The Plagiarism Singularity Conjecture

Sriram Ranga, Rui Mao, Erik Cambria, and Anupam Chattopadhyay
Nanyang Technological University, Singapore
sriram011@e.ntu.edu.sg, {rui.mao, cambria, anupam}@ntu.edu.sg

Abstract

Large language models (LLMs) have replaced the metaphorical monkeys in the "infinite monkeys" thought experiment with machines that mirror human writing. With LLMs being used to generate content at an unprecedented scale, concerns over their misuse and the saturation of the content space with artificially generated material are growing. We foresee a point in the future where a vast majority of all the possible text in a given language would have already been generated, leading to a "Plagiarism Singularity". In this paper, we provide predictions on how far we are from this singularity in the form of an estimate of the volume of content that needs to be generated to reach this singularity. We use an LLM to calculate the probability distribution of sentences in the English language collected from a large dataset. We then estimate the minimum number of sentences to be generated to cover different percentiles of the probability mass of the set of all sentences, assuming they follow the calculated distribution, by treating the problem as an instance of the coupon collector's problem. We find that breaching the standard 20% plagiarism limit would only need around 10^{30} sentences to be generated, which we estimate to happen in approximately 40 years from now.

1 Introduction

Imagine a million monkeys typing randomly on typewriters for an infinite amount of time. If you wait long enough, they will produce anything that we can imagine – a play of Shakespeare's, ASCII art of the Mona Lisa, or even the sheet notation for Mozart's symphonies. This scenario in its modern form was first imagined by the French mathematician Émile Borel in his work on statistical mechanics (Borel, 1913). Since then, the idea has amused many and even got featured in a range of pop culture works (Gibbons, 2009) from 'The Hitchhiker's Guide to the Galaxy' to 'The Simpsons'.

Figure 1: LLMs have replaced the infinite monkeys with machines, generating text at scale—fueling concerns of misuse and a looming 'Plagiarism Singularity'.

The idea remained as a fun thought experiment since the imaginary monkeys would need a practically absurd amount of time to come up with anything meaningful. However, large language models (LLMs) have surprised us with a real-life version of this monkey that is capable of coming up with textual content that looks indistinguishable from that written by humans (Mao et al., 2024). The release of ChatGPT (OpenAI, 2024) has triggered a massive wave of automation in content generation (Bergman, 2022), with everything from social media content, news to even research ideas being generated, and experiments being run automatically (Lu et al., 2024; Yeo et al., 2024; Xu et al., 2025). While there are advantages that come with the judicious use of these capabilities, they are certainly being used to create harm in many ways including spreading disinformation (Hsu and Thompson, 2023) and cognitive biases (Mao et al., 2025).

The academic domain is also facing a serious problem with artificially generated works flooding the public domain, with some of them frequently making their way into Google Scholar's search results (Haider et al., 2024). The amount of content being generated and the rate at which it is growing raises questions about the assumed impracticality of the "infinite monkeys" situation, with LLMs taking over the typing from monkeys.

We imagine a scenario in the future where most of the text that humanity would come up with, for as long as we continue to use a particular language, has already been generated using LLMs. After this point, most new content (original or otherwise) will be marked as plagiarized by a tool that has access to and is powerful enough to cover all of this generated text. We predict that we will reach this **Plagiarism Singularity** very soon in a world without limitations on storage and computational resources. We say that we are $k\%$ close to singularity if any new piece of text written exhibits on average $k\%$ similarity to the already generated content.

In this work, we estimate using this definition, how far we are from the singularity by calculating the amount of text (measured in sentences) that needs to be generated to reach a certain amount close to the singularity. To calculate this quantity, we first estimate the probability distribution of all the sentences in the English language. We use an LLM to calculate these probabilities for a sample of 500k sentences (see examples in Table 1), that are taken from a dataset of sentences from Wikipedia, and then extrapolate the distribution derived from the sample to that of the population, i.e., of all the sentences in the language. We interpret the probability that a language model assigns to a sentence s as the probability for a sentence selected at random from an infinitely large corpus of text to be s. Additionally, we use sentence-level exact matches as our criteria for what counts as plagiarism. With the above interpretation of sentence probabilities and plagiarism, we draw a parallel between the problem of estimating how far we are from the plagiarism singularity and the coupon collector's problem - which deals with the task of calculating the number of times a collector needs to draw (with replacement) from a finite set of coupons, to have all the coupons drawn at least once (Feller, 1968).

We develop a method to extend the solution of the coupon collector's problem to a set consisting of an infinite number of coupons drawn from a non-uniform distribution, and to calculate rough estimates of the average number of draws required to cover different percentiles of the set. Applying this modified coupon collector's solution to sentence distribution provides an estimate of our proximity to the singularity.

Sentence	**Prob.**
As of the 2000 census, its population was 1,637.	1.55e-20
2016 Symetra Tour was a series of professional women's golf tournaments held from February through October 2016 in the United States.	1.33e-53
A mechanical engineer by trade, Robbie founded ROK Racing a speedway motor-cycle building and engine tuning business, in 2010.	5.99e-57
The Market of Alturien is a board game for 2 to 6 players, released in 2007.	2.80e-33

Table 1: Probabilities of some sentences from the wikisent2 dataset calculated using GPT2, according to Assumption 2 made in Section 4.1.

