Understanding Transformers via N-gram Statistics

Timothy Nguyen
Google DeepMind
timothycnguyen@google.com

Abstract

Transformer based large-language models (LLMs) display extreme proficiency with language yet a precise understanding of how they work remains elusive. One way of demystifying transformer predictions would be to describe how they depend on their context in terms of simple template functions. This paper takes a first step in this direction by considering families of functions (i.e. rules) formed out of simple N-gram based statistics of the training data. By studying how well these rulesets approximate transformer predictions, we obtain a variety of novel discoveries: a simple method to detect overfitting during training without using a holdout set, a quantitative measure of how transformers progress from learning simple to more complex statistical rules over the course of training, a model-variance criterion governing when transformer predictions tend to be described by N-gram rules, and insights into how well transformers can be approximated by N-gram rulesets in the limit where these rulesets become increasingly complex. In this latter direction, we find that for 78% of LLM next-token distributions on TinyStories, their top-1 predictions agree with those provided by our N-gram rulesets.

1 Introduction

This paper is an attempt to answer the following

Question: *How does a transformer-based large language model (LLM) make use of its context when predicting the next token?*

Our approach proceeds via studying the statistical properties of training data. This is perhaps the most natural place to start even though it is not exhaustive (e.g. it does not include in-context learning (Brown et al., 2020)). The reasons to understand LLM behavior in terms of the statistics of their training data are plenty. First, the functional form of how LLMs use their training data is not well-understood (though there has been progress on understanding memorization (Nasr et al., 2023; Carlini et al., 2023)). Second, the over-reliance of LLMs on training data statistics leads to brittleness (e.g. the "reversal curse" (Berglund et al., 2024)) and the perpetuation of dataset biases (Gallegos et al., 2024). Understanding the nature of this statistical dependence can lead to improved and more informed dataset curation and training methods. Finally, in various scenarios, the performance of LLMs on downstream tasks are found to be correlated with frequency of relevant training data (Razeghi et al., 2022; Elazar et al., 2023; Kandpal et al., 2023; Kang and Choi, 2023). A better understanding of this phenomenon would allow better steering of models towards desired performance levels.

We can think of the complexity of an LLM next token prediction (regarded as a probability distribution over tokens) along two axes: form and selection. Form refers to the functional form of the prediction as a function of the context, e.g. whether the prediction is some explicit function of associated training data statistics (see Figure 1). Selection refers to which functional form, chosen from a set of functional templates, suitably describes the transformer prediction (supposing the choice set is sufficiently rich). As a first nontrivial step, one might hope that an approximate model for an LLM

Preprint. Under review.

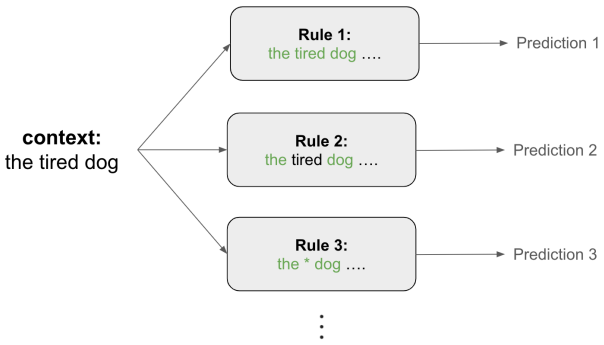

Figure 1: **Illustration of rule approximation.** Given a context, different N-gram based rules formed out of the context will yield different next-token predictive distributions. In the above example, the context consists of three tokens. The first rule uses all three tokens of the context and makes a prediction based on the corresponding 4-gram rule derived from the training data; the second rule uses only the first and last tokens to form a corresponding 3-gram rule (and so the next token "slept" will be assigned less weight than the first rule since the "tired" token is ignored); and the third rule makes a prediction using the N-gram statistics obtained from aggregating over three token contexts from the training data where the second token is arbitrary (i.e. the second token is marginalized). Given a list of such rules, one can ask which rule's predictive distribution best matches that of the transformer.

is that each of its next token predictions can be roughly described by simple statistical rules from the context (simple form) even if the mechanism for its rule selection remains hidden (complex selection)[1]. This paper is an attempt to see how far this perspective can be pushed, and fortuitously we obtain additional insights for understanding LLM behavior along the way. The statistical rules we consider, which are based on N-grams, are defined in Section 4, with Figure 1 showing some examples.

We perform our main investigations on the TinyStories (Eldan and Li, 2023) dataset, with supporting experiments on Wikipedia to confirm our results remain robust at larger scales. The use of TinyStories is for practical reasons: its small size makes training models and aggregating N-gram statistics computationally efficient, yet it is complex enough to capture basic natural language statistics (those occurring in simple children's stories).

Below is a summary of our observations and contributions:

1. (Approximation-Variance Association) We observe an approximation-variance association indicating that next token LLM predictions that have low variance (across different training runs[2]) tend to be well-approximated by N-gram rules. (Section 5)
2. (Curriculum Learning Dynamics) By grouping our N-gram rulesets in terms of complexity (as measured by the amount of context they use), we discover the various ways in which the learning dynamics of LLMs implement a statistical type of curriculum learning, in which easier rules are eventually supplanted by more complex ones. (Section 6.1)
3. (Overfitting Criterion) Based on our analysis of approximating LLM predictions by N-gram rules, we propose a simple and novel procedure for detecting overfitting of LLMs during training. The procedure makes no use of holdout data and it makes quantatively precise the intuition that overfitting corresponds to a model memorizing long context at the expense being able to generalize through making use of subcontext. (Section 6.2)
4. (Approximation Strength) We study how well LLM predictions can be approximated by our N-gram rulesets, noting that significant gains in top1-accuracy occur as we increase ruleset

[1]It is important to emphasize that we seek a *descriptive* approximation of a transformer, rather than an *explanatory* one. A description merely requires that we can provide a post-hoc, per-instance approximation of transformer predictions in terms of an available rule; an explanation means we provide reasons for and thus can predict in advance why and when a particular rule approximates transformer predictions. Hence, we make the distinction between form (description) and selection (explanation).

[2]Different runs have different dataset shuffles.

complexity and diversity, whereby we achieve up to 78% top-1 accuracy on TinyStories (Table 2). We also visually ground these approximations with concrete examples (Figure 5), which may form the basis for dataset attribution methods in future work. (Section 7)

2 Related Work

Rule extraction methods for neural networks have been studied in quite different settings, e.g. (Jacobsson, 2005; Mcmillan et al., 1991). Some recent works have performed N-gram analyses for large-language models in the setting of in-context learning (Akyürek et al., 2024) and associative recall (Arora et al., 2023). The "infini-gram" model (Liu et al., 2024) compares LLM predictions with the single N-gram rule given by retrieving the largest possible matching context from the training data. Our work uses shorter but more sophisticated N-gram rules. In (Voita et al., 2023), an approach to understanding how LLMs process N-grams is carried out at the level of individual neurons. This complements our dataset-based work, which treat models as a black box. In (Edelman et al., 2024), the evolution of the type of N-gram statistics that transformers learn during training is analyzed in the setting of synthetic Markov chain data, in contrast to our natural language setting. Other works studying the learning trajectory of language models include (Chen et al., 2024; Choshen et al., 2022). There is a large literature on building more sophisticated N-gram models, e.g. (Kneser and Ney, 1995; Goodman, 2001). Such models could have been incorporated into our set of rules, but for simplicity we choose not to include them.

3 Experimental Setup

We train standard decoder-only transformer models on the TinyStories (Eldan and Li, 2023) dataset (480M tokens) consisting of children's stories synthetically generated from GPT-4 using templated prompts. The value of this dataset lies in its linguistic simplicity, whereby it is possible to model language well on the dataset using very small models. Unless stated otherwise, our experiments use a 160M parameter model trained for 4 epochs, which achieves a loss of around 1.11 nats on the validation set. We train for 4 epochs since we use learning rate warmup and cosine learning rate decay and we want to ensure all datapoints receive updates with a high learning rate (this way all N-gram statistics have a fair chance of being learned during training). For overfitting experiments in Section 6.2, we train a 1.4B model for 10 epochs. In the Appendix, we include some additional corresponding experiments on Wikipedia (from MassiveText (Rae et al., 2022)) with a single epoch of training in order to validate that our results are of a general nature and extend to more complex datasets. For a fixed dataset, the only source of randomness among different runs are different dataset shuffles. Full experimental details are described in the Appendix.

4 N-Gram Rules

The attention layer within a transformer is in essence a soft context-selection mechanism. The N-gram rules we consider will be loosely modeled on this mechanism. Namely, given a context we will form a derived context in which each token will either be kept, discarded, or marginalized, which is meant to mimic positive attention, no attention, and semantic invariance[3], respectively. More formally, we proceed as follows:

Given a regular expression[4] α, all contexts from the training data can be retrieved which match the regular expression. This allows us to define a corresponding rule that defines for us a distribution over tokens t:

$$R_\alpha(t) = \frac{\#\{\alpha t\}}{\#\{\alpha *\}} \tag{1}$$

[3]For instance, the next token distribution for the context "... the tired dog" may be insensitive to replacing "tired" with "brown" or "furry". Statistics which thus marginalize over all extant substitutions for "tired" yield a crude but generally applicable way of capturing semantic invariance. One can imagine an attention mechanism for which there is a many-to-one mapping of keys to a particular value that might implement semantic invariance.

[4]Our regular expressions operate on tokens not string characters, since our contexts are formed out of sequences of tokens.

where the numerator and denominator are the counts for the N-grams from the training data matching the concatenated regular expressions αt and $\alpha *$, respectively, where $*$ is wildcard (single) character match[5]. (Thus the N-grams in the numerator end with t while those in the denominator can end with any token.) Observe that the next-token predictions of a vanilla N-gram model are obtained by letting $R_\alpha(t)$ vary over all ordinary token sequences α of length $N-1$.

