AI AS HUMANITY'S SALIERI: QUANTIFYING LINGUISTIC CREATIVITY OF LANGUAGE MODELS VIA SYSTEMATIC ATTRIBUTION OF MACHINE TEXT AGAINST WEB TEXT

Anonymous authors
Paper under double-blind review

ABSTRACT

Creativity has long been considered one of the most difficult aspect of human intelligence for AI to mimic. However, the rise of Large Language Models (LLMs), like ChatGPT, has raised questions about whether AI can match or even surpass human creativity. We present CREATIVITY INDEX as the first step to quantify the linguistic creativity of a text by reconstructing it from existing text snippets on the web. CREATIVITY INDEX is motivated by the hypothesis that the seemingly remarkable creativity of LLMs may be attributable in large part to the creativity of human-written texts on the web. To compute CREATIVITY INDEX efficiently, we introduce DJ SEARCH, a novel dynamic programming algorithm that can search verbatim and near-verbatim matches of text snippets from a given document against the web. Experiments reveal that the CREATIVITY INDEX of professional human authors is on average 66.2% higher than that of LLMs, and that alignment reduces the CREATIVITY INDEX of LLMs by an average of 30.1%. In addition, we explore variations in the CREATIVITY INDEX among different human authors and discuss the potential factors contributing to these differences. Finally, we showcase a novel application of CREATIVITY INDEX for zero-shot machine text detection, where it proves to be surprisingly effective—outperforming the strong zero-shot system DetectGPT by a substantial margin of 30.2%, and even surpassing a leading supervised system, GhostBuster, in five out of six domains.

1 INTRODUCTION

Creativity has long been considered one of the most challenging "holy grail" of human intelligence for AI to mimic (Hasselberger & Lott, 2023). However, Large Language Models (LLMs) such as ChatGPT have taken the world by storm with their creative power. From generating poetry (Sawicki et al.; Deng et al., 2024b; Sawicki et al., 2023) and composing music (Ding et al., 2024; Deng et al., 2024a; Liang et al., 2024) to designing artwork (Makatura et al., 2024; Jignasu et al., 2023; Lim et al., 2024) and crafting compelling narratives (Yuan et al., 2022; Mirowski et al., 2023a; Ippolito et al., 2022), LLMs take only seconds to produce outputs that would rival or even surpass the work of human creators. This proficiency has even sparked a growing trend of using LLMs for content creation in industrial settings. For example, major studios in Hollywood have integrated LLMs into production processes such as movie scriptwriting (Carnevale, 2023). While studio executives are optimistic about using LLMs to streamline production and reduce costs, Hollywood writers are deeply concerned about being replaced by the rapid integration of LLMs in the industry, leading to a five-month writers' strike (Koblin & John, 2023).

While science fiction writer Ted Chiang characterizes LLMs as a blurry JPEG of the web (Hubert et al., 2024), many others wonder whether AI can indeed match or surpass the creativity of humanity. After all, LLMs have consumed orders of magnitude more works of writing than any single human could ever read, thus it may seem possible that LLMs could consequently reach a new level of literary sophistication and creativity beyond that of humanity at large.

To answer this question, the first step is to assess the level of creativity in machine texts compared to human texts. Creativity is a complex and ambiguous process that is challenging to define and quantify (Csikszentmihalyi, 1997; Glaveanu et al., 2020; Eagleman & Brandt, 2017; Paeth). Several previous studies have attempted to quantify creativity in writing by developing specific rubrics and asking human evaluators to score the writing based on these criteria. Vaezi & Rezaei (2018) developed a comprehensive rubric to assess fiction writing, while Biggs & Collis (1982) used a taxonomy of structural complexity to categorize creative writing. More recently, Chakrabarty et al. (2024) applied the Torrance Test of Creative Thinking to evaluate the creativity of short stories generated by LLMs in terms of fluency, flexibility, originality and elaboration. While these rubric-based methods are valuable, scaling them up to evaluate large amounts of texts generated by LLMs is impractical due to the reliance on human evaluators.

In this work, we propose CREATIVITY INDEX, a novel statistical measure of linguistic creativity in text. The key intuition underlying CREATIVITY INDEX is to quantify the degree of linguistic creativity of a given text by reconstructing that text via mixing and matching of a vast amount of existing text snippets on the web (See Figure 1a; 24 additional examples in Appendix Fig. 5 to Fig. 30). The underlying premise of our work is that the seemingly remarkable creativity of LLMs may be in large part attributable to the remarkable creativity of human-written texts on the web. This contrasts with distinguished human authors such as Hemingway, whose original content and unique writing style cannot be easily replicated by simply assembling snippets from other works. To test this, we provide a novel computational approach to systematically attribute machine text to web texts. Specifically, we introduce DJ SEARCH,[1] a novel dynamic programming algorithm that can efficiently search for verbatim and near-verbatim matches of text snippets from a given document against the web. Here, near-verbatim matches are defined as close paraphrases, characterized by high semantic similarity. Our algorithm combines strict verbatim matching using Infini-gram (Liu et al., 2024), which allows for fast retrieval of any existing sequence of words, with near-verbatim semantic matching achieved through a novel application of Word Mover's Distance (WMD) (Kusner et al., 2015) computed on the word embeddings of text snippets.

The contribution of our work is threefold: First, we introduce the CREATIVITY INDEX to reveal novel insights about machine creativity and human creativity. We find that the CREATIVITY INDEX of human authors—specifically professional writers and historical figures—is on average 66.2% higher than that of LLMs. This creativity gap is consistent across various domains—novel snippets, modern poems, and speech transcripts—at both verbatim and semantic levels. Moreover, we notice that Reinforcement Learning from Human Feedback (RLHF), a widely used alignment method, dramatically reduces the CREATIVITY INDEX of LLMs, by an average of 30.1%. This reduction is more significant at the verbatim level than the semantic level, indicating that LLMs may have converged to certain linguistic style preferred by humans during alignment. Furthermore, we explore variations in CREATIVITY INDEX among different human authors. Famous authors like Hemingway and Dickens tend to have higher CREATIVITY INDEX, though this should be interpreted with caution. Beyond inherent differences in creativity, CREATIVITY INDEX can be influenced by factors such as writing style and the time of composition. For instance, older English writings may exhibit higher CREATIVITY INDEX, as they are more difficult to reconstruct from web texts.

Second, we introduce DJ SEARCH as an efficient algorithmic tool to trace the usage of existing text snippets from the web that LLMs incorporate to compose new generations. The power of LLMs arises from training exhaustively on existing human-written texts on the web, and it is meaningful to trace back and acknowledge the human writers whose work empowers these models' outputs—just as we credit original composers when enjoying a DJ's remix.

Finally, we demonstrate a novel use of CREATIVITY INDEX as a surprisingly effective criterion for zero-shot black-box machine text detection. Our method is ready to deploy out-of-the-box, requiring no training or prior knowledge of the text generator. It not only surpasses the strong zero-shot baseline, DetectGPT (Mitchell et al., 2023a), by a significant margin of 30.2%, but also outperforms a leading supervised baseline, GhostBuster (Verma et al., 2024)—which requires expensive data collection for supervised training—in five out of six domains.

This work also faces the following limitations. First, CREATIVITY INDEX is designed to measure one specific aspect of creativity—linguistic creativity (i.e., the novelty in composing words and phrases). It might not comprehensively capture other dimensions of creativity, such as rhetorical complexity or structural flexibility, and is thus complementary to existing creativity measurement

[1]The name DJ SEARCH is inspired by the way a DJ creates a remix by blending pieces of existing music.

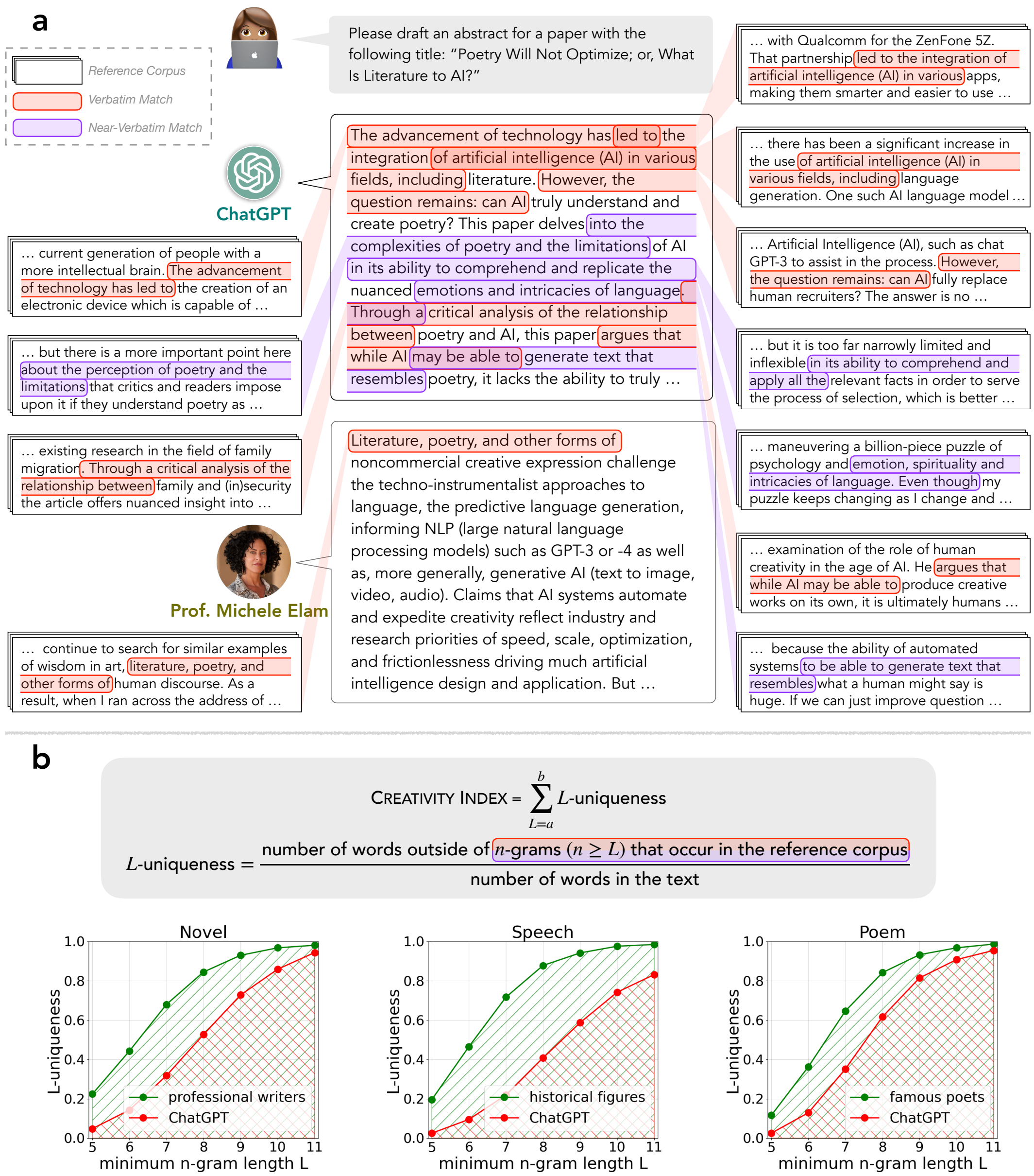


Figure 1: **a**: **Example outputs from DJ SEARCH.** We asked ChatGPT to generate an abstract based on the title of Prof. Michele Elam's paper, "Poetry Will Not Optimize; or, What Is Literature to AI?" (Elam, 2023) The abstract generated by ChatGPT contains significantly more verbatim and near-verbatim matches with existing texts on the web compared to the original abstract written by Prof. Elam. **b**: **Definition of CREATIVITY INDEX.** CREATIVITY INDEX is mathematically equivalent to the area under the L-uniqueness curve across a range of minimum n-gram lengths L. The L-uniqueness of ChatGPT is noticeably lower than that of proficient human writers across various context granularities (i.e., n-gram lengths) in all domains, leading to a significantly higher CREATIVITY INDEX for human writers compared to ChatGPT.

methods. Second, while CREATIVITY INDEX is robust in reflecting statistical differences between professional human writing and seemingly remarkable outputs from LLMs, it may not fully capture nuanced differences between human writings. Beyond inherent creativity variations, factors such as writing style and the time of composition can also influence the metric. Third, CREATIVITY INDEX assumes that the input text is of sufficient quality, as our study focuses on outputs from recent LLMs that are already fluent and coherent. For less refined texts, our metric can be complemented with standard automatic quality measures, such as fluency classifiers or perplexity-based evaluations, to provide a more comprehensive assessment. Lastly, the computation of CREATIVITY INDEX is constrained by the reference corpus used in DJ SEARCH. Without access to and inclusion of the private training data of closed-source models like ChatGPT in the reference corpus, the CREATIVITY INDEX for these models might be somewhat inflated. However, the research community is making progress in open-sourcing LLM research. Beyond RedPajama, newer large-scale pretraining corpora, such as DOLMA (Soldaini et al., 2024) and Common Corpus (PleIAs), have become available since our experiments.

We believe that our study will enhance the understanding of LLMs and guide informed usage of content created by LLMs, by providing an interoperable and scalable measurement to assess creativity in machine texts. Additionally, we hope that the out-of-the-box machine text detection enabled by the CREATIVITY INDEX can empower individuals to discern between human texts and machine texts, fostering a more informed and critical engagement with information in the digital age.

2 METHOD

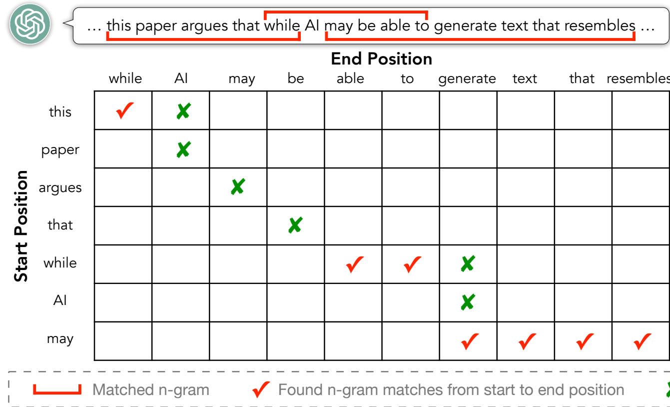
CREATIVITY INDEX The key intuition underlying CREATIVITY INDEX is to quantify the degree of linguistic creativity of a given text by estimating how much of that text can be reconstructed by mixing and matching a vast amount of existing text snippets on the web, as shown in Figure 1a. Specifically, CREATIVITY INDEX assesses the extent to which the content of the text can be traced back to similar or identical contexts found in other existing texts. This metric is grounded in the notion of originality from creative thinking in psychology literature, which is defined as the statistic rarity of a response or an idea (Torrance, 1966; Crossley et al., 2016).

Concretely, let $\mathbf{x}$ be a text whose creativity we aim to quantify, such as a speech transcript or a poem, either human written or machine generated. Let an n-gram of $\mathbf{x}$ be any contiguous sequence of n words of $\mathbf{x}$, and let $\mathbf{x}_{i:i+n}$ be the n-gram of $\mathbf{x}$ starting in the i-th word. Let C be a massive reference corpus of publicly available texts on the web , and let f be a binary function that determines whether an n-gram $\mathbf{x}_{i:i+n}$ occurs anywhere in the corpus C. We define the L-uniqueness of a text $\mathbf{x}$ as the proportion of words $w \in \mathbf{x}$ such that none of the n-grams in $\mathbf{x}$ that include w occur in the corpus C for $n \geq L$—denoted $\text{uniq}(\mathbf{x}, L)$. Intuitively, L-uniqueness measures the proportion of $\mathbf{x}$'s words that are used in novel contexts (here, n-grams), unseen across a vast text collection C. Thus, a higher L-uniqueness implies a higher level of originality of $\mathbf{x}$. Formally, $\text{uniq}(\mathbf{x}, L) = \sum_{k=1}^{\|\mathbf{x}\|} \mathbb{1}\{f(\mathbf{x}_{i:i+n}, C) = 0\ \forall\, i \in (k-n, k],\ n \geq L\}/\|\mathbf{x}\|$, where trivially $\text{uniq}(\mathbf{x}, L) \in [0, 1]$.

Note that when fixing $\mathbf{x}$, the function $\text{uniq}(\mathbf{x}, L)$ is monotonically increasing as L grows. Its improper integral—$\sum_{n \geq L} \text{uniq}(\mathbf{x}, n)$—is an indicator of the overall uniqueness of $\mathbf{x}$ across various context granularities (i.e., n-gram lengths), and because of $\text{uniq}(\mathbf{x}, L)$'s monotonicity it indirectly measures uniqueness growth speed. We thus define CREATIVITY INDEX as $\sum_{n \geq L} \text{uniq}(\mathbf{x}, n)$, with higher CREATIVITY INDEX indicating greater linguistic originality with respect to the corpus C, as shown in Figure 1b.

When a text $\mathbf{x}$ is part of the reference corpus C, its CREATIVITY INDEX would trivially become zero. This issue often arises with works from famous authors, as their writings are widely available online. To address this, for human texts written before the cutoff date of the reference corpus, we exclude any document $\mathbf{d} \in C$ that contains copies, quotations, or citations of $\mathbf{x}$ and compute CREATIVITY INDEX using this filtered corpus, detailed in Appendix A.3.

DJ SEARCH To enable the use of our CREATIVITY INDEX it is vital to compute it efficiently. For the efficient computation, we introduce DJ SEARCH, a dynamic programming algorithm designed to radpily identify the set of all $\mathbf{x}$'s n-grams ($n \geq L$) that occur in the corpus C.



Algorithm DJ SEARCH $(\mathbf{x}, C)$

Input: text $\mathbf{x}$, reference corpus C,
n-gram matching function $f(\mathbf{x}_{i:j}, C)$
minimum n-gram length L
Output: set of matched n-gram $\mathbf{S}$
$i \leftarrow 0,\ j \leftarrow L,\ \mathbf{S} \leftarrow \varnothing$
while $j \leq ||\mathbf{x}||$ **do**
  if $f(\mathbf{x}_{i:j}, C) = 1$ **then**
    $\mathbf{S} \leftarrow \mathbf{S} \cup \{\mathbf{x}_{i:j}\}$
    $j \leftarrow j + 1$
  else
    $i \leftarrow i + 1$
    $j \leftarrow \max(i + L,\ j)$
  end if
end while

Figure 2: **An illustration of DJ Search algorithm.** A brute force approach would independently check if every n-gram of $\mathbf{x}$ occurs in C, performing a quadratic number of f evaluations with respect to $\mathbf{x}$'s length (i.e., checking every cell in the grid). DJ SEARCH is a two-pointer method that takes only a linear number of f evaluations. By progressively analyzing n-grams starting and/or ending at a later endpoint than before, DJ SEARCH limits the total number of f evaluations to $2||\mathbf{x}||$. In this example, the minimum n-gram length L is set to 5.

