

Slaves to the Law of Large Numbers: An Asymptotic Equipartition Property for Perplexity in Generative Language Models

Avinash Mudireddy¹ Tyler Bell¹ Raghu Mudumbai¹

Abstract

We prove a new asymptotic equipartition property for the perplexity of long texts generated by a language model and present supporting experimental evidence from open-source models. Specifically we show that the logarithmic perplexity of any large text generated by a language model must asymptotically converge to the average entropy of its token distributions. This defines a “typical set” that all long synthetic texts generated by a language model must belong to. We show that this typical set is a vanishingly small subset of all possible grammatically correct outputs. These results suggest possible applications to important practical problems such as (a) detecting synthetic AI-generated text, and (b) testing whether a text was used to train a language model. We make no simplifying assumptions (such as stationarity) about the statistics of language model outputs, and therefore our results are directly applicable to practical real-world models without any approximations.

1. Introduction and Motivation.

Consider a generative model, defined as an algorithm that takes a user input \mathbf{X} and produces an output \mathbf{Y} that statistically resembles data from a natural source. A specific type of generative model is a large language model (LLM) whose output \mathbf{Y} is a body of text and the input is a text user prompt \mathbf{X} . State-of-the-art LLMs (Anthropic, 2024; OpenAI, 2024) are now able to produce detailed and information-rich text outputs such as entire screenplays and book-length manuscripts from a short and simple user prompt. Furthermore, LLMs produce text that can imitate human language well-enough to pass the Turing test (Pinar Saygin et al., 2000) i.e. resembles text created by humans well enough to be convincing to human observers. In this work, we

argue that an LLM is strongly constrained by statistical laws. Specifically, we show that the logarithmic *perplexity* of any large text produced by a language model must asymptotically converge to the average entropy of its token distributions. This means that any language model is constrained to only output text strings from a *typical set*, which is an exponentially vanishing subset of all possible grammatically correct strings.

1.1. Contribution: Equipartition Property for Perplexity

Perplexity, defined as an inverse likelihood function, is widely used as a performance metric for language models (Meister & Cotterell, 2021). It is closely related to the information-theoretic concepts of *surprisal* (Levy, 2008) and cross-entropy (Cover & Thomas, 2005b), and it also appears to capture linguistic (Miaschi et al., 2021; Gamallo et al., 2017) and cognitive (Demberg & Keller, 2008; Cohen & Pakhomov, 2020) phenomena at least partially. Many “AI detection” tools for identifying synthetic text from language models are based on observed differences between the perplexity of synthetic and natural text (Mitchell et al., 2023; Gehrmann et al., 2019).

Our main result is a generalization of the well-known Asymptotic Equipartition Theorem (AEP) (Cover & Thomas, 2005a) from information theory. The simplest version of the AEP states that a long sequence of independent and identically distributed (iid) random symbols is likely to be a *typical sequence* (Csiszar, 1998) defined by the property that the empirical distribution of the occurrence of different symbols within the sequence is very close to the distribution from which the symbols were drawn.

The AEP itself can be thought of as a variant of the Law of Large Numbers (LLN) (Idele, 2018) applied to a log likelihood function, and just like the LLN, there is an extensive literature on extending (Moy, 1961) the AEP to a sequence of nearly-independent but not identically distributed random variables (Nishiara & Morita, 2000; Timo et al., 2010). The outputs of non-trivial language models, however, cannot be reasonably modeled as a sequence of nearly-independent symbols (Brown et al., 2020).

^{*}Equal contribution ¹Department of Electrical & Computer Engineering, University of Iowa, Iowa City, USA. Correspondence to: Raghu Mudumbai <rmudumbai@engineering.uiowa.edu>.

Our main contribution is to formulate and provide theoretical and experimental justification for a version of the AEP that applies to real-world practical LLMs without making simplifying assumptions about their statistics. A secondary contribution is to show possible applications of this theory to practical problems such as AI text detection and LLM dataset inference.

2. Background and Definitions

Let M be a generative model described by $\mathbf{Y} = m(\mathbf{X}, \mathbf{W})$, where the output \mathbf{Y} consisting of a (potentially infinite) string of tokens $\mathbf{Y} = [Y_1, Y_2, \dots, Y_N, \dots]$ is a deterministic function of user prompt \mathbf{X} and a pseudo-random sequence \mathbf{W} . Each token is chosen from a finite set of tokens $Y_n \in \mathcal{T}$. The model M can produce a number of different output strings \mathbf{Y} for the same user prompt \mathbf{X} corresponding to different values of the pseudo-random sequence \mathbf{W} . This defines a probability distribution $P(\mathbf{Y}|\mathbf{X})$, or more formally, a sequence of probability distributions $P(\mathbf{Y}_N|\mathbf{X})$ over $\mathbf{Y}_N \in \mathcal{T}^N$ where $\mathbf{Y}_N \doteq [Y_1, Y_2, \dots, Y_N]$ is the substring of \mathbf{Y} consisting of the first N tokens.

2.1. Sequential LLM

Practical implementations of LLMs specify the probability distribution iteratively (Radford et al., 2018):

$$P(Y_1, Y_2|\mathbf{X}) = P(Y_1|\mathbf{X})P(Y_2|Y_1, \mathbf{X}) \quad (1)$$

and so on. Thus, the model M first draws a random value for the first token say $Y_1 = y_1$ by sampling from the distribution $P(Y_1|\mathbf{X})$. Then the model determines a distribution for the second token as a function of the initial prompt \mathbf{X} and the randomly chosen first token y_1 . Thus, the second token is randomly sampled from a distribution $P(Y_2|\mathbf{X}, Y_1 = y_1)$ and so on. We can write

$$P(\mathbf{Y}_N|\mathbf{X}) = p_1(Y_1)p_2(Y_2) \dots p_N(Y_N),$$

where $p_n(Y_n) = P(Y_n|Y_1, Y_2, \dots, Y_{n-1}, \mathbf{X})$ (2)

Given a string \mathbf{Y} , open-source LLMs can be programmed to print out the distributions $p_n(Y_n)$ from which its tokens were selected. Specifically, given a user prompt \mathbf{X} and a string of tokens $\mathbf{Y}_N \equiv [Y_1, Y_2, \dots, Y_N]$, it is possible to get a complete listing of the distributions $p_n(Y_n)$, $n = 1 \dots N$. Note that complete knowledge of the $p_n(y)$ is not the same as complete knowledge of $P(\mathbf{Y}|\mathbf{X})$ even for a fixed user prompt \mathbf{X} . As an example, the former requires knowledge of $p_3(y) \equiv P(y|Y_1, Y_2, \mathbf{X})$ for different values of $y \in \mathcal{T}$, but only conditioned on the specific values of Y_1, Y_2 contained in one specific string \mathbf{Y} .

Remark. Equation (2) is simply an application of the Bayes rule of probability theory and it always holds for any generative model regardless of whether the tokens Y_n are sequentially generated. However, the conditional distributions

$p_n(y)$ are not in general easily accessible, so while (2) is true for all generative models, it may only be *useful* for sequential models.

The perplexity $\text{perp}_M(\mathbf{Y}_N) \doteq \prod_{n=1}^N p_n(Y_n)^{-\frac{1}{N}}$ of a (finite length) text string $\mathbf{Y}_N = [Y_1, Y_2, \dots, Y_N]$ for a model M is defined as the per-token inverse likelihood of the string \mathbf{Y} . It is usually more convenient to work with the log-perplexity $l_M(\mathbf{Y}_N)$:

$$\begin{aligned} l_M(\mathbf{Y}_N) &\doteq \log_2(\text{perp}_M(\mathbf{Y}_N)) \\ &\equiv -\frac{1}{N} \sum_{n=1}^N \log_2(p_n(Y_n)) \end{aligned} \quad (3)$$

2.2. A Toy Problem

Let $\alpha(y)$, $\beta(y)$ be two fixed probability distributions over the (discrete) set \mathcal{T} of tokens. Consider a toy problem involving two language models A and B, that each generate a string of tokens $\mathbf{Y} = [Y_1, Y_2, \dots]$ where each token is generated iid from the distribution $\alpha(y)$ and $\beta(y)$ respectively. The iid assumption implies that the tokens Y_n can be thought of as being generated by a stationary and ergodic random process¹.