Given σ, a symbol from the the alphabet $\{*, -, +\}$, consider the following operation which maps a token t to a regular expression:

$$S_\sigma(t) = \begin{cases} t & \sigma = + \\ * & \sigma = * \\ \epsilon & \sigma = - \end{cases} \tag{2}$$

where ϵ is the empty regular expression. Given now a sequence $\sigma = \sigma_{-N} \cdots \sigma_{-2}\sigma_{-1}$, define S_σ on a sequence of tokens $C = C_{-N} \cdots C_{-2}C_{-1}$ by tokenwise application of (2) and concatenation[6]:

$$S_\sigma(C) = S_{\sigma_{-N}}(C_{-N}) \cdots S_{\sigma_{-2}}(C_{-2})S_{\sigma_{-1}}(C_{-1}). \tag{3}$$

Thus (3) defines a regular expression which we can think of as fuzzy matching for a subset of a context C (the fuzziness arising from the presence of wildcard matches). For notational convenience, we assume σ is left padded with $-$ symbols, so that we can define $S_\sigma(C)$ for $\mathrm{len}(\sigma) < \mathrm{len}(C)$. Finally, define

$$R_\sigma(t|C) = R_{S_\sigma(C)}(t) \tag{4}$$

for C with $\mathrm{len}(C) \geq \mathrm{len}(\sigma)$. The collection of (4) for various σ defines our N-gram rules under consideration[7]. Each such rule is a function which maps a context C to a next token distribution. We refer to $S_\sigma(C)$ as the rule context for $R_\sigma(t|C)$.

As concrete examples, let $\sigma = + - *+$. If $C = C_{-5}C_{-4}C_{-3}C_{-2}C_{-1}$, then $S_\sigma(C) = C_{-4} * C_{-1}$ and

$$R_{+-*+}(t|C) = \frac{\#\{C_{-4} * C_{-1}t\}}{\#\{C_{-4} * C_{-1}*\}} \tag{5}$$

is a rule which yields a next token distribution based on a particular combination of 4-gram model statistics: it retrieves all three token contexts in the training data whose first token is C_{-4} and last token is C_{-1} and marginalizes over the second token. Likewise, the rules

$$R_{++--}(t|C) = \frac{\#\{C_{-4}C_{-3}t\}}{\#\{C_{-4}C_{-3}*\}} \qquad R_{++**} = \frac{\#\{C_{-4}C_{-3} * *t\}}{\#\{C_{-4}C_{-3} * **\}} \tag{6}$$

are respectively a trigram model with context $C_{-4}C_{-3}$ (all other tokens receiving a $-$ are dropped) and a model which uses four tokens of context but marginalizes over the two most recent ones.

When σ consists of all $+$ symbols, we get vanilla N-gram rules derived from the suffix of C. When σ consists of $\pm$ symbols, we get vanilla N-gram rules derived from subsets of C. Varying the length and the entries of σ yields the following rulesets[8]:

$$\mathcal{R}_M^{\text{suffix}} = \{R_\sigma(t|\cdot) : |\sigma| \leq M, \sigma_i = + \text{ for all i}\} \tag{7}$$

$$\mathcal{R}_M^{\text{subgram}} = \{R_\sigma(t|\cdot) : |\sigma| \leq M, \sigma_i = \pm \text{ for all i}\} \tag{8}$$

$$\mathcal{R}_M^{\text{all}} = \{R_\sigma(t|\cdot) : |\sigma| \leq M\}. \tag{9}$$

The parameter M controls how much of the context is being used for the rules.

[5]We use $*$ (i.e. glob notation) instead of the standard . symbol for readability purposes.

[6]The empty regular expression does nothing under concatenation and does not contribute to the length of the resulting sequence.

[7]For $\sigma = \emptyset$, we define R_σ to be the unigram distribution.

[8]There is some redundancy among the σ's in terms of the rules they determine: for instance, in between any two $+$ consecutive symbols, permuting the order of $-$ and $*$ will determine the same rule. Also in practice, we can assume the first entry of σ is a $+$ since marginalizing the first token is equivalent to reducing the context length. From this, it follows that the number of distinct rules in $\mathcal{R}_M^{all}$ is $2, 5, 13, 34, 89, 233, 378$, for $M = 1, \ldots, 7$, respectively.

optimal rule distance: the minimum (possibly averaged over runs) distance between LLM predictions and rule predictions	$\min_{r\in\mathcal{R}} \text{avg}_i d(p_i(t\|C), p_r(t\|C))$
optimal rule: a rule achieving the optimal distance	$\text{argmin}_{r\in\mathcal{R}} \text{avg}_i d(p_i(t\|C), p_r(t\|C))$
model variance: the average of the pairwise distance between LLM predictive distributions over different runs	$\underset{\substack{i,j \\ \text{distinct runs}}}{\text{avg}} d(p_i(t\|C), p_j(t\|C))$

Table 1: Terminology associated to a context C. Here, $\mathcal{R}$ is some reference ruleset under consideration.

5 Approximating Transformer Predictions with Rules

Let $p(t|C)$ denote the next-token distribution of an LLM conditioned on the context C and for notational similarly, write $p_r(t|C)$ for $r(t|C)$, where r is one of the rules defined in Section 4. We wish to measure how similar these distributions are (higher similarity corresponds to a better rule description). To that end, we use the variational distance to measure the difference of two distributions (we discuss our choice in the Appendix):

$$d(p, q) = \frac{1}{2}\sum_i |p_i - q_i|. \tag{10}$$

Since variational distance may be lacking in concrete interpretability, we will sometimes use top1-accuracy to measure similarity, defined by

$$\text{top1-acc}(p, q) = \frac{|\text{argmax}(p) \cap \text{argmax}(q)|}{|\text{argmax}(p) \cup \text{argmax}(q)|} \tag{11}$$

(in general, the argmax of a probability distribution is a set due to potential ties among maximal probabilities). When the argmaxes in (11) are singletons, top1-accuracy just measures agreement between greedy predictions.

Given a context C, we want to understand how $d(p(t|C), p_r(t|C))$ varies with different rules r and in particular if it can be made small. To that end, we introduce some terminology:

We are interested in determining the optimal rule $p_r(t|C)$ (as defined in Table 1) and if it has small optimal rule distance then we regard the rule as being a good description of the corresponding transformer prediction(s) $p_i(t|C)$.[9] As a first step, note there is a distinguished rule

$$p_{\text{full}}(t|C) = \frac{\#\{Ct\}}{\#\{C*\}} \tag{12}$$

whose rule context is the full unmodified context C.[10] This is because (roughly) the language-modeling objective aims to make $p(t|C)$ similar to $p_{\text{full}}(t|C)$.[11] All other rules in our rulesets are "subleading" in that they drop or marginalize over tokens in the context C. Our goal is to quantify which rules, either (12) or subleading ones, are optimal rules and what their optimal rule distances are.

Our main finding is an *approximation-variance association*: contexts with low model-variance tend to have low optimal rule distance. The surprising aspect of this association is the sufficiency of low model-variance (necessity is a given).[12] We present the case of 7-gram contexts in Figure 2 to corroborate this association, with additional examples relegated to the Appendix. We sample

[9] In practice, we will only have one model available and our optimal rules are computed per-context and per-model. In this section, we have available five models from five runs for use in computing optimal quantities.

[10] That is, $p_{\text{full}}(t|C)$ is the invokation of the rule corresponding to $\sigma = + \cdots + \in \mathcal{R}^{\text{suffix}}_{|C|}$ applied to C.

[11] See Section C for additional discussion.

[12] Predictions which have high variance cannot be well approximated by a single model-independent rule. We use five runs in our analysis here since approximation by a rule that remains fixed across models yields a fortiori approximation by a per-model rule.

around six-thousand 7-grams from the training data, sampling from logarithmically spaced buckets based on counts, and plot various relations between counts, model variances, and rule distances. For simplicitly, we consider the ruleset $\mathcal{R} = \mathcal{R}_7^{\text{suffix}}$ to limit the number of rules under consideration. Our analysis of Figure 2 can be summarized as follows:

Plot (d) summarizes our approximation-variance association. There is a clear positive correlation and reasonable fit between optimal rule distance vs model variance when using the ruleset $\mathcal{R}_7^{\text{suffix}}$. This fit weakens due to many outliers if we only consider the single p_{full} rule (a vanilla 8-gram model) as shown in (b). These outliers correspond to many LLM predictions being poorly approximated by p_{full}. The transition from (b) to (d) is a way of visualizing LLMs performing back-off, whereby LLMs rely on statistics from subsets of the context. We include plots (a) and (c) to highlight how replacing model-variance with the count of a context would lead to a much worse fit. We highlight count of the context C because it is the most obvious and naive measure of how well one should expect $p(t|C)$ to match the rule $p_{\text{full}}(t|C)$. Nevertheless, the weak correlation between count and distance measures makes sense: an LLM can make predictions based on subcontexts of C and those subcontexts can have very different count statistics than those of C.

We believe our approximation-variance association and its corresponding analyses have significance beyond the experiments carried out here since they (i) highlight that naive count-based statistics do not provide the strongest signal in terms of how LLMs leverage dataset statistics (ii) suggest that LLM predictions that have low-variance are likely the ones that are amenable to description (or even explanation) by some underlying dataset statistic (with high-variance predictions being regarded as noise). We leave a more systematic exploration of (ii) to future work.

