Language Model Evaluation Beyond Perplexity

Clara Meister[🐨] **Ryan Cotterell**[🐨,🧸]
[🐨]ETH Zürich [🧸]University of Cambridge
`{first.last}@inf.ethz.ch`

Abstract

We propose an alternate approach to quantifying how well language models learn natural language: we ask how well they match the *statistical tendencies* of natural language. To answer this question, we analyze whether text generated from language models exhibits the statistical tendencies present in the human-generated text on which they were trained. We provide a framework—paired with significance tests—for evaluating the fit of language models to these trends. We find that neural language models appear to learn only a subset of the tendencies considered, but align much more closely with empirical trends than proposed theoretical distributions (when present). Further, the fit to different distributions is highly-dependent on both model architecture and generation strategy. As concrete examples, text generated under the nucleus sampling scheme adheres more closely to the type–token relationship of natural language than text produced using standard ancestral sampling; text from LSTMs reflects the natural language distributions over length, stopwords, and symbols surprisingly well.

1 Introduction

Neural language models[1] have become shockingly good at modeling natural language data in recent years (Merity et al., 2017; Conneau and Lample, 2019; Radford et al., 2019). Thus, to test just how well neural language models capture language NLP researchers have started to look beyond standard evaluation metrics such as perplexity, endeavoring to understand which underlying attributes of human language these models are learning. To this end, a nascent literature has emerged that focuses on probing language models (Belinkov and Glass, 2019), i.e., determining whether models encode linguistic phenomena. For the most part, these works have been limited to analyses of sentence-level phenomenon, such as subject–verb agreement (Gulordava et al., 2018) and garden path effects (van Schijndel and Linzen, 2018) among a myriad of other properties (Blevins et al., 2018; Chowdhury and Zamparelli, 2018, *inter alia*).

[1]In this work, we do *not* use the term language model to refer to cloze language models such as BERT (Devlin et al., 2019), which do not give us a distribution over strings.

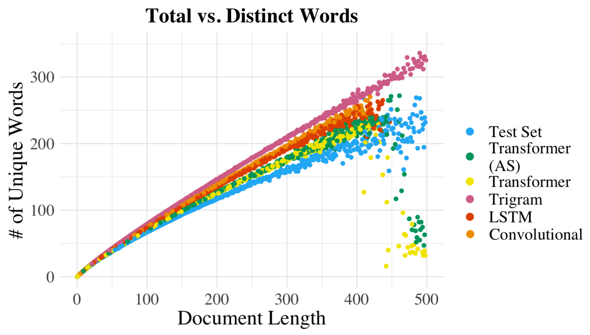

Figure 1: Average number of unique words vs. document length, i.e., type–token, in text sampled from language models. Values from models' test set are plotted for reference.

In this work, we attempt to understand which macro-level phenomena of human language today's language models reflect. That is, we pose the question: *Do neural language models exhibit the statistical tendencies of human language?* Phenomena that can be measured at this level provide an alternate view of a model's comprehension; for example, rather than exploring whether morphological agreement is captured, we look at whether our models learn the trends across a corpus as a whole, e.g., the token rank–frequency (Zipf's) relationship. In comparison to standard probing techniques, this framework does not require we know *a priori* how linguistic phenomena should manifest themselves. That is, when there is no law stating the theoretical tendencies of an attribute of natural language or we have reason to believe our language domain does not follow such a law, we can use the statistical tendencies present in *empirical* data as our baseline. This characteristic both allows us to assess a model's fit to highly corpus-dependent distributions—like the length distribution—and mitigates the biases introduced by our own preconceptions regarding properties of natural language.[2]

More concretely, our paper describes an experimental design and accompanying hypothesis tests to determine precisely whether text generated from language models follows the same empirical trends as human language. Our experiments reveal that adherence to natural language tendencies varies widely with both model architecture and generation strategy, e.g., Fig. 1 shows varying degrees of adherence to the empirical type–token relationship, an artifact that perplexity alone could not reveal. Our findings suggest this framework is a valuable tool for gaining a deeper understanding of where today's language models are succeeding and failing at capturing human language.

2 Language Models

Language models are probability distributions over natural language sentences. We define the support of a language model $p_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$ as

$$\mathcal{Y} := \{\text{BOS} \circ \mathbf{v} \circ \text{EOS} \mid \mathbf{v} \in \mathcal{V}^*\} \tag{1}$$

where $\mathcal{V}$ is the model's vocabulary and tokens EOS and BOS demarcate the beginning and end of a string, respectively, and $\mathcal{V}^*$ is the Kleene closure of $\mathcal{V}$. In this paper, we term vocabularies consisting of words **closed** and those consisting of BPE tokens (Sennrich et al., 2016) **open**.

In the case when $p_{\boldsymbol{\theta}}$ is locally normalized, which is the predominant case for language models, $p_{\boldsymbol{\theta}}$ is defined as the product of probability distributions:

$$p_{\boldsymbol{\theta}}(\mathbf{y}) = \prod_{t=1}^{|\mathbf{y}|} p_{\boldsymbol{\theta}}(y_t \mid \mathbf{y}_{<t}) \tag{2}$$

where each $p_{\boldsymbol{\theta}}(\cdot \mid \mathbf{y}_{<t})$ is a distribution with support over $\bar{\mathcal{V}} := \mathcal{V} \cup \{\text{EOS}\}$ and $\mathbf{y}_{<1} = y_0 := \text{BOS}$. To estimate model parameters $\boldsymbol{\theta}$, one typically optimizes the log-likelihood function over a corpus $\mathcal{C}_{\text{train}}$:

$$\mathcal{L}(\boldsymbol{\theta} \mid \mathcal{C}_{\text{train}}) = \sum_{\mathbf{y} \in \mathcal{C}_{\text{train}}} \log p_{\boldsymbol{\theta}}(\mathbf{y}) \tag{3}$$

where we call each string $\mathbf{y}$ a **document**. To determine the goodness of fit of a model to the empirical distribution (defined by $\mathcal{C}_{\text{train}}$), it is standard practice to measure perplexity on a held-out dataset, which is simply a monotonic function of average (per token) log-likelihood under that model. While low perplexity on an evaluation set undoubtedly reflects some level of fit to natural language, it does not give us a fine-grained view of which linguistic attributes a model has learned.