Consider a long string $\mathbf{A}_N \doteq [A_1, A_2, \dots, A_N]$, $N \gg 1$ randomly generated from model A. Let $p_{\mathbf{A}}(y)$ denote the empirical distribution of $y \in \mathcal{T}$ in the string \mathbf{A}_N :

$$p_{\mathbf{A}}(y) \doteq \frac{n_{\mathbf{A}}(y)}{N} \equiv \frac{1}{N} \sum_{n=1}^N \mathbf{1}(A_n \equiv y), \quad \forall y \in \mathcal{T} \quad (4)$$

where $\mathbf{1}(\cdot)$ denotes the indicator function and $n_{\mathbf{A}}(y)$ is the number of occurrences of token y in the (long) string \mathbf{A}_N . The log-perplexity of string \mathbf{A}_N for model A is:

$$\begin{aligned} l_A(\mathbf{A}_N) &= -\frac{1}{N} \sum_{n=1}^N \log_2(\alpha(Y_n)) \\ &\equiv -\sum_{y \in \mathcal{T}} \frac{n_{\mathbf{A}}(y)}{N} \log_2(\alpha(y)) \equiv H(p_{\mathbf{A}}, \alpha) \end{aligned} \quad (5)$$

where $H(\gamma_1, \gamma_2) \doteq -\sum_{y \in \mathcal{T}} \gamma_1(y) \log_2(\gamma_2(y))$ is the cross-entropy between two distributions $\gamma_1(y)$, $\gamma_2(y)$ over $y \in \mathcal{T}$.

It is well-known $H(\gamma_1, \gamma_2) \geq H(\gamma_1)$ with equality when $\gamma_1 \equiv \gamma_2$, where $H(\gamma) \doteq -\sum_{y \in \mathcal{T}} \gamma(y) \log_2(\gamma(y))$ is the entropy of distribution γ . Thus we have from (5):

$$\begin{aligned} l_A(\mathbf{A}_N) &\equiv H(p_{\mathbf{A}}, \alpha) \geq H(p_{\mathbf{A}}) \\ &\text{with equality iff } p_{\mathbf{A}}(y) \equiv \alpha(y), \quad \forall y \in \mathcal{T} \end{aligned} \quad (6)$$

¹Strictly speaking, a stationary random process “starts” at $n = -\infty$ rather than $n = 1$; here we ignore the tokens produced by the assumed random process before $n = 1$.

The simplest version of the classical Asymptotic Equipartition Theorem (AEP) (Cover & Thomas, 2005a) from information theory states that the log-perplexity of a long string \mathbf{A}_N of iid symbols is almost always very close to the entropy $H(\alpha)$ of the distribution $\alpha(y)$ the symbols are drawn from.

Proposition 2.1. Simple AEP. *For a long text string $\mathbf{A}_N = [A_1, A_2, \dots, A_N]$ of iid tokens A_n drawn from a distribution $\alpha(y)$, the log perplexity $l_A(\mathbf{A}_N)$ as defined in (3) is close to the entropy of the distribution $H(\alpha)$ of the distribution $\alpha(y)$ with high probability:*

$$\lim_{N \rightarrow \infty} \Pr[|l_A(\mathbf{A}_N) - H(\alpha)| > \epsilon] \equiv 0, \forall \epsilon > 0 \quad (7)$$

Proposition 2.1 can be thought of as a direct consequence of the (weak) Law of Large Numbers² applied to the random variable $l_M(\mathbf{A}_N)$:

$$l_M(\mathbf{A}_N) \rightarrow E(l_M(\mathbf{A}_N)) \equiv H(\alpha) \quad (8)$$

From (5) and (6), we see that (8) itself is equivalent to $p_{\mathbf{A}}(y) \equiv \frac{n_{\mathbf{A}}(y)}{N} \rightarrow \alpha(y)$, $\forall y \in \mathcal{T}$ i.e. the empirical distribution $p_{\mathbf{A}}(y)$ for long strings \mathbf{A} is very close to $\alpha(y)$ with high probability. Putting these observations together we have:

$$\begin{aligned} l_A(\mathbf{A}) &\equiv H(p_{\mathbf{A}}, \alpha) \rightarrow H(p_{\mathbf{A}}) \\ &\leq H(p_{\mathbf{A}}, \beta) \equiv l_B(\mathbf{A}) \end{aligned} \quad (9)$$

Intuitively, (9) states that a long string \mathbf{A} generated from model A is likely to have a lower perplexity for model A than for any other model B . This means that a model trained to have low perplexity over a set of reference texts will generate text strings that are statistically similar to the reference texts. This is the theoretical justification for using perplexity as a loss function for training language models and it is an excellent justification - provided only that we accept the Shannon model of language i.e. the idea that languages can be reasonably modeled as a stochastic sequence of tokens (Jakobson, 1961). This same reasoning also underlies popular tools for LLM-related applications as we discuss in Section 5.

2.3. A Modest Generalization

This theoretical analysis relies on Proposition 2.1 which only applies to toy models that generate strings of iid tokens. Of course, the simple AEP in Proposition 2.1 has been extended in various ways in the literature such as the following version (see (Tal et al., 2017), Appendix B.4), which we will make use of in the sequel.

²For all of our results in this paper, convergence is in the “weak” sense i.e. convergence in probability.

Proposition 2.2. Generalized AEP. *Consider a language model A' that generates text string $\mathbf{A}'_N = [A'_1, A'_2, \dots, A'_N]$ where the tokens A'_n are drawn independently from a sequence of distributions $\alpha_n(y)$, $n = 1, 2, \dots$. Assuming the entropies of distributions $\alpha_n(y)$ are uniformly bounded i.e. $H(\alpha_n(y)) < H_{max} < \infty$, $\forall n$, the log perplexity $l_{A'}(\mathbf{A}'_N)$ of the string \mathbf{A}'_N as defined in (3) is close to the average entropy of the distributions $\alpha_n(y)$, $n = 1, 2, \dots, N$ with high probability:*

$$\begin{aligned} \lim_{N \rightarrow \infty} \Pr[|l_{A'}(\mathbf{A}'_N) - H_{A'}(N)| > \epsilon] &\equiv 0, \forall \epsilon > 0, \\ \text{where } H_{A'}(N) &\equiv \frac{1}{N} \sum_{n=1}^N H(\alpha_n) \end{aligned} \quad (10)$$

The regularity condition that the entropies of $\alpha_n(y)$ are uniformly bounded is trivially true if the token dictionary \mathcal{T} is a finite set: $H(p(y)) \leq \log_2 |\mathcal{T}|$ for any distribution $p(y)$ over $y \in \mathcal{T}$.

While Proposition 2.2 does not lend itself to an intuitive interpretation in terms of the empirical distribution of the tokens A'_n , it too is a direct consequence of the Law of Large Numbers applied to the log-perplexity random variable. Much of the analysis in Section 2.2 can be extended to generalized models like A' in Proposition 2.2 that allows each token A'_n to be drawn from different distributions $\alpha_n(y)$. However, the tokens in these models must be drawn from *fixed distributions independent of past tokens*. This is still quite trivial compared to modern language models where the output tokens depend in highly complex and sophisticated ways on past tokens.

3. An Equipartition Property for Perplexity

We now propose a generalization of the theory described in Section 2.2 to a more interesting class of models. Consider a language model M and a random infinitely long text string \mathbf{Y} generated by M , whose probabilities for a given prompt \mathbf{X} is described by (2). Specifically, given a model M and a string \mathbf{Y} , (2) defines a sequence of probability distributions $p_n(y)$ over the set of tokens $y \in \mathcal{T}$. We will assume that the prompt \mathbf{X} is fixed and omit it from our notation in the sequel.

Definition 3.1. The empirical entropy $h_M(\mathbf{Y}_N)$ of model M for string \mathbf{Y}_N is defined as:

$$h_M(\mathbf{Y}_N) \doteq \frac{1}{N} \sum_{n=1}^N H(p_n) \quad (11)$$

where $H(p) \equiv -\sum_{y \in \mathcal{T}} p(y) \log_2(p(y))$ is the entropy of the distribution $p(y)$ over $y \in \mathcal{T}$.

Note that the empirical entropy $h_M(\mathbf{Y}_N)$ is not the same as $\frac{1}{N} H(P(\mathbf{Y}_N))$; the former expression is a random variable,

whereas the latter is the entropy of $P(\mathbf{Y}_N)$ over $\mathbf{Y}_N \in \mathcal{T}^N$ and thus is a constant number. Indeed, we can show that

$$E[l_M(\mathbf{Y}_N)] \equiv E[h_M(\mathbf{Y}_N)] \equiv \frac{1}{N} H(P(\mathbf{Y}_N)) \quad (12)$$

where the expectation is over random text strings \mathbf{Y}_N .