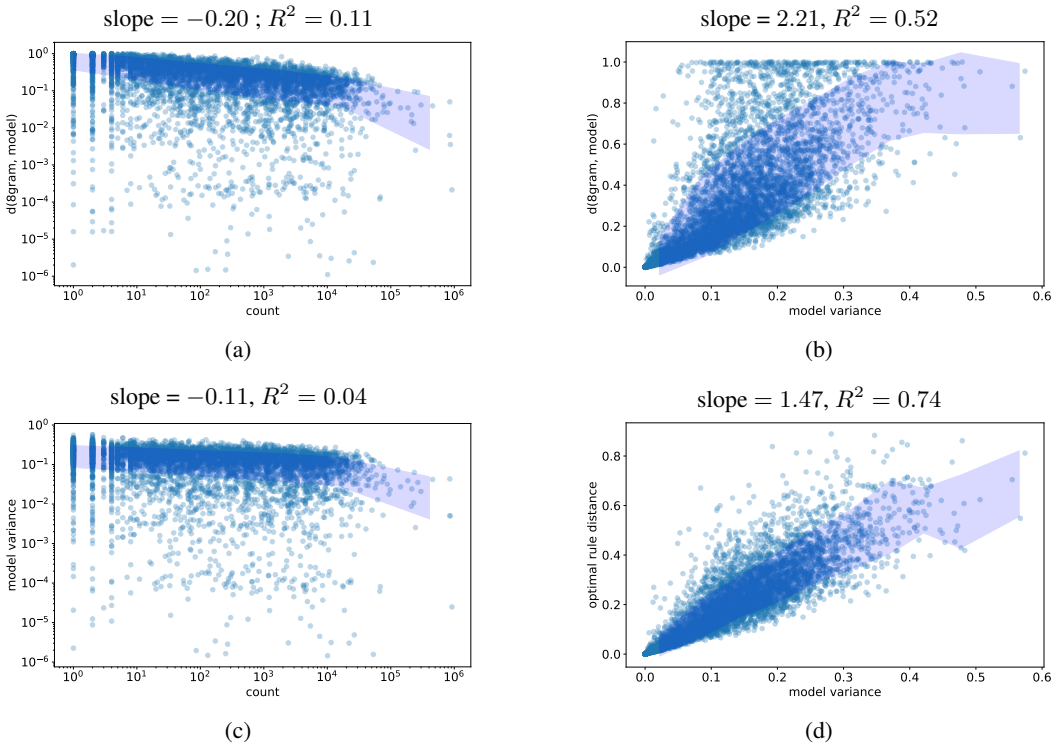

Figure 2: **TinyStories 7-grams**. Every point in the above plots represents a 7-gram context. Shaded regions are plots obtained by bucketing along the x-axis and computing one standard deviation within the mean along the y-axis. Slope and R^2 values of plots are with respect to the linear fit of the data given by their axes. Optimal rule distances and model variances are computed with respect to five model runs. *(a)*: $d(p(t|C), p_{\text{full}}(t|C))$ vs count of C. *(b)*: $d(p(t|C), p_{\text{full}}(t|C))$ vs model variance. *(c)*: model variance vs count of C. *(d)*: similar to (b) but now the y-axis is optimal rule distance of the optimal rule from $\mathcal{R}_7^{\text{suffix}}$.

6 Learning Dynamics

6.1 Curriculum Learning

We can track how well LLM predictions are described by our N-gram rules over the course of training by tracking optimal rule distance as a function of train step. Here optimal rule distance is defined as in Table 1 with $\mathcal{R}$ any of the rulesets (7)-(9), and we will measure how optimal rule distances vary with maximum context length M (the resulting analyses are similar for "all", "subgram", and "suffix" rules so we show our analysis for "all").

Figure 3 summarizes our results. Early in training, LLM predictions acquire structure and thus become approximable by rule predictors. However, with further training, LLM predictions eventually diverge from simpler rules (small context length) while continuing to increase in similarity with more complex rules (larger context length). Moreover, the rightmost plot of Figure 3 shows that top1-acc$(p(t|C), p_r(t|C))$ increases over the course of training for optimal $r \in \mathcal{R}_M^{\text{all}}$ (for $M > 1$) showing that the rule selection improves with time. Altogether, this shows that LLMs undergo a curriculum style learning, in which their predictions gradually move away from simpler rules to more complex and effective rules.

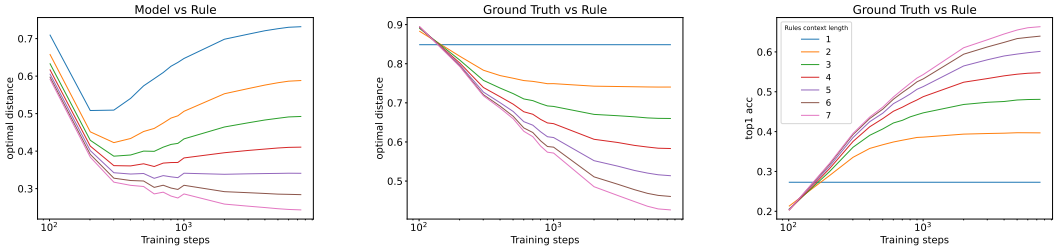

Figure 3: **Training Dynamics.** *Left:* Models reach their lowest distance to more complex rules later in training. For rules with four tokens of context or less, the variational distance eventually starts increasing later in training. For six and seven tokens of context, the variational distance continues to decrease. *Center & Right:* The optimal rule selected always has nonincreasing distance and nondecreasing top1-accuracy relative to the ground truth (interpreted as a one-hot distribution), despite distances eventually increasing or plateauing for rules with less than six tokens of context. This shows that the optimal rule selection is improving with additional training even if the optimal rule distance with respect to model predictions is not improving. (One can imagine the rule predictions as a mesh in probability space, with LLM predictions navigating this space through training. The distance to the mesh may plateau but which rule is closest can continue to change.)

6.2 Early Stopping Criterion

Our investigations of approximating LLMs with rules given by limited contexts naturally lead us to consider LLMs with limited context. The latter have predictive distributions given by

$$p_n(t|C) = p(t|C_{-n} \cdots C_{-1}) \tag{13}$$

where n is the maximum context length. In Figure 4, we plot the loss of an LLM trained to overfit (train loss decreases while validation loss increases) along with its limited context versions for $1 \leq n \leq 7$. For the limited context models with $n > 1$, we see that on *both* the train and validation set, the two respective loss curves track each other closely and both eventually go up. This suggests the following picture: an overfitting LLM is spending capacity to minimize train loss by memorizing the full context at the expense of using capacity to learn statistics of subcontext, i.e. the reduced context in (13). This manifests itself both during training (where subcontext arises from a subset of a larger memorized context) and during validation (where subcontext arises from the partial overlap between novel context and the train set).

Our discovery suggests a simple and computationally inexpensive early stopping criterion: during training, evaluate the transformer on train data consisting of short contexts and when this quantity begins increasing, stop training. Significantly, this method involves no holdout set and is a training dataset intrinsic criterion.

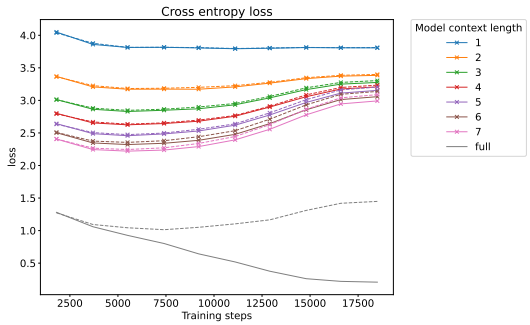

Figure 4: **Overfitting Detection.** We plot both train loss (solid lines) and validation loss (dashed lines) for the full transformer and limited context length transformers (the latter are marked with an "x" for emphasis) on TinyStories. Unlike the full transformer which overfits, those with limited context length have train and validation loss curves closely following each other.

7 Rule Peformance

Finally, addressing our main question from the introduction, we track how well our rulesets describe LLM predictions (in the sense of Section 5) as a whole at inference time. Here, the utility of our N-gram rules defined in Section 4 becomes apparent, since on a holdout set, there will be novel contexts and being able to drop or marginalize context tokens aid in being able to retrieve or aggregate corresponding training dataset statistics. In Table 2, we show the average top1-accuracy between the optimal rule from our various rulesets and LLM predictions on 100 random stories from the validation set. Here, we include as baseline backoff$_M$, the single rule given by the predictive model which performs "stupid backoff" (Brants et al., 2007) using M tokens of context.[13]

We see significant gains in accuracy at large M when adding additional types of rules (for $M = 7$ we gain 6% each time in going from "suffix" to "subgram" to "all"). In the end, we are able to obtain 78% top1-accuracy between the per-prediction optimal rule and the LLM predictions, averaged over all tokens. This is perhaps a remarkably high figure, considering that the top1 accuracy of the model with respect to the ground truth on the validation set is 69.6%. At minimum, we have provided a precise quantification of structure in LLM next-token predictions: they are often matched (as measured by top token prediction) by some simple N-gram rule derived from the training data. See Section D.1 for some supplementary analysis.

To ground our rule optimization procedure, we provide Figure 5 which shows side-by-side how LLM predictions compare with ground truth and optimal rule predictions in an example heldout story. For instance, for the target token "`climb`" in "`... Roxy loved to climb`", both the LLM and optimal rule R_α predict "`play`", where $\alpha =$ "`. * loved to`". For target token "`climb`" in "`... She climbed`", the LLM predicts "`would`" whereas the ground truth and R_α predict "`climb`", where $\alpha =$ "`loved to climb * She`". In general, optimal rules provide the closest statistical match from the training data to the given LLM predictive distributions (from amongst our rulesets), and their top1-predictions can agree or disagree agree (as indicated by target token color). Additional examples, including those from Wikipedia, are shown in Section D. For interpretability purposes, we re-emphasize that our optimal rules currently only provide descriptions, not explanations. We leave the possibility of the latter for future work.

8 Conclusions and Limitations

Our work provides quantitative measures of how well the predictions of transformer-based LLMs are described (i.e. approximated) by simple N-gram rules. Such rules were motivated by the simplest token-level operations applied to the context (keep, ignore, or marginalize). The results we obtained in Section 7 imply that, at least on simple datasets like TinyStories and Wikipedia, LLM predictions contain much quantifiable structure insofar that they often can be described in terms of our simple

[13] That is, $p_{\text{backoff}_M}(t|C) = p_{\text{full}}(t|C_{-k}\cdots C_{-1})$ where $k \leq M$ is the largest value for which $C_{-k}\cdots C_{-1}$ occurs in the training data.