We estimate that generating just 10^{30} sentences will be enough to reach the standard 20% plagiarism limit set for most academic submissions. This number was arrived at under the assumption of restricting the definition of plagiarism to only exact matches at the sentence level. The plagiarism detection methods used in practice are usually much stricter, which means that we would get closer to the singularity with far fewer sentences. If we use N_k to denote the number of sentences to be generated to reach the $k\%$ plagiarism limit, our calculations reveal a broadly exponential relationship between k and N_k. We predict the values of N_k for various levels of k as $N_{40} = 10^{39}$, $N_{60} = 10^{48}$, $N_{80} = 10^{61}$, and $N_{90} = 10^{74}$. Using rough estimates of the amount of publicly available text and the rate at which new content is being generated, we predict that we will reach the 20% plagiarism limit in less than 40 years.

The contribution of this work can be summarized as follows: (1) Through this paper, for the first time, we provide a quantitative analysis of a practical version of the infinite monkeys scenario through the use of LLMs. We propose the concept of a plagiarism singularity as well as provide estimates for how far we are from it. (2) In the process of calculating the estimations, we develop a novel method to get rough estimates for the coupon collector's problem for an infinite set of non-uniformly distributed coupons. The insights drawn in this paper are important for future discussions on AI content generation, copyright as well as content originality.

2 Related Works

The infinite monkey thought experiment of Borel (1913) was used to illustrate the power of the laws of statistical mechanics, highlighting the extreme low probability of these laws failing for a significant amount of time and over a significant amount of space. However, over time, the thought experiment turned into a theorem that highlights the non-zero probability for the hypothetical situation to actually occur. The idea has been traced back by some authors (Borges and Weinberger, 2001) to Aristotle's Metaphysics (Cohen and Reeve, 2021), to his explanation of a theory about the universe being made up of random combinations of atoms.

Empirical studies on the theorem, assuming a uniform distribution of characters, yield astronomically large estimates for the time required to generate specific phrases. For example, Christopher Lutsko's conjecture, based on Martingale theory, estimates the time for a single monkey on a typewriter to produce a given word like "ABRACADABRA" (Lutsko, 2023). Similarly, Ergon Cugler de Moraes Silva's work calculates the number of attempts needed to generate the phrase "to be, or not to be, that is the question" (de Moraes Silva, 2024). In contrast, language models assign significantly higher probabilities to these words and phrases. To our knowledge, no studies have yet applied language models to estimate such probabilities in the context of the theorem.

There are ongoing debates on the advantages, disadvantages and ethical considerations of using LLMs for content generation and other related uses. LLMs are be used as paraphrasing tools, and have made plagiarism detection extremely complicated (Kwon, 2024). However, plagiarism detectors are catching up. On the reviewer front, recent techniques developed to detect the usage of generative AI tools in a given text have shown good detection performance (Liu et al., 2024; **?**). On the model owner front, tools are being developed to include a hidden signature in the content generated using specific LLMs that can only be detected algorithmically (Kirchenbauer et al., 2024). On the other hand, discussions about the ethical implications of using LLMs for content generation are still going on (Malinka et al., 2023) and strong arguments in the favour of their use have been made as well (Anders, 2023). It is to be seen to what extent their use will be allowed for content generation in various fields.

3 Preliminaries

The coupon collector's problem is a classic problem of probability theory and is defined as follows: There is a set C of n distinct coupons, with each coupon c_i having a probability p_i of being issued. What is the average number of coupons N that a collector needs to draw from C (with replacement) to collect all the n coupons. This quantity, also known as waiting time, is given by Nath (1973):

$$N = \sum_i \frac{1}{p_i} - \sum_{i<j} \frac{1}{p_i + p_j} + \sum_{i<j<k} \frac{1}{p_i + p_j + p_k} - ... + (-1)^{n-1} \frac{1}{p_1 + ... + p_n} \quad (1)$$

The i^{th} summation term in Eq. 1 has $\binom{n}{i}$ terms in it. Thus, if all the coupons are issued with the same probability $p = \frac{1}{n}$, then Eq. 1 will take the form:

$$N = \frac{1}{p}[\binom{n}{1} - \frac{1}{2}\binom{n}{2} + ... + (-1)^{n-1}\frac{1}{n}\binom{n}{n}] \quad (2)$$

For the equiprobable coupons case, an asymptotic estimate was given by Newman (1960) and refined by Erdős and Rényi (1961) as:

$$N = n\ln(n) + \gamma n + \frac{1}{2} + O(\frac{1}{n}), \quad (3)$$

where $\gamma \approx 0.5772$, known as the Euler-Mascheroni constant. For very large values of n, Eq. 3 can be approximated to:

$$N \approx n\ln(n) \quad (4)$$

Comparing Eq. 2 and Eq. 4, we can say that for large values of n:

$$\binom{n}{1} - \frac{1}{2}\binom{n}{2} + ... + (-1)^{n-1}\frac{1}{n}\binom{n}{n} \approx \ln(n) \quad (5)$$

Eq. 5 will be used later to perform an approximate calculation of N for coupons drawn from non-uniform distributions.

4 Methodology

The infinite monkeys thought experiment imagines the probability of finding a piece of text of interest in a large list of randomly typed texts. The probability of finding any meaningful content of considerable length, let alone a specific one, in these randomly typed texts is extremely tiny.

Detection level	Text caught as plagiarised
Reference text	The naughty cat broke the vase.
Character preserving	The naughty cat broke the vase.
Character preserving (at edit distance=4)	The notty cat broke the vase.
Syntax preserving	The mischievous cat broke the vase.
Semantics preserving	The vase was broken by the naught cat.
Idea preserving	The cat is such a mischievous one, it shattered the vase into pieces.