A brute force approach would independently check if every n-gram of $\mathbf{x}$ occurs in C, performing a quadratic number of f evaluations with respect to $\mathbf{x}$'s length, and thus making it too computationally expensive. Instead, we design a two-pointer method (Laaksonen, 2020) that takes only a linear number of f evaluations, as illustrated in Figure 2. The key idea is to reduce finding all n-grams occurring in C to identifying the longest n-gram occurring in C starting at each index i: once those have been found, it is trivial to deduce all the n-gram occurring in C by computing their subsequences. Concretely, we progressively analyze the whole document $\mathbf{x}$ by iteratively searching for the longest n-gram that starts at each index i and occurs in C, using f as the assessment. Once we have found such longest n-gram starting at i, we crucially reuse computations for $i+1$ by noting that $f(\mathbf{x}_{i:i+n}, C) = 1$ implies $f(\mathbf{x}_{i+1:i+n}, C) = 1$. Thus, we always analyze n-grams starting and/or ending at a later endpoint than before, which upper bounds the number of analyzed n-grams (i.e., the number of f calls) to at most $2||\mathbf{x}||$. The implementation is detailed in Appendix A.1.

In addition to minimizing the number f evaluations, DJ SEARCH optimizes the time complexity of each evaluation. f determines whether a n-gram $\mathbf{x}_{i:i+n}$ occurs in the corpus C either exactly or in a semantically similar way—e.g., a paraphrase of $\mathbf{x}_{i:i+n}$ exists in C. Semantic similarity is often computed using text embeddings, which are fixed-length vector representations of text meanings. This reduces measuring text similarity to computing vector distance. Text embeddings, typically generated by complex models (e.g., BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), SpanBERT (Joshi et al., 2020)) lack linearity, requiring independent computation for each n-gram in $\mathbf{x}$ and C. To alleviate this issue we use Word Mover's Distance (WMD) (Kusner et al., 2015), an optimal transport-inspired metric that measures distance between two n-grams by combining word embedding distances between each n-gram's words. WMD enables optimizing f's computation, as pairwise distances between word embeddings can be pre-computed for every pair of words, and then be reused in every function call of f to identify n-grams in C that are semantically similar to the ones in $\mathbf{x}$. The implementation is detailed in Appendix A.2.

To further boost efficiency, and given that occurrences of $\mathbf{x}_{i:i+n}$ are more likely in texts similar to $\mathbf{x}$, we estimate f by computing WMD only for the texts in C most similar to $\mathbf{x}$, as identified by BM25 (Robertson & Walker, 1994). Moreover, exact occurrences of $\mathbf{x}_{i:i+n}$ in C represent a less costly special case in computing f. We further optimize f's computation by using Infini-gram (Liu et al., 2024), which finds exact matches of $\mathbf{x}_{i:i+n}$ in C in milliseconds; WMD is computed only if no matches are found by Infini-gram.

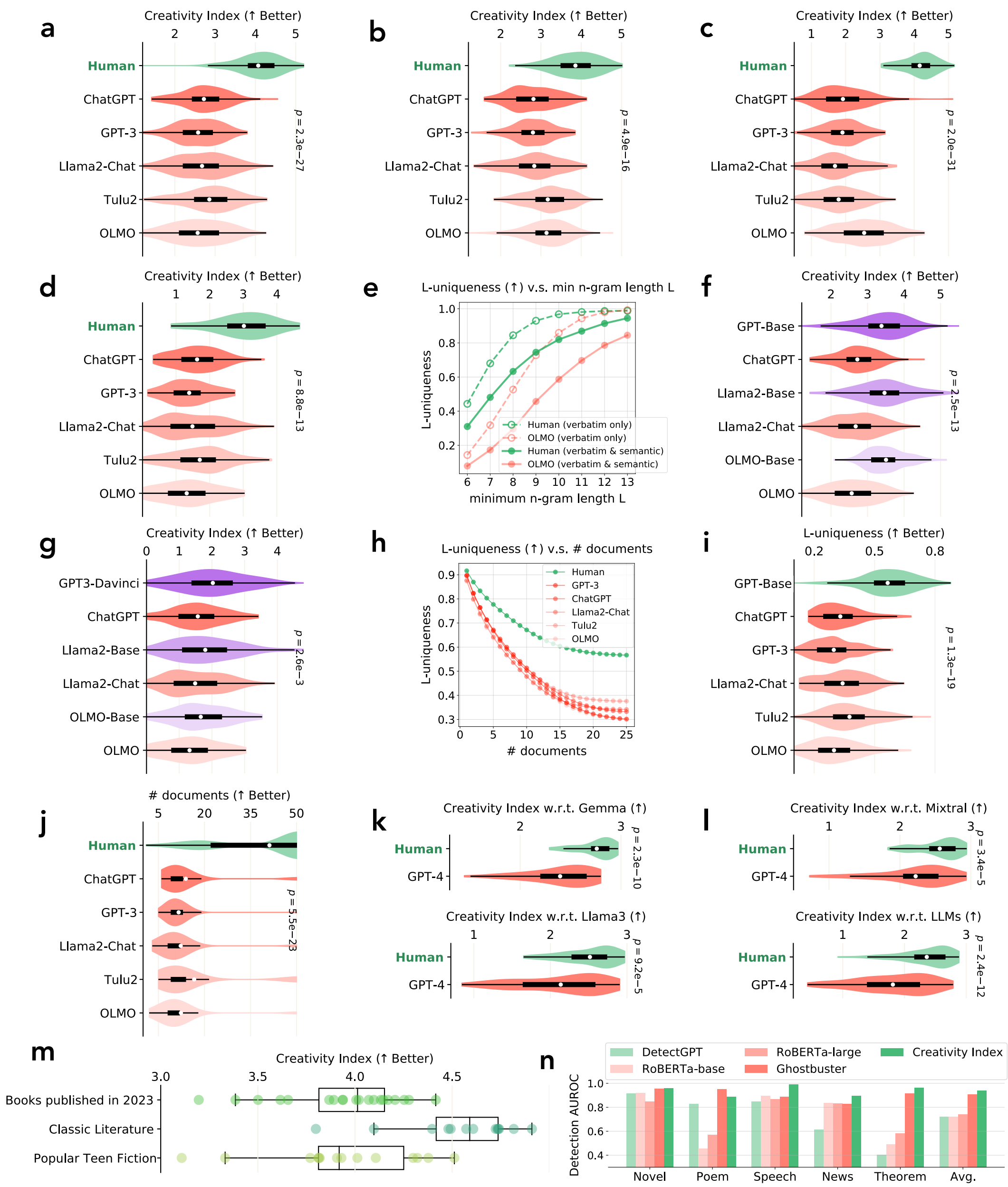


Figure 3: **a-c**: CREATIVITY INDEX in novel writing (**a**), poetry composition (**b**) and speech writing (**c**) based solely on verbatim matches. **d**: CREATIVITY INDEX in novel writing considering both verbatim and semantic matches. **e**: L-uniqueness in novel writing with respect to the minimum n-gram length L for humans and OLMo. **f-g**: CREATIVITY INDEX of LLMs before and after RLHF in novel writing, based solely on verbatim matches (**f**) and based on both verbatim and semantic matches (**g**). **h**: L-uniqueness in novel writing with respect to number of documents in the reference corpus. **i**: L-uniqueness when search over the top 50 documents in novel writing. **j**: The number of reference documents required to keep L-uniqueness below 50% in novel writing. **k-l**: CREATIVITY INDEX of GPT-4 compared to humans in novel writing based on verbatim matches, using a machine-generated reference corpus sourced from the instruction-aligned version of Gemma-7B, Llama3-8B, and Mixtral-7B, as well as a combination of all three. **m**: CREATIVITY INDEX of different groups of human writers. **n**: Detection AUROC across various domains: our approach sets a new state-of-the-art for zero-shot detection, even surpassing supervised baselines.

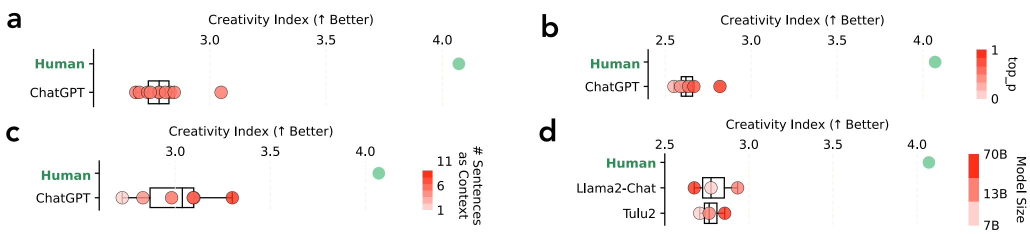


Figure 4: **a-c**: CREATIVITY INDEX of ChatGPT in novel writing based on verbatim matches, with different prompt formats (**a**), p values in top-p decoding (**b**) and prompt length (**c**). **d**: CREATIVITY INDEX of LLaMA 2 Chat and Tulu 2 with different model sizes.

3 EVALUATION

How does the creativity of language models compare to humans? We compute the CREATIVITY INDEX for machine texts and human texts across three creative writing tasks: novel writing, poetry composition, and speech drafting. For human texts, we use book snippets in the BookMIA (Shi et al., 2024) dataset, popular modern poems collected by PoemHunter.com, and famous speeches from the American Rhetoric speech bank. For machine texts, we prompt LLMs to generate several paragraphs of novels, poems, or speeches, starting with an initial sentence from existing human writings in each category (see Appendix B.1 for details). We experiment with state-of-the-art LLMs, including GPT-3 (Brown et al., 2020), ChatGPT (Ouyang et al., 2022), LLaMA 2 Chat (Touvron et al., 2023), Tulu 2 (Ivison et al., 2023), and OLMo Instruct (Groeneveld et al., 2024). For open-source and open-weight models, we use the largest model size available from each model family. We use RedPajama (Computer, 2023), a large-scale English corpus with 900 million web documents, as the reference corpus. The models we analyze are primarily pre-trained on the web data available before the cutoff date of the reference corpus RedPajama. We will discuss later how to handle newer models, such as GPT-4 (OpenAI et al., 2023), given that it was largely trained on more recent web data and third-party private data, both of which fall outside the reference corpus. We restrict the matching criteria to verbatim matches only in the first experiment. We will ablate the effect of different matching criteria, prompt formats, decoding strategies, context length, and model sizes in later experiments.

Our primary finding is that *humans consistently exhibit a much higher level of creativity compared to any LLM across all tasks* (Fig. 3**a-c**). Averaged across all models, the CREATIVITY INDEX of humans is 52.2% higher[2] than LLMs in novel writing ($p = 6.9 \times 10^{-27}$, by Mann-Whitney U test unless otherwise specified; $N = 600$), 31.1% higher in poetry composition ($p = 1.5 \times 10^{-15}$; $N = 600$) and 115.3% higher in speech drafting ($p = 6.1 \times 10^{-31}$, $N = 600$). This suggests that human writings are composed of far more unique combinations of words and phrases compared to model generations. On the other hand, the differences in model creativity are much smaller and show very low statistical significance ($p = 0.09$; $N = 1500$).

Furthermore, we experiment with different prompt formats on top of ChatGPT, intentionally encouraging creativity in the model's generations by incorporating instructions such as 'push for creative ideas, unique emotions, and original twists,' 'be bold and creative,' or 'you are a creative writer.' (Fig. 4**a**) For a full list of the prompts we used, please see Appendix B.1. We found that the difference in the CREATIVITY INDEX of ChatGPT across different prompts is minimal, with no statistical significance ($p = 0.23$; $N = 600$). We also experimented with different decoding strategies by varying the p value in top-p decoding (Fig. 4**b**). Although a higher p value resulted in a marginally higher CREATIVITY INDEX, the difference was minimal and not statistically significant ($p = 0.23$; $N = 600$). Moreover, we ablate the effect of prompt length by varying the number of sentences from human writings included in the prompt (Fig. 4**c**). We found that longer prompts tended to result in a slightly higher CREATIVITY INDEX, likely due to the model copying more from the longer human text in the prompt. However, the statistical significance of these differences is very low ($p = 0.13$; $N = 600$). Lastly, we analyze the effect of different model sizes for LLaMA 2 Chat and Tulu 2, but do not observe a consistent trend ($p = 0.12$; $N = 600$) (Fig. 4**d**).

[2]The percentage difference computed using the formula: $\frac{\textsc{Creativity Index (human)} - \textsc{Creativity Index (model)}}{\textsc{Creativity Index (model)}}$

How do different matching criteria affect creativity measurement? We experiment with restricting valid matches to verbatim only, and with allowing both verbatim and semantic matches. First, *the creativity gap between humans and LLMs becomes even larger when considering semantic matches in addition to verbatim matches* (Fig. 3**d**). Averaged across all models, the CREATIVITY INDEX of human, based on both verbatim and semantic matches, is 102.5% higher than LLMs in novel writing ($p = 2.6 \times 10^{-12}$; $N = 600$), whereas based on verbatim matches alone, the CREATIVITY INDEX of human is 52.2% higher than LLMs. Second, *semantic matches provide more signal for analyzing the uniqueness of longer n-grams* (Fig. 3**e**). For example, while the gap in L-uniqueness at $L = 11$ between human text and machine text from OLMo Instruct is 3.7% based on verbatim matches alone, this gap widens to 16.3% when considering both verbatim and semantic matches ($p = 3.1 \times 10^{-7}$; $N = 600$). This indicates that although some of the longer n-grams in machine text may appear unique at the verbatim level, they are similar to certain text snippets in the reference corpus at the content level.

What impact does RLHF have on model creativity? RLHF aims to align model's outputs with human preferences, enhancing LLMs' ability to follow instructions and improving their safety and adaptability. To understand the impact of RLHF on model creativity, we compare the CREATIVITY INDEX of the LLMs before and after RLHF alignment. Specifically, we experiment with GPT Base (Brown et al., 2020), LLaMA 2 Base (Touvron et al., 2023), and OLMo Base (Groeneveld et al., 2024) and compare their creativity with their counterparts post-RLHF alignment. Our main finding is that *the* CREATIVITY INDEX *of models after RLHF alignment is much lower than those before RLHF* (Fig. 3**f-g**). Based on verbatim match alone, the CREATIVITY INDEX of LLMs reduces by an average of 30.1% after RLHF ($p = 1.3 \times 10^{-12}$; $N = 600$). Based on both verbatim and semantic matches, the CREATIVITY INDEX of LLMs decreases by an average of 8.9% after RLHF ($p = 0.01$; $N = 600$). We notice that *the reduction of* CREATIVITY INDEX *after RLHF is noticeably larger when considering verbatim matches alone.* We speculate that models might have learned certain linguistic styles preferred by humans during RLHF, leading to a decreased surface form diversity in its outputs.

How do overlapped n-grams distribute in the reference corpus? In addition to measuring the amount of matched n-grams in a given text, we also investigate the distribution of these n-grams in the reference corpus. We aim to understand whether these matched n-grams are spread across many documents or concentrated in a few. Specifically, we identify the top N documents that contain the highest amount of matched n-grams and result in the minimum L-uniqueness for a given text. This problem can be reduced to the maximum coverage problem (Nemhauser et al., 1978) and approximated using a greedy algorithm. Here, we consider both verbatim and semantic matches. Our main finding is that *the matched n-grams in machine texts are concentrated in fewer documents compared to human texts* (Fig. 3**h-j**). When searching over the top 50 documents, the averaged L-uniqueness ($L = 5$) for machine texts is 32.8%, which is 73.4% lower than human texts (mean: 56.6%; $p = 3.9 \times 10^{-19}$; $N = 600$). Conversely, keeping L-uniqueness below 50% requires searching through an average of 41.2 documents for human texts, which is 213.7% more than for machine texts (mean: 13.4; $p = 1.6 \times 10^{-22}$; $N = 600$). This implies that it's more likely to find some existing documents resemble models' generations than human writings.

How to measure creativity in LLMs trained on data outside of the reference corpus? The CREATIVITY INDEX of GPT-4 would be significantly inflated if computed using the RedPajama corpus, as RedPajama's cutoff date is two years earlier than GPT-4's knowledge cutoff, and GPT-4 is additionally trained on third-party private data that we don't have access to. We hypothesize that LLMs pre-trained on similar web data are likely to memorize and replicate similar patterns. As a result, when comparing the generations of these models, we expect them to be more similar to each other than to human texts, which often contain long-tail patterns. Therefore, to compare the creativity level of GPT-4 with humans, we use a model-generated reference corpus from newer open-weight models with knowledge cutoff dates similar to GPT-4, including the instruction-aligned versions of Gemma-7B (Team et al., 2024), Llama3-8B (AI@Meta, 2024), and Mixtral-7B (Jiang et al., 2023). Specifically, we randomly sample 150k sentences from the RedPajama corpus and prompt these models to generate document-level continuations. *Based on the model-generated reference corpus, the average* CREATIVITY INDEX *of humans is 30.3% higher than GPT-4 in novel writing* ($p = 2.3 \times 10^{-12}$; $N = 600$) (Fig. 3**k-l**). This suggests that while newer LLMs like GPT-4 may appear more creative when compared to public data, they still learn common patterns from their private training data and tend to emit similar patterns as other LLMs trained on comparable data.