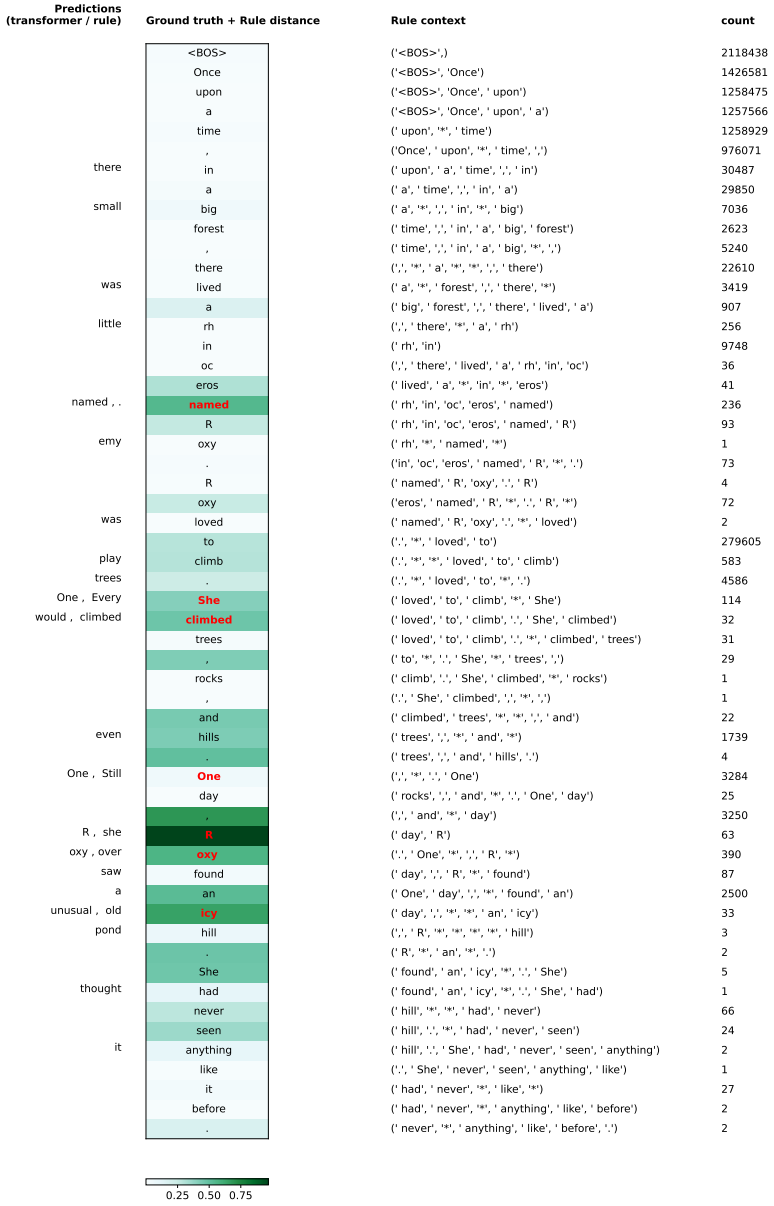

Figure 5: **Rule selection for a TinyStories validation sequence.** The above is a sequence from a heldout story. In the second and third columns are the ground truth, token by token, along with the rule context (as defined in Section 4) associated to the optimal rule from $\mathcal{R}_7^{all}$. The heatmap indicates the variational distance between optimal rule and LLM next token distributions at the given token. The first column shows at most two tokens, which are chosen as follows: If the LLM top-1 prediction disagrees with the ground truth, the LLM prediction is shown. If in addition, the rule selected makes a different top-1 prediction from the transformer, that token is also shown and the corresponding ground truth token is colored red. Thus red tokens are precisely the locations of disagreement between LLM and optimal rule greedy predictions. The last column shows the number of contexts supporting the optimal rule.

statistical rules. Along the way, we also obtained novel discoveries into the statistical nature of overfitting, the occurrence of curriculum learning, and the relation between model-variance and approximability by N-gram rules. Altogether then, our work provides various avenues of progress in understanding how simple dataset statistics are reflected in LLM behavior.

ruleset / M	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$\mathcal{R}_M^{\text{all}}$	30.1	44.9	54.3	62.4	68.8	74.0	77.9
$\mathcal{R}_M^{\text{subgram}}$	30.1	44.6	53.1	60.0	64.8	68.4	71.0
$\mathcal{R}_M^{\text{suffix}}$	30.1	44.4	52.2	57.8	61.5	63.8	65.5
backoff_M	30.1	42.5	48.7	52.6	54.6	55.8	56.6

Table 2: **Approximation Strength**. We look at the average top1-accuracy of the optimal rule versus LLM predictions for rules of varying strength and maximum context length. We compute this average over each token prediction from 100 random validation stories (around 22K tokens total).

On the other hand, it is intuitively clear that current state-of-the-art LLMs go well beyond invoking N-gram rules. A typical request to perform a nontrivial task (e.g. "Write a thirty line poem about mathematics that rhymes") requires a high-level conceptual understanding of language that goes beyond simple literal token-level associations between the context and the training data that we consider here. Nevertheless, one can speculate that an analogue of our work could still apply: in general, an LLM might be performing some high-level rule application, whereby statistics formed out of distributional categories (Pereira et al., 1993) instead of individual tokens are leveraged from the context. Formulating a correct and parsimonious set of rules, if it is at all possible, would be a nontrivial challenge to overcome and one which we leave to future work. Addressing such a challenge and being able to promote the descriptive approximations provided here to explanatory ones would provide a next step towards understanding how LLMs work.

Acknowledgements

The author would like to especially thank Senthooran Rajamanoharan for numerous conversations and an exceptionally discerning eye, which greatly improved the paper from earlier drafts. The author also thanks Jonathan Hale, Marcus Hutter, Matthew McGill, Nick Roy, Avraham Ruderman, and Daniel Tanis for helpful feedback and discussions. Finally, the author thanks Frank Perbet and Daniel Tanis for engineering support.

References

Akyürek, E., Wang, B., Kim, Y., and Andreas, J. (2024). In-context language learning: Architectures and algorithms.

Arora, S., Eyuboglu, S., Timalsina, A., Johnson, I., Poli, M., Zou, J., Rudra, A., and Ré, C. (2023). Zoology: Measuring and improving recall in efficient language models.

Berglund, L., Tong, M., Kaufmann, M., Balesni, M., Stickland, A. C., Korbak, T., and Evans, O. (2024). The reversal curse: Llms trained on "a is b" fail to learn "b is a".

Brants, T., Popat, A. C., Xu, P., Och, F. J., and Dean, J. (2007). Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramèr, F., and Zhang, C. (2023). Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Chen, A., Shwartz-Ziv, R., Cho, K., Leavitt, M. L., and Saphra, N. (2024). Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*.

Choshen, L., Hacohen, G., Weinshall, D., and Abend, O. (2022). The grammar-learning trajectories of neural language models. In Muresan, S., Nakov, P., and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8281–8297, Dublin, Ireland. Association for Computational Linguistics.

Edelman, B. L., Edelman, E., Goel, S., Malach, E., and Tsilivis, N. (2024). The evolution of statistical induction heads: In-context learning markov chains.

Elazar, Y., Kassner, N., Ravfogel, S., Feder, A., Ravichander, A., Mosbach, M., Belinkov, Y., Schütze, H., and Goldberg, Y. (2023). Measuring causal effects of data statistics on language model's 'factual' predictions.

Eldan, R. and Li, Y. (2023). Tinystories: How small can language models be and still speak coherent english?

Gallegos, I. O., Rossi, R. A., Barrow, J., Tanjim, M. M., Kim, S., Dernoncourt, F., Yu, T., Zhang, R., and Ahmed, N. K. (2024). Bias and fairness in large language models: A survey.

Goodman, J. (2001). A bit of progress in language modeling. *CoRR*, cs.CL/0108005.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Vinyals, O., Rae, J. W., and Sifre, L. (2024). Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Jacobsson, H. (2005). Rule extraction from recurrent neural networks: Ataxonomy and review. *Neural Computation*, 17(6):1223–1263.

Kandpal, N., Deng, H., Roberts, A., Wallace, E., and Raffel, C. (2023). Large language models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Kang, C. and Choi, J. (2023). Impact of co-occurrence on factual knowledge of large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *1995 International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 181–184 vol.1.

Liu, J., Min, S., Zettlemoyer, L., Choi, Y., and Hajishirzi, H. (2024). Infini-gram: Scaling unbounded n-gram language models to a trillion tokens.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Mcmillan, C., Mozer, M., and Smolensky, P. (1991). The connectionist scientist game: Rule extraction and refinement in a neural network.

Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., and Lee, K. (2023). Scalable extraction of training data from (production) language models.

Pereira, F., Tishby, N., and Lee, L. (1993). Distributional clustering of English words. In *31st Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Columbus, Ohio, USA. Association for Computational Linguistics.

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., Rutherford, E., Hennigan, T., Menick, J., Cassirer, A., Powell, R., van den Driessche, G., Hendricks, L. A., Rauh, M., Huang, P.-S., Glaese, A., Welbl, J., Dathathri, S., Huang, S., Uesato, J., Mellor, J., Higgins, I., Creswell, A., McAleese, N., Wu, A., Elsen, E., Jayakumar, S., Buchatskaya, E., Budden, D., Sutherland, E., Simonyan, K., Paganini, M., Sifre, L., Martens, L., Li, X. L., Kuncoro, A., Nematzadeh, A., Gribovskaya, E., Donato, D., Lazaridou, A., Mensch, A., Lespiau, J.-B., Tsimpoukelli, M., Grigorev, N., Fritz, D., Sottiaux, T., Pajarskas, M., Pohlen, T., Gong, Z., Toyama, D., de Masson d'Autume, C., Li, Y., Terzi, T., Mikulik, V., Babuschkin, I., Clark, A., de Las Casas, D., Guy, A., Jones, C., Bradbury, J., Johnson, M., Hechtman, B., Weidinger, L., Gabriel, I., Isaac, W., Lockhart, E., Osindero, S., Rimell, L., Dyer, C., Vinyals, O., Ayoub, K., Stanway, J., Bennett, L., Hassabis, D., Kavukcuoglu, K., and Irving, G. (2022). Scaling language models: Methods, analysis & insights from training gopher.

Razeghi, Y., RobertL.Logan, I., Gardner, M., and Singh, S. (2022). Impact of pretraining term frequencies on few-shot numerical reasoning. In *Conference on Empirical Methods in Natural Language Processing*.

Voita, E., Ferrando, J., and Nalmpantis, C. (2023). Neurons in large language models: Dead, n-gram, positional.