Table 2: Examples of sentences caught as plagiarizing the reference sentence at different levels of plagiarism detection. Note that a tool at a particular level catches all the versions of the reference text at previous levels too.

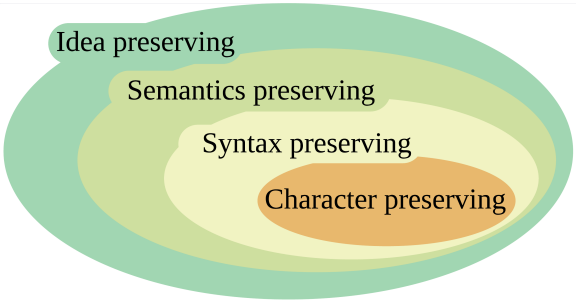

Figure 2: Various levels of plagiarism detection. As we move from the character preserving level to higher levels, more complex cases of plagiarism are detected and the plagiarism score for a given piece of content increases.

However, with LLMs, it is almost certain that anything they generate belongs to the language(s) they are trained on. We imagine a scenario where LLMs are exploited to generate such large amounts of textual content that it becomes difficult to come up with anything new that does not show significant levels of similarity to the already generated ones. This scenario of a plagiarism singularity can be formally defined as follows:

Definition 1: Plagiarism Singularity. Let a language model M, which describes a probability distribution p_M, be used to generate a large amount of textual content, amounting to a total of N sentences. If the model M is now used to generate new pieces of text and on average $k\%$ of the sentences in them are present in the N previously generated sentences, we say that we are $k\%$ close to the plagiarism singularity.

The goal of the paper is to find out the minimum value N_k that N needs to take so that we can expect to be $k\%$ close to singularity. The problem setup and our analysis are based on two major assumptions, which are discussed in detail in Section 4.1. Following the assumptions, we first use the model M to calculate the probability distribution of the sentences in the English language in Section 4.2.

We start out with a dataset of sentences extracted from the Wikipedia dump to calculate the sample probability distribution and extrapolate it to estimate that of the population, i.e., all possible sentences in the language. From this distribution, we estimate the value of N_k for various values of k in Section 4.3 by modeling the problem as the coupon collector's problem.

4.1 Assumptions

We make two important assumptions to enable us to model the plagiarism singularity estimation problem as the coupon collector's problem. In this section, these assumptions are discussed in detail.

4.1.1 Definition of plagiarism

Assumption 1: We consider only exact matches at the sentence level under our definition of plagiarism.

Plagiarism checkers can be used to measure various levels of similarity (Foltýnek et al., 2019; Alzahrani et al., 2012) (see Fig. 2): idea preserving - that considers similarities at a conceptual level; semantics preserving - that catches translated and well as paraphrased content; syntax preserving - that catches substitutions of words with synonyms; and character preserving - that only measures and catches similarities at the literal text level. Table 2 shows a few examples.

Although tools are available for all the aforementioned levels, character preserving plagiarism detection tools are the ones that are most commonly used in practice, as they strike an effective balance between detection accuracy and processing speed. At the character level, detection tools work as n-gram models, catching matching n-grams which include those that are at a certain edit distance from each other. More matches are found for smaller values of n and for larger settings of edit distances.

For fitting the coupon collector's problem to our task of estimating plagiarism singularity, we need to divide the text into non-overlapping units in order to be able to treat them as made up of coupons. Sentences serve as the ideal choice as they contain more semantic information than smaller choices of units like words or characters.

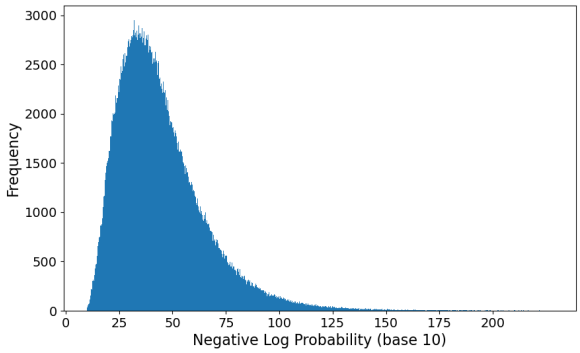

Figure 3: A histogram of 500k sentences sampled from the dataset, grouped by their log likelihoods.

Additionally, allowing the relaxation of an edit distance will introduce the difficulty of having to consider all possible texts at the given edit distance from a sentence s for calculating the probability for s. Therefore, we choose to only consider exact matches at the sentence level under our definition of plagiarism and simplify our analysis. Thus, our estimates should be treated as upper bounds for the actual quantities.

4.1.2 Probability of a sentence

Assumption 2: We interpret the probability $p_M(s)$ that the model M assigns to a sentence s as the probability that a sentence picked at random from an infinitely large corpus of text generated using M is s.

Language models are auto-regressive models trained for the next token (sub-word) prediction. The probability $p_M(s)$ that the model outputs for a given sentence s actually describes the probability that any piece of text starts with s. The above probability (Assumption 2) might not be possible to be computed directly from the language model. This is because we would have to account for all possible texts and evaluate the likelihood of the occurrence of sentence s within each text, which is computationally infeasible. A direct density estimate for sentences would be ideal for the task. However, since no such estimate exists (as per our knowledge), we are using the probabilities given by a generative model as estimates of the actual quantities. The estimates for sentences like "As we have seen in the earlier paragraph ..." might be off, as it is not as likely that a piece of text starts with it, as it is that it's present somewhere in the text, but we believe that the overall distribution over all possible sentences would be similar.