Definition 3.2. Let the log-deviation $\lambda(p)$ of a probability distribution $p(y)$ over $y \in \mathcal{T}$ be defined as:

$$\lambda(p) \doteq \sqrt{S(p) - (H(p))^2},$$

$$\text{where } S(p) \doteq \sum_{y \in \mathcal{T}} p(y) (\log_2 p(y))^2 \quad (13)$$

Note that the entropy $H(p)$ and log-deviation $\lambda(p)$ are the mean and standard deviation of the log-likelihood random variable $l_p(y) \equiv -\log_2 p(y)$ under distribution $p(y)$.

Definition 3.3. The log-deviation $\lambda_M(\mathbf{Y}_N)$ of a string \mathbf{Y}_N for model M is defined as:

$$\lambda_M(\mathbf{Y}_N) \doteq \frac{1}{N} \sqrt{\sum_{n=1}^N \lambda^2(p_n)} \quad (14)$$

We can again interpret $H(p_n)$, $\lambda(p_n)$ as the *conditional* mean and standard deviation of $-\log_2(\Pr(Y_n))$ conditioned on \mathbf{Y}_{N-1} . Clearly, if the log-deviations $\lambda(p_n)$ of the distributions $p_n(y)$ for a string \mathbf{Y} are uniformly upper-bounded, $\lambda_M(\mathbf{Y}_N)$ asymptotically vanishes. We are now ready to state our main result.

Proposition 3.1. AEP for Perplexity. *For a long text string \mathbf{Y} i.e. $N \gg 1$ generated from a language model with a given prompt \mathbf{X} , with high probability, the log perplexity $l_M(\mathbf{Y})$ is close to the empirical entropy $h_M(\mathbf{Y})$. More precisely:*

$$\lim_{N \rightarrow \infty} \Pr[|l_M(\mathbf{Y}_N) - h_M(\mathbf{Y}_N)| > \epsilon] \equiv 0, \forall \epsilon > 0 \quad (15)$$

We want to argue that $\lambda_M(\mathbf{Y}_N)$ is the standard deviation of $l_M(\mathbf{Y}_N) - h_M(\mathbf{Y}_N)$, and therefore asymptotically vanishing $\lambda_M(\mathbf{Y}_N)$ implies $l_M(\mathbf{Y}_N) \rightarrow h_M(\mathbf{Y}_N)$. This, however, is very wrong - for one thing $\lambda_M(\mathbf{Y}_N)$ is itself a random variable. In the sequel, we propose an *Indifference Principle* that allows us to carry through reasoning like the above to prove Proposition 3.1 and potentially other results. But first we present a direct proof of Proposition 3.1.

3.1. Formal Proof: an LLN for LLMs

We will use a version of the Weak Law of Large Numbers that applies to sequences of random variables with no restrictions on their dependence. Such a Law is presented in Appendix B (Proposition B.1). In fact, it turns out that a weaker version of Proposition B.1 namely Lemma B.2 is sufficient for our purposes.

Proof of Proposition 3.1. Consider the sequence of conditional likelihood random variables $\{l_n\}$ defined as:

$$l_n \doteq -\log_2 \Pr(Y_n | \mathbf{Y}_{n-1}) \equiv -\log_2(p_n(Y_n)) \quad (16)$$

We note that the random variables $Z_n \doteq l_n$ satisfy all of the conditions for Lemma B.2, if we impose the restriction that they are strictly bounded i.e. $|l_n| < L$, for some finite L . This is equivalent to requiring that all non-zero token probabilities $p_n(y)$ are above some lower bound e.g. $p_n(y) \geq 10^{-10}$. In practice, this is easily satisfied by any practical LLM. Then from Lemma B.2, we have for any $\epsilon > 0$:

$$\lim_{N \rightarrow \infty} \Pr\left(\left|\frac{1}{N} \sum_{n=1}^N (l_n - H(p_n))\right| > \epsilon\right) = 0 \quad (17)$$

From the definition (16), we note that $\frac{1}{N} \sum_{n=1}^N l_n \equiv l_M(\mathbf{Y}_N)$, and likewise $\frac{1}{N} \sum_{n=1}^N H(p_n) \equiv h_M(\mathbf{Y}_N)$. Combining these observations with (17) gives (15). \square

3.2. Informal Proof: Roads not Traveled

We now present an alternative proof of Proposition 3.1 using a more informal argument. Specifically, we introduce the conceptual device of an auxiliary language model that will allow us to apply the traditional Law of Large Numbers to a sequence of dependent random variables.

3.2.1. CONCEPT OF AN AUXILIARY LANGUAGE MODEL

Let us consider a fixed string \mathbf{S} . For this particular string \mathbf{S} and model M , we have a fixed set of distributions $q_n(y) \doteq \Pr(S_n = y | \mathbf{S}_{n-1})$. Now define an auxiliary generative model M'_S that generates a random string $\mathbf{Y}' \doteq [Y'_1, Y'_2, \dots, Y'_N, \dots]$ where the tokens Y'_n are generated *independently* from the distributions $q_n(y)$. If we represent a language model M as a probability tree, an auxiliary model M'_S represents one specific subtree corresponding to the string \mathbf{S} that defines the model. This is illustrated in Fig. 1. The auxiliary model M'_S is much simpler than the general language model M , because its output tokens Y'_n are generated independently of other tokens. Our informal proof of Proposition 3.1 is based on the following idea.

Proposition 3.2. The Indifference Principle. *For any random event E defined entirely on the subtree of a particular string \mathbf{S} , $\Pr(E)$ will be the same for generative model M as for the auxiliary model M'_S .*

The idea behind Proposition 3.2 is that while calculating $\Pr(E)$ for some event E , we can change any part of the probability tree that do not affect the event E (“the roads not traveled”) to make our calculation easier e.g. by allowing us to work with an auxiliary language model M'_S which

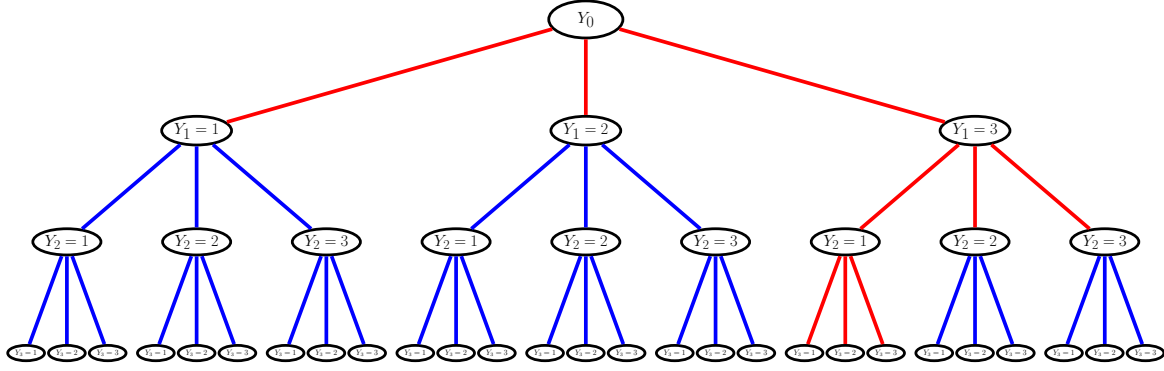


Figure 1. Probability tree for the first 3 tokens of a language model M that selects each token Y_n from a dictionary \mathcal{T} of three tokens. Also shown in red is the probability tree of an auxiliary model M'_S for a string S with $S_1 = 3$, $S_2 = 1$ and so on.

generates strings of independent tokens. For our purposes, Proposition 3.2 applies to all events in the σ -algebra generated by the conditional likelihood random variables l_n as defined in (16).

Informal Proof of Proposition 3.1. Using the Proposition 3.2, we can evaluate the distribution of random variables l_n assuming the tokens Y_n are generated by the auxiliary model M'_S instead of the original model M . Then the random variables l_m , l_n are independent of each other and $E[l_n] \equiv H(q_n)$. The weak Law of Large Numbers for the sequence of independent zero-mean random variables $l_n - H(q_n)$ immediately gives us (17). \square

Proposition 3.2 allows us to use traditional statistical laws that apply to sequences of independent random variables directly without having to prove bespoke versions of those laws such as Proposition B.1. For instance, we may be able to use Proposition 3.2 to formulate a version of the Central Limit Theorem for $\frac{1}{\lambda_M(S_N)}(l_M(S_N) - h_M(S_N))$. We delegate a full development of these ideas to future work and limit ourselves to one simple non-asymptotic result to illustrate the method.