A Choice of Distance Measure

We choose variational distance since it is a symmetric and bounded distance function (unlike the KL divergence). Symmetry means we do not have to make a choice between computing the distance between model predictions and rule predictions or vice versa. Boundedness ensures that when we measure average distance across tokens, large outliers do not dominate the average. In fact, for the KL divergence, since $KL(p||q)$ is infinite when $p > 0$ wherever $q = 0$, were we to use KL divergence, we would have to set p equal to rule predictions and q equal to model predictions (since rule predictions are typically sparse). To avoid such constraints and potential pathologies, we choose the variational distance as our metric. We find that other measures like Jensen-Shannon distance give similar results. It is worth noting that while the L^∞-metric often gives similar results, it has a failure mode when comparing two very high entropy distributions. If p and q are two distributions such that p_i and q_i are all small, then their L^∞ distance will be small even though their variational distance can be large.

B Additional Experimental Details

Our transformer architecture and training procedure is based on that of Chinchilla (Hoffmann et al., 2024). The architecture hyperparameters are as follows:

Model	Layers	Number Heads	$\mathbf{d_{key}}/\mathbf{d_{value}}$	$\mathbf{d_{model}}$
160M	12	16	64	896
1.4B	24	16	128	2048

Table 3: Model Specifications

We use a linear learning rate warmup of 1000 steps up to a maximum value of 2×10^{-4} and then use a cosine learning rate decay. We use weighted Adam optimizer (Loshchilov and Hutter, 2017) with weight decay 10^{-4}. Our models are trained using TPU accelerators. The 160M models use 16 TPU accelerators while the 1.4B models use 64 TPU accelerators (to exploit data parallelism) per run. We use a batch size of 128 sequences with each sequence consisting of 2048 tokens.

Our training datasets (TinyStories and MassiveText Wikipedia) are prepared as follows. After tokenizing the individual documents (stories for TinyStories and articles for Wikipedia), we concatenate them all into one long sequence, with each document separated by the $\langle$BOS$\rangle$ token[14]. The full sequence is then subdivided into contiguous sequences of length 2048 (with padding as needed) and then shuffled to form a static dataset of shuffled sequences. We refer to the previous procedure as "chunking". Crucially, observe that chunking results in most sequences not starting with the $\langle$BOS$\rangle$ token (hence a model will be trained to predict the next token conditioned on incomplete contexts, as desired).

For TinyStories experiments, we train 160M models for 4 epochs except for the overfitting experiments where we train 1.4B models for 10 epochs. We use the train and validation splits provided by HuggingFace[15]. For Wikipedia experiments, we train a 1.4B model for a single epoch. We have train and validation splits based on using choosing random sets of disjoint documents. Our Wikipedia train set has 4.4B tokens. In places where we perform several training runs (Section 5), the only source of variance (randomness) among the runs are different dataset shuffles.

Our tokenizer[16] uses byte-pair encoding trained on MassiveText with a vocabulary size of 32,678.

B.1 N-gram statistics

The computation of N-gram statistics of the training data is formed after chunking (as described above), so that they correspond to the N-gram statistics seen by models during training. In particular, tokens which are contiguous in a story but separated by the chunking will not contribute to the

[14]Attention masks are used so that tokens only attend to those from the same document

[15]Available at https://huggingface.co/datasets/roneneldan/TinyStories

[16]Trained using https://github.com/google/sentencepiece

N-gram statistics. We used a distributed map-reduce system to tabulate N-gram counts in the most naive manner. Using sliding windows of size N and aggregating across train documents, we are able to compute N-gram counts for all occurring N-grams and store them in SQL databases. (We ignore those invalid N-grams where $\langle \text{BOS} \rangle$ occurs not as the first token). Note that the number of rows of such N-gram databases is bounded by at most the size of the training corpus times N.

As an aside, we note that for the analysis in Section 6.1, we used our static N-gram rules computed from the entire training data. We do not compute statistics based on the training dataset seen up to the point in training. However, for the purposes of our analysis, this distinction is immaterial (and in practice, the distinction between two sets of statistics will, for the dominant N-grams, be small with sufficiently large batch size).

C Additional Approximation-Variance Association Analysis

We provide additional commentary and experimental settings for our analysis in Section 5.

C.1 Full-context vs Subcontext

As noted in footnote 11, there is usually a mismatch between the contexts that N-gram rules and LLMs receive during training: the latter can receive very long contexts (up to one less than the number of tokens in a document) while the former typically receives very short contexts (in our case, up to 7 tokens). Concretely, while a bigram model is trained on consecutive pairs of tokens (c, t), an LLM is rarely trained so as to optimize $p(t|c)$. Indeed, given a training sequence x, only the target for the first token of x has context consisting of a single token; the other targets will have more tokens of context accordingly. Thus, it is unclear how well LLM predictions $p(t|c)$ should match bigram rule predictions as c varies over the vocabulary set, since LLMs almost always receive c within a much larger context. More generally, it is unclear how well $p(t|C)$ matches $p_{\text{full}}(t|C)$. Nevertheless, because in practice LLMs learn how to use context effectively, LLMs manage to learn $p(t|C)$ despite being optimized for $p(t|\tilde{C})$ with $\tilde{C}$ a context containing C as a suffix.

As a measure of how much training context "dilutes" the LLM ability to learn the bigram distribution, in Figure 6 we plot the distance between LLM predictions and the bigram rule for two LLMs: one trained in the usual fashion with full context and one trained with only one token of context (concretely, a token can only attend to itself in attention layers). In both cases, we have the same

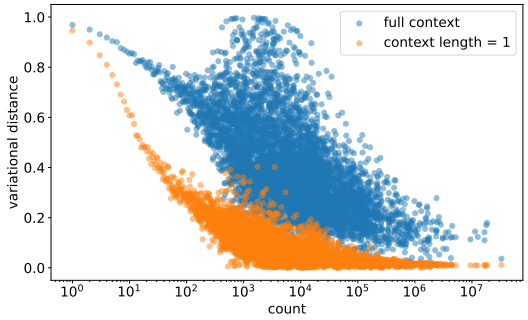

Figure 6: **Comparison with TinyStories bigram model.** We evaluate transformer models (trained with either full context or context length equal to one) on all 22.8K distinct unigrams of TinyStories and record the corresponding variational distance with the bigram rule. Grouping unigrams based on count and averaging the variational distances result in the above scatterplots.

pattern of increased count leads to lower rule distance. However, the context length equal to one transformer has much lower distances since it cannot learn anything else other than the bigram rule. The difference between the variational differences of the two models is thus a measure of the dilution an LLM has in learning a bigram rule owing to receiving surrounding context.

As an aside, we note how for both models, a context with low count has difficulty being learned. In this way, one can regard the inability to learn rules for low count contexts as being due to a failure of optimization, something that could be addressed in the future by improved optimization methods.

C.2 TinyStories Unigram Context

We repeat Figure 2 for the simplest case of unigram context. In this case, there is only one rule (the bigram rule) and so there are only three plots to consider. It turns out also the correlation between optimal distance and count is slightly stronger than with model variance. However, given the unigram context case is extreme (in the sense that there is only a single token of context), we treat this case as an edge case.

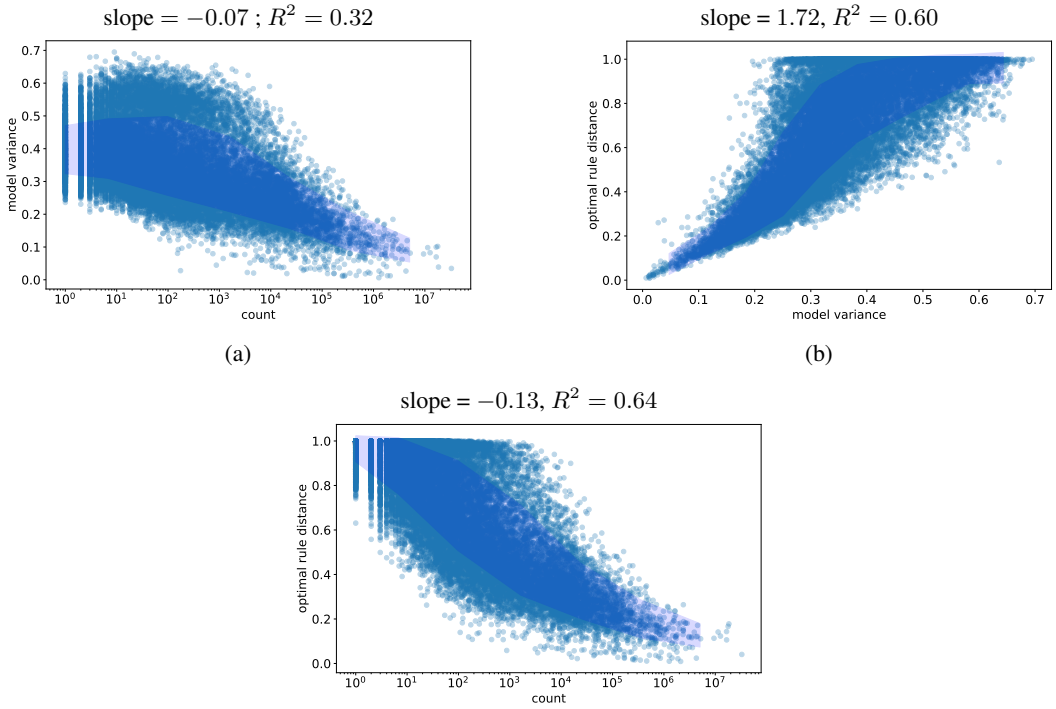

Figure 7: **TinyStories** 1**-grams**. Every point in the above plots represents a 1-gram context (all 22.8K from TinyStories). Shaded regions are plots obtained by bucketing along the x-axis and computing one standard deviation within the mean along the y-axis. Slope and R^2 values of plots are with respect to the linear fit of the data given by their axes. Optimal rule distances and model variances are computed with respect to five model runs. *(a)*: model variance vs count of C *(b)*: $d(p(t|C), p_{\text{full}}(t|C))$ vs model variance *(c)*: $d(p(t|C), p_{\text{full}}(t|C))$ vs count

C.3 Tinystories Bigram Context

Next, consider the case when there are two tokens of context. To get a more fine-grained analysis, we consider the case of full-context bigrams, i.e. those starting with the $\langle$BOS$\rangle$ token. This is because such bigrams do not appear within a larger context and so a transformer's corresponding predictions are more fair to compare with those of N-gram models (both are trained using equal contexts). Conveniently, there are only 691 full-context bigrams in this case and so we do not have to randomly sample a subset.