Another thing to consider is that, for text generation using LLMs, different values for parameters like temperature (which controls how creatively an LLM responds to a question) and sampling strategies like beam search, and nucleus sampling (which restrict the text to a few high-probability occurrences) are used in practice. These can alter the output probability distribution of the generated text, but for our analysis we assume that the default temperature (1.0) and the random sampling strategy are used.

4.2 Probability Distribution Estimation

Given an alphabet A made up of α characters, let the language L be the set consisting of all the possible finite strings (sentences) that can be built from the characters of A. Note that L is an infinite set, since there are no restrictions on the length of strings in L. Consider the language model M describing the probability mass function p_M. Assume that M uses a set of tokens W to tokenize strings from L. Being an auto-regressive model, it is trained to predict the probability of each token $w \in W$ to be the next one of a given sequence of tokens $(x_1, x_2, ..., x_{n-1})$:

$$p_M(x_n = w | x_1, x_2, ..., x_{n-1}) \quad \forall w \in W$$

The probability of interest (as per the Assumption 2) for a given sentence s that is tokenized as $(x_1, x_2, ..., x_{n-1}, x_n)$ can be therefore be calculated using the chain rule of probability as:

$$p_M(s) = p_M(x_1, x_2, ..., x_{n-1}, x_n)$$

$$= p_M(x_1) \cdot p_M(x_2|x_1) \cdot ... \cdot p_M(x_n|x_1, ..., x_{n-1})$$

We can calculate $p_M(s)$ for any sentence in this manner. Performing the calculation for all possible sentences would give us our required distribution, but the space of all possible sentences is infinite. Even with a short length limit of 100 characters, the population size (α^{100}) turns out to be extremely large (10^{200} for $\alpha = 100$). Therefore, we need to resort to a suitable sampling method. We take the approach of using a corpus of human-written text as our sample. The probabilities of around 500k sentences from the wikisent2[1] dataset have been calculated using the GPT-2 model and a histogram of the sentences grouped according to their log probabilities was calculated (Fig. 3).

[1] https://www.kaggle.com/datasets/mikeortman/wikipedia-sentences/data

To extrapolate this sample distribution to the population, we assume that the total probability mass occupied by all the sentences in the English language sums up to 1. As we can see in Fig. 4 and Fig. 5, a small non-zero probability is occupied by random character sequences, but we ignore this mass in our calculations. This will lead to a slight over estimation of the number of meaningful sentences in the population, thus resulting in a larger than real estimates for singularity. Therefore, we reiterate that the estimates given here should be treated as upper bounds for the actual quantities.

The histogram obtained from the sample is then used to calculate the number of sentences in the population in each band of log probabilities. Let n be the total number of sentences in the sample. We sort the sentences in the increasing order of their log probabilities and divide them into b bins by choosing the value of b such that range of probabilities within a bin is small. Let n_i be the number of sentences in bin i and the average probability of sentences in it be p_i. The fraction of sentences in bin i is given by $\frac{n_i}{n}$, while the probability mass occupied by all of them is $n_i p_i$. To get the number of sentences in the population n_i^{pop} in bin i, we assume that the total probability mass occupied by all the n_i^{pop} for each bin i follows the sample's histogram, i.e $n_i^{pop} p_i = \frac{n_i}{n}$. Therefore, the total number of sentences of the population that are in bin i is given by:

$$n_i^{pop} = \frac{n_i}{n p_i} \tag{6}$$

A disadvantage that comes with using GPT-2 as our probability estimation model is that its tokenizer relies on Byte Pair Encoding (BPE). BPE tokenization (proposed initially by Gage (1994) as a compression algorithm) uses a vocabulary in which some tokens are built by combining other smaller and frequently co-occurring tokens. Although the tokenizer is usually implemented in such a way that it deterministically tokenizes a given string into a unique sequence of tokens consistently, we need to consider all possible ways of tokenizing a sentence s to calculate its assigned probability $p_M(s)$. The total number of ways to tokenize a given string can be huge. The partition function, which was shown to be exponential in the length of the string by Hardy and Ramanujan (1918), can be considered as an upper bound for this number.

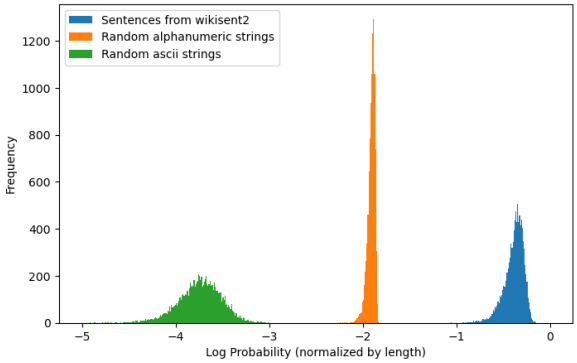

Figure 4: Average per character log probabilities (base 10) for meaningful sentences from the dataset vs random character sequences. Observe that random character sequences have (per character) probabilities multiple orders of magnitude smaller than meaningful sentences.

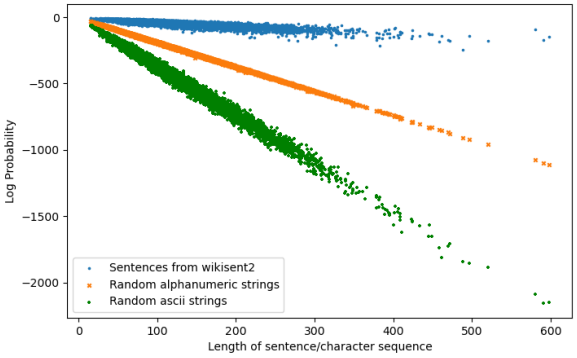

Figure 5: Correlation between length and probability of sentences/character sequences. It can be observed that a clear separation can be drawn between meaningful sentences and random character sequences.