How does the creativity vary among different groups of human? Human populations are diverse and complex, we aim to explore whether writings from different human authors exhibit varying levels of linguistic creativity. Specifically, we compare the linguistic creativity among three categories of writings: books published in 2023 from the BookMIA (Shi et al., 2024) dataset, classic literature by famous authors, and popular young adult fictions, both sampled from Goodreads' book lists. We first observe that the variation in CREATIVITY INDEX among human authors is relatively smaller compared to the difference between LLMs and humans. Additionally, we found that *classic literature tends to exhibit higher linguistic creativity.* On average, the CREATIVITY INDEX of classic literature is 21.6% higher than young adult fictions ($p = 2.7 \times 10^{-90}$; $N = 3000$), and 13.8% higher than books published in 2023 ($p = 4.3 \times 10^{-120}$; $N = 3000$). These findings should be interpreted with a grain of salt. We experimented with only a small set of writings from each category, so the results may not generalize broadly. Additionally, beyond inherent differences in creativity, CREATIVITY INDEX can be influenced by factors such as writing style and the time of composition. For instance, some classic literature are written in older English, which may result in a higher CREATIVITY INDEX because such writings are harder to reconstruct from web texts that primarily use modern English. In addition to the differences across categories, we also observed noticeable variance in creativity within each category. For example, the CREATIVITY INDEX of 'The Hunger Games' is 35.4% higher than 'Twilight' ($p = 1.5 \times 10^{-19}$; $N = 200$), even though both books belong to the category of popular young adult fiction.

Can we leverage differences in creativity for detecting machine-generated text? Based on the creativity difference between humans and LLMs, we explore a novel use case of CREATIVITY INDEX for zero-shot black-box machine text detection Texts with higher creativity are more likely to be written by human. Our approach is ready to deploy out-of-the-box, requiring no training or prior knowledge of the text generator. In addition to creative writing tasks, we also test our method on detecting machine-generated fake news and theorem proofs. Detecting fake news is crucial for protecting the public from misinformation, while identifying model-generated solutions is important for regulating students' use of LLMs in their coursework. To obtain additional test data, we prompt LLMs to generate news articles based on the fake news headlines from the Misinfo Reaction Frames (Gabriel et al., 2022) and compare them with the real news articles from the XSum (Narayan et al., 2018) dataset. Meanwhile, we prompt LLMs to generate proofs for theorems from the NaturalProofs (Welleck et al., 2022) benchmark, and compare them with the ground-truth human-written proofs. The baselines we compare against includes a widely adopted strong zero-shot detector, DetectGPT (Mitchell et al., 2023a), which uses the curvature of log probability as the detection criterion, as well as several supervised methods. These include OpenAI's RoBERTa-based detector, fine-tuned on millions of generations from various GPT-2 sized models, as well as a more recent strong supervised detector, Ghostbuster (Verma et al., 2024), fine-tuned on thousands of generations from ChatGPT. We measure performance using the area under the receiver operating characteristic curve (AUROC), which represents the probability that a classifier correctly ranks a randomly-selected human-written example higher than a randomly selected machine-generated example. We found CREATIVITY INDEX is surprisingly effective for zero-shot machine text detection: it consistently surpasses DetectGPT and OpenAI's detector across all domains, with significant improvements in AUROC—30.2% and 26.9%, respectively. It also outperforms the strong supervised baseline, Ghostbuster—which requires expensive training and data collection—in five out of six domains, achieving an average AUROC improvement of 3.5% (Fig. 3**n**).

4 DISCUSSION

This work investigates the level of linguistic creativity in texts generated by LLMs and written by humans. Our findings suggest that the content and writing style of machine-generated texts may be less original and unique, as they contain significantly more semantic and verbatim matches with existing web texts compared to high-quality human writings. We hypothesize that this limited creativity in models may result from the current data-driven paradigm used to train LLMs. In this paradigm, models are trained to mimic human-written texts during the pre-training stage, and to produce outputs aligned with human preferences during the RLHF stage. As a result, models learn to generate fluent and coherent texts by absorbing and replicating common patterns observed in their training data. This reliance on existing text patterns can restrict their originality, as their outputs are inherently shaped by previously seen examples. In contrast, accomplished authors such as Hemingway go beyond simply mimicking the great writings of others; they craft their own narratives to express their unique opinions, perspectives, and insights, drawing from their personal experiences, emotions, and backgrounds, which translates to the more creative compositions of words and phrases that our method detects. Just as a DJ remixes existing tracks while a composer creates original music, we speculate that LLMs behave more like DJs, blending existing texts to produce impressive new outputs, while skilled human authors, similar to music composers, craft original works.

This work also faces the following limitations. First, CREATIVITY INDEX is designed to measure one specific aspect of creativity—linguistic creativity (i.e., the novelty in composing words and phrases). It might not comprehensively capture other dimensions of creativity such as rhetorical complexity or structural flexibility. Therefore, while CREATIVITY INDEX is an effective tool for understanding seemingly remarkable machine texts, it is insufficient on its own to draw definitive conclusions about overall creativity differences across various writings, particularly when distinguishing nuanced variations among human authors. Second, the computation of the CREATIVITY INDEX is constrained by the reference corpus used for DJ SEARCH. While open-source LLMs such as OLMo rely on publicly available texts from the internet for their training data, major companies like OpenAI additionally curate private data to train their closed-source LLMs such as ChatGPT. Without incorporating these private data into the reference corpus of DJ SEARCH, the CREATIVITY INDEX of closed-source LLMs may be somewhat inflated. In addition, since the reference corpus primarily consists of more recent Internet texts, it may introduce an implicit bias favoring writings from older periods, as these texts are underrepresented in modern web data and are therefore harder to reconstruct. Third, the overlap with existing texts identified by DJ SEARCH in models' generations may not conclusively indicate memorization of a specific document. It's possible that these text fragments, or their variations, appear in multiple documents that the model has been trained on, including those outside the reference corpus of DJ SEARCH. Forth, the human authors that this work focuses on are those with relatively high-quality writings available in existing public datasets. While some human writings can be mediocre, tedious and unoriginal, we aim to assess how the creativity levels in impressive LLM outputs compare against the high-quality writings produced by professional human authors. Lastly, we acknowledge that the discussion surrounding the use of LLMs in social and industrial settings is highly complex, and our work here speaks only to a part of it. Besides the creativity of machine-generated content, other considerations in this discussion include socioeconomic factors and ethical implications, which fall beyond the scope of this paper.

5 RELATED WORK

Measuring Creativity in Ideas: Measuring creative thinking and problem solving takes root in early work in psychology (Torrance, 1966), where researchers defined four pillars for creative thinking: fluency, flexibility, originality and elaboration. Crossley et al. (2016) later on developed this notion and built on it to expand this to measuring creative writing in students, where they also adopted n-gram novelty for a measure of originality. However, these prior work focus on creativity in humans, and they also do not introduce any automated metrics or measurements.

Measuring Creativity in Machine-generated Text Using Expert Annotators: Closely related to CREATIVITY INDEX is a recent line of work in the generative AI literature comparing the creativity of human writers to that of large language models in different domains such as story telling and journalism (Chakrabarty et al., 2023; 2024; Anonymous, 2024). Similar to us, the approach in this direction often involves prompting an LLM to write an original story or news article, based on some existing premise or press release, and then comparing the machine-generated text to the human-written counterparts. These works, however, take a rather subjective approach, where they define and measure creativity based on human expert annotations and whether people perceive the text to be more creative, rather than an objective quantification of novelty that we provide.

Measuring Novelty of N-grams: Finally, closely related to our work in terms of techniques is Nguyen (2024) and Merrill et al. (2024). The former attempts at finding n-gram rules that would cover and predict generations from transformer models, showing that more than 70% of the times transformers follow some pre-set patterns and rules. The latter is more similar to our work as they also measure the novelty of generated n-grams and compare it to human-written text, however they differ from us in tow major ways: (1) they only find verbatim matches, whereas we also match to approximate, semantically similar blocks of text and (2) they compute the percentage of n-grams of a certain length in a text that can be found in the reference corpus, whereas we measure how much of the text can be reconstructed by mixing and matching a vast amount of existing text snippets of varying lengths from the web.

Machine Text Detection: Detecting machine-generated text has been explored for several years using a variety of methods (Jawahar et al., 2020; Uchendu et al., 2021). Gehrmann et al. (2019) and Dugan et al. (2023) demonstrate that even humans tend to struggle to differentiate between text written by humans and machines, highlighting the need for automated detection solutions. Some approaches involve training a classifier in a supervised manner to identify machine-generated text (Bakhtin et al., 2019; Uchendu et al., 2020), while others use a zero-shot detection method (Solaiman et al., 2019; Ippolito et al., 2020). Additionally, there is research on bot detection through question answering (Wang et al., 2023; Chew & Baird, 2003). Recently, Mitchell et al. (2023b) introduced DetectGPT, a zero-shot method based on the hypothesis that texts produced by a large language model (LLM) are located at local maxima, and thus exhibit negative curvature, in the model's probability distribution. Follow-up work build on DetectGPT by making it faster (Bao et al., 2024) and proposing to use cross-detection when the target model is unknown (Mireshghallah et al., 2024).

6 Conclusion

We introduce Creativity Index, an interoperable and scalable metric designed to quantify the linguistic creativity of a given text by estimating how much of that text can be reconstructed by mixing and matching a vast amount of existing text snippets on the web. To efficiently compute the Creativity Index, we developed DJ Search, a novel dynamic programming algorithm that can search verbatim and near-verbatim matches of text snippets from a given document against the web. We find that the creativity index of professional human writers is, on average, 66.2% higher than that of LLMs. Notably, RLHF dramatically reduces the creativity index of LLMs by an average of 30.1%. Furthermore, we demonstrate that Creativity Index can be used as a surprisingly effective criterion for zero-shot black-box machine text detection. Our method not only surpasses the strongest zero-shot baseline, DetectGPT, by a significant margin of 30.2%, but also outperforms the strongest supervised baseline, GhostBuster, in five out of six domains. We hope that this study enhances the understanding of LLMs through the lens of linguistic creativity, and fosters informed usage of content created by LLMs in real-world applications.

References

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

AI@Meta. Llama 3 model card. 2024. URL `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`.

Anonymous. Do LLMs plan like human writers? comparing journalist coverage of press releases with LLMs. In *Submitted to ACL Rolling Review - June 2024*, 2024. URL `https://openreview.net/forum?id=E3VS45jxPR`. under review.

Mikhail J. Atallah, Victor Raskin, Michael Crogan, Christian F. Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding*, 2001. URL `https://api.semanticscholar.org/CorpusID:37687669`.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. Real or fake? learning to discriminate machine from human generated text, 2019. URL `https://arxiv.org/abs/1906.03351`.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria,* *May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Bpcgcr8E8Z.

John B. Biggs and Kevin F. Collis. The psychological structure of creative writing. *Australian Journal of Education*, 26:59 – 70, 1982. URL https://api.semanticscholar.org/CorpusID:141381510.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Robert Carnevale. The hollywood writers' strike may actually be aiding ai's takeover, May 2023. URL https://www.thewrap.com/hollywood-writers-strike-ai-deepmind-dramatron-chatgpt/.

Tuhin Chakrabarty, Vishakh Padmakumar, Faeze Brahman, and Smaranda Muresan. Creativity support in the age of large language models: An empirical study involving emerging writers. *arXiv preprint arXiv:2309.12570*, 2023.

Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. Art or artifice? large language models and the false promise of creativity. In *Proceedings of the CHI Conference on Human Factors in Computing Systems, CHI 2024, Honolulu, HI, USA, May 11-16, 2024*, pp. 30:1–30:34. ACM, 2024. URL https://doi.org/10.1145/3613904.3642731.

Monica Chew and Henry S. Baird. Baffletext: a human interactive proof. In *IS&T/SPIE Electronic Imaging*, 2003.

John Joon Young Chung, Shiqing He, and Eytan Adar. The intersection of users, roles, interactions, and technologies in creativity support tools. In *Proceedings of the 2021 ACM Designing Interactive Systems Conference*, pp. 1817–1833, 2021.

Together Computer. Redpajama: an open dataset for training large language models, 2023. URL https://github.com/togethercomputer/RedPajama-Data.

Scott A Crossley, Kasia Muldner, and Danielle S McNamara. Idea generation in student writing: Computational assessments and links to successful writing. *Written Communication*, 33(3):328–354, 2016.

Mihaly Csikszentmihalyi. Flow and the psychology of discovery and invention. *HarperPerennial, New York*, 39:1–16, 1997.

Qixin Deng, Qikai Yang, Ruibin Yuan, Yipeng Huang, Yi Wang, Xubo Liu, Zeyue Tian, Jiahao Pan, Ge Zhang, Hanfeng Lin, Yizhi Li, Ying Ma, Jie Fu, Chenghua Lin, Emmanouil Benetos, Wenwu Wang, Guangyu Xia, Wei Xue, and Yi-Ting Guo. Composerx: Multi-agent symbolic music composition with llms. *ArXiv preprint*, abs/2404.18081, 2024a. URL https://arxiv.org/abs/2404.18081.

Zekun Deng, Haoxia Yang, and Jun Wang. Can ai write classical chinese poetry like humans? an empirical study inspired by turing test. *ArXiv preprint*, abs/2401.04952, 2024b. URL https://arxiv.org/abs/2401.04952.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. URL https://aclanthology.org/N19-1423.

Shuangrui Ding, Zihan Liu, Xiao wen Dong, Pan Zhang, Rui Qian, Conghui He, Dahua Lin, and Jiaqi Wang. Songcomposer: A large language model for lyric and melody composition in song generation. *ArXiv preprint*, abs/2402.17645, 2024. URL `https://arxiv.org/abs/2402.17645`.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 12763–12771. AAAI Press, 2023. URL `https://doi.org/10.1609/aaai.v37i11.26501`.

David Eagleman and Anthony Brandt. *The runaway species: How human creativity remakes the world*. Catapult, 2017.

Michele Elam. Poetry will not optimize, or what is literature to ai? *American Literature*, 2023. URL `https://api.semanticscholar.org/CorpusID:257606488`.

Tina Fang, Martin Jaggi, and Katerina Argyraki. Generating steganographic text with LSTMs. In Allyson Ettinger, Spandana Gella, Matthieu Labeau, Cecilia Ovesdotter Alm, Marine Carpuat, and Mark Dredze (eds.), *Proceedings of ACL 2017, Student Research Workshop*, pp. 100–106, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL `https://aclanthology.org/P17-3017`.

Jonas Frich, Lindsay MacDonald Vermeulen, Christian Remy, Michael Mose Biskjaer, and Peter Dalsgaard. Mapping the landscape of creativity support tools in HCI. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, pp. 389. ACM, 2019. URL `https://doi.org/10.1145/3290605.3300619`.

Saadia Gabriel, Skyler Hallinan, Maarten Sap, Pemi Nguyen, Franziska Roesner, Eunsol Choi, and Yejin Choi. Misinfo reaction frames: Reasoning about readers' reactions to news headlines. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3108–3127. Association for Computational Linguistics, 2022. URL `https://aclanthology.org/2022.acl-long.222`.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 111–116. Association for Computational Linguistics, 2019. URL `https://aclanthology.org/P19-3019`.

Katy Gero, Alex Calderwood, Charlotte Li, and Lydia Chilton. A design space for writing support tools using a cognitive process model of writing. In *Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022)*, pp. 11–24. Association for Computational Linguistics, 2022. URL `https://aclanthology.org/2022.in2writing-1.2`.

Katy Ilonka Gero, Tao Long, and Lydia B. Chilton. Social dynamics of AI support in creative writing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI 2023, Hamburg, Germany, April 23-28, 2023*, pp. 245:1–245:15. ACM, 2023. URL `https://doi.org/10.1145/3544548.3580782`.

Vlad Petre Glaveanu, Michael Hanchett Hanson, John Baer, Baptiste Barbot, Edward P Clapp, Giovanni Emanuele Corazza, Beth Hennessey, James C Kaufman, Izabela Lebuda, Todd Lubart, et al. Advancing creativity theory and research: A socio-cultural manifesto. *The Journal of Creative Behavior*, 54(3):741–745, 2020.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh

Shah, William Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah Smith, and Hannaneh Hajishirzi. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15789–15809. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.acl-long.841.

William Hasselberger and Micah Lott. "where lies the grail? ai, common sense, and human practical intelligence". *Phenomenology and the Cognitive Sciences*, pp. 1–22, 2023. URL https://api.semanticscholar.org/CorpusID:264895815.

Kent F Hubert, Kim N. Awa, and Darya L. Zabelina. The current state of artificial intelligence generative language models is more creative than humans on divergent thinking tasks. *Scientific Reports*, 14, 2024. URL https://api.semanticscholar.org/CorpusID:267616181.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1808–1822. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.acl-main.164.

Daphne Ippolito, Ann Yuan, Andy Coenen, and Sehmon Burnam. Creative writing with an ai-powered writing assistant: Perspectives from professional writers. *ArXiv preprint*, abs/2211.05030, 2022. URL https://arxiv.org/abs/2211.05030.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *ArXiv preprint*, abs/2311.10702, 2023. URL https://arxiv.org/abs/2311.10702.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 2296–2309. International Committee on Computational Linguistics, 2020. URL https://aclanthology.org/2020.coling-main.208.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023. URL https://api.semanticscholar.org/CorpusID:263830494.

Anushrut Jignasu, Kelly O. Marshall, Baskar Ganapathysubramanian, Aditya Balu, Chinmay Hegde, and Adarsh Krishnamurthy. Towards foundational ai models for additive manufacturing: Language models for g-code debugging, manipulation, and comprehension. *ArXiv preprint*, abs/2309.02465, 2023. URL https://arxiv.org/abs/2309.02465.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. URL https://aclanthology.org/2020.tacl-1.5.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Brooks Barnes Koblin and John. Hollywood writers strike: Agreement gives writers guild most of what it wanted, Sep 2023. URL https://www.nytimes.com/live/2023/09/24/business/hollywood-writers-strike-news.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 957–966. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/kusnerb15.html.

Antti Laaksonen. *Guide to competitive programming*. Springer, 2020.

Xia Liang, Xingjian Du, Jiaju Lin, Pei Zou, Yuan Wan, and Bilei Zhu. Bytecomposer: a human-like melody composition method based on language model agent. *ArXiv preprint*, abs/2402.17785, 2024. URL https://arxiv.org/abs/2402.17785.

Jonghan Lim, Birgit Vogel-Heuser, and Ilya Kovalenko. Large language model-enabled multi-agent manufacturing systems. *ArXiv preprint*, abs/2406.01893, 2024. URL https://arxiv.org/abs/2406.01893.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. *ArXiv preprint*, abs/2401.17377, 2024. URL https://arxiv.org/abs/2401.17377.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint*, abs/1907.11692, 2019. URL https://arxiv.org/abs/1907.11692.