3.3. Chebyshev Bound on Deviations

We note that for the auxiliary model M'_S , the variance of the conditional log likelihood random variable l_n is $\lambda^2(q_n)$ as defined in (13). Assuming independence of l_n using Proposition 3.2, we can show that the variance of $l_M(S_N)$ is $\lambda_M^2(S_N)$ as defined in (14). We can then write using the Chebyshev's Inequality (Cohen, 2015) for any $\alpha > 0$:

$$\Pr(|l_M(S_N) - h_M(S_N)| > \alpha \lambda_M(S_N)) \leq \frac{1}{\alpha^2} \quad (18)$$

3.4. Typical Sets are Vanishingly Small

Let us now formally define the *typical set* $\mathcal{P}_M^n(\epsilon) \subset \mathcal{T}^n$ of *typical strings* \mathbf{Y}_n for a model M with $\epsilon > 0$:

$$\begin{aligned} \mathcal{P}_M^n(\epsilon) &\doteq \{\mathbf{Y}_n : |l_M(\mathbf{Y}_n) - h_M(\mathbf{Y}_n)| < \epsilon\} \\ &\equiv \left\{ \mathbf{Y}_n : \left| h_M(\mathbf{Y}_n) + \frac{1}{n} \log_2(\Pr(\mathbf{Y}_n)) \right| < \epsilon \right\} \end{aligned} \quad (19)$$

Proposition 3.1 asserts that with high probability a stochastic model M is constrained to only output typical strings from this set $\mathcal{P}_M^n(\epsilon)$ for any $\epsilon > 0$ if we consider sufficiently long strings. This observation has profound consequences. We will now show one such consequence, namely that the typical set $\mathcal{P}_M^n(\epsilon)$ is very small:

Proposition 3.3. *The size of the typical set \mathcal{P}_M^n is an exponentially vanishing fraction $2^{-\gamma n}$, $\gamma > 0$ of all grammatically correct sequences.*

The argument below closely tracks standard information-theoretic results (Cover & Thomas, 2005a) about the size of the typical set. We start by restating Proposition B.1 informally as:

$$\Pr(\mathbf{Y}_n) \equiv 2^{-nl_M(\mathbf{Y}_n)} \rightarrow 2^{-nh_M(\mathbf{Y}_n)} \quad (20)$$

Let $T_g(n)$ be the total number of grammatically correct strings \mathbf{Y}_n of length n . The average entropy $h_M(\mathbf{Y})$ is a measure of the richness of the model that generated the string. Language models typically have a “temperature” parameter T in the range $T \in [0, 1]$ that can be used to tune the richness of the model’s output. Higher temperature means less “predictable” (i.e. less deterministic) and more “creative” (i.e. more random) outputs. This corresponds to higher values for $h_M(\mathbf{Y})$.

Consider the *maximally rich* model of length n that has the highest possible entropy $h_M(\mathbf{Y}_n) \equiv h_{max}(n)$ which corresponds to choosing all possible (grammatically correct)

strings of length n with equal probability³. By definition, for the maximally rich model, the probability of choosing any valid string is $2^{-nh_{max}(n)}$ and this means the number of possible strings of length n is:

$$T_g(n) \equiv 2^{nh_{max}(n)} \quad (21)$$

We can take (21) as the defining relation for $h_{max}(n)$. In practical LLMs, even at the highest temperature setting i.e. $T = 1$, the token probability is typically concentrated in a small number of tokens and the token probabilities are determined by far more complex rules than the richness of their sub-trees. Let us formalize this as an assumption.

Assumption 3.1. *Practical LLMs are not entropy-maximizing.* Any language model M has an entropy gap $\Delta h > 0$ such that the following holds for all sufficiently long strings:

$$h_M(\mathbf{Y}_n) < h_{max}(n) - \Delta h, \forall \mathbf{Y}_n \in \mathcal{T}^n \quad (22)$$

Proof of Proposition 3.3. From (20), for $n \gg 1$, we have $\Pr(\mathbf{Y}_n) \equiv 2^{-n l_M(\mathbf{Y}_n)} \rightarrow 2^{-n h_M(\mathbf{Y}_n)} > 2^{-n(h_{max}(n) - \Delta h)}$ where we used Assumption 3.1 in the last step. Since this holds for any typical string of length n , we must have for small $\epsilon > 0$ and correspondingly large n :

$$|\mathcal{P}_M^n(\epsilon)| < 2^{n(h_{max}(n) - \Delta h)} \equiv 2^{-n \Delta h} |T_g(n)| \quad (23)$$

Thus, the typical set $\mathcal{P}_M^n(\epsilon)$ represents an exponentially vanishing fraction of all grammatically correct strings. \square

We emphasize that all we needed to establish Proposition 3.3 was Assumption 3.1 which only requires that the model assigns higher probability to some sentences than to others.

3.5. Over-Typical Sets are Vanishingly Smaller Still

The obverse of Proposition 3.3 is that almost all grammatically correct strings are non-typical for a given language model M . A string can be non-typical in two different ways depending on whether its perplexity is abnormally high (“under-typical”) or abnormally low (“over-typical”). Formally, we define the set $\mathcal{V}_M^n(\epsilon)$ of over-typical strings of length n for a model M as:

$$\begin{aligned} \mathcal{V}_M^n(\epsilon) &\doteq \{\mathbf{Y}_n : (h_M(\mathbf{Y}_n) - l_M(\mathbf{Y}_n)) > \epsilon\} \\ &\equiv \{\mathbf{Y}_n : \Pr(\mathbf{Y}_n) > 2^{-n(h_M(\mathbf{Y}_n) - \epsilon)}\} \end{aligned} \quad (24)$$

Proposition 3.4. *The size of the over-typical set $\mathcal{V}_M^n(\epsilon)$ is an exponentially vanishing fraction of all grammatically correct sequences.*

³Note that this does not mean that the token distributions $p_m(y)$ are uniform over all valid tokens at each step m ; indeed tokens corresponding to richer sub-trees as shown in Fig. 1 should be chosen with higher probability to maximize the overall richness of the model’s output.

Proof. For any $\mathbf{Y}_n \in \mathcal{V}_M^n(\epsilon)$, we have from (24) and (22):

$$\Pr(\mathbf{Y}_n) > 2^{-nh_{max}(n)} \times 2^{n(\Delta h + \epsilon)} \quad (25)$$

Therefore, we must have

$$|\mathcal{V}_M^n(\epsilon)| < |T_g(n)| 2^{-n(\Delta h + \epsilon)}$$

where we used the definition of h_{max} in (21). \square

We have shown that almost all long grammatically correct strings are *under-typical* for any language model M . We illustrate these ideas for a simple Markov language model in Appendix A.

4. Experiments with Open-Source Models

We performed a series of experiments to verify the ideas described in Section 3. The basic idea is to use a local instance of a pre-trained open-source model to evaluate the empirical entropy of a given string using the model’s conditional probability distributions and compare the result with the log-perplexity of the string again calculated using the model’s probability distributions. We ran experiments on the smaller GPT-2 model (Radford et al., 2019) as well as the more recent, larger and more capable Llama 3.1 8B (Touvron et al., 2023) model.

4.1. Perplexity of Self-Generated Strings

Our first set of experiments consisted of repeatedly generating a long text string from a model (GPT-2 or Llama 3.1 8B) and then evaluating its log-perplexity, empirical entropy and log-deviations for the same model.

We initialized the prompt string and the seed of random number generator with fixed values to allow for later reproduction. We then used top-k sampling with $k = 100$ to generate the probability distribution for the next token. We normalized the probabilities of the top-k tokens to sum to unity. We then created and saved a table consisting of token ID, token string and selection probability of each of the top k tokens. Finally, a random next token is chosen by sampling from the normalized distribution, and the new token is appended to the prompt to repeat the process for N_{max} total tokens. We want the number of tokens N_{max} to be large enough to observe the asymptotic behavior of the perplexity of the generated string. In practice, N_{max} is constrained by certain limitations of the models as we discuss in the sequel.

The random token Y_n chosen at each step along with the probability distribution $p_n(y)$ from which they were chosen were saved into a json file for later processing. We ran this experiment many times for different seed values for the random number generator, different choices of sampling distributions and a number of different initial prompts.

For each generated string, we can then analyze the information archived in the json file to test our claim in Proposition 3.1. Specifically, for each token $N \in 1 \dots N_{max}$ we calculated the log-perplexity $l_M(\mathbf{Y}_N)$, empirical entropy $h_M(\mathbf{Y}_N)$ and the log-deviation $\lambda_M(\mathbf{Y}_N)$ of the substring \mathbf{Y}_N consisting of the first N tokens of the generated string.