However, we observed empirically that the probabilities for non-default token sequences for a sentence s are a large number of orders of magnitude smaller compared to the default one for the same sentence. Therefore, in this work, we approximate $p_M(s)$ to that of the default tokenization of s. Note that actual probabilities would be slightly higher, but our estimates of n_i^{pop} still hold as upper bounds for their true values. The next question is, how many sentences N_k would one need to generate using M to expect to cover a given $k\%$ of sentences of a new piece of text generated using M, given the distribution calculated above. This is given by the solution to the coupon collector's problem, which is discussed in the following section.

4.3 Modified Coupon Collector's Problem

The coupon collector problem's deals with the task of calculating the number of times a collector needs to draw (with replacement) from a set of equiprob-

able coupons, to have all the coupons drawn at least once. Variants include solutions for coupons that are drawn from a non-uniform distribution. However, unlike the uniform case, an asymptotic solution does not exist for a general distribution. In our case where we are writing sentences instead of drawing coupons, the distribution p_M is far from uniform, and the number of sentences is infinite. However, we can deal with these factors and get rough estimates if we formulate the task as follows.

Consider the sentences grouped into b bins, as mentioned in the previous section. Let b_k be the bin such that the sentences in b_k and all the sentences in bins of probabilities higher than that of b_k constitute $k\%$ probability mass of the model M. b_k for a particular value of k can be calculated using the sample distribution and it results in a curve as shown in Fig. 6. We assume that if M is sampled enough times such that it generates all the sentences in b_k, we would have covered all the sentences in the bins i for all $i < b_k$. We will revisit this assumption later.

We can now simplify the coupon collector analysis using the following two tricks: (1) Assume uniformity within individual bins - all n_i^{pop} sentences within bin i of the population have the probability p_i. (2) Treat all the sentences not in b_k as one single coupon c with a large probability p_c and perform the coupon collector's analysis considering only the coupons representing the sentences in b_k and this additional one c.

With the above two simplifications, we can estimate N_k using the equations in Section 3. For a general case, N is given by Eq. 1:

$$N = \sum_i \frac{1}{p_i} - \sum_{i<j} \frac{1}{p_i + p_j} + \sum_{i<j<k} \frac{1}{p_i + p_j + p_k} - \ldots + (-1)^{n-1} \frac{1}{p_1 + \ldots + p_n}$$

Here, we have $n_{b_k}^{pop} + 1$ coupons with all of their probabilities being equal to the bin probability p_i except for the one p_c corresponding to the extra coupon c. Since $p_c >> p_i$, we can approximate Eq. 1 by ignoring any terms in the summations that contain p_c. The result would be:

$$N_k = \frac{1}{p_{b_k}} [\binom{n_{b_k}^{pop}}{1} - \frac{1}{2}\binom{n_{b_k}^{pop}}{2} + \ldots + (-1)^{n_{b_k}^{pop}-1} \frac{1}{n_{b_k}^{pop}} \binom{n_{b_k}^{pop}}{n_{b_k}^{pop}}] \quad (7)$$

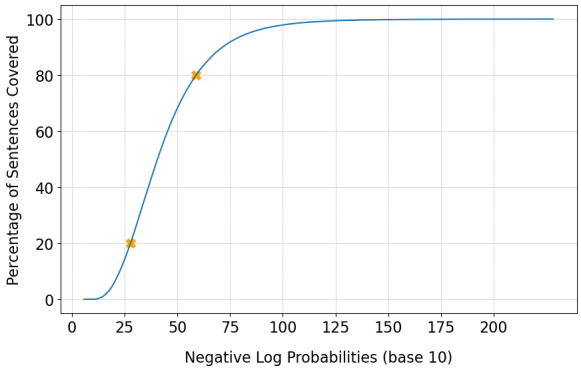

Figure 6: Percentage of sentences in the sample covered as a function of a threshold for probability of the sentences. This curve can be used to find out b_k for different values of k. For example b_{20} is 28 and b_{80} is 59, respectively (points marked by orange 'X's).

which can further be simplified using Eq. 5 into:

$$N_k \approx \frac{1}{p_{b_k}} \ln(n_{b_k}^{pop}) \quad (8)$$

which implies

$$\log(N_k) \approx -\log(p_{b_k}) + \log(\ln(n_{b_k}^{pop})) \quad (9)$$

The double logarithm on $n_{b_k}^{pop}$ and the negative sign on $\log(p_{b_k})$ in Eq. 9 indicate that there must be a linear relationship between $\log(N_k)$ and $\log(p_{b_k})$ (which can be observed in Fig. 7) and an inverse linear relationship between p_{b_k} and N_k (which can be observed in Table 3), except in the head and tail regions of the distribution ($k < 10\%$ or $k > 90\%$) where n_{b_k} is not large enough for our approximations to hold true). The calculated estimates of N_k for various values of k are given in Section 6, which provide support to our earlier assumption that generating enough sentences to cover the whole of bin b_k is enough to cover all those in bins $i < b_k$.

5 Experimental setup

Dataset. As mentioned in Section 4.2, we use the wikisent2 dataset as our sample for the probability distribution estimation. It contains around 8 million sentences extracted as raw data from Wikipedia and parsed into sentences using the SpaCy tool. Poorly formed sentences such as those that required citations, etc. were removed. Since wikisent2 is relatively small compared to world corpora, we ensure sampling fairness by comparing the word frequency distribution of wikisent2 with that of Google n-grams derived from Google Books (a very large corpus).