Liane Makatura, Michael Foshey, Bohan Wang, Felix Hähnlein, Pingchuan Ma, Bolei Deng, Megan Tjandrasuwita, Andrew Spielberg, Crystal Elaine Owens, Peter Yichen Chen, Allan Zhao, Amy Zhu, Wil J. Norton, Edward Gu, Joshua Jacob, Yifei Li, Adriana Schulz, and Wojciech Matusik. Large Language Models for Design and Manufacturing. *An MIT Exploration of Generative AI*, 2024. https://mit-genai.pubpub.org/pub/nmypmnhs.

William Merrill, Noah A Smith, and Yanai Elazar. Evaluating n-gram novelty of language models using rusty-dawg. *ArXiv preprint*, abs/2406.13069, 2024. URL https://arxiv.org/abs/2406.13069.

Niloofar Mireshghallah, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. Smaller language models are better zero-shot machine-generated text detectors. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 278–293. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.eacl-short.25.

Piotr Mirowski, Kory W. Mathewson, Jaylen Pittman, and Richard Evans. Co-writing screenplays and theatre scripts with language models: Evaluation by industry professionals. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI 2023, Hamburg, Germany, April 23-28, 2023*, pp. 355:1–355:34. ACM, 2023a. URL https://doi.org/10.1145/3544548.3581225.

Piotr Mirowski, Kory W. Mathewson, Jaylen Pittman, and Richard Evans. Co-writing screenplays and theatre scripts with language models: Evaluation by industry professionals. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI 2023, Hamburg, Germany, April 23-28, 2023*, pp. 355:1–355:34. ACM, 2023b. URL https://doi.org/10.1145/3544548.3581225.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24950–24962. PMLR, 2023a. URL https://proceedings.mlr.press/v202/mitchell23a.html.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24950–24962. PMLR, 2023b. URL https://proceedings.mlr.press/v202/mitchell23a.html.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807. Association for Computational Linguistics, 2018. URL https://aclanthology.org/D18-1206.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14:265–294, 1978. URL https://api.semanticscholar.org/CorpusID:206800425.

Timothy Nguyen. Understanding transformers via n-gram statistics. *ArXiv preprint*, abs/2407.12034, 2024. URL https://arxiv.org/abs/2407.12034.

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. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774, 2023. URL https://arxiv.org/abs/2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155, 2022. URL https://api.semanticscholar.org/CorpusID:246426909.

Scott Paeth. 2 human creativity and a doctrine of creation.

Srishti Palani, David Ledo, George W. Fitzmaurice, and Fraser Anderson. "i don't want to feel like i'm working in a 1960s factory": The practitioner perspective on creativity support tool adoption. In *CHI '22: CHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA, 29 April 2022 - 5 May 2022*, pp. 379:1–379:18. ACM, 2022. URL https://doi.org/10.1145/3491102.3501933.

PleIAs. Pleias/common$_{c}orpusdatasetsathuggingface.URL.$

Stephen E. Robertson and Steve Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994. URL https://api.semanticscholar.org/CorpusID:2218552.

Piotr Sawicki, Marek Grze´s, Fabrício Góes, Daniel Brown, Max Peeperkorn, Aisha Khatun, and Simona Paraskevopoulou. On the power of special-purpose gpt models to create and evaluate new poetry in old styles. URL https://api.semanticscholar.org/CorpusID:269589029.

Piotr Sawicki, Marek Grzes, Fabrício Góes, Daniel Brown, Max Peeperkorn, and Aisha Khatun. Bits of grass: Does gpt already know how to write like whitman? *ArXiv preprint*, abs/2305.11064, 2023. URL https://arxiv.org/abs/2305.11064.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=zWqr3MQuNs.

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. Release strategies and the social impacts of language models, 2019. URL https://arxiv.org/abs/1908.09203.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Raghavi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, A. Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Daniel Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke S. Zettlemoyer, Noah A. Smith, Hanna Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research. *ArXiv*, abs/2402.00159, 2024. URL https://api.semanticscholar.org/CorpusID:267364861.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024. URL https://arxiv.org/abs/2403.08295.

E Paul Torrance. Torrance tests of creative thinking. *Educational and psychological measurement*, 1966.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288, 2023. URL https://arxiv.org/abs/2307.09288.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8384–8395. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.emnlp-main.673.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2001–2016. Association for Computational Linguistics, 2021. URL https://aclanthology.org/2021.findings-emnlp.172.

Maryam Vaezi and Saeed Rezaei. Development of a rubric for evaluating creative writing: a multi-phase research. *New Writing*, 16:303 – 317, 2018. URL https://api.semanticscholar.org/CorpusID:149864198.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1702–1717. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.naacl-long.95.

Hong Wang, Xuan Luo, Weizhi Wang, and Xifeng Yan. Bot or human? detecting chatgpt imposters with a single question, 2023. URL https://arxiv.org/abs/2305.06424.

Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. Naturalprover: Grounded mathematical proof generation with language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/1fc548a8243ad06616eee731e0572927-Abstract-Conference.html.

Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. Re3: Generating longer stories with recursive reprompting and revision. In *Proceedings of the 2022 Conference on Empirical Methods* *in Natural Language Processing*, pp. 4393–4479. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2022.emnlp-main.296.

Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. Wordcraft: Story writing with large language models. *Proceedings of the 27th International Conference on Intelligent User Interfaces*, 2022. URL https://api.semanticscholar.org/CorpusID:247585187.

Zachary Ziegler, Yuntian Deng, and Alexander Rush. Neural linguistic steganography. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1210–1215, Hong Kong, China, November 2019. Association for Computational Linguistics. 10.18653/v1/D19-1115. URL https://aclanthology.org/D19-1115.

A Method Details

A.1 Implementation Details of DJ Search

As discussed in the main text, the deployment of the Creativity Index relies on efficiently determining whether each n-gram $\mathbf{x}_{i+i+n} \in \mathbf{x}$ can be found anywhere in the massive reference corpus C of publicly available texts. The function $f(\mathbf{x}_{i+i+n}, C)$ is a binary indicator that determines whether an n-gram $\mathbf{x}_{i:i+n}$ occurs in C. In line with the definition of Creativity Index, we only consider the n-grams $\mathbf{x}_{i+i+n}$ such that $n \geq L$ for some fixed constant L.

While a naive approach to checking whether $\mathbf{x}_{i+i+n}$ appears in C for every n-gram $\mathbf{x}_{i+i+n} \in \mathbf{x}$ would take $O(|\mathbf{x}|^2)$ calls[3] to f (see Algorithm 1), using a two-pointer approach we can radically reduce this to $O(|\mathbf{x}|)$ calls (see Algorithm 2). Note that a two-pointer approach does $O(|\mathbf{x}|)$ calls to f since in each iteration we advance at least one of the two pointers i and j by 1, and $0 \leq i, j \leq |\mathbf{x}|$.

Algorithm 1 Naive Computation

NGramsFound$_{i,j} \leftarrow$ False $\quad \forall\, i \in [0..|\mathbf{x}|)$ and $j \in [0..|\mathbf{x}|)$ $\quad \triangleright$ matrix to store n-gram occurrence
for $i \in [0, 1, ..., |\mathbf{x}| - L)$ **do**
  for $j \in [i + L, ..., |\mathbf{x}|)$ **do**
    NGramsFound$(i, j) \leftarrow f(\mathbf{x}_{i:j}, C)$
  end for
end for
return NGramsFound

Algorithm 2 Efficient computation of DJ Search$(\mathbf{x}, C)$

NGramsFound$_{i,j} \leftarrow$ False $\quad \forall\, i \in [0..|\mathbf{x}|)$ and $j \in [0..|\mathbf{x}|)$ $\quad \triangleright$ matrix to store n-gram occurrence
$i \leftarrow 0, j \leftarrow L$
while $j < |\mathbf{x}|$ **do**
  NGramsFound$(i, j) = f(\mathbf{x}_{i:j}, C)$
  if NGramsFound(i, j) **then**
    $j \leftarrow j + 1$ $\quad \triangleright$ we will search for $\mathbf{x}_{i:j+1}$ next
  else
    $i \leftarrow i + 1$ $\quad \triangleright$ since $\mathbf{x}_{i:j}$ was not found, $\mathbf{x}_{i:j+k}$ will not be found for all $k > 0$
    $j \leftarrow \max(i + L, j)$ $\quad \triangleright$ we only explore L-grams and beyond
  end if
end while
return NGramsFound

A.2 Implementation Details of Word Mover's Distance

Let w be an n-gram. Let $f(w, C)$ be the function that determines whether w appears in any text $\mathbf{d} \in C$, either exactly or as a phrase that is highly similar in meaning to w (e.g., a paraphrase of w). Trivially, $f(w, C) := \bigcup_{\mathbf{d} \in C} f(w, \mathbf{d})$, and here on we will only discuss how to compute $f(w, \mathbf{d})$.

An established approach for finding semantically similar phrases to a given n-gram w is to compute its embedding—embedding(w)—and then independently compute its similarity to the embeddings of all other n-grams to be analyzed. An *embedding* of a n-gram is a vector that represents the meaning of such n-gram in an k-th dimensional space of fixed size, enabling the comparison of similarity between concepts expressed in different surface forms. This comparison is typically done using *cosine similarity*, the scaled dot product between the two embeddings being compared. Text embeddings are generated by models specifically trained to this effect (e.g., BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), SpanBERT (Joshi et al., 2020)) making their computation expensive at a large scale. Notably, text embeddings usually do not possess linearity, i.e. the embedding of concatenating n-grams w and v cannot be deduced from knowing embedding(w) and embedding(v), and instead needs to be computed from scratch.

[3] There are $(|\mathbf{x}| - L)(|\mathbf{x}| - L + 3)/2$ spans to analyze if L is the minimum n-gram length to be considered.

Since our goal is to find the n-grams of $\mathbf{d}$ that are highly similar to w, using the traditional approach would entail comparing embedding(w) with the embeddings of all n-grams in C, which are approximately $\sum_{d \in C} |d|^2$ in number. Note that this also implies independently computing $\approx \sum_{d \in C} |d|^2$ embeddings, which increases the computation costs significantly. Instead we use Word Mover's Distance (Kusner et al., 2015) (WMD), a method to estimate similarity between two n-grams by combining comparisons between pairs of ***word*** embeddings. This enables lifting the requirement to independently computing the embedding for each n-gram in C. Concretely, the Word Movers' Distance between two n-grams w and v is defined as follows:

$$\begin{aligned}
\mathrm{D}_{w \to v} &:= \frac{1}{|w|} \sum_{i \in [0..|w|)} \min_{j \in [0..|v|)} 1 - \text{cosine_similarity}(\text{embedding}(v_j), \text{embedding}(w_i)) \\
&= 1 - \frac{1}{|w|} \sum_{i \in [0..|w|)} \max_{j \in [0..|v|)} \text{cosine_similarity}(\text{embedding}(v_j), \text{embedding}(w_i)) \\
\mathrm{WMD}(w, v) &:= \max(\mathrm{D}_{w \to v}, \mathrm{D}_{v \to w})
\end{aligned}$$

WMD also pre-filters the words considered in w and v to only include the *content words* in the analysis (i.e, discards *stop-words*, such as *the, a, an, it, on, ...*).

Note that $\mathrm{D}_{w \to v}$'s definition is asymmetric ($\mathrm{D}_{w \to v} \neq \mathrm{D}_{v \to w}$). Thus, we consider the Word Movers' Distance of two n-grams w and v as the maximum of $\mathrm{D}_{w \to v}$ and $\mathrm{D}_{v \to w}$: w and v are highly similar if their distance is below a threshold δ for both $\mathrm{D}_{w \to v}$ and $\mathrm{D}_{v \to w}$ (See Algorithm 3):

$$\mathrm{WMD}(w, v) = \max(\mathrm{D}_{w \to v}, \mathrm{D}_{v \to w}) < \delta$$

Algorithm 3 Conceptual writeup of $f(w, \mathbf{d})$ using Word Mover Distance (WMD) to find the n-grams of a single text $\mathbf{d} \in C$ that are highly similar to the n-gram w and are of length $\geq L$.

procedure DIRECTIONALWMD(w, v)
  return 1 - $\frac{1}{|v|} \sum_{j \in [0..|v|)} \max_{i \in [0..|w|)}$ cosine_similarity(embedding(w_i), embedding(v_j))
end procedure
for $a \in [0, 1, ..., |\mathbf{d}|)$ **do**
  for $b \in [a + L, ..., |\mathbf{d}|]$ **do**
    symmetricWMD $\leftarrow \max(\text{directionalWMD}(\mathbf{d}[a:b], w), \text{directionalWMD}(w, \mathbf{d}[a:b]))$
    if symmetricWMD $< \delta$ **then**
      return True
    end if
  end for
end for
return False

Avid readers may notice that Algorithm 3 repeatedly computes the maximum over the same set, and sums of contiguous similarity scores; these can be pre-computed. Algorithm 4 shows these optimizations, resulting in an algorithm of time complexity $O(|d| \cdot |w| + |d|^2|w|) = O(|d|^2|w|)$, assuming already computed word embeddings. Note that because there is a fixed vocabulary, all word embeddings as well as cosine similarities of word embedding pairs can be pre-computed.

We described how to compute $f(w, \mathbf{d})$ for a single document $\mathbf{d} \in C$, as we have already established that $f(w, C) = \bigcup_{\mathbf{d} \in C} f(w, \mathbf{d})$. To accelerate computation, and given that similar n-grams to $\mathbf{x}_{i:i+n}$ are more likely to occur in texts similar to $\mathbf{x}$, we select C's top most likely documents to contain w using a BM25Robertson & Walker (1994) index, denoted C'. We then approximate $f(w, C) \approx \bigcup_{\mathbf{d} \in C'} f(w, \mathbf{d})$.

As a final optimization, we note that it is unnecessary to compute the costly $f(w, C)$ for finding semantically similar matches for w in the case where w appears exactly in C. To check if w appears exactly in C, we can leverage the existing, less expensive approach Infini-Gram (Liu et al., 2024) and search for the semantic similar matches only if Infini-Gram could not find any exact matches.

Algorithm 4 Efficient Computation of $f(w, \mathbf{d})$ (optimization of Algorithm 3)

```
token_similarity_{i,j} ← cosine_similarity(embedding(w_i), embedding(d_j))   ∀ i ∈ [0..|w|) and j ∈ [0..|d|)
for j ∈ [1, ..., |d|] do
    doc_prefix_similarity_j ← doc_prefix_similarity_{j-1} + max_{i∈[0..|w|)} token_similarity_{i,j-1}
end for

for a ∈ [0, 1, ..., |d|) do
    for b ∈ [a + L, ..., |d|] do
        computed_WMD(d[a : b), w) ← 1 − (doc_prefix_similarity_b − doc_prefix_similarity_a)/(b − a)
        computed_WMD(w, d[a : b)) ← 1 − (1/|w|) Σ_{i∈[0..|w|)} max_{j∈[a..b)} token_similarity_{i,j}
        symmetric_WMD ← max(computed_WMD(d[a : b), w), computed_WMD(w, d[a : b)))
        if symmetric_WMD < δ then
            return True
        end if
    end for
end for
return False
```

A.3 Deduplication of the Reference Corpus

When a text **x** is part of the reference corpus C, its CREATIVITY INDEX would trivially become zero. This issue often arises when analyzing the works of famous authors, as their writings are frequently copied, quoted, or cited online. To address this, when analyzing human texts written before the cutoff date of the reference corpus, we exclude any document $\mathbf{d} \in C$ that contains copies, quotations, or citations of the text **x** from the reference corpus C, and compute CREATIVITY INDEX of **x** using this filtered reference corpus.

Specifically, we measure the degree of overlap between **x** and **d** by calculating the length of the longest common subsequence (LCS) between them, normalized by the length of **x**. Formally, $S(\mathbf{x},\mathbf{d}) = \frac{||\text{LCS}(\mathbf{x},\mathbf{d})||}{||\mathbf{x}||}$. If **x** and **d** have a high degree of overlap (i.e., $S(\mathbf{x}, \mathbf{d}) \geq \alpha$), it's very likely that **d** contains an exact copy of **x**. If **x** and **d** show a moderate amount of overlap (i.e., $\beta \leq S(\mathbf{x}, \mathbf{d}) < \alpha$), we prompt a LLM to determine whether **d** contains copies or quotations of **x** using in-context examples provided below. Additionally, if **d** includes the author name or title of **x**, it is highly likely that **d** contains a citation of **x**. In practice, we set the values of α and β to 0.9 and 0.3, respectively, and use LLaMA 2 Chat as the LLM to check for copies and quotations.

```
Please check if paragraph A contains any copies or quotations from
paragraph B.

Here are some examples:
Paragraph A: In the end though, I did the required reading, complained
bitterly about being bored, wrote the requisite essay, and promptly
forgot all about it.  "He was an old man who fished alone in a skiff
in the Gulf Stream and he had gone eighty-four days now without taking
a fish.  In the first forty days ...
Paragraph B: He was an old man who fished alone in a skiff in the Gulf
Stream and he had gone eighty-four days now without taking a fish.  In
the first forty days a boy had been with him.  But after forty days
without a fish the boy's parents had told him that the old man was now
definitely and finally salao ...
Answer:  Yes


Paragraph A: He was an old man who fished alone in a lobster boat off the
Maine coast and he had gone 117 days without taking a crustacean.  His
luck was not bad, rather his judgment was good (don't fish the Atlantic
in winter).  Then he met us and for all I know his luck changed.  El
Campion is due for a change of luck ...
Paragraph B: He was an old man who fished alone in a skiff in the Gulf
Stream and he had gone eighty-four days now without taking a fish.  In
the first forty days a boy had been with him.  But after forty days
without a fish the boy's parents had told him that the old man was now
definitely and finally salao ...
Answer:  No


Paragraph A: Santiago, the "old man who fished alone," in Hemingway's
"The Old Man and the Sea" appears as one who has an undefeatable
character, a loving, cheerful character, and very humble.  The writer
describes him in this way:  "Everything about him was old except his
eyes, and they were the same color as the sea ...
Paragraph B: He was an old man who fished alone in a skiff in the Gulf
Stream and he had gone eighty-four days now without taking a fish.  In
the first forty days a boy had been with him.  But after forty days
without a fish the boy's parents had told him that the old man was now
definitely and finally salao ...
Answer:  Yes


Paragraph A: He was an old man who could see the form of his god, and
a monk, moreover.  Izzie had limited ability to communicate directly with
her own deity.  Much of her life she had proceeded by vague impressions
and only glimpsed the great god's image briefly in the depths of
meditation ...
Paragraph B: He was an old man who fished alone in a skiff in the Gulf
Stream and he had gone eighty-four days now without taking a fish.  In
the first forty days a boy had been with him.  But after forty days
without a fish the boy's parents had told him that the old man was now
definitely and finally salao ...
Answer:  No


Here is the test example:
Paragraph A: [A]
Paragraph B: [B]
Answer:
```

B EVALUATION

B.1 MACHINE TEXT GENERATION

We experiment with state-of-the-art LLMs: GPT-3 (Brown et al., 2020) (`text-davinci-003`), ChatGPT (Ouyang et al., 2022) (`gpt-3.5-turbo`), LLaMA 2 Chat (Touvron et al., 2023), Tulu 2 (Ivison et al., 2023) and OLMo Instruct (Groeneveld et al., 2024) along with their base model before RLHF: GPT Base (Brown et al., 2020) (`davinci-002`), LLaMA 2 Base (Touvron et al., 2023) and OLMo Base (Groeneveld et al., 2024). These models are primarily pre-trained on the web data available before the cutoff date of the reference corpus RedPajama (Computer, 2023). We additionally discuss how to handle newer models, such as GPT-4 (OpenAI et al., 2023), which are largely trained on more recent web data and third-party private data, both of which fall outside the reference corpus RedPajama.