4.2. Perplexity of Externally Generated Strings

We also used a slightly modified version of the procedure described in Section 4.1 to calculate the log-perplexity and empirical entropy of an arbitrary string (that was not generated by the model itself) on the GPT-2 and Llama 3.1 8B models. We first choose a text string that we wish to analyze. We fix an initial fragment of the string as our prompt \mathbf{X} . We tokenize the remaining portion of the string excluding the prompt using the model’s tokenizer. Let Y_n , $n = 1 \dots N_{max}$ denote the resulting sequence of tokens.

Stating with $n = 1$, we list the probability distribution $p_n(y)$ of the next token Y_n for the model using top-k with $k = 100$ sampling for the n ’th token. We can then calculate the empirical entropy $h_M(\mathbf{Y}_N)$ and log-deviation $\lambda_M(\mathbf{Y}_N)$ as in Section 4.1. However, to calculate the log-perplexity $l_M(\mathbf{Y}_N)$, we also need the probability that our model will output the actual next token Y_n from the string \mathbf{Y} which may not be included in the list of top- k tokens⁴. When this happens, we replaced the lowest probability entry in the list of top- k tokens with the actual next token from our string along with its probability.

The normalized probability of each of the top- k tokens along with their token-ID and token strings, as well as the probability and ID of the actual next token are saved in a json file. This json file is then processed in exactly the same way as in Section 4.1 to compute the log-perplexity $l_M(\mathbf{Y}_N)$, empirical entropy $h_M(\mathbf{Y}_N)$ and the log-deviation $\lambda_M(\mathbf{Y}_N)$ of the sub-string \mathbf{Y}_N for the GPT-2 model for $N = 1 \dots N_{max}$.

4.3. Results and Discussion

Figure 2 shows the log-perplexity and empirical entropy for sub-strings \mathbf{Y}_N for the GPT-2 or Llama 3.1 8B model as a function of N for several text strings generated by the same model. The plots also shows the range $h_M(\mathbf{Y}_N) \pm \lambda_M(\mathbf{Y}_N)$ and $h_M(\mathbf{Y}_N) \pm 2\lambda_M(\mathbf{Y}_N)$. We see that the log-perplexity $l_M(\mathbf{Y}_N)$ stays within the range $h_M(\mathbf{Y}_N) \pm 2\lambda_M(\mathbf{Y}_N)$ consistently in all cases. The asymptotic convergence is more clearly seen in the long strings from the Llama 3.1 8B

⁴Note that our choice of top- k sampling is a compromise. The number of tokens in GPT-2 or Llama is quite large ($|\mathcal{T}| \approx 40,000$). If we exhaustively list the full probability distribution of all tokens, our experiment becomes very computationally expensive and slow.

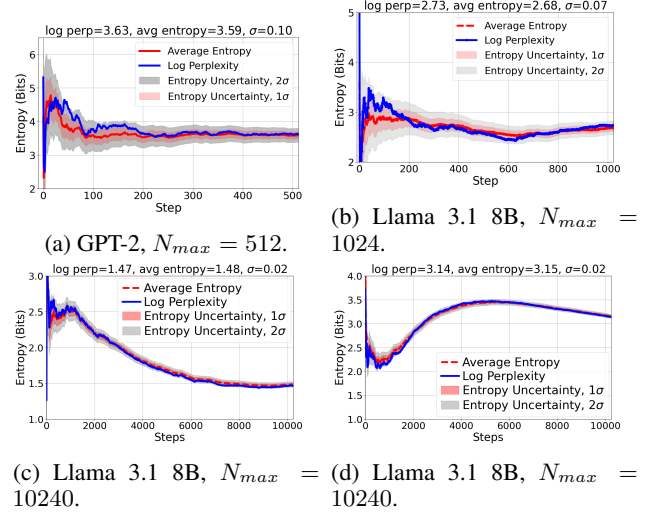


Figure 2. Log-perplexity and empirical entropy for GPT-2 or Llama 3.1 8B of self-generated text, top-100 sampling.

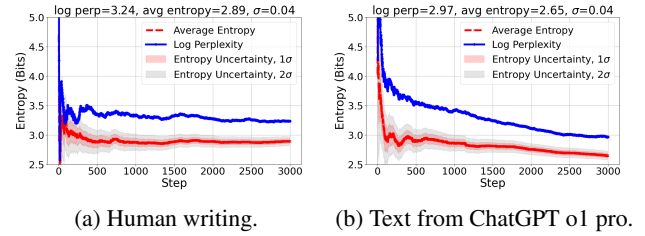


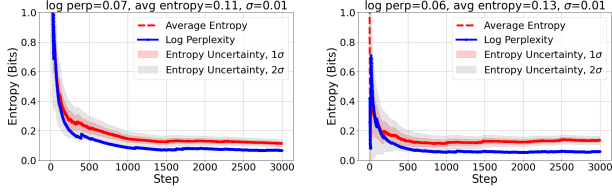
Figure 3. Log-perplexity and empirical entropy for Llama 3.1 8B of non-Llama-generated strings \mathbf{Y} .

model⁵ in Fig. 2c, 2d. We find that very long synthetic texts such as those in Fig. 2c, 2d can be ungrammatical, repetitive and generally of poor quality, as reported elsewhere (Holtzman et al., 2020). However, Proposition 3.1 holds regardless of the quality of the text as seen in Fig. 2.

Figure 3 shows the log-perplexity and empirical entropy for the Llama 3.1 8B model for two long, non-Llama generated strings. The text used in Fig. 3a is an excerpt from a recent article (Pelly, 2025) in Harper’s Magazine, a respected literary publication. The string used in Fig. 3b was generated by ChatGPT o1 Pro (OpenAI, 2024).

We see from Fig. 3a and Fig. 3b that the log-perplexity $l_M(\mathbf{Y}_N)$ for Llama 3.1 8B is several standard deviations $\lambda_M(\mathbf{Y}_N)$ larger than the empirical entropy $h_M(\mathbf{Y}_N)$ showing that the corresponding strings are clearly *under-typical* for the Llama 3.1 8B model. While this non-convergence is not *required* by our theory - we have not considered whether the converse to Proposition 3.1 is true - it is strongly suggested by Propositions 3.3 and 3.4 if we assume that the

⁵The GPT-2 model cannot generate similarly long texts.



(a) Alice in Wonderland excerpt. (b) The Great Gatsby excerpt.

Figure 4. Log-perplexity and empirical entropy for Llama 3.1 8B of excerpts from literary classics.

strings in Fig. 3 are randomly sampled from the set of all grammatically correct texts.

4.4. More Catholic than the Pope

This brings us to the remarkable plots in Fig. 4 which shows the log-perplexity of excerpts from classic texts for Llama 3.1 8B, specifically from Alice in Wonderland and The Great Gatsby in Figs. 4a and 4b respectively. The average entropy for the token distributions in Fig. 4a is an astonishingly small 0.11 bits, which means that the model on average assigns a very high probability ($\sim 99\%$) to the precise sequence of tokens from the excerpt for its predicted next tokens. In other words, *this excerpt has been memorized by the Llama 3.1 8B model*. The same observation also applies to Fig. 4b. We also see from both plots that the log-perplexity is several standard deviations *below* the empirical entropy making the excerpts slightly, but unambiguously, *over-typical* for the Llama-8B model: the model judges these texts to be more perfect than its own creations!

From Proposition 3.4, we can conclude that these excerpts are among the small number of very special strings to be so memorized by the model. Clearly, these texts were part of the training set of the Llama-8B model and they were esteemed highly by its cost function.

Interestingly, the over-typicality of these excerpts means that it is pretty much impossible for the Llama-8B model itself to generate them verbatim - even with a 99% probability that each next token will exactly match the excerpt, a randomly generated string can be expected to diverge from the excerpt after only about 100 tokens.

Additional plots and more details about these experiments are presented in Appendix C.1.

5. Practical Applications

Consider the problem of AI text detection i.e. determining whether a string \mathbf{Y} was generated by a model M . The intuition behind (9) suggests a simple method for doing this: check if the perplexity is small enough. Specifically, declare ‘yes’ if $l_M(\mathbf{Y}) < l_{th}$ for a threshold l_{th} . The same test

can also be used for the problem of LLM dataset inference i.e. determining whether a string \mathbf{Y} was used as part of the training dataset for the model M .