3 Statistical Tendencies of Language

Human languages are thought to exhibit statistical tendencies, several of which are explicitly quantified by laws (Altmann and Gerlach, 2016). In this section, we review a subset of these distributions–both with and without well-established forms—over which we subsequently perform analyses.

3.1 Classical Laws

Rank–Frequency. Zipf's law (1949), otherwise known as the rank–frequency law, states that the frequency of a word in a corpus decays exponentially in the frequency rank of that word, i.e., the frequency $\omega(\cdot)$ of the k^{th} most frequent word w_k follows the power-law distribution: $\omega(w_k) \propto k^{-s}$. When fit to natural language text, the free parameter s is typically close to 1. Zipf's law also has a probabilistic interpretation: the marginal probability that a random word in our corpus takes on the value of thek^{th} most frequent can be expressed as

$$p_{\text{zipf}}(W = w_k) = \frac{1}{\zeta(s)} k^{-s} \tag{4}$$

where $\zeta(s) = 1/\sum_{k=1}^{\infty} k^{-s}$ is the normalizing constant of our probability mass function (pmf). The adherence of language to Zipf's law has been widely studied and is considered one of the canonical laws of quantitative linguistics (Baroni, 2009; Li et al., 2010; Moreno-Sánchez et al., 2016).

Estimating s from an observed set of rank–frequency pairs can be done using standard estimation techniques. Here we use the maximum-likelihood estimate[3] (MLE), employing numerical optimization to solve for s since the MLE of the discrete power law lacks a closed form solution.

Type–Token. Heaps' law (Herdan, 1960), also known as the type–token relationship, states that

[2]Such biases are naturally introduced by many probing techniques that e.g., draw conclusions from carefully constructed challenge tasks.

[3]Derivation in App. A. We may also estimate s using, e.g., least squares over the original or log–log transform of our distribution. However, it has been empirically observed that least-squares estimates under this paradigm are not reliable (Clauset et al., 2009) and further, directly incorporate assumptions that contradict power law behavior (Schluter, 2020).

the number of additional unique tokens (i.e., number of types) in a document diminishes as its length increases. Formally, we can express the expected number of types $u(\cdot)$ as a function of the length $l(\cdot)$ of the string $\mathbf{y}$ via the relationship $u(\mathbf{y}) \propto l(\mathbf{y})^\beta$ where $\beta < 1$ is a free parameter. Types may be, e.g., unigrams or bigrams.

The above formulation of Heaps' law lacks an obvious probabilistic interpretation. However, if we frame Heaps' law as modeling the *expected value* of the number of types for any given length document, then we can model the relation as a Poisson process, where the marginal distribution over document length follows Heaps' proposed power law. Specifically, we model the number of types for a document of a given length as a non-homogeneous Poisson process (NHPP; Ross, 1996) where our rate parameter $\lambda(l(\mathbf{y}))$ is Heaps' power law relation. The probability that there are k types in a document of length t is then

$$p_{\text{heaps}}(u(\mathbf{y}_{\leq t}) = k) = \frac{\lambda(t)^k}{k!}\exp(-\lambda(t)) \quad (5)$$

for $\lambda(l(\mathbf{y})) = \alpha \cdot l(\mathbf{y})^\beta$. Similarly to Eq. (4), we can fit parameters α, β using MLE (see App. A).

3.2 Other Tendencies

Natural language has other quantifiable distributions, e.g., over document length or unigrams. While there may not exist well-established laws for the behavior of these (often highly corpus-dependent) distributions, we can observe their *empirical* distributions w.r.t. a corpus. We review a few here and leave the exploration of others to future work.

Length. Using notation from earlier, we estimate the pmf of the distribution over the length of documents in a corpus $\mathcal{C}$ as

$$\hat{p}_l(l(\mathbf{y}) = k) \propto \sum_{\mathbf{y}\in\mathcal{C}} \mathbb{1}\{l(\mathbf{y}) = k\} \quad (6)$$

We can additionally compute statistics of this distribution, such as sample mean: $\hat{\mu}_l(\mathcal{C}) = 1/|\mathcal{C}| \sum_{\mathbf{y}\in\mathcal{C}} l(\mathbf{y})$.

Unigram. Notably, the rank–frequency law of §3.1 leaves the categorical distribution over words unspecified, i.e., it defines the frequency for thek^{th} ranked word without specifying the word itself. In order to make explicit comparisons, we define the unigram distribution w.r.t. corpus $\mathcal{C}$ as

$$\hat{p}_{\text{uni}}(w) \propto \sum_{w'\in\mathcal{C}} \mathbb{1}\{w' = w\} \quad (7)$$

Stopwords and Symbols. Certain percentages of words in a string consist of either symbols, i.e., numbers and punctuation, or stopwords, i.e., common words such as "that" or "so" that primarily serve a syntactic function. We can model this percentage as a (continuous) random variable S and estimate its probability density function (pdf) as

$$\hat{p}_{\text{stop}}(s < S \leq s+\delta) \quad (8)$$
$$\propto \sum_{\mathbf{y}\in\mathcal{C}} \mathbb{1}\left\{\frac{\#\text{stop}(\mathbf{y})}{l(\mathbf{y})} \in (s, s+\delta]\right\}$$

The pdf for symbols is defined similarly. As with our length distribution, we can compute the means $\hat{\mu}_{\text{stop}}, \hat{\mu}_{\text{sym}}$ of these distributions.