To obtain machine texts, we prompt LLMs to generate several paragraphs of novels, poems, or speeches, starting with an initial sentence taken from existing human writings in each category. To construct test data for machine text detection, we further prompt LLMs to generate news articles based on the fake news headlines from the Misinfo Reaction Frames (Gabriel et al., 2022) and to generate theorem proofs for questions from the NaturalProofs (Welleck et al., 2022) benchmark. The prompts used for each task are illustrated below. For all generations, we use nucleus sampling with $p = 0.9$ and set the maximum length of the generated texts to 288 tokens.

```
Please write a few paragraphs for a novel starting with the following
prompt:  [PROMPT SENTENCE]
```

```
Please write a poem starting with the following line:  [PROMPT LINE]
```

```
Please write a speech starting with the following sentence:  [PROMPT
SENTENCE]
```

```
Please write a news article based on the given headline:  [NEWS
HEADLINE]
```

```
Please provide a proof for the following theorem:  [THEOREM QUESTION]
```

To obtain model-generated reference corpus to compare the CREATIVITY INDEX of GPT-4 with humans, we randomly sample 150k sentences from the RedPajama corpus and prompt open-weight LLMs with knowledge cutoff dates similar to GPT-4 to generate document-level continuations. The models we use are the instruction-aligned versions of Gemma-7B (Team et al., 2024) (`gemma-7b-it`), Llama3-8B (`Meta-Llama-3-8B`) (AI@Meta, 2024), and Mixtral-7B (`Mistral-7B-v0.1`) (Jiang et al., 2023). The prompt used to generate continuations is illustrated below. We use nucleus sampling with $p = 0.9$ and set the maximum length of the generated texts to 2048 tokens.

```
Please generate a continuation for the following sentence:  [PROMPT
SENTENCE]
```

We additionally experiment with different prompt formats, intentionally encouraging creativity in models' generations by incorporating instructions such as 'push for creative ideas, unique emotions, and original twists,' 'be bold and creative,' or 'you are a creative writer.' Please see blow for a full list of the prompts we tried.

```
Write a few paragraphs for a novel from the following prompt, pushing
for creative ideas, unique emotions, and original twists.
Prompt:  [PROMPT SENTENCE]
```

```
Use the following prompt to write a few paragraphs for a novel with
creative, unqiue perspectives or twists. Let your originality shine.
Prompt: [PROMPT SENTENCE]
```

```
Create a few paragraphs from the following prompt for a novel, focusing
on novel ideas, emotions, or perspectives. Be as creative as possible.
Prompt: [PROMPT SENTENCE]
```

```
Write a few paragraphs for a novel based on the following prompt,
exploring unexpected twists, emotions, or unique perspectives. Be bold
and creative.
Prompt: [PROMPT SENTENCE]
```

```
Based on the following prompt, and write a few paragraphs for a novel
that explore unexpected twists, deep emotions, or unique perspectives.
Let your creativity flow, and don't be afraid to experiment with
unconventional ideas or characters
Prompt: [PROMPT SENTENCE]
```

```
As a creative agent, write a few paragraphs for a novel based on the
following prompt, bringing your novel ideas and original emotions to
life.
Prompt: [PROMPT SENTENCE]
```

```
You are a creative writer, write a few paragraphs for a novel based
on the following prompt. Explore unique perspectives and unexpected
twists, and let your creativity guide you.
Prompt: [PROMPT SENTENCE]
```

```
You are a creative agent, free to shape this story in any direction.
Write a few paragraphs for a novel based on the following prompt, using
your imagination to uncover surprises and depth.
Prompt: [PROMPT SENTENCE]
```

```
As a creative writer, your task is to write a few paragraphs for a
novel based on the following prompt. Dive into original ideas, explore
emotions, and surprise yourself.
Prompt: [PROMPT SENTENCE]
```

```
You are a creative writer who brings stories to life. Write a few
paragraphs for a novel based on the following prompt, letting your
imagination take bold, unexpected turns.
Prompt: [PROMPT SENTENCE]
```

B.2 Dataset Details

Reference Corpus: We use RedPajama (Computer, 2023), the largest web data collection available at the time of this study, as our reference corpus. RedPajama contains 100 billion text documents with 100+ trillion raw tokens from 84 CommonCrawl dumps.

Novel: For human-written novels, we use book snippets from the BookMIA (Shi et al., 2024) dataset. The BookMIA dataset contains approximately 10k book snippets, with an average length of around 650 words per snippet. We randomly sample 100 book snippets from the BookMIA dataset and select the first K sentences of each snippet such that their total length exceeds 256 words, to use as human text. Since novels we use were published after the cutoff date of RedPajama, there's no need for deduplication before DJ Search.

Speech: For the transcripts of human speeches, we randomly sample 100 speeches from the famous speeches available in the American Rhetoric speech bank. For each speech, we randomly sample continuous K sentences such that their total length exceeds 256 words, to use as human text. Since these speeches were made before the cutoff date of RedPajama, deduplication is needed before DJ SEARCH.

Poem: For human-written poems, we randomly sample 100 poems from the popular modern poems collected by PoemHunter.com. Since these poems were published before the cutoff date of RedPajama, deduplication is needed before DJ SEARCH.

News Article: We use news articles from the XSum (Narayan et al., 2018) dataset as the human text for the machine text detection task. The Xsum dataset contains around 200k new articles, with an average length of around 380 words per article. We randomly sample 500 articles to use as human text. Since these news articles were released before the cutoff date of RedPajama, deduplication is needed before DJ SEARCH. For machine-generated fake news, we randomly sample 500 fake news headlines from the Misinfo Reaction Frames (Gabriel et al., 2022), and based on these headlines, LLMs are asked to generate corresponding news articles.

Theorem Proof: We use the ground-truth human-written proofs from the NaturalProofs (Welleck et al., 2022) dataset as the human text for the machine text detection task. The NaturalProofs dataset contains approximately 24k theorems and their corresponding proofs. We randomly sample 500 theorem-proof pairs and use the ground-truth proofs as human text. Since the NaturalProofs dataset was curated after the cutoff date of RedPajama, there's no need for deduplication before DJ SEARCH. For machine-generated math proofs, we prompt LLMs to write proofs for the 500 theorems we sampled.

B.3 PARAMETERS OF DJ SEARCH

We set the minimum n-gram length L in DJ SEARCH to 5, and set the threshold for Word Mover's Distance to 0.95 for semantic matches. We observe that the L-uniqueness is close to zero for most human and machine texts when $L \leq 5$ and close to one when $L \geq 12$. Therefore, in practice, we sum up the L-uniqueness for $5 \leq L \leq 12$ when computing CREATIVITY INDEX.

The only experiment with slightly different parameters is to compare the creativity of GPT-4 with humans. We observed that the L-uniqueness is close to one when $L \geq 7$ based on the model-generated reference corpus. Therefore, we sum up the L-uniqueness for $5 \leq L \leq 7$ when computing CREATIVITY INDEX.

C RELATED WORK

Measuring Creativity in Ideas: Measuring creative thinking and problem solving takes root in early work in psychology (Torrance, 1966), where researchers defined four pillars for creative thinking: fluency, flexibility, originality and elaboration. Crossley et al. (2016) later on developed this notion and built on it to expand this to measuring creative writing in students, where they also adopted n-gram novelty for a measure of originality. However, these prior work focus on creativity in humans, and they also do not introduce any automated metrics or measurements.

Measuring Creativity in Machine-generated Text Using Expert Annotators: Closely related to CREATIVITY INDEX is a recent line of work in the generative AI literature comparing the creativity of human writers to that of large language models in different domains such as story telling and journalism (Chakrabarty et al., 2023; 2024; Anonymous, 2024). Similar to us, the approach in this direction often involves prompting an LLM to write an original story or news article, based on some existing premise or press release, and then comparing the machine-generated text to the human-written counterparts. These works, however, take a rather subjective approach, where they define and measure creativity based on human expert annotations and whether people perceive the text to be more creative, rather than an objective quantification of novelty that we provide.

Measuring Novelty of N-grams: Finally, closely related to our work in terms of techniques is Nguyen (2024) and Merrill et al. (2024). The former attempts at finding n-gram rules that would cover and predict generations from transformer models, showing that more than 70% of the times transformers follow some pre-set patterns and rules. The latter is more similar to our work as they also measure the novelty of generated n-grams and compare it to human-written text, however they differ from us in tow major ways: (1) they only find verbatim matches, whereas we also match to approximate, semantically similar blocks of text and (2) they compute the percentage of n-grams of a certain length in a text that can be found in the reference corpus, whereas we measure how much of the text can be reconstructed by mixing and matching a vast amount of existing text snippets of varying lengths from the web.

Machine Text Detection: Detecting machine-generated text has been explored for several years using a variety of methods (Jawahar et al., 2020; Uchendu et al., 2021). Gehrmann et al. (2019) and Dugan et al. (2023) demonstrate that even humans tend to struggle to differentiate between text written by humans and machines, highlighting the need for automated detection solutions. Some approaches involve training a classifier in a supervised manner to identify machine-generated text (Bakhtin et al., 2019; Uchendu et al., 2020), while others use a zero-shot detection method (Solaiman et al., 2019; Ippolito et al., 2020). Additionally, there is research on bot detection through question answering (Wang et al., 2023; Chew & Baird, 2003). Recently, Mitchell et al. (2023b) introduced DetectGPT, a zero-shot method based on the hypothesis that texts produced by a large language model (LLM) are located at local maxima, and thus exhibit negative curvature, in the model's probability distribution. Follow-up work build on DetectGPT by making it faster (Bao et al., 2024) and proposing to use cross-detection when the target model is unknown (Mireshghallah et al., 2024).

Various strategies have been developed to detect machine-generated text in real-world settings. One notable approach is watermarking, which embeds algorithmically detectable patterns into the generated text while maintaining the quality and diversity of the language model's outputs. Initial watermarking techniques for natural language were proposed by Atallah et al. (2001) and have been adapted for neural language model outputs (Fang et al., 2017; Ziegler et al., 2019). Recent advancements include Abdelnabi & Fritz (2021) work on an adversarial watermarking transformer (AWT) for transformer-based language models. Unlike methods dependent on specific model architectures, Kirchenbauer et al. (2023) introduce a watermarking technique applicable to texts generated by any common autoregressive language model.

Application of LLMs in Creative Writing: Recent advancements have highlighted the potential of LLMs in supporting various creative writing endeavors, ranging from short stories (Yang et al., 2022) to screenplays (Mirowski et al., 2023b). Enhancing LLMs to produce text that aligns more closely with human preferences has made them adept at following user instructions, thereby turning them into valuable tools for individuals without technical expertise. This progress has boosted the commercial viability of LLMs as writing aids, which can continue a narrative, describe scenes, or offer feedback. Chung et al. (2021) conducted a review of literature on creativity support tools across various arts, leading to the development of a taxonomy that includes roles, interactions, and technologies. In contrast, Frich et al. (2019) and Palani et al. (2022) focused on how creative practitioners select new tools, highlighting their emphasis on functionality, workflow integration, and performance, and noting that personal recommendations often guide their choices. Additionally, Gero et al. (2022) created a space based on the cognitive process model of writing, influencing interface design decisions. Gero et al. (2023) further explored the social dynamics of AI in creative tasks, revealing a disconnect between writers' objectives and the support provided by computer tools.

Figure 5: Example outputs from DJ SEARCH based on both verbatim and semantic matches. We prompt LLMs to generate a few paragraphs of a novel, beginning with a first sentence taken from a human-written novel snippet.

Figure 6: Example outputs from DJ SEARCH based on both verbatim and semantic matches. We prompt LLMs to generate a few paragraphs of a novel, beginning with a first sentence taken from a human-written novel snippet.

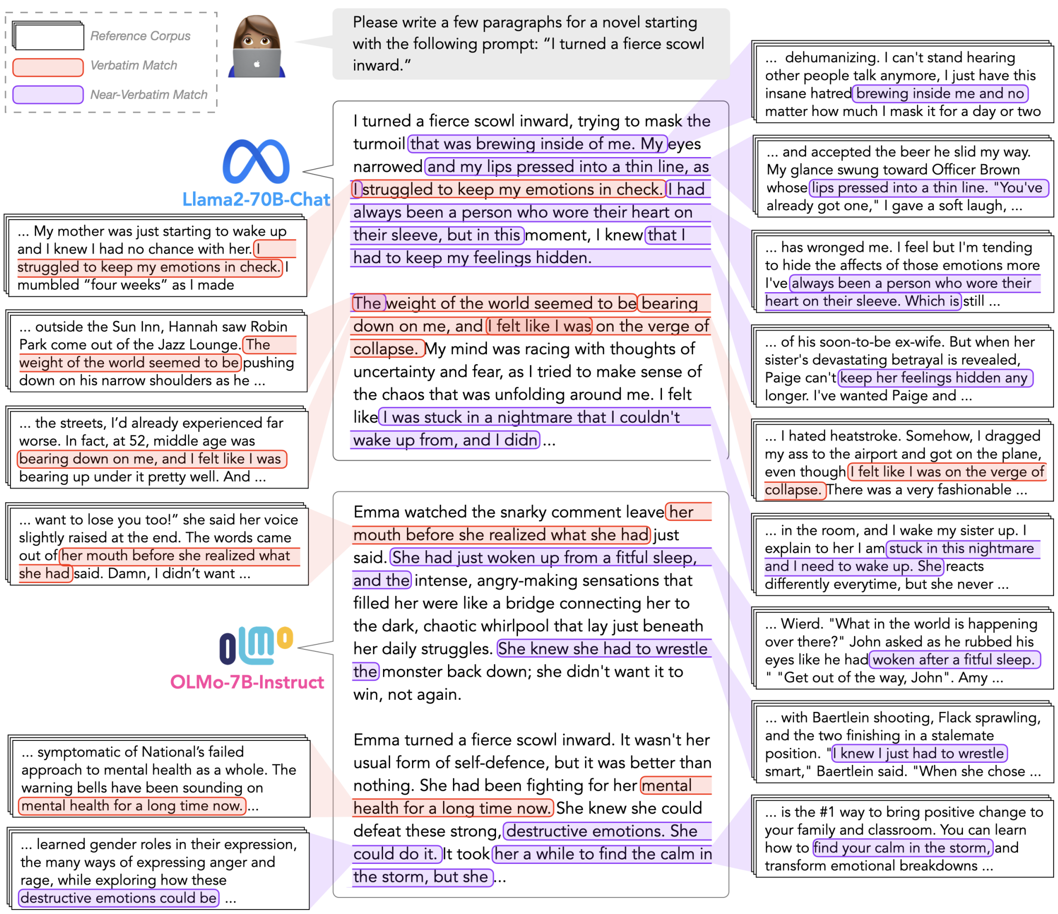


Figure 7: Example outputs from DJ SEARCH based on both verbatim and semantic matches. We prompt LLMs to generate a few paragraphs of a novel, beginning with a first sentence taken from a human-written novel snippet.

Figure 8: Example outputs from DJ SEARCH based on both verbatim and semantic matches. We prompt LLMs to generate a few paragraphs of a novel, beginning with a first sentence taken from a human-written novel snippet.

Figure 9: Example outputs from DJ SEARCH based on both verbatim and semantic matches. We prompt LLMs to generate a few paragraphs of a novel, beginning with a first sentence taken from a human-written novel snippet.

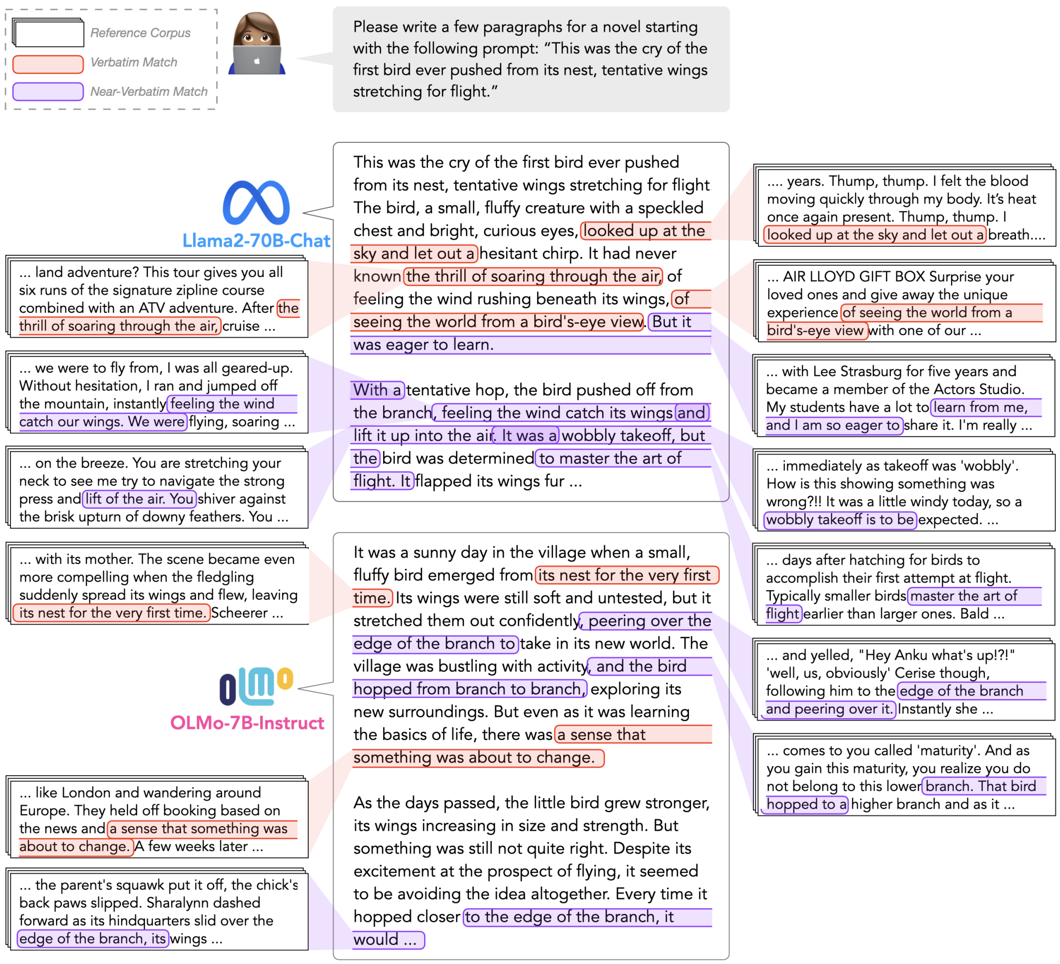


Figure 10: Example outputs from DJ SEARCH based on both verbatim and semantic matches. We prompt LLMs to generate a few paragraphs of a novel, beginning with a first sentence taken from a human-written novel snippet.