While, this idea is not new and indeed variations of it are behind popular solutions for both problems (Mitchell et al., 2023; Maini et al., 2024), our results in this paper provide a rigorous theoretical foundation for it. It also provides a principled way to choose the classification threshold l_{th} which previously had to be computed empirically (Carlini et al., 2021). As an example, with $l_{th} = h_M(\mathbf{Y}) + 3\lambda_M(\mathbf{Y})$, Fig. 2 shows that we can accurately distinguish between synthetic text generated by Llama 3.1 8B and non-Llama text such as in Fig. 3 even with relatively short strings of 1000 tokens. We can also provide performance guarantees. Indeed, a simple lower-bound on the probability of a false negative (i.e. failing to detect a Llama-generated string) can be obtained from (18). Similar comments apply to the dataset inference problem: while it is well-known that text strings memorized by an LLM have low perplexity, our theory allows us to rigorously derive threshold values and performance guarantees. Extrapolating from Fig. 4, we posit that *if a long excerpt from a human-written book is over-typical for a language model, it is a near-statistical certainty that the book was part of the training data set for that model*.

6. Conclusions

We proposed an asymptotic property that must be satisfied by the perplexity of any long string generated by a language model and provided theoretical and experimental arguments in support of the property. While this paper focused on a rigorous exposition of the AEP property, it opens up many questions for further study. An important open problem is to design practical algorithms for AI text detection and dataset inference using the typical set concept. Tighter non-asymptotic bounds on deviations that improve on (18) will significantly benefit practical applications by providing better performance guarantees. More fundamentally, Proposition 3.3 leads naturally to the question of how much do the typical sets of different models overlap. The theory presented in this paper is neutral on this question which must be answered empirically and both possible answers have interesting implications. No overlap between typical sets means that each model can be associated with a unique statistical signature. On the other hand, independently trained models converging on a common set of typical strings, would be a notable linguistic phenomenon, though it is hard to speculate on what this might mean. The AEP is also just one example of the broader observation that generative AI models are stochastic machines subject to statistical laws and the discovery of other statistical regularities in the outputs of generative models is an important topic for future work.

References

- Anthropic. Claude 3 Opus. <https://claude.ai>, 2024. URL <https://www.anthropic.com>.
- Brown, T., Mann, B., and et al. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, Ú., Oprea, A., and Raffel, C. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650. USENIX Association, August 2021. ISBN 978-1-939133-24-3. URL <https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting>.
- Cohen, J. E. Markov’s inequality and chebyshev’s inequality for tail probabilities: A sharper image. *The American Statistician*, 69(1):5–7, 2015. doi: 10.1080/00031305.2014.975842.
- Cohen, T. and Pakhomov, S. A tale of two perplexities: Sensitivity of neural language models to lexical retrieval deficits in dementia of the Alzheimer’s type. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1946–1957, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.176. URL <https://aclanthology.org/2020.acl-main.176>.
- Cover, T. and Thomas, J. Asymptotic equipartition property. In *Elements of Information Theory*, chapter 3, pp. 57–69. John Wiley & Sons, Ltd, 2005a. ISBN 9780471748823. doi: <https://doi.org/10.1002/047174882X.ch3>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/047174882X.ch3>.
- Cover, T. and Thomas, J. Entropy, relative entropy, and mutual information. In *Elements of Information Theory*, chapter 2, pp. 13–55. John Wiley & Sons, Ltd, 2005b. ISBN 9780471748823. doi: <https://doi.org/10.1002/047174882X.ch2>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/047174882X.ch2>.
- Csiszar, I. The method of types [information theory]. *IEEE Transactions on Information Theory*, 44(6):2505–2523, 1998. doi: 10.1109/18.720546.
- Demberg, V. and Keller, F. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210, 2008. ISSN 0010-0277. doi: <https://doi.org/10.1016/j.cognition.2008.07.008>. URL <https://www.sciencedirect.com/science/article/pii/S0010027708001741>.
- Gamallo, P., Campos, J. R. P., and Alegria, I. A perplexity-based method for similar languages discrimination. In *Proceedings of the fourth workshop on NLP for similar languages, varieties and dialects (VarDial)*, pp. 109–114, 2017.
- Gehrmann, S., Strobelt, H., and Rush, A. M. 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, 2019.
- Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=rygGQyrFvH>.
- Idele, S. I. The Law of Large Numbers: Some Issues of Interpretation and Application for Beginners. *Journal of the Royal Statistical Society Series D: The Statistician*, 28(3):209–217, 12 2018. ISSN 2515-7884. doi: 10.2307/2987870. URL <https://doi.org/10.2307/2987870>.
- Jakobson, R. Linguistics and communications theory. 1961.
- Levy, R. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177, 2008. ISSN 0010-0277. doi: <https://doi.org/10.1016/j.cognition.2007.05.006>. URL <https://www.sciencedirect.com/science/article/pii/S0010027707001436>.
- Maini, P., Jia, H., Papernot, N., and Dziedzic, A. Llm dataset inference: Did you train on my dataset?, 2024. URL <https://arxiv.org/abs/2406.06443>.
- Meister, C. and Cotterell, R. Language model evaluation beyond perplexity. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5328–5339, 2021.
- Miaschi, A., Brunato, D., Dell’Orletta, F., and Venturi, G. What makes my model perplexed? a linguistic investigation on neural language models perplexity. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 40–47, 2021.

- Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., and Finn, C. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24950–24962. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/mitchell23a.html>.
- Moy, S.-T. C. Generalizations of shannon-mcmillan theorem. *Pacific Journal of Mathematics*, 11(2):705–714, 1961.
- Nishiara, M. and Morita, H. On the aep of word-valued sources. *IEEE Transactions on Information Theory*, 46(3):1116–1120, 2000. doi: 10.1109/18.841193.
- OpenAI. Openai o1 system card, 2024. URL <https://arxiv.org/abs/2412.16720>.
- OpenAI. Hello GPT-4o. <https://openai.com/index/hello-gpt-4o>, 2024.
- Pelly, L. The ghosts in the machine. *Harper’s Magazine*, pp. 25–32, Jan 2025.
- Pinar Saygin, A., Cicekli, I., and Akman, V. Turing test: 50 years later. *Minds and machines*, 10(4):463–518, 2000.
- Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- Tal, O., Tran, T. D., and Portegies, J. From typical sequences to typical genotypes. *Journal of Theoretical Biology*, 419:159–183, 2017. ISSN 0022-5193. doi: <https://doi.org/10.1016/j.jtbi.2017.02.010>. URL <https://www.sciencedirect.com/science/article/pii/S0022519317300656>.
- Timo, R., Blackmore, K., and Hanlen, L. Word-valued sources: An ergodic theorem, an aep, and the conservation of entropy. *IEEE Transactions on Information Theory*, 56(7):3139–3148, 2010. doi: 10.1109/TIT.2010.2046251.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023. URL <https://arxiv.org/abs/2302.13971>.

A. A Simple Markov Language Model

We now illustrate the ideas from Section 3 for a simple Markov language model, M_1 with a binary token space consisting of only two tokens $\mathcal{T} = \{t_1, t_2\}$ and a parameter $\rho \in (0, 1)$. The user prompt \mathbf{X} consists of a single token which we can set to be t_1 without loss of generality: $\mathbf{X} = [t_1]$. The model M_1 generates a text string of length N following a simple probabilistic algorithm: the n 'th token Y_n is set equal to the $n - 1$ 'th token with probability ρ . The prompt serves as the initial token i.e. $Y_0 \doteq t_1$. Let $h_b(\rho) \doteq -\rho \log_2 \rho - (1 - \rho) \log_2(1 - \rho)$ denote the binary entropy function with argument ρ .

It is easy to see that $H(p_n) \equiv h_b(\rho)$, $\forall n$ and $H_{M_1}(\mathbf{Y}) \equiv h_b(\rho)$ for any text string \mathbf{Y} . For $\rho = 0.9$, $H_{M_1}(\mathbf{Y}) \equiv h_b(0.9) \approx 0.47$

Example A.1. Consider the string \mathbf{Y}_1 with all tokens $Y_n = t_1$, $\forall n = 1 \dots N$. The log-perplexity is given by $l_{M_1}(\mathbf{Y}_1) = -\log_2(\rho)$. For $\rho = 0.9$, $l_{M_1}(\mathbf{Y}_1) = -\log_2(0.9) \approx 0.15$.

Example A.2. Consider the string \mathbf{Y}_2 with alternating token values: $\mathbf{Y}_2 \equiv [t_2, t_1, t_2, t_1, t_2, t_1 \dots]$. The log-perplexity is given by $l_{M_1}(\mathbf{Y}_2) = -\log_2(1 - \rho)$. For $\rho = 0.9$, $l_{M_1}(\mathbf{Y}_2) = -\log_2(0.1) \approx 3.3$.