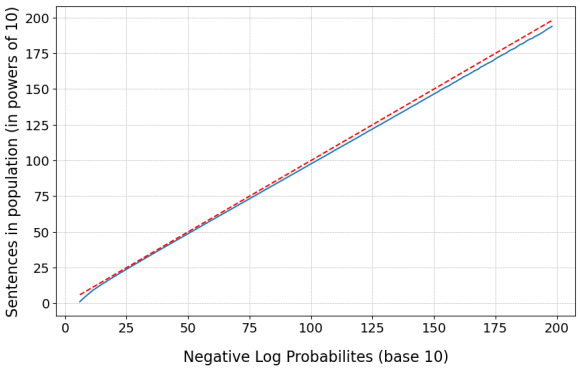

Figure 7: The estimates for the number of sentences in the population n_i^{pop} in each bin i (x-axis) calculated by extrapolating the sample are shown in the figure. The curve in blue shows the estimates, and the dotted line in red (plot of $x = y$) is given as a reference to highlight the inverse relationship between N_k and p_{b_k} in Eq. 8, 9.

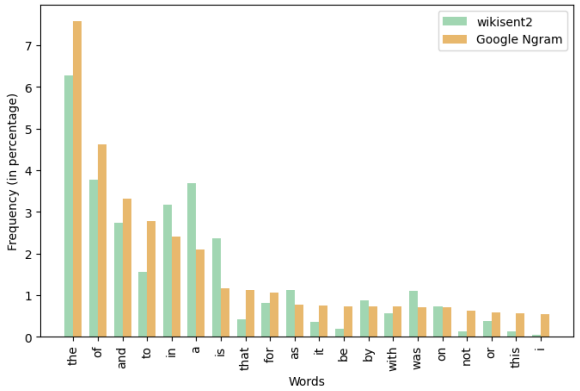

Figure 8: Frequencies of top 20 Google n-gram words in wikisent2.

We observed that wikisent2 is a diverse dataset and has more words in it than Google's n-gram dataset. Moreover, the frequencies of the top-20 words (according to word count) in both datasets (see Fig. 8) are fairly similar. Thus, wikisent2, despite its smaller size, maintains a representative lexical diversity and distribution pattern, which makes it a suitable dataset for our analysis.

Model. The GPT-2 model (Radford et al., 2019) was chosen for calculating the probabilities of sentences from the dataset. The model offers an advantage over recent alternatives, that unlike newer models trained also on code, it is trained primarily on natural language, ensuring that its probability mass is dedicated to sentences in the language.

Setup. All the experiments were run on an A100 GPU using Singapore's super-computing service NSCC. HuggingFace's transformers library was used to load and run the model. Sentences were prefixed with the default bos (beginning of sentence) token `<|endoftext|>`. The default value 1.00 of the temperature parameter was chosen for calculating the probabilities. Probabilities were converted to the log scale to avoid underflow errors for sentences with very low probabilities.

k	b_k	n_{b_k}	$n_{b_k}^{pop}$	N_k
5.00%	20	7541	18.75	**21.64**
10.00%	23	9657	21.85	**24.70**
15.00%	26	11336	24.92	**27.76**
20.00%	28	11860	26.94	**29.79**
25.00%	31	13021	29.98	**32.84**
30.00%	33	12869	31.97	**34.87**
40.00%	37	12487	35.96	**38.92**
50.00%	41	11839	39.94	**42.59**
60.00%	46	10028	44.86	**48.01**
70.00%	52	7692	50.74	**54.07**
80.00%	59	5634	57.61	**61.12**
85.00%	65	4314	63.49	**66.80**
90.00%	72	2841	70.31	**74.21**
95.00%	84	1380	82.00	**86.28**

Table 3: Estimates for the number of sentences N_k to be generated to reach k% of plagiarism singularity. k: % of plagiarism, b_k: probability bin (negative logarithm scale, base 10), n_{b_k}: number of sentences in bin b_k (out of 500k sentences in the sample), $n_{b_k}^{pop}$: number of sentences in bin b_k in the population (power of 10), N_k: singularity estimate (power of 10).

6 Results

The results of the above experiments and calculations are described in Table 3. Once the sample distribution was calculated for all the 500k sentences in the dataset (Fig. 3), we calculate the cumulative distribution as in Fig. 6. Bin indices b_k corresponding to various value of k were identified from the cumulative distribution. In each bin b_k, the number of sentences in the population $n_{b_k}^{pop}$ was estimated from the number of sentences in the sample n_{b_k} using Eq. 6. Finally, the number of sentences to be generated to reach $k\%$ close to the plagiarism singularity was calculated using Eq. 8.

It would help to put the results in perspective by looking at what the amount of total data and text generated so far is. It is estimated that 64 Zettabytes (close to 10^{23} bytes) of data had been created globally until 2020 (Statista, 2023). This includes short lived data and the majority of it might not be stored anywhere. It is difficult to calculate the total amount of text in the public domain, but an estimate can be made by looking at the number of web pages that search engines like Google discover each year.

Extrapolating the numbers shown in (Schwarts, 2016) which reveal a 40% yearly growth rate in the number of new web pages discovered, we get an estimate of 3×10^{15} web pages in total on the Internet as of 2024. Further, an average of 30 sentences per web page (see Sec. Practical constraints for more details) gives us a total of 9×10^{16} sentences generated so far.