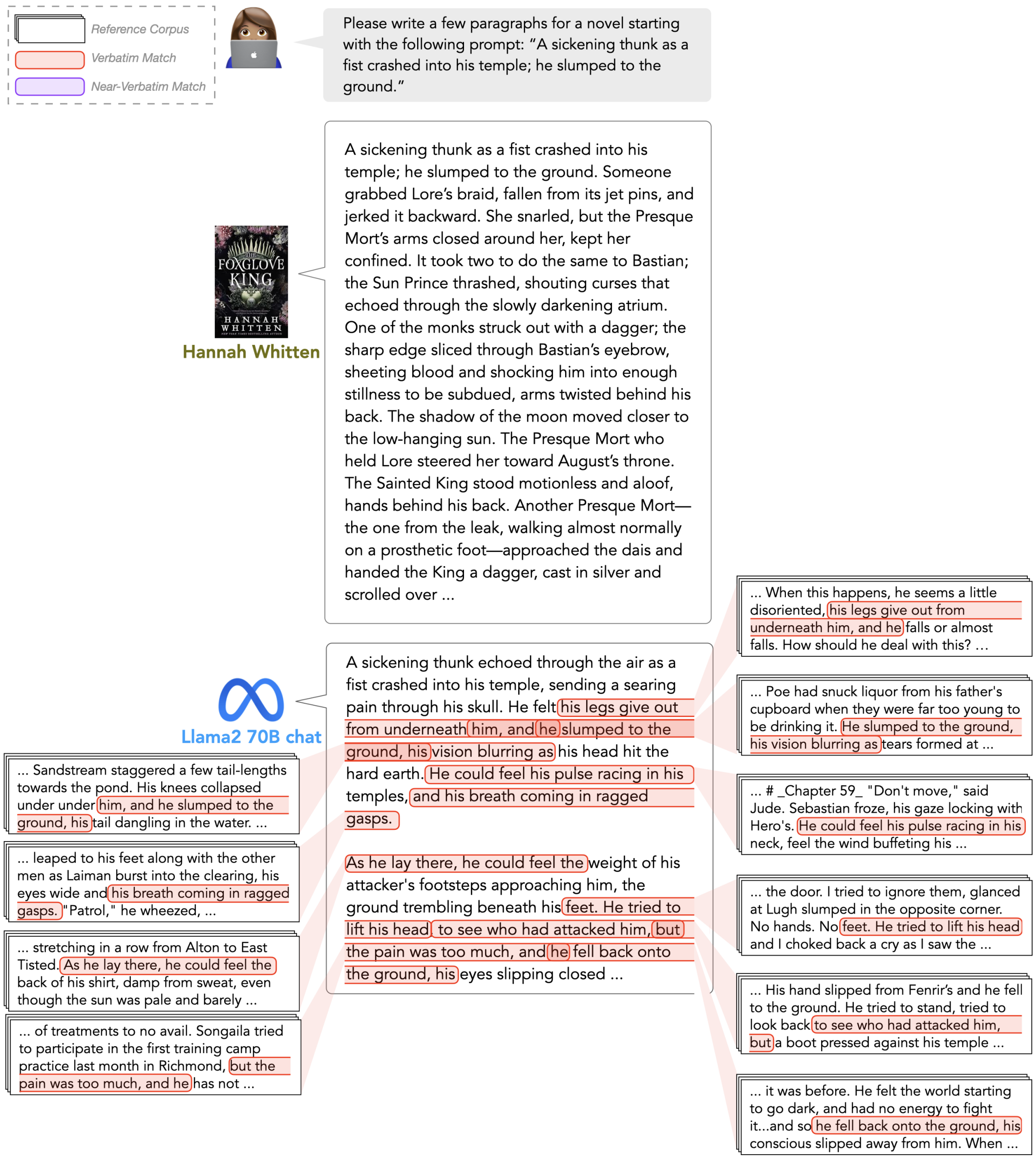


Figure 11: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a few paragraphs of a novel, beginning with a first sentence taken from a human-written novel snippet.

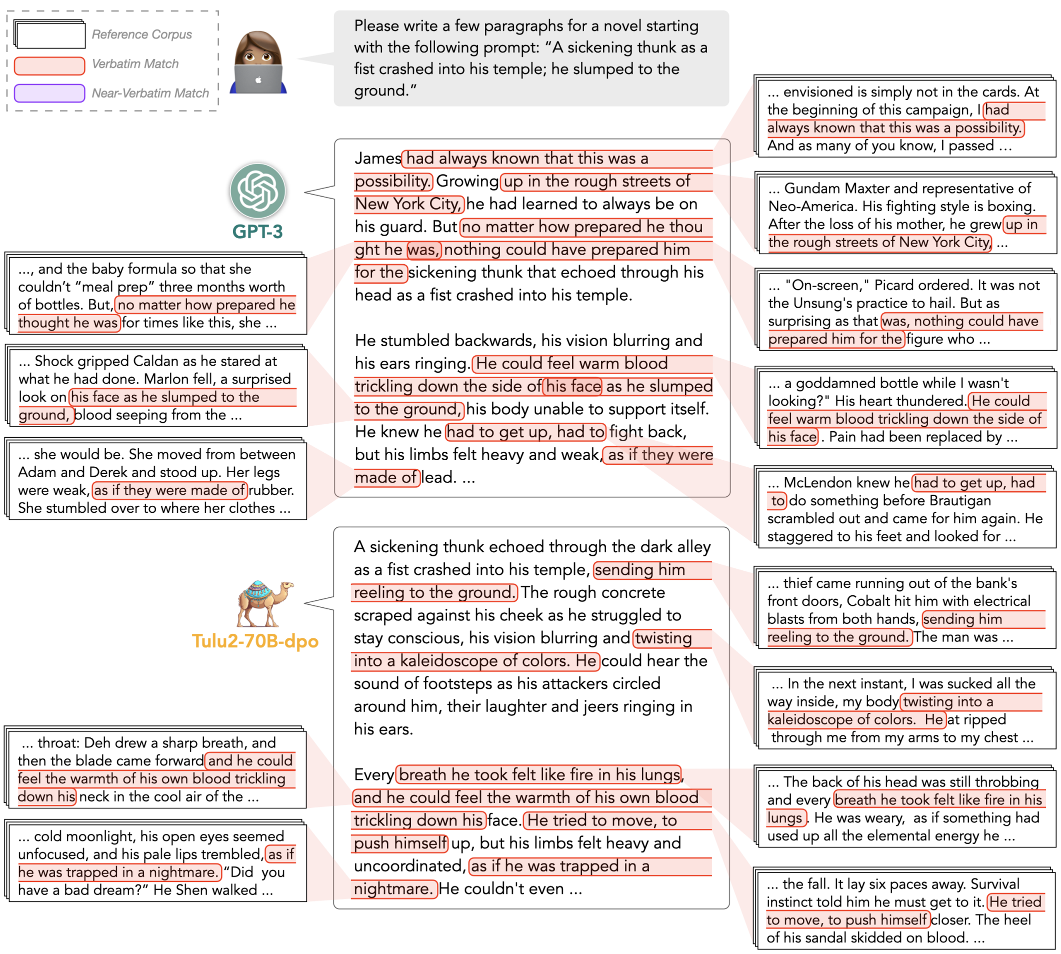


Figure 12: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a few paragraphs of a novel, beginning with a first sentence taken from a human-written novel snippet.

Figure 13: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a few paragraphs of a novel, beginning with a first sentence taken from a human-written novel snippet.

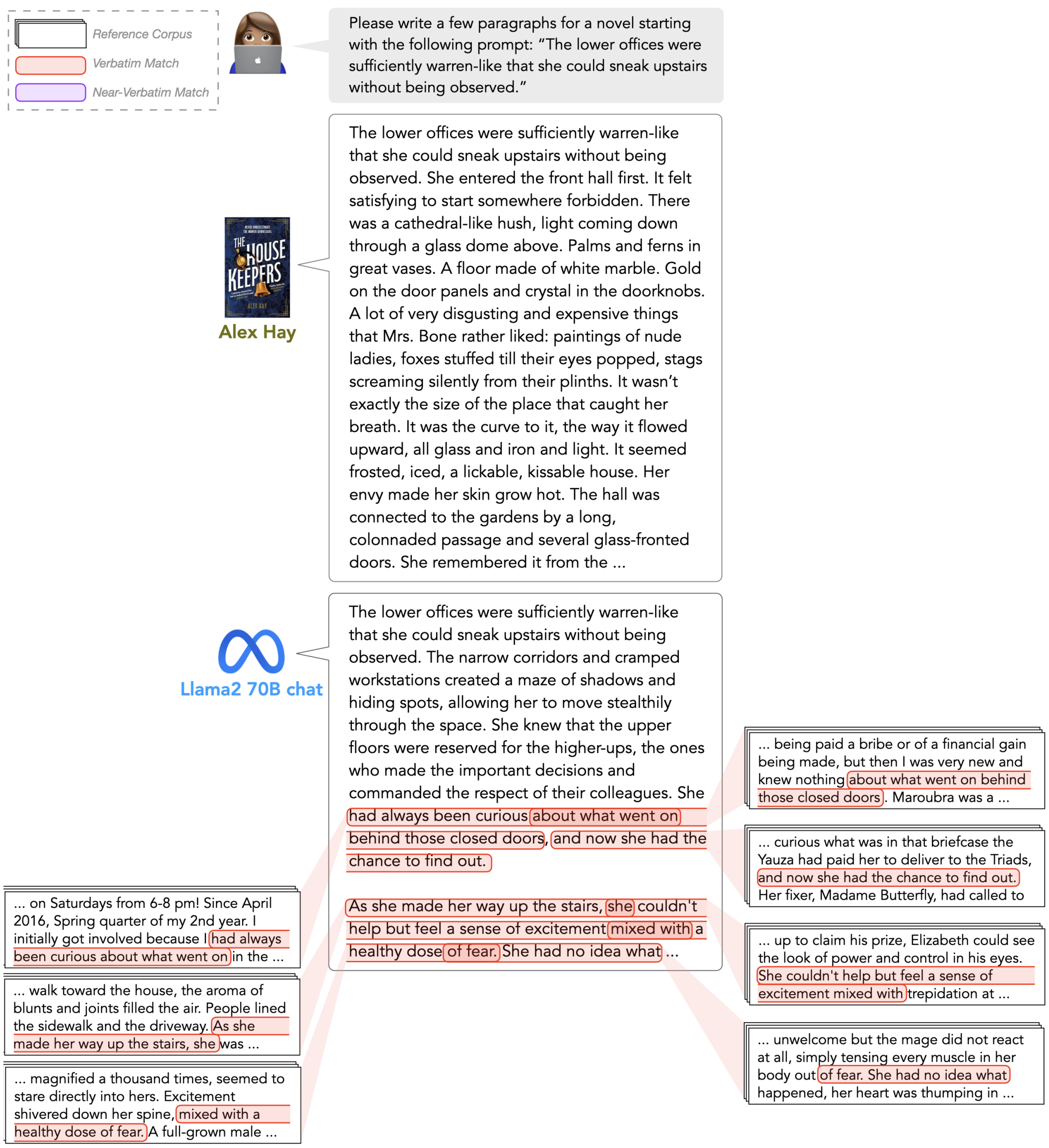


Figure 14: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a few paragraphs of a novel, beginning with a first sentence taken from a human-written novel snippet.

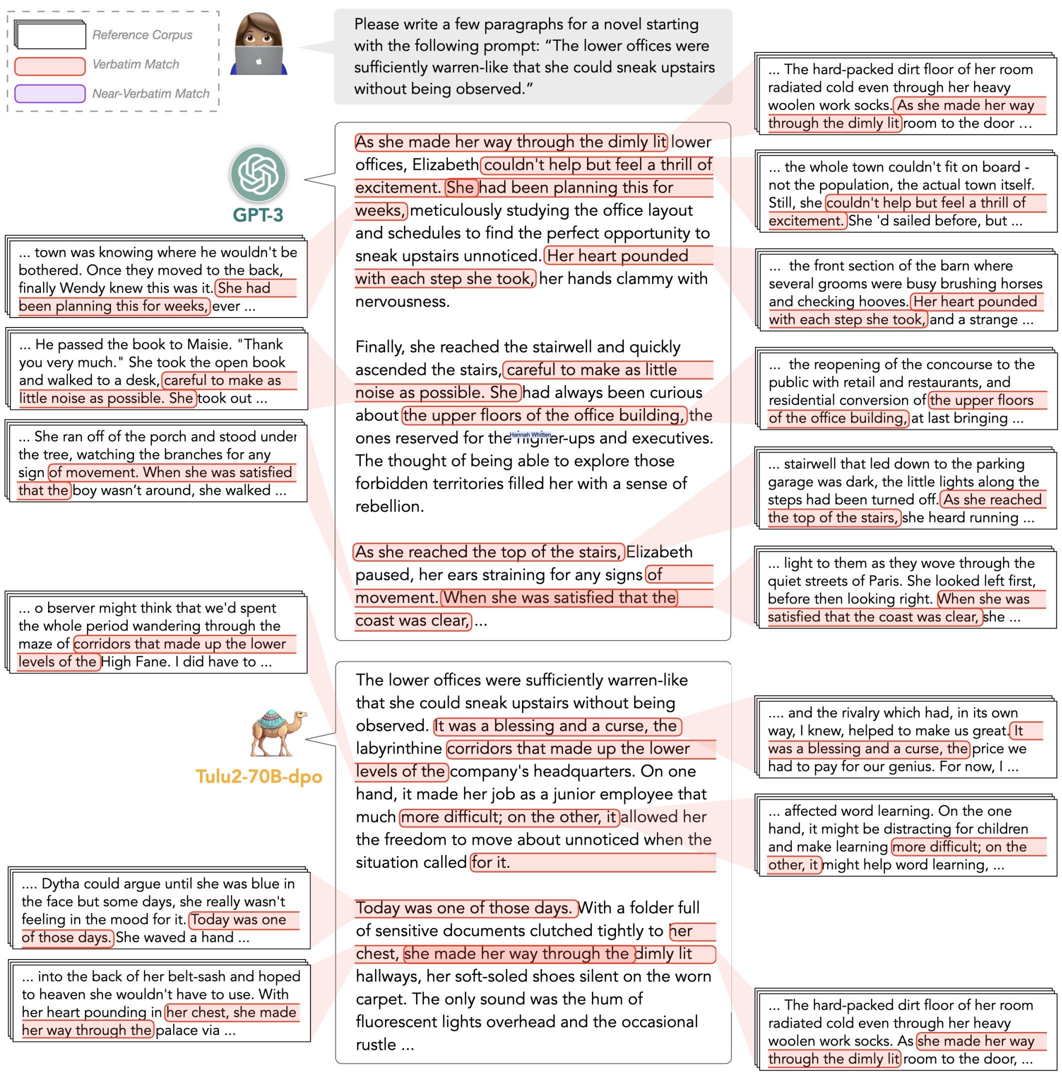


Figure 15: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a few paragraphs of a novel, beginning with a first sentence taken from a human-written novel snippet.

Figure 16: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a few paragraphs of a novel, beginning with a first sentence taken from a human-written novel snippet.

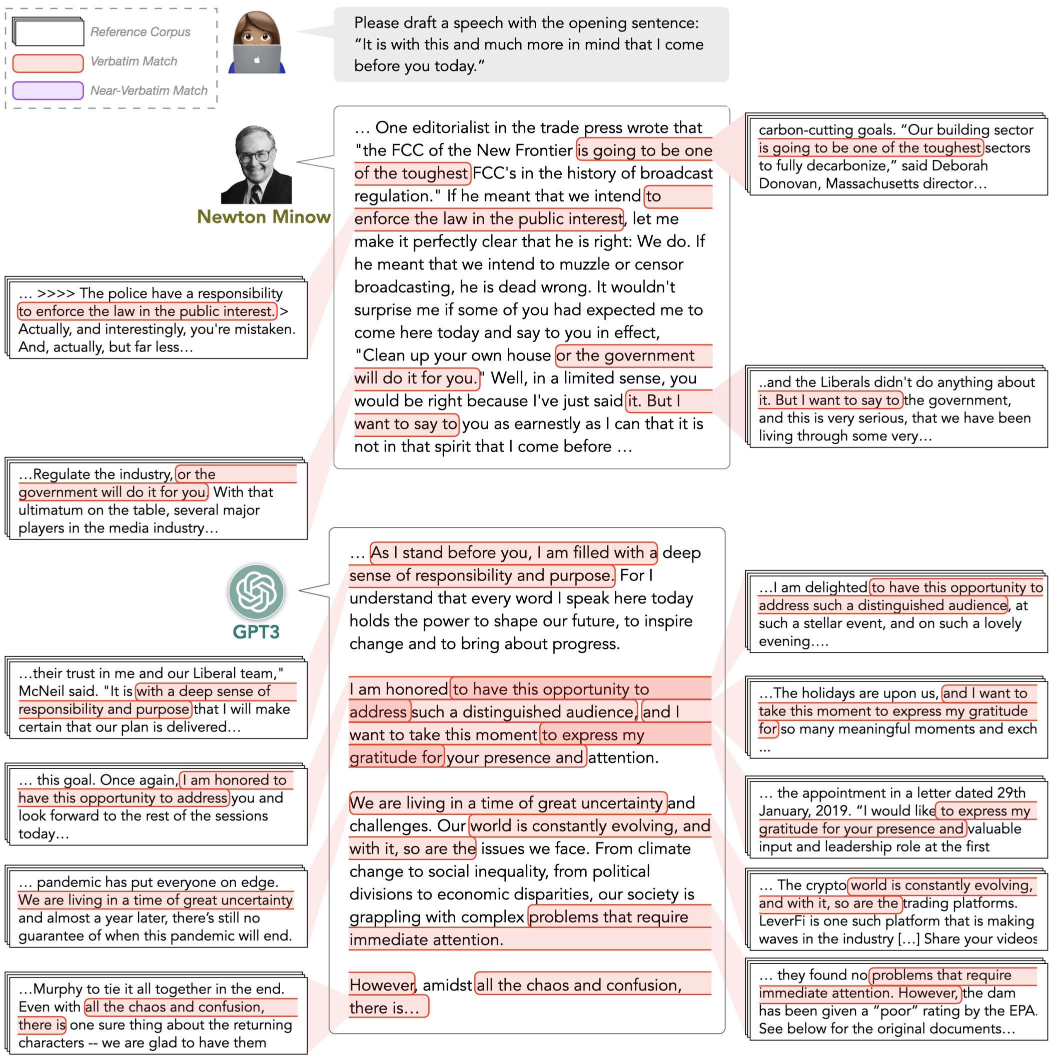


Figure 17: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a speech starting with the opening sentence of a human speech transcript.