We will consider the ruleset $\mathcal{R}_2^{\text{all}}$ for which there are three relevant N-gram rules of interest: one which uses the entire bigram of context (a trigram model), one which uses only the last token (a bigram model), and one which uses only the first token (the next token distribution of $\langle$BOS$\rangle$).[17] We will refer to these as the trigram, bigram, and $\langle$BOS$\rangle$ rule respectively.

[17] It turns out that the $\langle$BOS$\rangle*$ rule (given by R_{+*}) in $\mathcal{R}_2^{\text{all}}$ never occurs as an optimal rule for full-context bigrams and so can be ignored in this example.

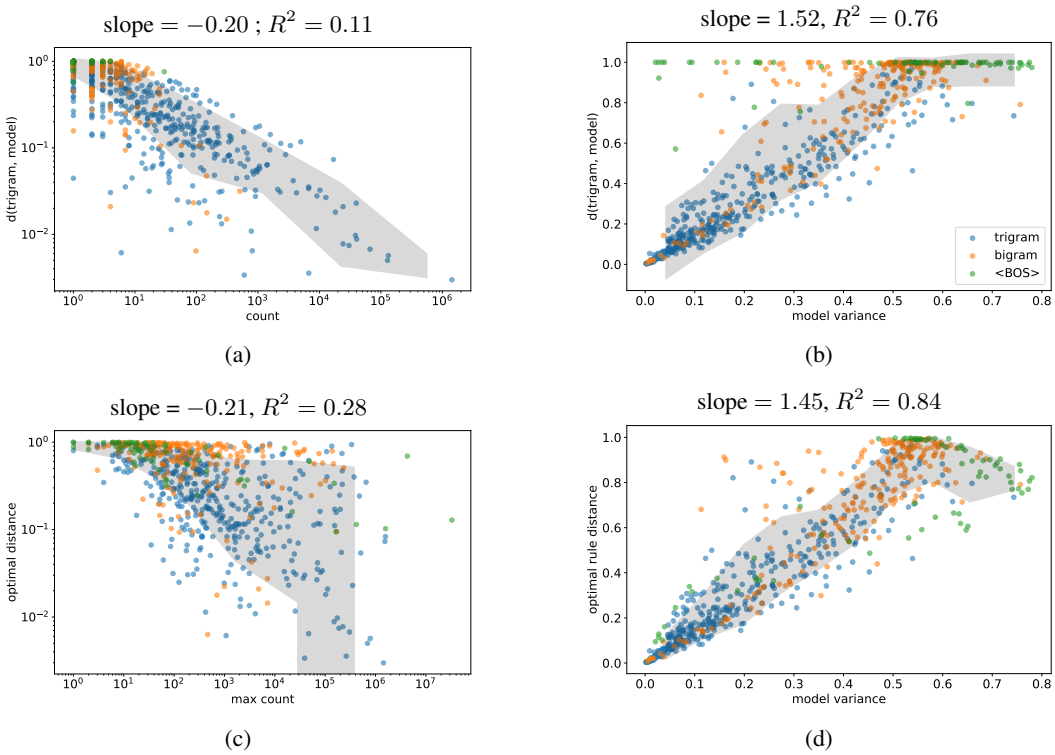

Figure 8: **TinyStories full-context bigrams**. Every point in the above plots represents a full-context bigram C from among the 691 distinct ones in TinyStories. Points are colored by which N-gram rule is the optimal rule, among those in $\mathcal{R}_2^{\text{all}}$, for transformer prediction for the given context. Shaded regions are plots obtained by bucketing along the x-axis and computing one standard deviation within the mean along the y-axis. *(a):* $d(p(\cdot|C), p_{\text{trigram}}(\cdot|C))$ vs count of C. *(b):* $d(p(\cdot|C), p_{\text{trigram}}(\cdot|C))$ vs trigram-model predictions. *(c):* Optimal rule distance vs the greater of the bigram count of C and the unigram count of C_{-1}. *(d):* Similar to upper right but now the y-axis is optimal rule distance. Five model runs are used to compute optimal rule distance and model variance.

We plot an analog of Figure 2 for full-context bigrams in Figure 8. Given the small number of rules, we now color code the optimal rule of each full-context bigram (as indicated by the legend in (b)). In passing from (b) to (d), we see how the outliers in the upper left of (b) move towards the bottom once the large distance from the trigram model is replaced with the optimal rule distance. These are bigrams whose rules are well approximated by bigram or $\langle$BOS$\rangle$ rules and are misspecified when trying to be approximated by the trigram rule. As before, count based correlations in (a) and (c) are weak as indicated by low R^2 values. In (c), we plot a variation in which the x-axis is the maximum of the count of C and the unigram C_{-1}. What the poor fit in (c) indicates is that whether a prediction is well-described by a rule is not a simply determined by whether a subcontext of C occurs often.

C.4 Wikipedia 6-gram contexts

We plot the analog of Figure 2 but for contexts consisting of 6-grams from Wikipedia. We also subsample as before, from logarithmically spaced buckets, for a total of around 6.8K total contexts. We get nearly identical behavior as with TinyStories. Our approximation-variance association is thus not specific to small datasets like TinyStories.

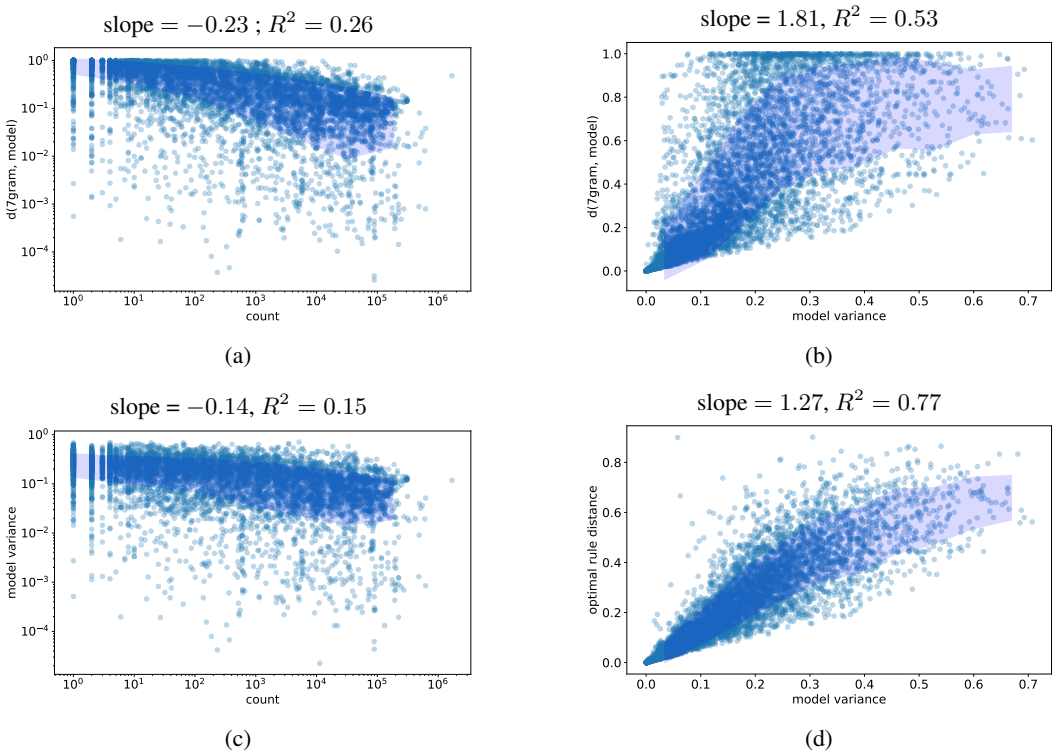

Figure 9: **Wikipedia 6-grams**. Every point in the above plots represents a 6-gram context. Shaded regions are plots obtained by bucketing along the x-axis and computing one standard deviation within the mean along the y-axis. Slope and R^2 values of plots are with respect to the linear fit of the data given by their axes. Optimal rule distances and model variances are computed with respect to five model runs. *(a)*: $d(p(t|C), p_{\text{full}}(t|C))$ vs count of C. *(b)*: $d(p(t|C), p_{\text{full}}(t|C))$ vs model variance. *(c)*: model variance vs count of C. *(d)*: similar to (b) but now the y-axis is optimal rule distance of the optimal rule from $\mathcal{R}_6^{\text{suffix}}$.

D Rule Performance: Additional Analysis and Examples

D.1 Rule Approximation: A Closer Look

We supplement Table 2 with Table 4 to show how optimal rule distances decrease with increasing rule strength. This is to preclude a trivial situation in which by having sufficiently many rules (say a one-hot distribution for every vocabulary token), one can have a ruleset that for any model prediction always returns an optimal rule with 100% top-1 accuracy! Such coarse rules will not in general yield small optimal distances however[18] and our variational distances decreasing in Table 4 shows that our rulesets are truly better approximating the predictions with increasing strength.

ruleset / M	**1**	**2**	**3**	**4**	**5**	**6**	**7**
$\mathcal{R}_M^{\text{all}}$	0.738	0.596	0.507	0.433	0.369	0.315	0.273
$\mathcal{R}_M^{\text{subgram}}$	0.738	0.597	0.512	0.448	0.398	0.361	0.334
$\mathcal{R}_M^{\text{suffix}}$	0.738	0.598	0.519	0.464	0.425	0.399	0.381

Table 4: **Average Optimal Rule distance**. We look at the average optimal rule distance with LLM predictions for rules of varying strength and maximum context length. We compute this average over each token prediction from 100 random TinyStories validation stories (around 22K tokens total).