Note that \mathbf{Y}_1 and \mathbf{Y}_2 are the most likely and least likely string of length N respectively for $\rho = 0.9$ (or indeed for any $\rho > 0.5$). However, it is clear that neither string is typical for any reasonable ϵ with the perplexity of \mathbf{Y}_1 being abnormally low and that of \mathbf{Y}_2 abnormally high respectively.

Example A.3. Consider the string \mathbf{Y}_3 created by flipping a fair coin repeatedly to generate a vector whose elements are iid randomly selected with $\Pr(t_1) = \Pr(t_2) = 0.5$. Since \mathbf{Y}_3 is a random string, its perplexity $l_{M_1}(\mathbf{Y}_3)$ is also a random variable. However, for long strings i.e. $N \gg 1$, with high probability, $l_{M_1}(\mathbf{Y}_3) \approx -0.5 \log_2 \rho - 0.5 \log_2(1 - \rho)$. For $\rho = 0.9$, $l_{M_1}(\mathbf{Y}_3) \approx 1.74$ which is incidentally the arithmetic mean of $l_{M_1}(\mathbf{Y}_1)$ and $l_{M_1}(\mathbf{Y}_2)$.

Clearly, with high probability, the string \mathbf{Y}_3 is not typical. Indeed, from these examples, we can see that typical strings for the model M_1 are precisely those that contain close to $(1 - \rho)N$ token 'flips' i.e. instances when $Y_{n+1} \neq Y_n$.

B. A Generalized Law of Large Numbers for Random Variables with Arbitrary Dependence

Consider a infinitely long string $\mathbf{Y} = [Y_1 Y_2 \dots Y_n \dots]$ consisting of a sequence of symbols $Y_n \in \mathcal{Y}$ generated stochastically from a finite dictionary \mathcal{Y} . We will denote the substring consisting of the first N symbols as $\mathbf{Y}_N \equiv [Y_1 Y_2 \dots Y_N]$ and the joint distribution of the symbols $P(\mathbf{Y}_N)$. We make no assumptions about the distribution $P(\mathbf{Y}_N)$ from which the symbols Y_n are generated. In particular, we do not assume that the symbols are independent or weakly dependent even asymptotically. We also do not assume stationarity i.e. the marginal distributions of symbols Y_n and Y_m can be completely different. We also do not assume any Markovian or conditional independence properties.

B.1. Conditional Means and Sample Means for Sequences of Dependent Random Variables

Consider a sequence of random variables $\mathbf{Z} = [Z_1 Z_2 \dots Z_n \dots]$ whose n 'th element Z_n is a function of the first n symbols Y_1, Y_2, \dots, Y_n i.e. $Z_n \doteq g_n(\mathbf{Y}_n)$ and $g_n : \mathcal{Y}^n \rightarrow \mathbb{R}$ are functions that we will assume to be uniformly bounded i.e. $|g_n(\mathbf{Y}_n)| \leq G < \infty, \forall n \in \mathbb{Z}^+, \mathbf{Y}_n \in \mathcal{Y}^n$.

We define the conditional mean and conditional variance of the random variables Z_n :

$$\mu_n^{(\mathbf{Y})} \doteq \sum_{y \in \mathcal{Y}} p_n^{(\mathbf{Y})}(y) g_n(\mathbf{Y}_n) \equiv E[Z_n | \mathbf{Y}_{n-1}] \quad (26)$$

$$V_n^{(\mathbf{Y})} \doteq \sum_{y \in \mathcal{Y}} p_n^{(\mathbf{Y})}(y) \left(g_n(\mathbf{Y}_n) - \mu_n^{(\mathbf{Y})} \right)^2 \equiv E[Z_n^2 | \mathbf{Y}_{n-1}] - \left(\mu_n^{(\mathbf{Y})} \right)^2 \quad (27)$$

where the distribution $p_n^{(\mathbf{Y})}(y) \doteq \Pr(Y_n = y | \mathbf{Y}_{n-1})$. Since the random variables Z_n are discrete-valued and bounded, they have bounded moments for any distribution:

$$|\mu_n^{(\mathbf{Y})}| \leq G, V_n^{(\mathbf{Y})} \leq G^2 \quad (28)$$

A more formal statement. Let \mathcal{F}_n be the σ -algebra defined by $\mathbf{Y}_n \equiv [Y_1 Y_2 \dots Y_n]$ i.e. \mathcal{F}_n is a collection that includes all (mathematically reasonable) events defined on the first n symbols. Then the random variables Z_n as we have defined them are defined by the condition that Z_m is \mathcal{F}_m -measurable, and $\mu_m^{(\mathbf{Y})} \equiv E[Z_m | \mathcal{F}_{m-1}]$.

We need some intermediate results before presenting our main result. First, we define the partial sums of the sequence Z_n and conditional means:

$$S_n \doteq \sum_{m=1}^n Z_m, \mu_{S_n}^{(\mathbf{Y})} \doteq \sum_{m=1}^n \mu_m^{(\mathbf{Y})} \quad (29)$$

Clearly, $E[Z_n] \equiv E[\mu_n^{(\mathbf{Y})}]$ and $E[S_n] \equiv E[\mu_{S_n}^{(\mathbf{Y})}]$.

Lemma B.1. *The variance of the zero-mean random variable $S_n - \mu_{S_n}^{(\mathbf{Y})}$ can be written as sums of contributions from each of the Z_m 's. Specifically, we have:*

$$V_{S_n} \doteq E \left[\left(S_n - \mu_{S_n}^{(\mathbf{Y})} \right)^2 \right] = \sum_{m=1}^n E \left[V_m^{(\mathbf{Y})} \right] \quad (30)$$

Proof. We will use the method of induction to prove (30). For $n = 1$, (30) holds trivially. Now, let's assume (30) holds for $n = l$ i.e. assume that:

$$V_{S_l} \equiv \sum_{m=1}^l E \left[V_m^{(\mathbf{Y})} \right] \quad (31)$$

Now consider $S_{l+1} \equiv S_l + Z_{l+1}$. Its variance can be written as:

$$V_{S_{l+1}} = E \left[\left(S_l + Z_{l+1} - \mu_{S_l}^{(\mathbf{Y})} - \mu_{l+1}^{(\mathbf{Y})} \right)^2 \right] \quad (32)$$

$$= V_{S_l} + E \left[V_{l+1}^{(\mathbf{Y})} \right] - 2 E \left[\left(S_l - \mu_{S_l}^{(\mathbf{Y})} \right) \left(Z_{l+1} - \mu_{l+1}^{(\mathbf{Y})} \right) \right] \quad (33)$$

We will now show that the last term in (33) is zero. We apply the law of iterated expectations to write:

$$E \left[\left(S_l - \mu_{S_l}^{(\mathbf{Y})} \right) \left(Z_{l+1} - \mu_{l+1}^{(\mathbf{Y})} \right) \right] = E_{\mathbf{Y}_l} \left[E \left[\left(S_l - \mu_{S_l}^{(\mathbf{Y})} \right) \left(Z_{l+1} - \mu_{l+1}^{(\mathbf{Y})} \right) \mid \mathbf{Y}_l \right] \right] \quad (34)$$

$$= E_{\mathbf{Y}_l} \left[\left(S_l - \mu_{S_l}^{(\mathbf{Y})} \right) E \left[\left(Z_{l+1} - \mu_{l+1}^{(\mathbf{Y})} \right) \mid \mathbf{Y}_l \right] \right] \equiv 0 \quad (35)$$

where we used the fact that $S_l - \mu_{S_l}^{(\mathbf{Y})}$ is a deterministic function of \mathbf{Y}_l and $\mu_{l+1}^{(\mathbf{Y})} \equiv E[Z_{l+1} \mid \mathbf{Y}_l]$.

Remark. An alternative way to arrive at (35) is to treat $\mu_{l+1}^{(\mathbf{Y})} \equiv E[Z_{l+1} \mid \mathbf{Y}_l]$ as an estimate of Z_{l+1} given \mathbf{Y}_l , and then use the famous Orthogonality Principle to argue that the estimation error $(Z_{l+1} - \mu_{l+1}^{(\mathbf{Y})})$ must be uncorrelated with any function of \mathbf{Y}_l . More formally, $(S_l - \mu_{S_l}^{(\mathbf{Y})})$ is \mathcal{F}_l -measurable and is therefore orthogonal to $(Z_{l+1} - E[Z_{l+1} \mid \mathcal{F}_l])$.