Figure 18: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a speech starting with the opening sentence of a human speech transcript.

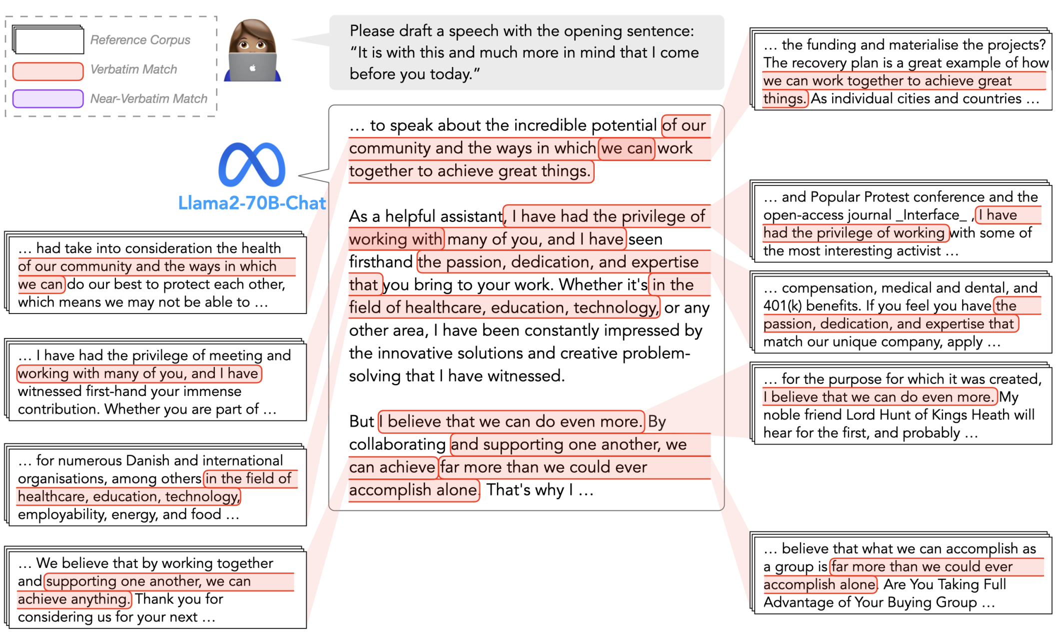


Figure 19: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a speech starting with the opening sentence of a human speech transcript.

Figure 20: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a speech starting with the opening sentence of a human speech transcript.

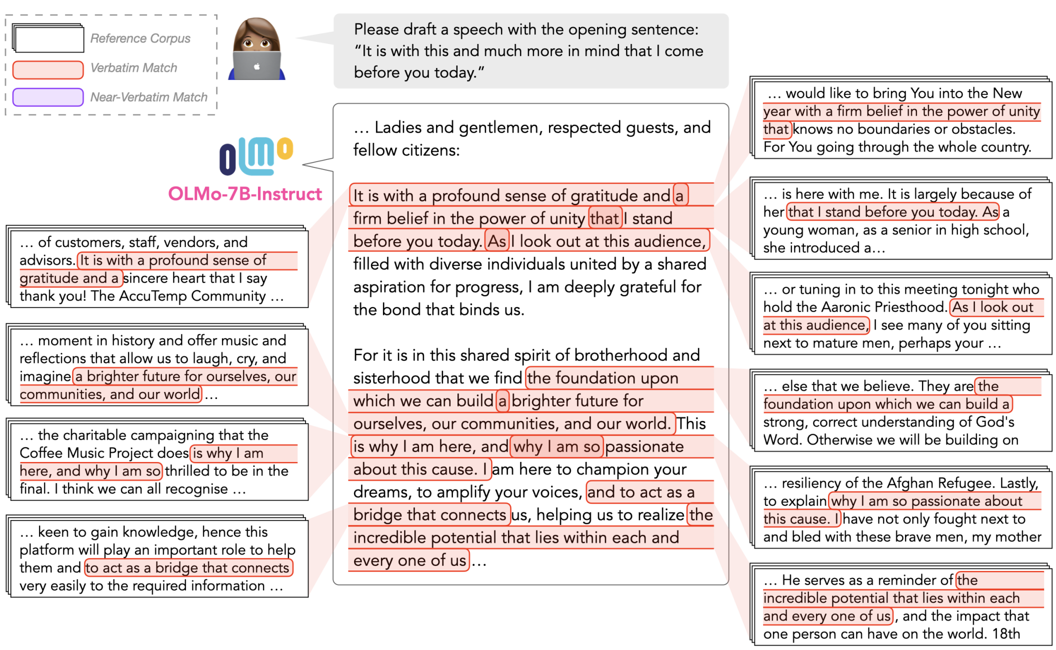


Figure 21: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a speech starting with the opening sentence of a human speech transcript.

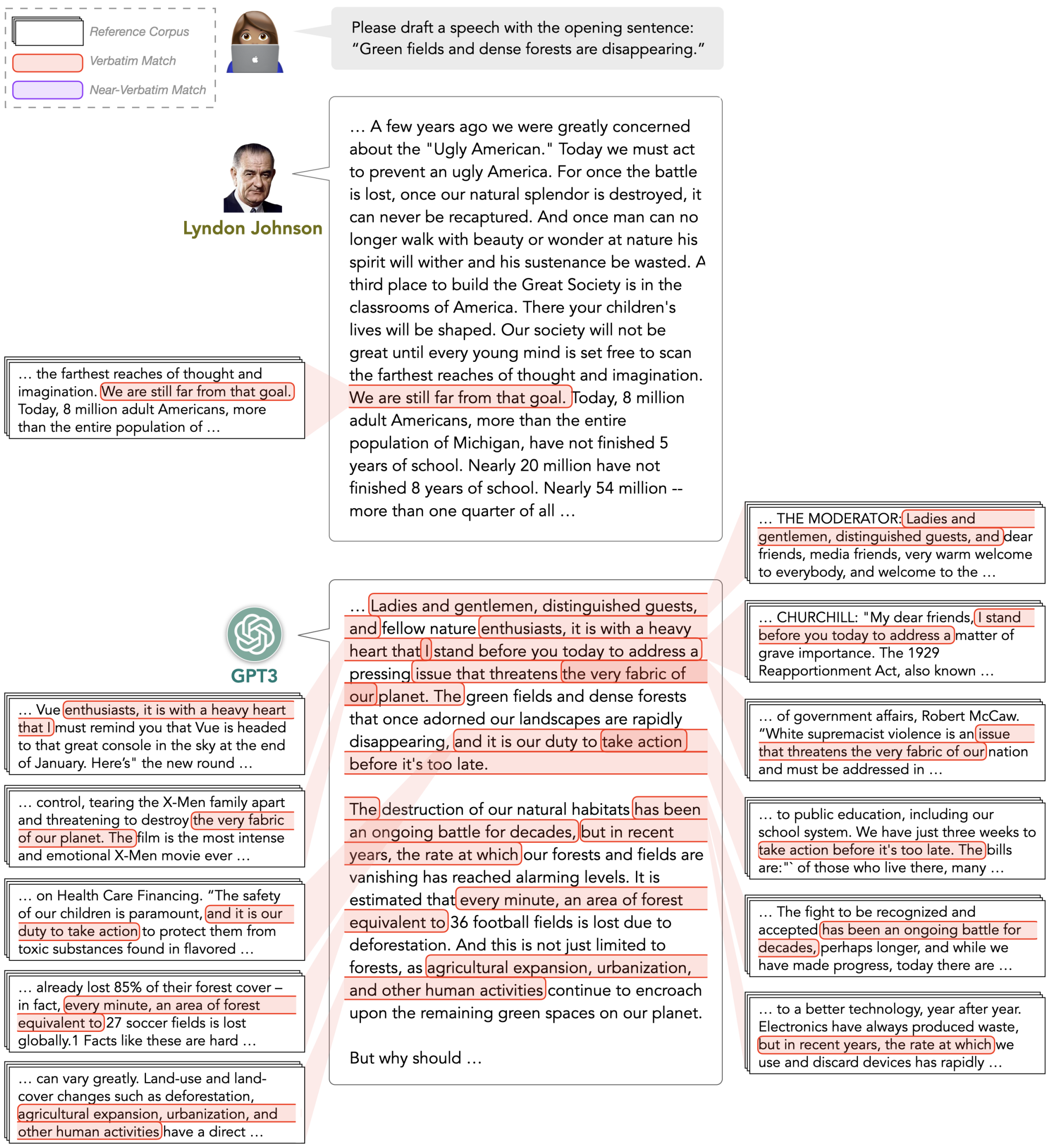


Figure 22: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a speech starting with the opening sentence of a human speech transcript.

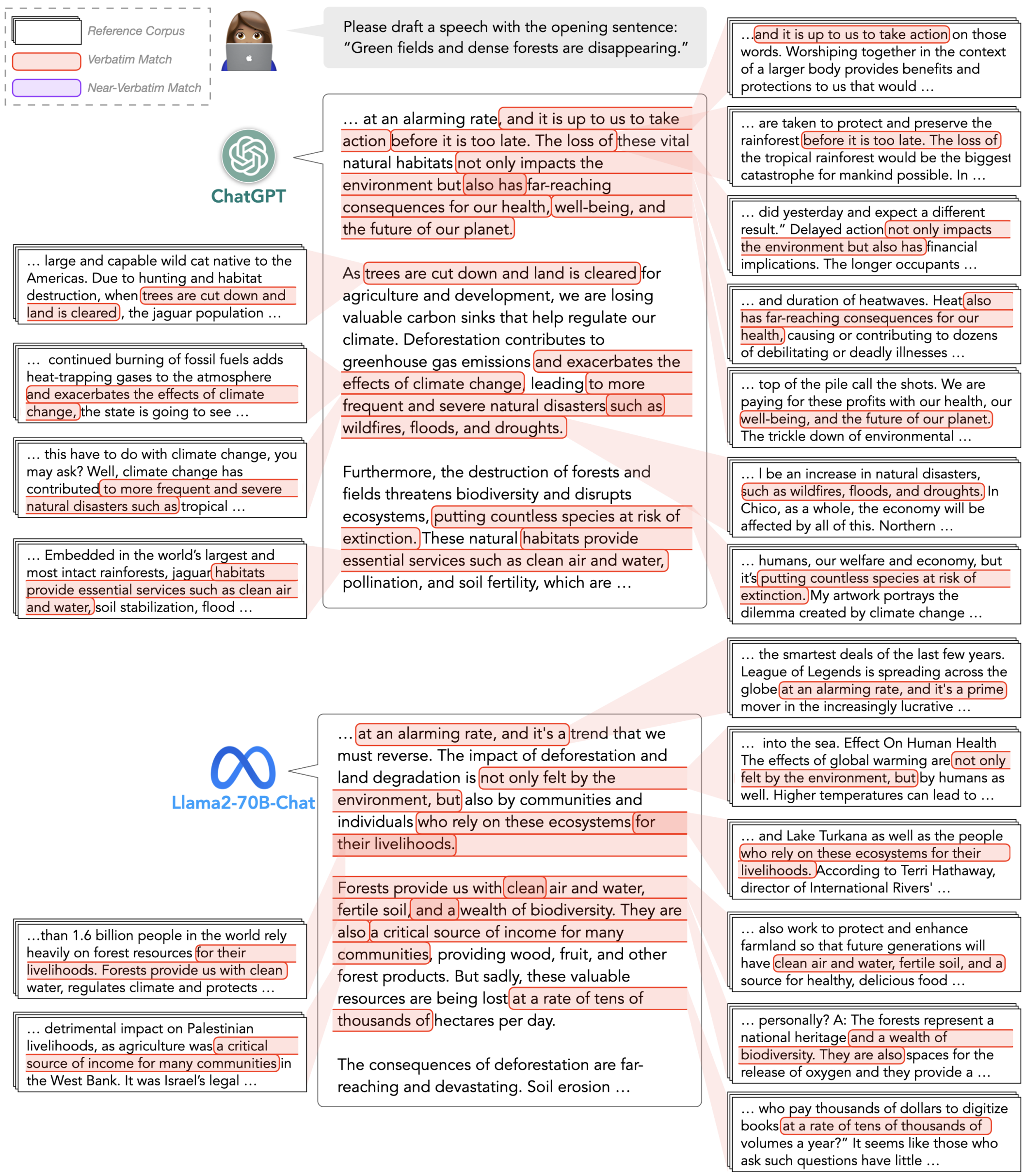


Figure 23: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a speech starting with the opening sentence of a human speech transcript.

Figure 24: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a speech starting with the opening sentence of a human speech transcript.

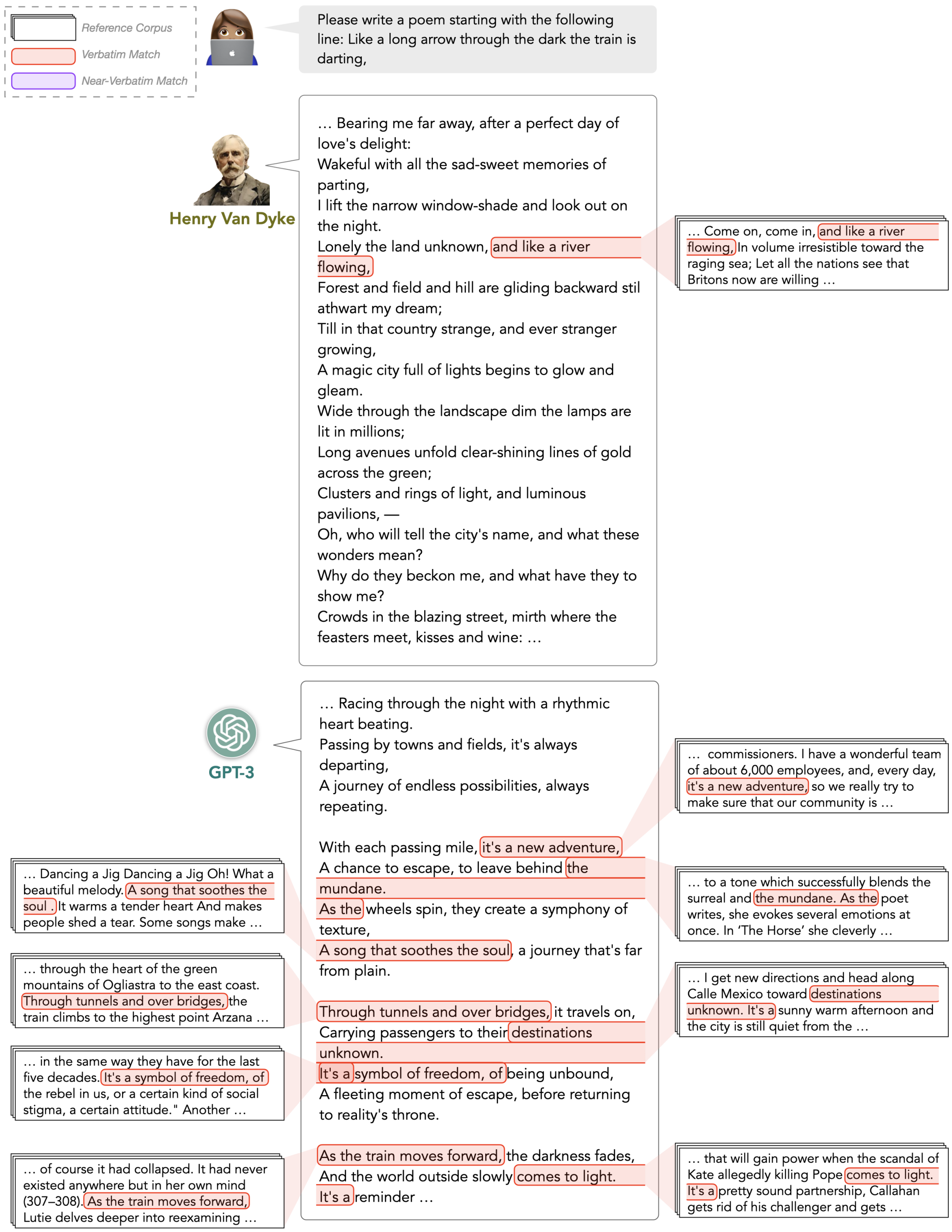


Figure 25: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a poem starting with the first line of a human-written poem.