4 Statistical Distances

In this work, we aim to quantify the degree to which the linguistic distributions of text generated from language models match—or differ from—those of natural language. To this end, we propose the use of several probability metrics (Mostafaei and Kordnourie, 2011; Rachev et al., 2013) as our notion of statistical distance.[4] For each of these metrics, we present *nonparametric* statistical significance tests, i.e., tests that may be used when the underlying distribution of observed data is not known.

4.1 Primary Metrics

Perhaps the simplest method for measuring the distance between two random variables is through differences in expectations, e.g., means or variances. (Semi-)distances of this nature are formally called **primary metrics**. To estimate this distance, we can use observations from random samples $\mathcal{S}_1$ and $\mathcal{S}_2$, e.g., $\mu_1 - \mu_2 \approx \phi(\mathcal{S}_1, \mathcal{S}_2) = \hat{\mu}(\mathcal{S}_1) - \hat{\mu}(\mathcal{S}_2)$.

Observing a value of $\phi(\mathcal{S}_1, \mathcal{S}_2) \neq 0$ on its own is not enough to confirm a difference between μ_1 and μ_2; we need to assess whether the observed distance is *significantly* above or below 0. Formally, our null and alternative hypotheses are:

$$\boxed{\begin{aligned} &\mathrm{H}_0 : \phi(\mathcal{S}_1, \mathcal{S}_2) = 0 \\ &\mathrm{H}_a : \phi(\mathcal{S}_1, \mathcal{S}_2) \neq 0 \end{aligned} \quad (9)}$$

[4]Some of these metrics are formally pseudo-distances, as they are not necessarily symmetric.

In our setting, we typically do not know the theoretical distributions of the random variables generating $\mathcal{S}_1$ and $\mathcal{S}_2$, nor of an arbitrary test statistic ϕ. Consequently, we use resampling techniques to construct the sampling distribution of $\phi(\mathcal{S}_1, \mathcal{S}_2)$.

Permutation Tests. In a nutshell, a permutation test provides a simple method for constructing the sampling distribution of a test statistic ϕ through empirical observations. The method uses the value of ϕ over all possible rearrangements of the observed data points to represent the distribution of the test statistic under the null hypothesis. Using this distribution, we can determine the probability of observing a value of the test statistic (or a more extreme value), which if low, may give us reason to reject a specific null hypothesis. In this work, we only consider statistics $\phi(\cdot, \cdot)$ over two samples. We provide pseudocode for this case in App. B.[5]

4.2 Simple Metrics

Primary metrics provide only a weak measure of the sameness of random variables as they are completely dependent on a single statistic of a distribution. On the other hand, we know a random variable can be completely described by its distribution function. As such, we turn to **simple metrics** of distance between random variables.

Given cumulative density functions (cdfs) P_1 and P_2 over one-dimensional random variables, the Kolmogorov–Smirnov (KS) metric is

$$D(P_1, P_2) = \sup_y |P_1(y) - P_2(y)| \tag{10}$$

where $D \in [0, 1]$ and $D(\cdot, \cdot) = 0$ indicates the distributions are identical. However, not all random variables can be described in terms of a cdf. For categorical distributions where the support of our random variable is not ordinal, the natural counterpart to the KS metric is the Chi-square distance. This metric has a number of drawbacks (discussed in App. C)—primarily that its value can be hard to interpret and so we instead turn to the total variation distance (TVD)—a widely used metric of distance between probability distributions.

Given two pmfs p_1 and p_2, we define TVD as

$$\text{TVD}(p_1, p_2) = \sup_y |p_1(y) - p_2(y)| \tag{11}$$

[5]When the number of possible permutations of the data is computationally prohibitive, we may instead use a MC sampling approach, where we sample from the set of possible permutations (Good, 2000).

where similarly to the KS metric, TVD is bounded above by 1 and a value of 0 indicates identical distributions. In our setting, we consider two use cases for the KS metric and TVD: as distance metrics between an empirical and theoretical distribution (one-sample) and between two empirical distributions (two-sample). The corresponding hypotheses that we can test with these metrics are:

One-Sample Case: (12)

H$_0$: Sample $\mathcal{S}$ is drawn from p

H$_a$: Sample $\mathcal{S}$ is *not* drawn from p

Two-Sample Case: (13)

H$_0$: Samples $\mathcal{S}_1$ and $\mathcal{S}_2$ are drawn from same p

H$_a$: Samples $\mathcal{S}_1$ and $\mathcal{S}_2$ are not drawn from same p

where in the two-sample case, the exact form of p does not need to be known. These hypotheses require the following tests.

The Kolmogorov–Smirov Test. The KS test (Smirnov, 1948) is a nonparametric goodness-of-fit test originally designed to assess the fit of a continuous cdf to empirically-observed data; the two-sample version tests whether two samples come from the same distribution. The method has since been extended to discrete distributions and is regarded as one of the most widely applicable nonparametric goodness-of-fit tests for comparing two distributions (Horn, 1977; Moreno-Sánchez et al., 2016). The test uses the KS metric D as its test statistic; under our null hypothesis, D converges to 0 almost surely in the limit as our number of samples $n \to \infty$ by the Glivenko–Cantelli theorem.[6] We may reject the null hypothesis if our test statistic is greater than the critical value, which is computed based off of our sample size and a desired significance level.[7]

A Test for TVD. Unlike the KS metric, we do not have a (theoretical) limiting distribution for TVD between samples from the same distribution that holds for all density functions (Devroye and Győrfi, 1990). However, we can construct this distribution using resampling techniques. Formally, when $\mathcal{S}_1$ and $\mathcal{S}_2$ are drawn from the same distribution p—where p need not be known—then the test statistic $\text{TVD}(p_{\mathcal{S}_1}, p_{\mathcal{S}_2})$ follows the sampling distribution $\mathcal{Z}_p$, i.e., $\text{TVD}(p_{\mathcal{S}_1}, p_{\mathcal{S}_2}) \sim \mathcal{Z}_p$. The distribution of $\mathcal{Z}_p$ can

[6]Also known as the "fundamental theorem of statistics."