[18]The variational distance between a one-hot distribution and a distribution which is uniform on n tokens is at least $\frac{n-1}{n}$. Thus, whenever an LLM has at least two roughly valid options, we expect a one-hot distribution to be at least of distance 0.5 from the LLM prediction.

D.2 Rule Approximation: Another Interpretation

Our rule approximation is formulated as a retrodictive procedure in which after an LLM prediction is made, one uses the optimization procedure defined in Section 5 to select an optimal rule for describing the prediction. This retrodictive viewpoint however can be replaced with an alternative interpretation:

Regard the LLM output next-token distribution as a value-vector and each rule prediction as a key-vector in probability space. We use nearest-neighbors with respect to variational distance to select the closest key to the given LLM value-vector. This key then becomes the resulting predicted probability distribution of this joint system of an LLM plus N-gram rules. The joint system is then a *predictive* model that forces LLM predictions through a "bottleneck layer" of a small set of N-gram rules. Our results from Section 7 can be interpreted as saying that such an N-gram bottleneck layer achieves 78% fidelity with respect to the original LLM predictions on TinyStories as measured by top-1 accuracy.

D.3 TinyStories

Here we supplement our example in Figure 5 by showing how the smaller rulesets $\mathcal{R}_7^{\text{subgram}}$ and $\mathcal{R}_7^{\text{suffix}}$ compare in Figures 10 and 11. As expected, the top1 accuracy between transformer predictions and optimal rule predictions decrease with smaller rulesets.

D.4 Wikipedia

We present analogous results in Section 7 for a 1.4B model trained on Wikipedia. In Table 5 we present the analogue of Table 2 (except we go up to maximum context length $M = 6$). To investigate sensitiy to the choice of distance measure on probability distributions, for this section, optimal rules are chosen using the L^∞ metric

$$d_\infty(p, q) = \max_i |p_i - q_i| \tag{14}$$

instead of the variational distance.

The top1-accuracy when using optimal rules from $\mathcal{R}_6^{\text{all}}$ is 64.5%. As with TinyStories, we see significant gains in accuracy when we increase rule strength. Achieving the number 64.5% (versus the corresponding 74.0% number for TinyStories from Table 1), perhaps a surprisingly a high score, is the result of two competing factors: on the one-hand, Wikipedia has a much greater diversity of N-gram statistics (which makes prediction harder), while on the other hand, the training data has more N-grams for use by the rules. Note that our reference LLM (a 1.4B model) achieves 50.1% top-1 accuracy on the 100 validation stories and a train loss of around 2.1 nats at the end of training.

ruleset / M	**1**	**2**	**3**	**4**	**5**	**6**
$\mathcal{R}_M^{\text{all}}$	26.1	41.4	50.5	56.0	60.5	64.5
$\mathcal{R}_M^{\text{subgram}}$	26.1	40.7	48.6	52.5	55.0	57.2
$\mathcal{R}_M^{\text{suffix}}$	26.1	40.5	47.7	50.4	51.3	51.9
backoff$_M$	26.1	38.6	43.2	43.3	42.6	42.5

Table 5: **Approximation Strength for Wikipedia**. We look at the average top1-accuracy between optimal rule and LLM predictions for rules of varying strength and maximum context length. We compute this average over each token prediction from 10 holdout Wikipedia sequences each consisting of 2048 tokens.

We also ground our rule approximation on Wikipedia by providing two concrete examples in Figures 12 and 13.

E Broader Impacts

Large language-models are having significant impacts on society, due to their use as question-answer tools and natural language generators. A better understanding of such language models will only serve to improve their capabilities. Our work here presents steps towards a fundamental understanding

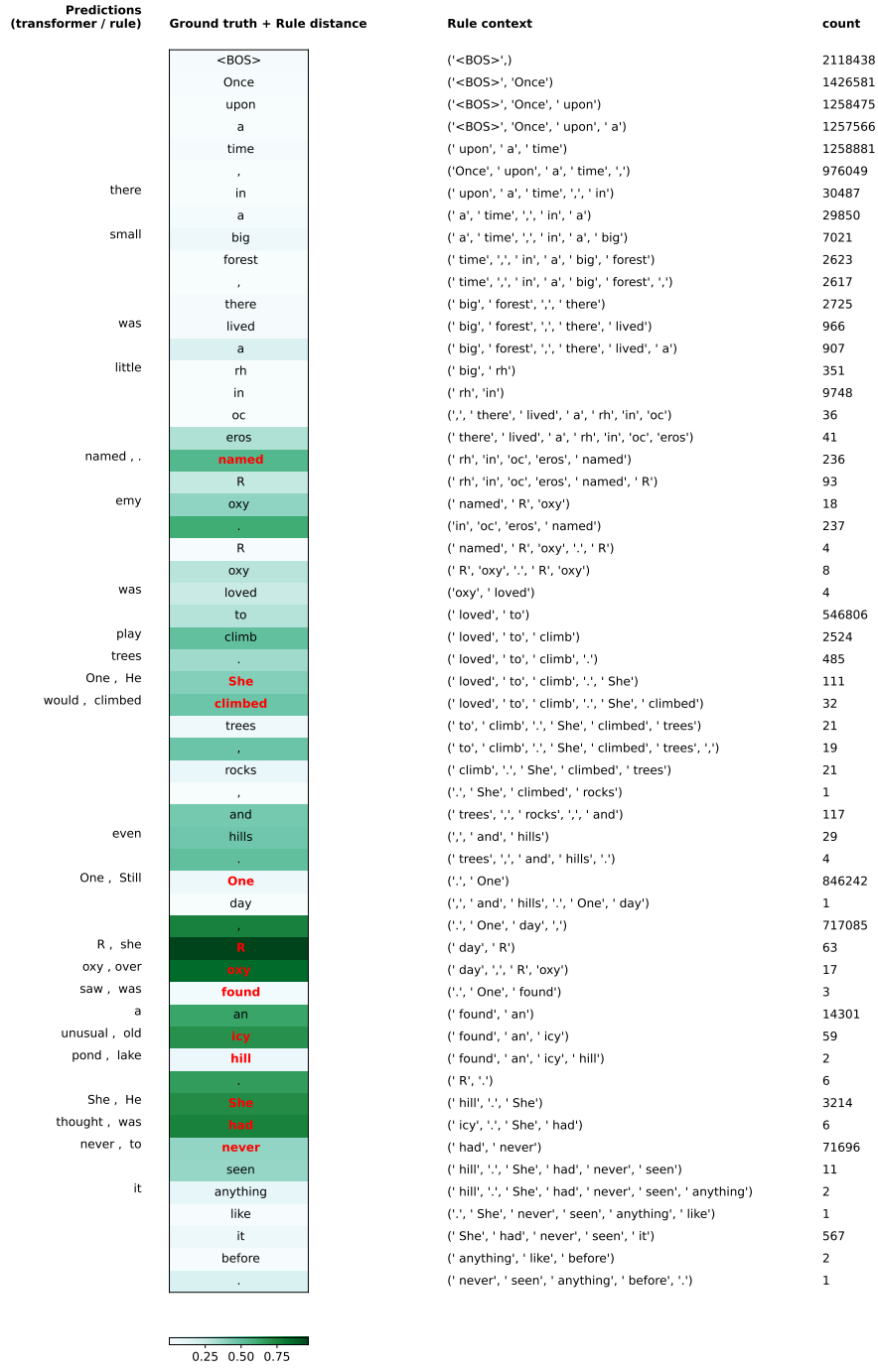

Figure 10: **Rule selection for a TinyStories heldout sequence using $\mathcal{R}_7^{\text{subgram}}$.** Analogous to Figure 5 but with optimal rule chosen from $\mathcal{R}_7^{\text{subgram}}$ instead of $\mathcal{R}_7^{\text{all}}$.

of language models, albeit in a small-scale regime far removed from those relevant for production systems. Given how far removed our work is from realistic datasets and use cases, we do not anticipate any direct negative broader impacts of our work.