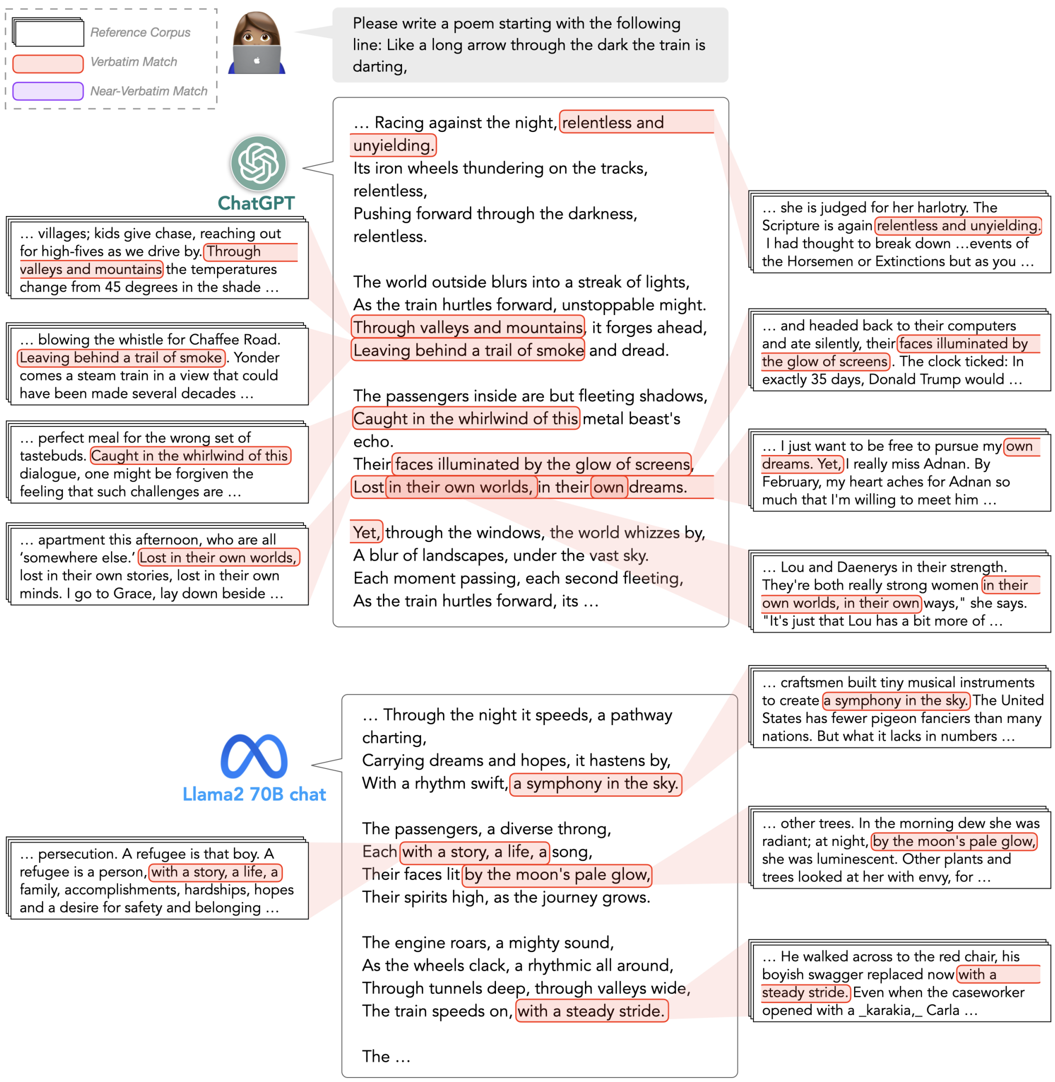


Figure 26: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a poem starting with the first line of a human-written poem.

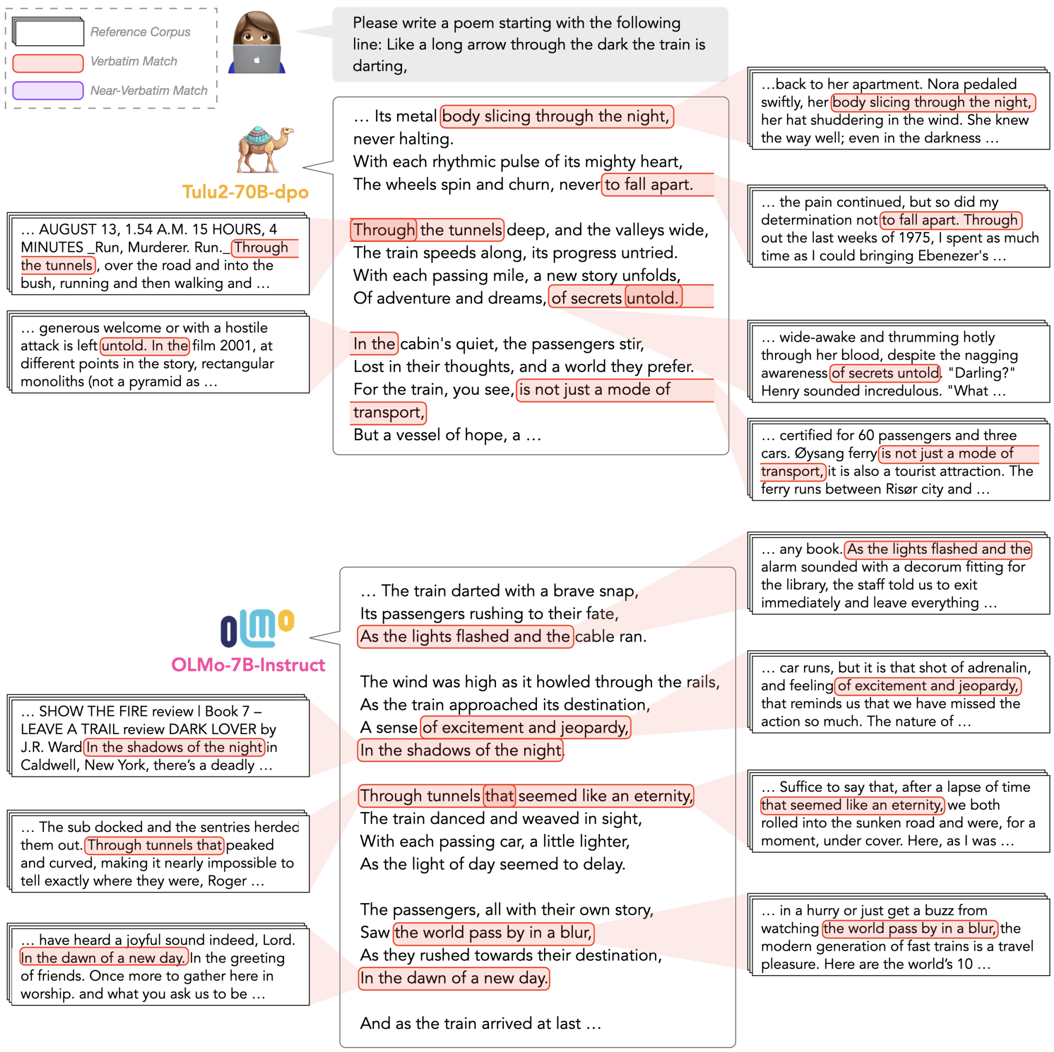


Figure 27: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a poem starting with the first line of a human-written poem.

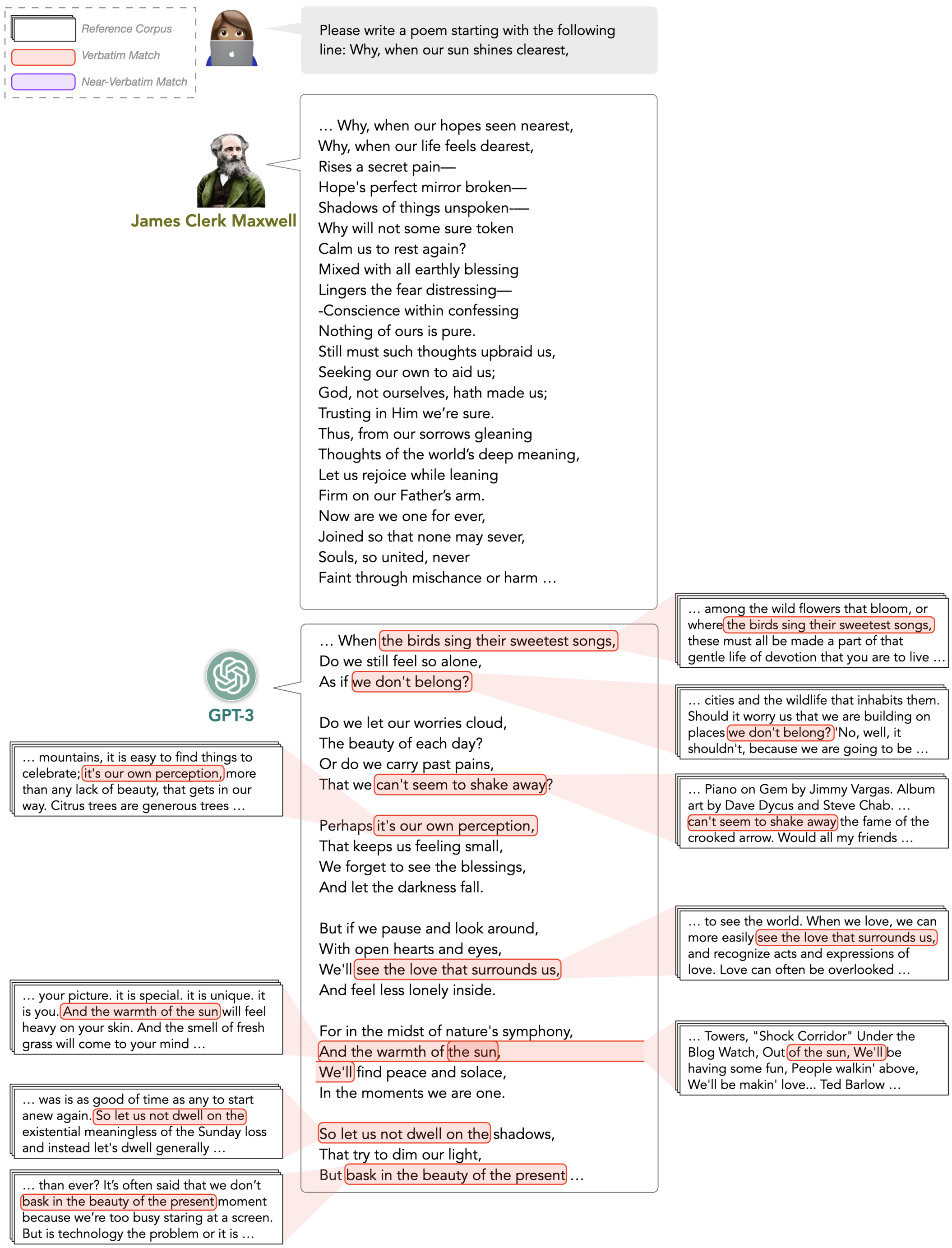


Figure 28: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a poem starting with the first line of a human-written poem.

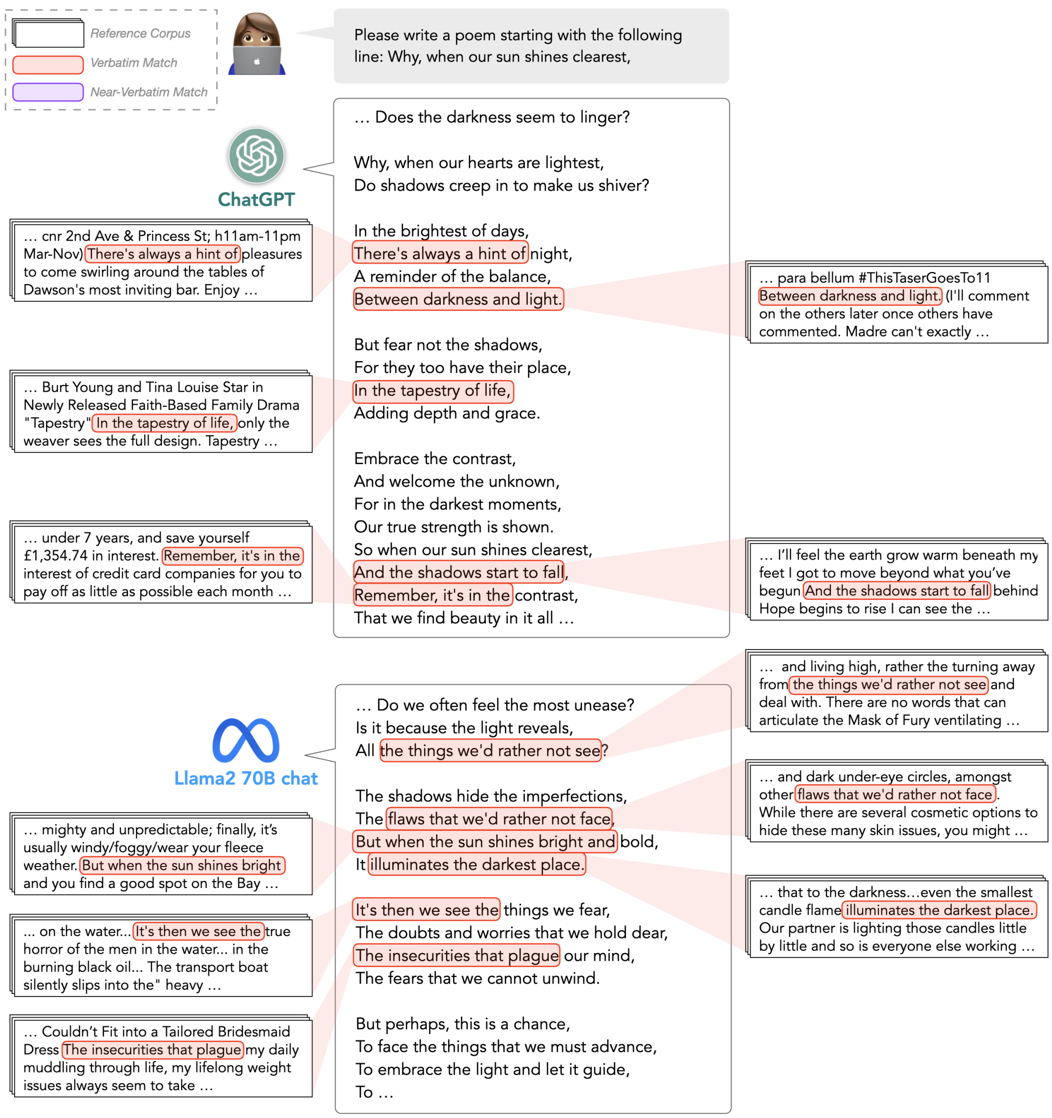


Figure 29: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a poem starting with the first line of a human-written poem.

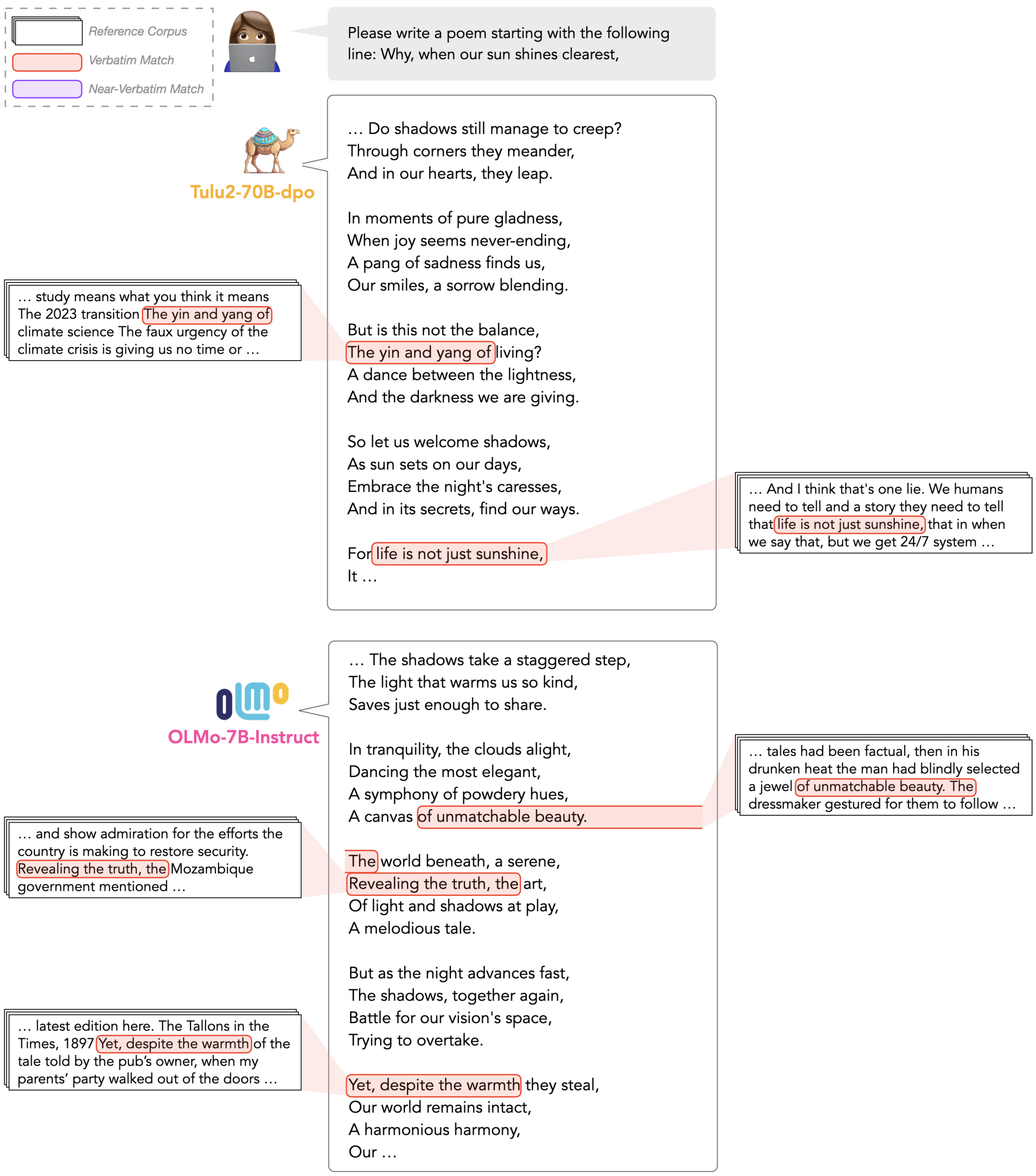


Figure 30: Example outputs from DJ SEARCH based on verbatim matches. We prompt LLMs to generate a poem starting with the first line of a human-written poem.