[7]Under the null hypothesis, our text statistic D follows a Kolmogorov distribution. In the two sample case, the critical value is dependent on the size of both samples.

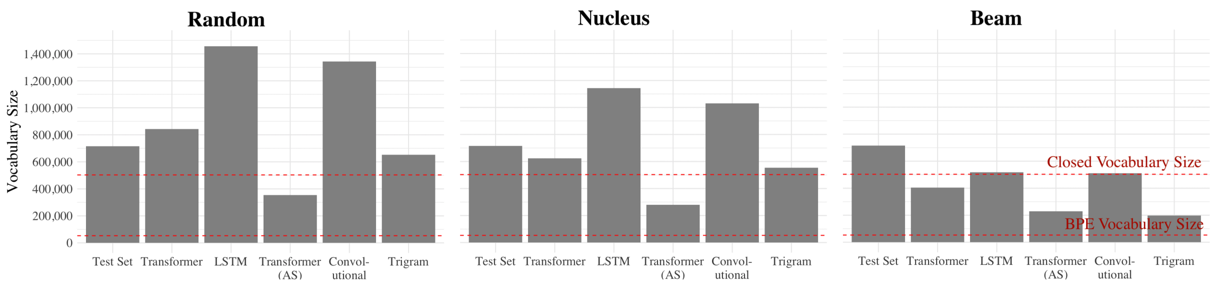

Figure 2: Vocabulary sizes of test set and model-generated samples. Training set (not shown) has vocabulary size of 53.2e5. Only Transformer (AS) and trigram models have a closed vocabulary; the higher red line is the size of the former.

be computed using permutations of our samples, in the same manner as defined in §4.1.

5 Experiments

We use the above framework to assess the degree to which language models learn various distributions of natural language, i.e., we report metrics outlined in §4 measured over the distributions and quantities defined in §3. We compare samples generated from language models to a reserved test set taken from the same corpus as the model's training data. Each set contains 1 million samples.[8] We tokenize all samples using the Moses decoder toolkit (Koehn et al., 2007). All text is lower-cased and only complete unigrams are considered, i.e., when BPE is used, only the detokenized unigram is considered. Length of a string is computed as the number of tokens separated by whitespace. Note that when reporting the KS metric (D), we always report the metric between (a) an empirical cdf computed over the respective model-generated samples and (b) a reference cdf, where D_p indicates direct comparison with empirical cdf of the test set. $D_{p_{\boldsymbol{\theta}}}$ and $D_{\hat{p}}$ indicate comparison with cdfs of a parametric distribution, whose parameters are estimated on the model and test set, respectively.

Natural Language Corpus. We use English Wikipedia Dumps,[9] preprocessing data following the steps used for XLM (Conneau and Lample, 2019) albeit with a 44.7e6 train–1e4 valid–1e6 test split. The test set is used in all statistical tests, however, we estimate standard deviations for statistics in Tab. 4 (in the Appendix) using samples from the training set; see this table for e.g., parameter estimates over test set.

Simulating Corpora from Language Models. Given the distribution $p_{\boldsymbol{\theta}}$, we may exactly compute statistics and distributions for language models over the entire set $\mathcal{Y}$, weighting examples by the probability assigned to each string; however, doing so is infeasible due to the size of the output space and non-Markovian structure of most neural models. Rather, we turn to sampling to create a representative set $\mathcal{S} = \langle \mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(N)} \rangle$ from $p_{\boldsymbol{\theta}}$. We explore three sampling schemes: ancestral random sampling (**Random**), nucleus sampling (**Nucleus**), and beam sampling (**Beam**).[10]

In ancestral random sampling, $\mathbf{y}^{(i)}$ are constructed iteratively according to the distribution

$$y_t^{(i)} \sim p_{\boldsymbol{\theta}}(\cdot \mid \mathbf{y}_{<t}^{(i)}) \tag{14}$$

where $y_0 = \textsc{bos}$. Under the local normalization scheme of Eq. (2), sampling according to Eq. (14) is equivalent to sampling $\mathbf{y}^{(i)}$ directly from $p_{\boldsymbol{\theta}}$. In nucleus sampling, our distribution is truncated to the most probable items covering portion $n \in (0, 1]$ of the probability mass. Formally, we now sample

$$y_t^{(i)} \sim \begin{cases} p_{\boldsymbol{\theta}}(\cdot \mid \mathbf{y}_{<t}^{(i)})/Z & \textbf{if } y_t^{(i)} \in \mathcal{V}_n(p_{\boldsymbol{\theta}}(\cdot \mid \mathbf{y}_{<t}^{(i)})) \\ 0 & \textbf{otherwise} \end{cases} \tag{15}$$

where $\mathcal{V}_n(p) \subseteq \bar{\mathcal{V}}$ is the smallest subset such that $\sum_{y \in \mathcal{V}_n(p)} p(y) \geq n$ and $Z := \sum_{y \in \mathcal{V}_n(p)} p(y)$. Beam sampling uses Eq. (14) as the sampling distribution, but extends a "beam" of k sequences at each sampling iteration. I.e., k extensions are sampled from $p_{\boldsymbol{\theta}}(\cdot \mid \mathbf{y}_{<t}^{(i)})$ and the k most probable of the k^2 sampled items remain on the beam; note that unlike standard beam search, this is a *stochastic* procedure.[11] We use a beam size of 5 in all experiments.