Using (35), we can now simplify (33) to:

$$V_{S_{l+1}} = V_{S_l} + E[V_{l+1}^{(\mathbf{Y})}] \equiv \sum_{m=1}^{l+1} E[V_m^{(\mathbf{Y})}] \quad (36)$$

where we used the induction assumption (31). We have shown that if (30) holds for $n = l$, then (30) must also hold for $n = l + 1$, thus completing the induction. \square

Lemma B.2. *The sequence $\frac{1}{n} (S_n - \mu_{S_n}^{(\mathbf{Y})})$ converges to zero in probability:*

$$\lim_{n \rightarrow \infty} \Pr \left(\frac{1}{n} \left| S_n - \mu_{S_n}^{(\mathbf{Y})} \right| > \epsilon \right) = 0, \forall \epsilon > 0 \quad (37)$$

Proof. Using Lemma B.1, we can write:

$$V_{S_n} = \sum_{m=1}^n E[V_m^{(\mathbf{Y})}] \leq nG^2 \quad (38)$$

using the simple bounds in (28). The variance of $\frac{1}{n} (S_n - \mu_{S_n}^{(\mathbf{Y})})$ is simply $\frac{1}{n^2} V_{S_n} \leq \frac{G^2}{n} \rightarrow 0$ as $n \rightarrow \infty$. Equation (37) now follows immediately using the Chebyshev Inequality (Cohen, 2015). \square

Lemma B.2 states that the sample means of the two random sequences $\{Z_n\}$ and $\{\mu_n^{(\mathbf{Y})}\}$ must be close to each other asymptotically. Since we have made very few assumptions about the random variables involved, we cannot assume that either of the sequences $\{Z_n\}$ or $\{\mu_n^{(\mathbf{Y})}\}$ themselves converge in any sense to an asymptotic limit. Neither can we assume that their sample means separately converge to a limit.

With an additional assumption stating the existence of one such limit, we can establish a Weak Law of Large Numbers generalized to its most extreme i.e. a statement that the sample mean of a sequence of random variables $\{Z_n\}$ with arbitrary dependence converges asymptotically to a suitably defined average quantity in probability.

Proposition B.1. A Weak Law of Large Numbers. *If the average of the conditional means converges to a limit i.e. $\mu^{(\mathbf{Y})} \equiv \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{m=1}^n \mu_m^{(\mathbf{Y})}$ exists, then the sequence $\frac{1}{n} S_n$ converges in probability to $\mu^{(\mathbf{Y})}$:*

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{m=1}^n Z_m \rightarrow_p \mu^{(\mathbf{Y})} \quad \text{or} \quad \lim_{n \rightarrow \infty} \Pr \left(\left| \frac{1}{n} S_n - \mu^{(\mathbf{Y})} \right| > \epsilon \right) = 0, \forall \epsilon > 0 \quad (39)$$

Proof. Combining Lemma B.2 with the observation $\frac{1}{n} \mu_{S_n}^{(\mathbf{Y})} \rightarrow \mu^{(\mathbf{Y})}$ completes the proof. \square

B.2. Discussion

Proposition B.1 is remarkable because unlike traditional versions of the LLN, *it makes no assumptions whatsoever about the dependence or correlation* between the symbols Y_n . It does not even require for instance that the correlation between Y_1 and Y_n should decrease as n becomes large.

The two extreme cases of strong dependence and complete independence for the symbols Y_n turn out to be two simple special cases for our proposed LLN:

- If the symbols Y_n are independent, Proposition B.1 simply reduces to the classical weak LLN.
- If the symbols are perfectly correlated, the Z_m become deterministic functions of the first token, and the LLN then becomes a trivial observation.

C. Supplemental Material

C.1. Additional Experimental Context and Results

In this work, all experiments were ran on a high-performance computing server running Linux (Ubuntu). AMD EPYC 7413 24-Core Processor, 48 logical cores, 96 GB RAM, four NVIDIA A100-SXM4-80GB GPUs. On this setup, each run of a text generation (i.e., running GPT-2 to generate $N_{\max} = 512$ tokens, Llama-8B to generate $N_{\max} = 1024$ tokens and $N_{\max} = 10240$ tokens, their distributions, etc.), file I/O, data analysis and plotting takes several seconds to minutes, with each experiment taking about 90 minutes to run in the case of 10,240 steps. We were able to run a few hundred of these experiments for this paper from which selected examples are presented in Section 4.3.

Figure 5 shows additional results similar to Fig. 2 where we naively varied the seed value of the random number generator from 209 – 214 for Llama-8B. We present these to show that Fig. 2’s results are representative of the general case and were not cherry-picked. In all cases, $K = 100$.

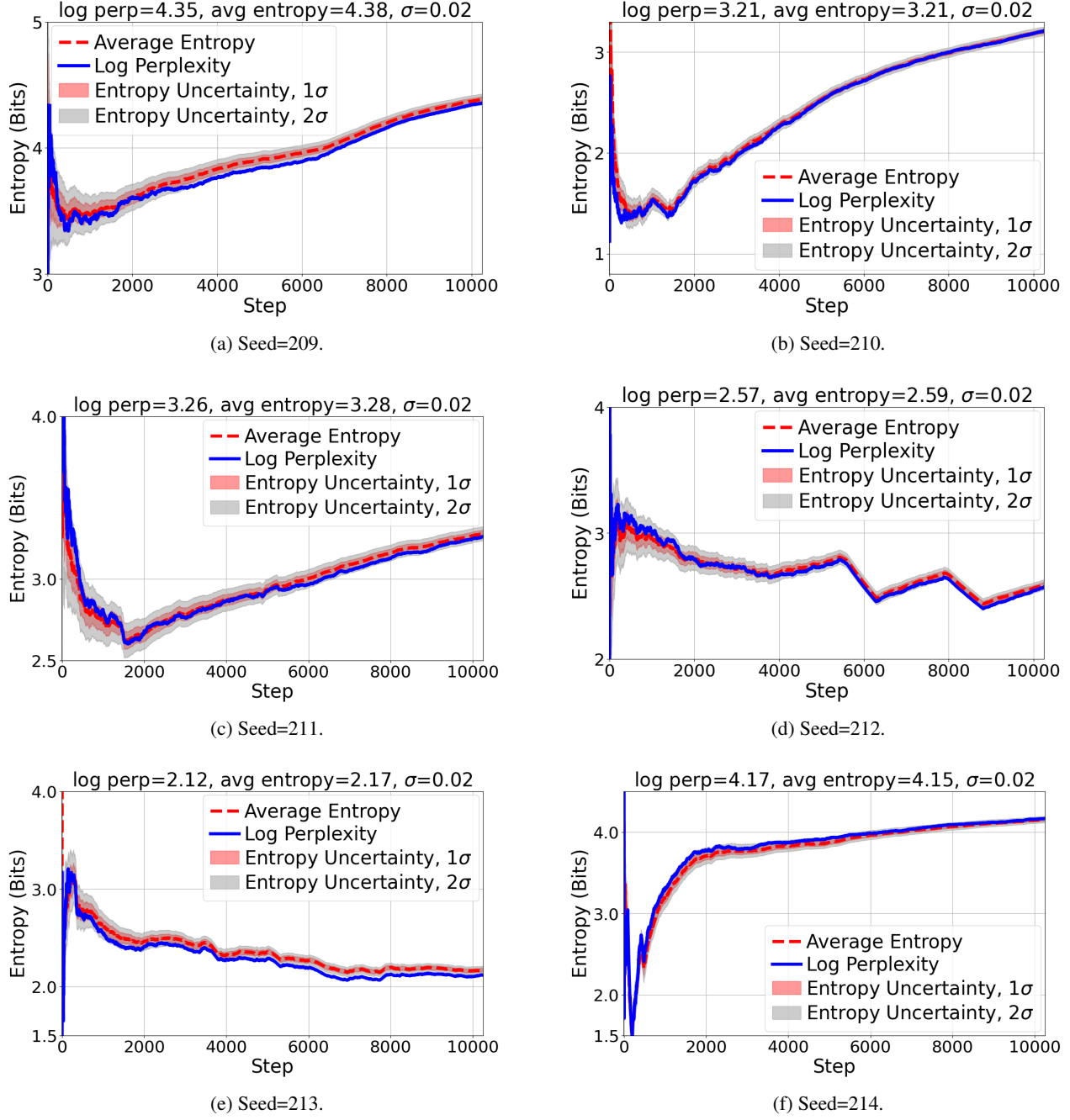


Figure 5. Log-perplexity and empirical entropy for Llama 3.1 of strings generated by the same model with top-100 sampling.