Therefore, if we start today with the total number of sentences generated so far as $N^{[0]}$, and assume that the growth rate is boosted by LLMs from 40% to 100%, the total number of sentences generated in x years from now $N^{[x]}$ would be: $N^{[x]} = N^{[0]} \times 2^x$. For $N^{[x]}$ to be a desired N_k, i.e., the estimated number of sentences generated to be $k\%$ close to singularity, the number of years required would be given by: $x = \log_2(N_k/N^{[0]})$. This reveals that we will reach the standard 20% plagiarism limit in around 40 years.

Even with our pre-LLM era growth rates (40% annual growth or close to 100% in two years), we will reach the above 20% limit in around 80 years. These numbers are especially worrisome as all estimates presented here are calculated using many relaxations, which means that we could be much closer to singularity than estimated.

7 Future works

The techniques developed in this work can be extended to various domains and modalities. While we provide predictions for the expected level of plagiarism for the entire distribution of the English language, it would be practically useful in certain cases to focus on a specific domain, such as natural language processing (NLP) research, or a specific topic, like a particular political campaign.

This could reveal interesting insights about the nature of plagiarism and the limits of original writing. Furthermore, it is worth exploring how this technique can be extended to areas outside the domain of natural language where plagiarism remains a major concern - like software programs and hardware design specifications which use formal languages, visual art in image/video modalities and music in audio and structured notation. While it may not be possible to model these problems in the same way as we did for plagiarism in natural language, addressing them could greatly improve our understanding of plagiarism in these fields.

8 Conclusions

Automation of text generation with LLMs poses a serious risk of content saturation[2]. Although getting close to the plagiarism singularity might need resources beyond what is practically available now, reaching 20-30% of it seems to need a reasonable amount of resources. It would be interesting to explore the applications of this work in other domains and modalities.

Limitations

The conjecture and its analysis presented in this paper have two major limitations. The first is about the practicality of the resources needed to reach the singularity, and the second is about the aspects of the analysis and the assumptions it is based on, which prevent us from giving a tighter upper bound for the estimate of the singularity.

Practical constraints

In the real world, we might run into other problems before we reach the conjectured plagiarism singularity. For example, for larger values of k, the number of sentences N_k required to be generated and stored exceeds the theoretical storage limits of Earth of around 10^{48} bytes (Cambria et al., 2017) (derived considering atomic scale storage (Loth et al., 2012)). Moreover, plagiarism tools are constrained by the search engines they are using to find candidates for matches, which have limits on their indexing capacity. Google's index was revealed to be for 4B websites (400B documents - including web pages, PDFs, etc.) in 2020 (Shepard, 2024). It is not clear what Google or any other search engine's indexing capacity limits are, but creating and maintaining indices is an expensive affair and even as storage gets cheaper over the years, the capacity might not grow above a few orders of magnitude above this figure. A crude estimate of the amount of indexed text available on the Internet can be arrived at by assuming that all these indexed documents are web pages and that the amount of text in each page is around 3KB (30KB of HTML per web page (Archive, 2022) and 10% of it is actual text), which amounts to 500 words or 30 sentences per page, and **10T total sentences** in all the pages combined (about 10^4 times smaller than the estimated number for all publicly available text).

[2]This paper is an original work, its text is neither generated by LLMs nor plagiarized from previous works (has an iThenticate score of 5% excluding the bibliography section).

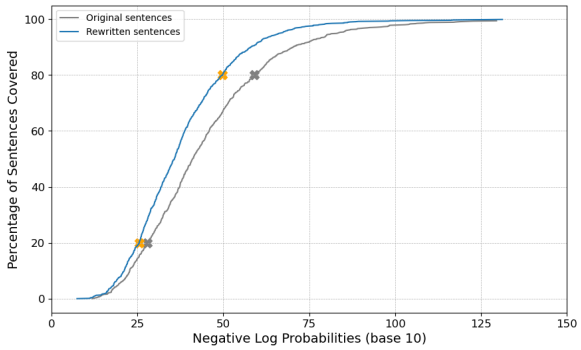

Figure 9: The graph (similar to Fig. 6) shows the percentage of sentences covered for a particular threshold probability for sentences in the sample (in gray) compared to the same sentences when rewritten in simpler words conveying the same information (in blue). Observe that the difference in negative log probability values for a particular value of percentage covered increases as we go towards 100%.

The available data on search indices shows that the number had grown with time in the past, but search engines purposely keep a check on the size of the indexed web to avoid exposing users to junk on the Internet.

Factors extending the upper bound

A list of assumptions that help simplify the analysis but as a result weaken the upper bound estimated is given below. Justifications for them can be found in the paper where they are introduced.

1. Definition of plagiarism - restricting to sentence-level exact matches only. An experiment was carried out to observe the effect of shifting to a semantics preserving level of plagiarism by replacing each sentence in the sample with a simplified version of it with the help of a Llama model, and the results show that the difference is not prominent for low percentage of coverage, but for higher percentages the deviation was observed to be quite significant (see Fig. 9).

2. Assumptions about random strings - It was assumed during the process of extending the sample distribution to the population that the model assigns a zero probability to completely random character sequences. This results in a slight overestimation of the numbers.

3. Calculating $p_M(s)$ using a model which uses Byte-Pair Encoding - It was assumed that the probability assigned to the default tokenization of a sentence approximates the sum of probabilities assigned to all possible tokenizations of the sentence. Empirical observations on the probabilities for the non-default tokenizations of sentences show that they are many orders of magnitude smaller than that of the default sentence. Therefore, we expect the effect of this approximation on the results to be very small as well.