[8]Due to our large sample sizes, we should anticipate that our results will almost always be significant, even when effect sizes are trivially small. As such, we will almost assuredly reject our null hypotheses that model-generated samples come from the same distribution as natural language ones. While in this light, the presentation of hypothesis tests in §4 may seem pointless, we provide them for cases where generating many samples for each model setting is computationally prohibitive.

[9]`dumps.wikimedia.org/`

[10]The latter two sampling designs do not result in samples drawn according to our original $p_{\boldsymbol{\theta}}$. As such, the schemes lead

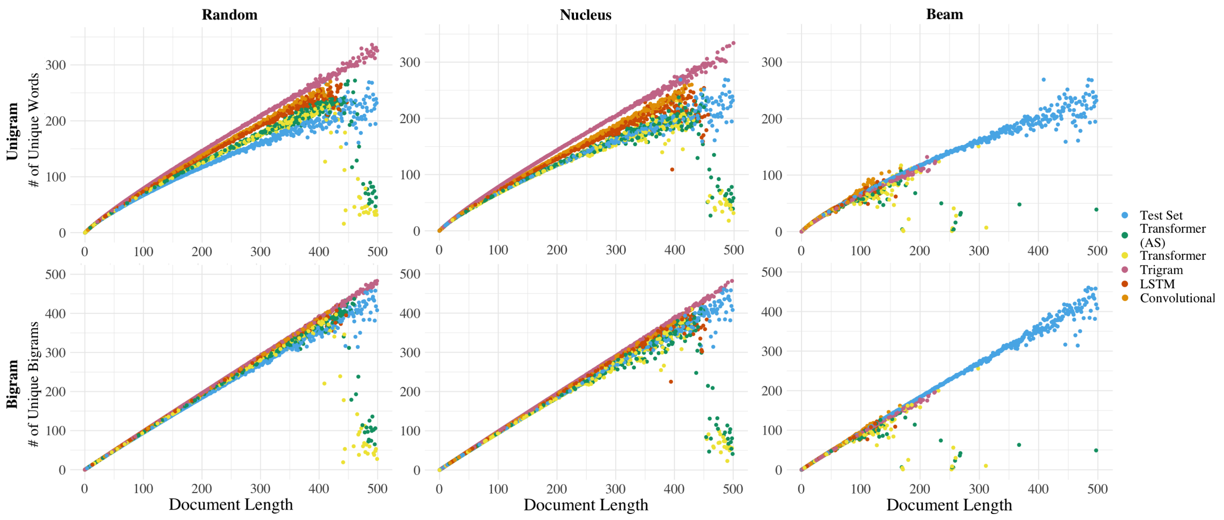

Figure 3: Distinct vs. unique token distributions (unigram and bigram) for test set and text generated from models.

	# params (millions)	Test Set Perplexity
Transformer	205	23.52
Transformer (adaptive softmax)	315	32.66
Gated Convolutional Network	133	48.96
LSTM (3 decoder layers)	59	49.29

Table 1: Neural language model statistics.

Models. We perform our tests on neural models with three different architectures: a transformer (Vaswani et al., 2017; Baevski and Auli, 2019) (only decoder portion), LSTM (Hochreiter and Schmidhuber, 1997), and Convolutional Neural Network (Dauphin et al., 2017). All models are implemented and trained using `fairseq`.[12] We train models on corpora processed both with and without BPE. We include details for each model in Tab. 1. We additionally estimate a trigram model on the training data; formally, we build a model where the probability of observing token $x \in \bar{\mathcal{V}}$ at position i of the text is estimated as

$$p(x \mid x_{i-2}, x_{i-1}) = \frac{c(\langle x_{i-2}, x_{i-1}, x\rangle)}{\sum_{x' \in \bar{\mathcal{V}}} c(\langle x_{i-2}, x_{i-1}, x'\rangle)} \tag{16}$$

where $c(\cdot)$ denotes the function counting occurrences of a sequence in some implicit $\mathcal{C}$. Note that we do not employ smoothing techniques in this model, thus, perplexity over a held-out dataset may diverge and so is not reported in Tab. 1. Vocabulary statistics for each sample are shown in Fig. 2. We provide samples of model-generated text in App. E.

to two "new" distributions, $p_{\boldsymbol{\theta}}^{(n)}$ and $p_{\boldsymbol{\theta}}^{(b)}$, respectively.

[11]Note that this is the default sampling scheme for language generation in the `fairseq` library.

[12]`github.com/pytorch/fairseq/`

5.1 Rank–Frequency

To understand the rank–frequency relationship implicitly learned by language models—and how it relates to the rank–frequency distribution present in natural language—we compute the three KS metrics previously described: $D_{p_{\boldsymbol{\theta}}}$, $D_{\hat{p}}$, and D_p. Specifically, for the first two values, we use the cdf of a Zipfian distribution parameterized by s as our reference—where s is estimated using model generated samples or the test set, respectively.[13] These metrics give us a sense of how well the rank–frequency distribution under our language models match a Zipfian distribution. Since the power-law behavior of the token rank–frequency distribution is known to fall off at higher ranks (Piantadosi, 2014; Moreno-Sánchez et al., 2016), we consider solely the first 10,000 ranks in each sample, including when computing D_p. We report these values in Tab. 2. Values of estimates of s and plots of rank–frequency are shown in App. D.