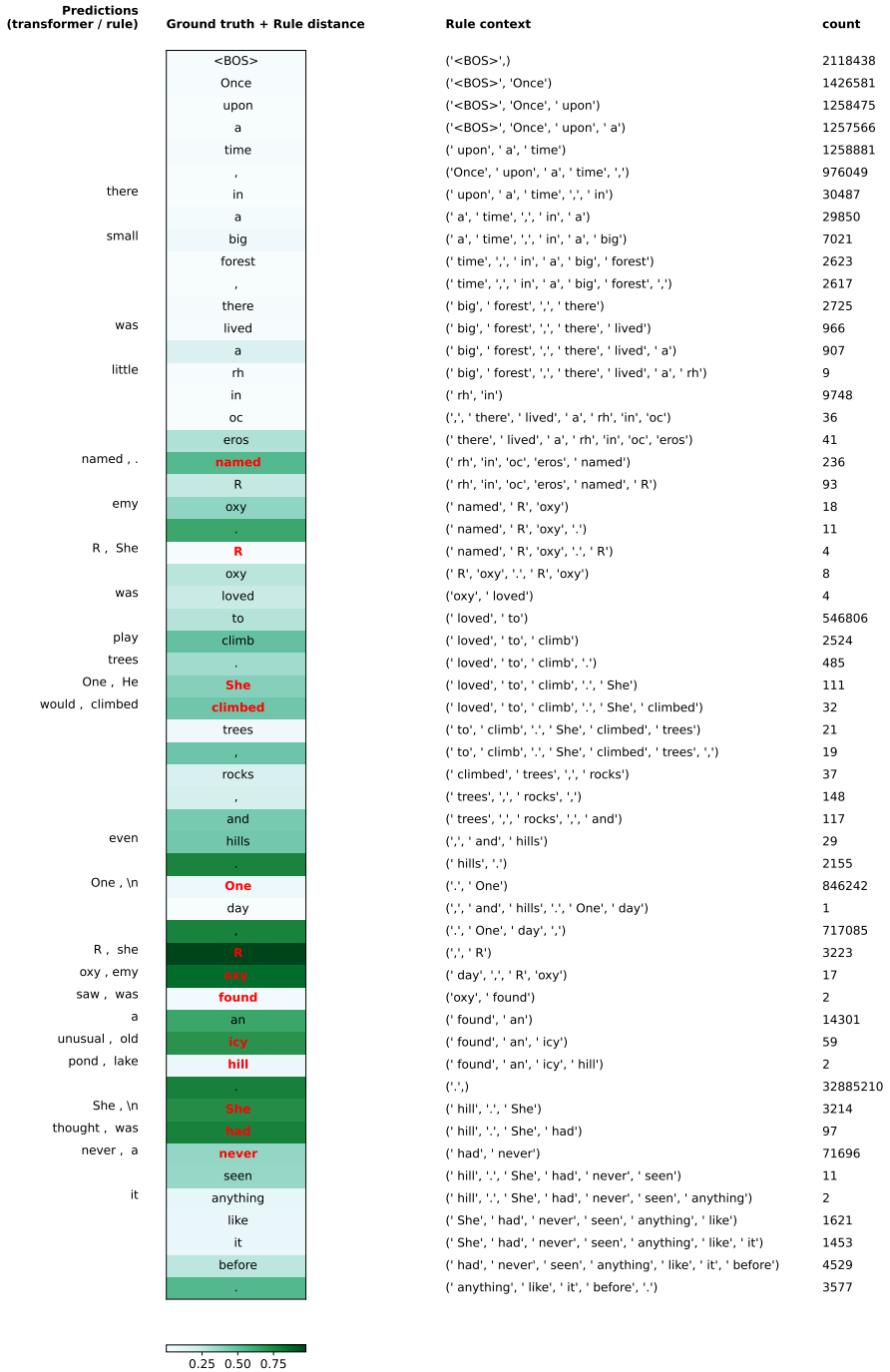

Figure 11: **Rule selection for a TinyStories heldout sequence using $\mathcal{R}_7^{\text{suffix}}$.** Analogous to Figure 5 but with optimal rule chosen from $\mathcal{R}_7^{\text{suffix}}$ instead of $\mathcal{R}_7^{\text{all}}$.

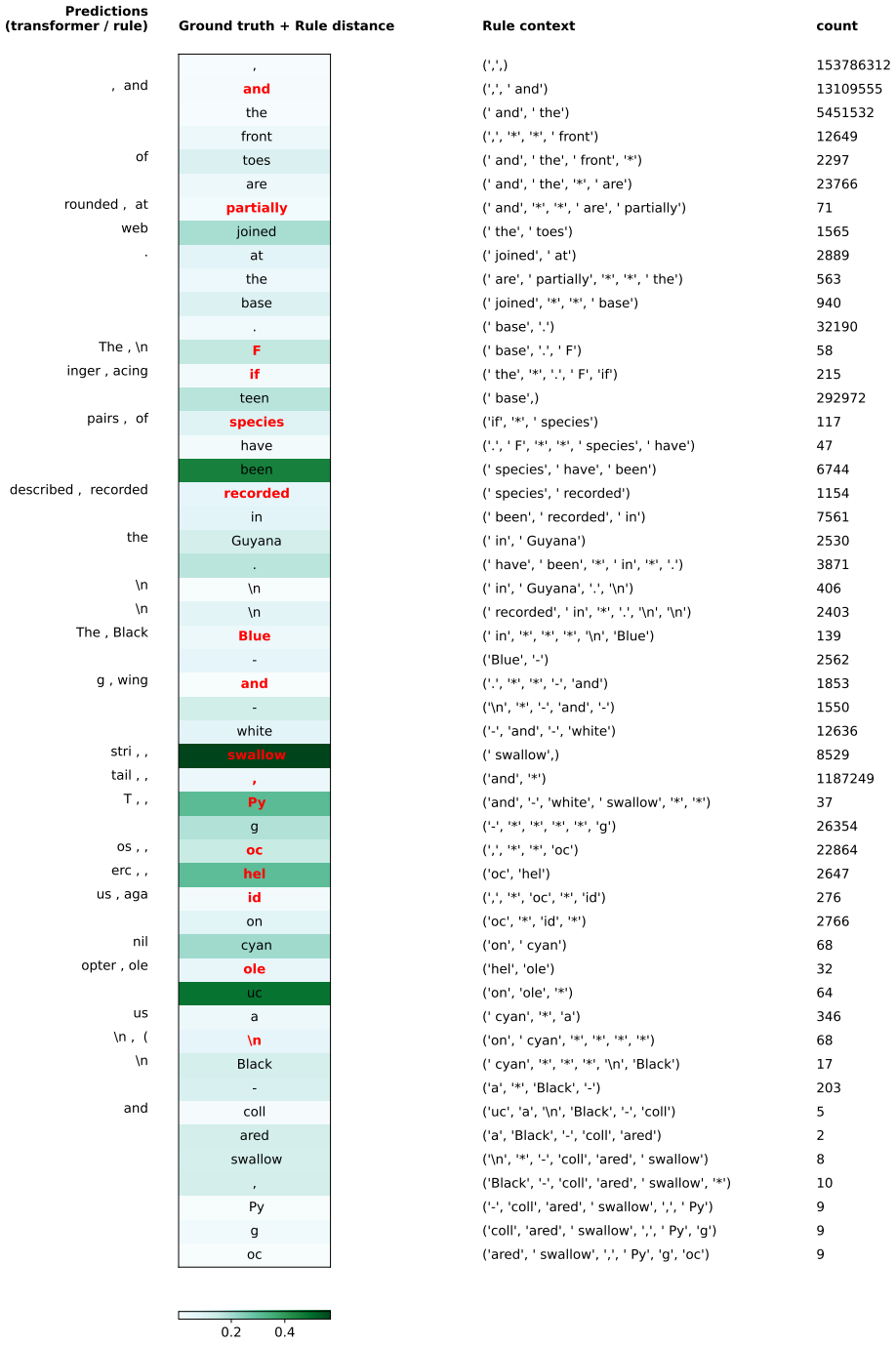

Figure 12: **Rule selection for a Wikipedia heldout sequence.** Analogous to Figure 5 but with optimal rule chosen from $\mathcal{R}_6^{\text{all}}$ and with variational distance replaced with the L^∞ metric for measuring distances between probability distributions.

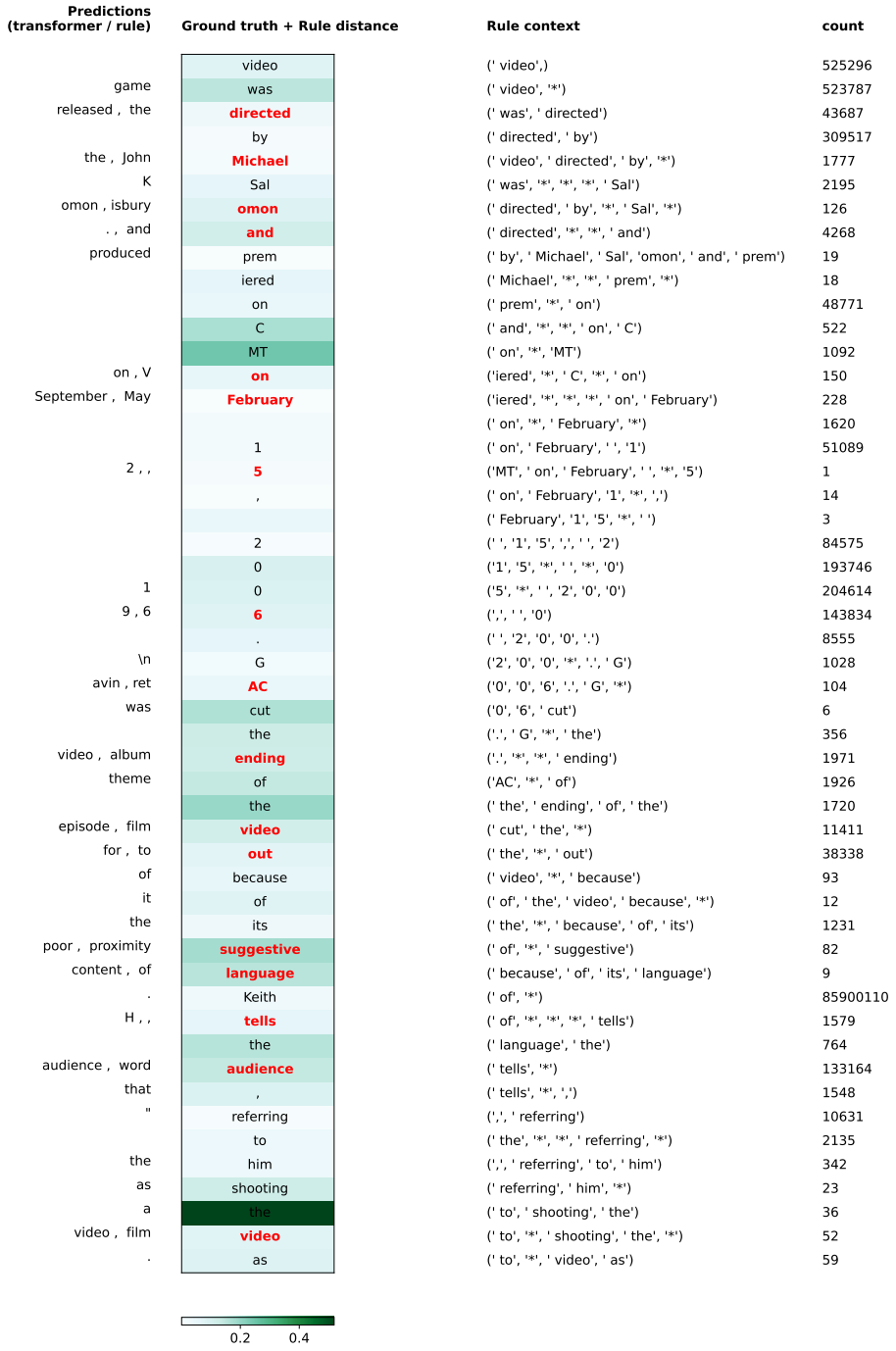

Figure 13: **Rule selection for a Wikipedia heldout sequence.** Analogous to Figure 5 but with optimal rule chosen from $\mathcal{R}_6^{\text{all}}$ and with variational distance replaced with the L^∞ metric for measuring distances between probability distributions.