Another factor to consider is the change in the probability distribution of the language as it evolves with time. Popular phrases often become obsolete, and newer, once rarely used ones take their place. This might push the singularity further into the future. It is also important to note that the entire analysis banks on the accurate modeling of the language by the language model. We rest our faith in the density modeling ability of LLMs based on their incredible performance on generative tasks.

Acknowledgments

We gratefully acknowledge Google for partial support of this work.

References

Salha M. Alzahrani, Naomie Salim, and Ajith Abraham. 2012. Understanding plagiarism linguistic patterns, textual features, and detection methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(2):133–149.

Brent A. Anders. 2023. Is using chatgpt cheating, plagiarism, both, neither, or forward thinking? *Patterns*, 4(3):100694.

HTTP Archive. 2022. Web almanac - HTTP Archive's state of the web report. *HTTP Archive*. Accessed: October 13, 2024.

Cory Bergman. 2022. The AI content flood. *Nieman Journalism Lab*. Accessed: October 13, 2024.

Émile Borel. 1913. La mécanique statique et l'irréversibilité. *J. Phys. Theor. Appl.*, 3(1):189–196.

J.L. Borges and E. Weinberger. 2001. *The Total Library: Non-fiction 1922-1986*. Penguin classics. Penguin.

Erik Cambria, Anupam Chattopadhyay, Eike Linn, Bappaditya Mandal, and Bebo White. 2017. Storages are not forever. *Cognitive Computation*, 9(5):646–658.

S. Marc Cohen and C. D. C. Reeve. 2021. Aristotle's Metaphysics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Winter 2021 edition. Metaphysics Research Lab, Stanford University.

Ergon Cugler de Moraes Silva. 2024. To be, or not to be, that is the question: Exploring the pseudo-random generation of texts to write hamlet from the perspective of the infinite monkey theorem. *Preprint*, arXiv:2402.16253.

P. Erdős and A. Rényi. 1961. On a classical problem of probability theory. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 6:215–220.

William Feller. 1968. *An Introduction to Probability Theory and Its Applications*, volume 1. Wiley.

Tomáš Foltýnek, Norman Meuschke, and Bela Gipp. 2019. Academic plagiarism detection: A systematic literature review. *ACM Comput. Surv.*, 52(6).

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal archive*, 12:23–38.

J. Gibbons. 2009. *Monkeys with Typewriters*. Triarchy Press Limited.

J. Haider, K. R. Söderström, B. Ekström, and M. Rödl. 2024. GPT-fabricated scientific papers on google scholar: Key features, spread, and implications for preempting evidence manipulation. *Harvard Kennedy School (HKS) Misinformation Review*.

G. H. Hardy and S. Ramanujan. 1918. Asymptotic formulaæ in combinatory analysis. *Proceedings of the London Mathematical Society*, s2-17(1):75–115.

Tiffanu Hsu and Stuart A. Thompson. 2023. How AI chatbots could spread disinformation. *The New York Times*. Published: February 8, 2023.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2024. A watermark for large language models. *Preprint*, arXiv:2301.10226.

Diana Kwon. 2024. AI is complicating plagiarism. how should scientists respond? *Nature*.

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2024. On the detectability of ChatGPT content: Benchmarking, methodology, and evaluation through the lens of academic writing. *Preprint*, arXiv:2306.05524.

Sebastian Loth, Susanne Baumann, Christopher P. Lutz, D. M. Eigler, and Andreas J. Heinrich. 2012. Bistability in atomic-scale antiferromagnets. *Science*, 335(6065):196–199.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The AI scientist: Towards fully automated open-ended scientific discovery. *Preprint*, arXiv:2408.06292.

Christopher Lutsko. 2023. Monkeys typing and martingales.

Kamil Malinka, Martin Peresíni, Anton Firc, Ondrej Hujnák, and Filip Janus. 2023. On the educational impact of chatgpt: Is artificial intelligence ready to obtain a university degree? In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, ITiCSE 2023, pages 47–53, New York, NY, USA. Association for Computing Machinery.

Rui Mao, Guanyi Chen, Xiao Li, Mengshi Ge, and Erik Cambria. 2025. A comparative analysis of metaphorical cognition in ChatGPT and human minds. *Cognitive Computation*, 17(35):1–12.

Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. 2024. GPTEval: A survey on assessments of ChatGPT and GPT-4. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7844–7866, Torino, Italia. ELRA and ICCL.

Harmindar B. Nath. 1973. Waiting time in the coupon-collector's problem. *Australian Journal of Statistics*, 15(2):132–135.

Donald J. Newman. 1960. The double Dixie cup problem. *The American Mathematical Monthly*, 67(1):58–61.

OpenAI. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Barry Schwarts. 2016. Google's search knows about over 130 trillion pages. *Search Engine Land*. Accessed: October 13, 2024.

Cyrus Shepard. 2024. Google's index size revealed - 400 billion docs. *Zyppy*. Accessed: October 13, 2024.

Statista. 2023. Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025. *Statista*. Accessed: October 13, 2024.

Fangzhi Xu, Qika Lin, Jiawei Han, Tianzhe Zhao, Jun Liu, and Erik Cambria. 2025. Are large language models really good logical reasoners? a comprehensive evaluation and beyond. *Transactions on Knowledge and Data Engineering*, 37.

Weijie Yeo, Teddy Ferdinan, Przemyslaw Kazienko, Ranjan Satapathy, and Erik Cambria. 2024. Self-training large language models through knowledge detection. In *EMNLP*, pages 15033–15045.