Our results indicate that our models' empirical rank–frequency distributions do not adhere very closely to a standard Zipfian distribution (as shown by $D_{p_{\boldsymbol{\theta}}}$ and $D_{\hat{p}} \gg 0$), despite appearing to at a superficial level (see App. D). However, the same is true for our test ($D_{\hat{p}} = 0.148$), which suggests that our models fit a Zipfian distribution perhaps *no more poorly* than natural language does. Rather, the model produces qualitatively worst text (see App. E)—a trigram model under the beam sampling generation strategy—follows a power law trend the most closely of any of our samples. On the other hand, the small values of D_p suggest our

[13]s is known to vary with the corpus size $|\mathcal{C}|$ (Powers, 1998), however $|\mathcal{C}|$ is the same for all sets, so this should not affect our analysis.

	Rank–Frequency									Unigram		
	$D_{p_{\boldsymbol{\theta}}}$			$D_{\hat{p}}$			D_p			TVD		
Model	**R**	**N**	**B**	**R**	**N**	**B**	**R**	**N**	**B**	**R**	**N**	**B**
Transformer	0.150	0.145	0.170	0.150	**0.142**	0.170	**3.7e-3**	0.029	0.024	6.9e-3	6.9e-3	6.9e-3
Transformer (AS)	0.145	0.142	0.150	0.143	**0.142**	**0.142**	0.013	0.041	0.046	0.014	0.014	0.038
CNN	0.145	0.142	0.167	0.144	**0.142**	0.167	0.013	0.039	0.022	6.9e-3	6.9e-3	8.6e-3
LSTM	0.147	0.143	**0.175**	0.144	**0.142**	**0.178**	0.016	0.043	0.034	3.4e-3	0.010	9.2e-3
Trigram	0.151	0.148	**0.119**	0.154	0.146	0.152	4.9e-3	0.020	**0.251**	**2.9e-3**	3.0e-3	**0.075**

Table 2: KS metrics (lower implies closer fit) between models' empirical cdf and reference cdfs for the rank–frequency relationship. $D_{p_{\boldsymbol{\theta}}}$ and $D_{\hat{p}}$ indicate statistical distance from a Zipfian distribution, where parameter s is estimated using the model and test sets, respectively. D_p indicates direct comparison with empirical cdf of test set. p-values (estimated using Monte Carlo simulations (Wood and Altavela, 1978)) for all KS metrics are $\ll 0.001$. For the unigram distribution, we report TVD between empirical cdfs of model and test set. All p-values are < 0.001 (see App. D).

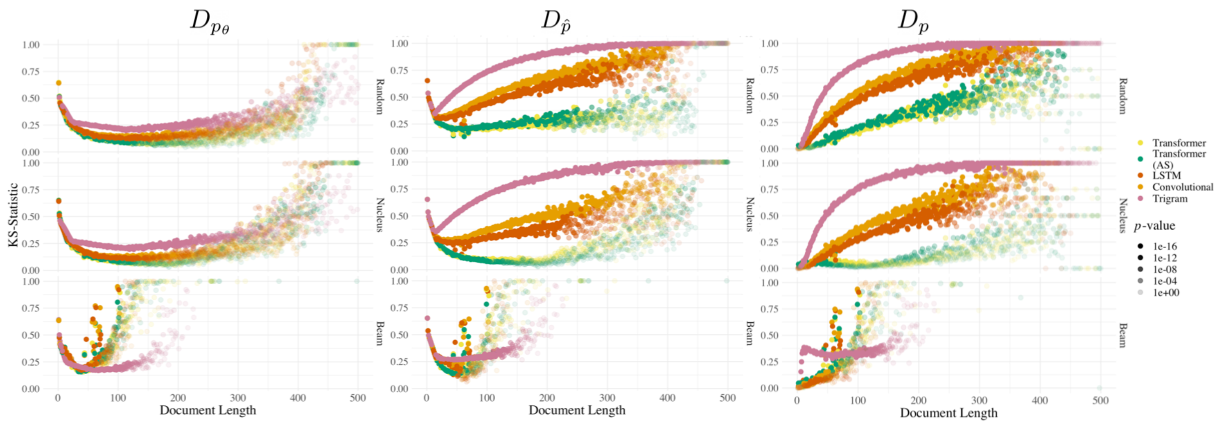

Figure 4: KS metrics (lower implies closer fit) with reference distributions for the type–token relationship as a function of document length. $D_{p_{\boldsymbol{\theta}}}$ and $D_{\hat{p}}$ statistical distance from NHPP distribution for params fit to model text and test sets, respectively; D_p is computed directly against the empirical cdf of test set. Shading indicates significance of the statistic.

models learn the *empirical* rank–frequency trends of human text quite well, something that would not be evident by simply looking at adherence to a Zipfian distribution. The combination of these results suggest the limitation of using adherence to Zipf's law as a gauge for a model's consistency with natural language.

5.2 Type–Token

Fig. 3 shows the type–token trend for all corpora and generation schemes. While most models appear not to follow the same trend as the natural language distribution (as depicted by our test set), we observe that transformers under the nucleus sampling generation scheme match it most closely. Indeed, both models based on the transformer architecture exhibit remarkably similar trends in these experiments, despite having different vocabulary sizes and hyperparameters: both in their generally close fit to the natural language type–token distribution and in their visible fall-off for longer length sequences. The latter observation reveals a deficiency that is seemingly specific to the transformer architecture—one that may be linked to observations in natural language generation tasks. More specifically, we take this as quantitative evidence for recent qualitative observations that when left to generate lots of text, neural language models based on the transformer architecture tend to babble repetitively (Holtzman et al., 2020; Cohen and Beck, 2019; Eikema and Aziz, 2020).

To provide a more mathematically rigorous analysis, we compute KS metrics,[14] again presenting three values: $D_{p_{\boldsymbol{\theta}}}$, $D_{\hat{p}}$, and D_p. In Fig. 4, we can see that model-generated text follows a NHPP parameterized by Heaps' law moderately well ($D_{p_{\boldsymbol{\theta}}}$); there are larger divergences at the tails of document length. However, most do not follow an NHPP with the same parameters as our test set ($D_{\hat{p}}$). Further, in contrast to rank–frequency, the type–token distribution is *more* disparate from the empirical natural language distribution than our parameterized ones, as shown by high values of D_p. While both transformers exhibit the closest fit for all document lengths, which is in-line with our observations in Fig. 3, statistical distance from the natural language distribution for all models and in all settings increases with document length.

[14]§3.1 provides motivation for comparing distributions at individual time steps rather than collectively over time; analyzing Eq. (5) for all document lengths simultaneously would not give us a sense of how the power-law fit changes as a function of document length.

5.3 Unigram Distribution

Because we do not have a well-established law dictating the form of the natural language unigram distribution, we compare only empirical pmfs from model-generated samples and the test set directly. Further, as the distribution over unigrams is categorical, we employ TVD following §4.2. Our results in Tab. 2 indicate that language models generally capture the unigram distribution quite well. The transformer (AS), which has a closed vocabulary, consistently performs poorly in comparison to other models. While we might speculate this outcome is a result of disparate tails between empirical cdfs—i.e., the part of the distribution over infrequent words, which may have been omitted from the closed vocabulary but could still be generated using BPE—the TVD metric in this setting should generally be robust to tail probabilities.[15] This suggests that BPE (or similar) vocabulary schemes may lead to models that can better fit this natural language distribution.

5.4 Length, Stopwords and Symbols

Similarly to the unigram distribution, for length, stopwords and symbols, we compare solely empirical cdfs. We use the set of English stopwords defined by NLTK (Bird et al., 2009). We define the set of symbols as tokens consisting solely of punctuation and numerical values. Our results in Tab. 3 demonstrate that our language models—at least when using random and nucleus sampling—mimic these natural language distributions quite well. Notably, text generated from an LSTM using random sampling follows all three distributions the closest of any model, suggesting LSTMs may have an inductive bias that is helpful for capturing these distributions. On the other hand, using beam sampling leads to strong divergence from natural language distributions across the board. Results for differences in distribution means in the permutation testing framework can be found in App. D.

With respect to the length distribution, these results are perhaps surprising: the local-normalization scheme used by the majority of language generation models (and by those in these experiments) has been claimed to result in models that favor shorter than typical sequences (Sountsov and Sarawagi, 2016; Murray and Chiang, 2018). The results in Tab. 3 and Fig. 5 suggest otherwise.

[15]We observe this empirically; calculating TVD between distributions truncated to the (union of the) first 1000 ranked unigrams lead to almost the exact same result.

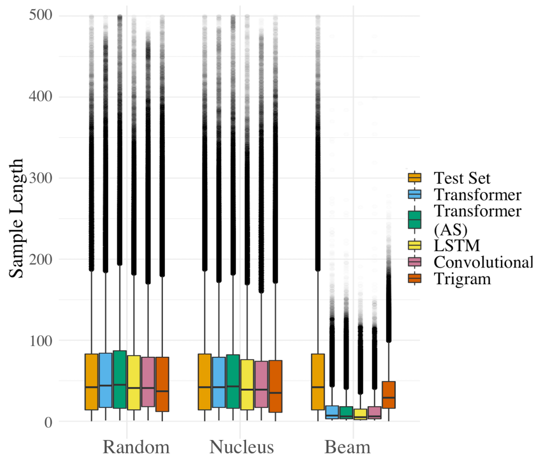

Figure 5: Boxplots showing the distribution of sample length per model and generation scheme. Distribution of test set is repeated in each group for reference.

Specifically, we see that our models fit the natural language length distribution of our corpus quite closely, in terms of both overall distributions and means (see App. D). Rather, it appears that the generation strategy may be the cause of prior observations. This finding raises further questions: since models capture the length distribution well, is a language model more likely to produce degenerate text (e.g., repetitions) than the EOS token if only long documents are used in training? We posit that corpus preprocessing should perhaps be more carefully considered in light of these results.

5.5 Consistent Trends

Across results, we observe that text generated using the nucleus sampling decoding scheme often aligns with natural language more closely than text produced using other generation strategies. This suggests that nucleus sampling performs a helpful alteration to a standard distribution learned via MLE, which may in turn provide motivation for recent efforts to employ truncated or sparse probability distributions directly at training time, e.g., truncated loss (Kang and Hashimoto, 2020) or α-entmax loss (Peters et al., 2019).

We additionally observe large discrepancies in both §5.1 and §5.2 between the results when using empirical natural language cdfs vs. parametric ones. We take this as a warning that assumptions about the forms of linguistic distributions—such as the ones employed by challenge tasks in probing—can have significant effects on results.

6 Related Work

In the last few years, a number of works have extended language model analysis beyond simple

Model	Length			Stopword			Symbol		
	Random	Nucleus	Beam	Random	Nucleus	Beam	Random	Nucleus	Beam
Transformer	0.031	0.034	0.481	0.023	0.062	0.323	0.081	0.065	0.205
Transformer (AS)	0.037	0.041	0.477	0.047	0.015	0.378	0.083	0.072	0.252
CNN	0.034	0.051	0.491	0.036	0.102	0.324	0.069	0.054	0.213
LSTM	**0.014**	0.036	**0.516**	**0.008**	0.069.	0.382	**0.037**	0.048	**0.271**
Trigram	0.093	0.084	0.214	0.126	0.145	**0.490**	0.044	**0.037**	0.061

Table 3: KS metrics (D_p) between empirical length, stopword, and symbol distributions of test set and model generated text. p-values (estimated using Monte Carlo simulations (Wood and Altavela, 1978)) for all KS metrics are $\ll 0.001$.

evaluation metrics—like perplexity—in order to understand what attributes of human language these models are learning. Some use task-based approaches, i.e., they design a set of tasks that require a specific subset of linguistic knowledge then evaluate model performance on these tasks (Linzen et al., 2016; Gulordava et al., 2018; Jiang et al., 2020, *inter alia*). Others use model-based approaches, where a separate model is trained to perform some auxiliary task on representations learned by the model under test (Blevins et al., 2018; Giulianelli et al., 2018; Sorodoc et al., 2020, *inter alia*). We direct readers to Belinkov and Glass (2019) for a full survey of probing methods.

These approaches have drawbacks; for example, introducing a secondary model to determine what the original model has learned presents confounding factors (Hewitt and Liang, 2019). The designing of auxiliary tasks for assessing linguistic knowledge requires large manual effort and lends itself to implicit bias about how linguistic phenomena should manifest. In contrast, our work allows us to take a hands-off approach to analyzing language models. We see the benefit of this in §5, where our results *without* an assumed model of statistical tendencies give us a much different sense of which empirical properties of human-generated text our models have learned.

Our work is closest to that of Takahashi and Tanaka-Ishii (2017, 2019) who use model generated text to visually analyze whether language models reflect well-established statistical tendencies. In contrast, our work provides a quantitative framework, along with appropriate significance tests,[16] for evaluating distribution fits. We additionally assess the fit of language models to our test set directly, rather than solely to established laws. Further, our analysis includes different generation strategies, multiple neural architectures, and a wider variety of empirical language distributions.

[16]In this respect, our work is similar to Dror et al. (2018), whom also present statistical tests for use in NLP.

7 Conclusion and Future Directions

In this work, we present a framework for determining the linguistic properties learned by language models through analysis of statistical trends in generated text. We find that neural language models accurately capture only a subset of natural language distributions and that this subset is highly dependent on both model architecture and generation strategy; no one configuration stands out as capturing all linguistic distributions. Ultimately, we see this analysis framework as a means for a more fine-grained evaluation of language models than perplexity alone can provide. Uncovering which linguistic properties language models have learned—and which they have not—should help us to understand both the inductive biases of various models and via which avenues they can still be improved.

There are a number of important axes of variation that this work does not explore: perhaps most importantly, our results are limited to a single corpora in the English language. A cross-linguistic analysis may reveal whether different model architectures exhibit inductive biases compatible with different languages; observing how these metrics change as a function of corpus size would have implications about the effects of data availability. An exploration of the correlation of these metrics with other quantifications of model performance, such as perplexity or a model's ability to capture sentence level phenomenon, may help us understand how comprehensive other evaluation metrics are. We leave these analyses as future work.

Acknowledgements

We thank Adhi Kuncoro for helpful discussion and feedback in the middle stages of our work and Tiago Pimentel, Jason Wei, and our anonymous reviewers for insightful feedback on the manuscript. We additionally thank B. Bou for his concern.

References

Eduardo G. Altmann and Martin Gerlach. 2016. *Statistical Laws in Linguistics*, pages 7–26. Springer International Publishing.

Alexei Baevski and Michael Auli. 2019. Adaptive input representations for neural language modeling. In *Proceedings of the 7th International Conference on Learning Representations*.

Marco Baroni. 2009. Distributions in text. *Corpus Linguistics: An International Handbook*, 2:803–821.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*, 1st edition. O'Reilly Media, Inc.

Terra Blevins, Omer Levy, and Luke Zettlemoyer. 2018. Deep RNNs encode soft hierarchical syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19. Association for Computational Linguistics.

Shammur Absar Chowdhury and Roberto Zamparelli. 2018. RNN simulations of grammaticality judgments on long-distance dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 133–144. Association for Computational Linguistics.

Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. 2009. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703.

Eldan Cohen and Christopher Beck. 2019. Empirical analysis of beam search performance degradation in neural sequence models. In *Proceedings of the International Conference on Machine Learning*, volume 97.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 933–941.

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

Luc Devroye and László Győrfi. 1990. No empirical probability measure can converge in the total variation sense for all distributions. *The Annals of Statistics*, 18(3):1496–1499.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392. Association for Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? The inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520. International Committee on Computational Linguistics.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248. Association for Computational Linguistics.

Phillip I. Good. 2000. *Permutation Tests : A Practical Guide to Resampling Methods for Testing Hypotheses*, 2nd edition. Springer.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205. Association for Computational Linguistics.

Gustav Herdan. 1960. *Type–token Mathematics: A Textbook of Mathematical Linguistics*. The Hague: Mouton.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 2733–2743. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. *Proceedings of the International Conference on Learning Representations*.

Susan Dadakis Horn. 1977. Goodness-of-fit tests for discrete data: A review and an application to a health impairment scale. *Biometrics*, 33(1):237–247.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Daniel Kang and Tatsunori Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, page 177–180. Association for Computational Linguistics.

Wentian Li, Pedro Miramontes, and Germinal Cocho. 2010. Fitting ranked linguistic data with two-parameter functions. *Entropy*, 12(7):1743–1764.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing LSTM language models. *CoRR*, abs/1708.02182.

Isabel Moreno-Sánchez, Francesc Font-Clos, and Álvaro Corral. 2016. Large-scale analysis of Zipf's law in english texts. *PLOS ONE*, 11(1):1–19.

Hamidreza Mostafaei and Shaghayegh Kordnourie. 2011. Probability metrics and their applications. *Applied Mathematical Sciences*, 5:181–192.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 212–223. Association for Computational Linguistics.

Ben Peters, Vlad Niculae, and André F. T. Martins. 2019. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519. Association for Computational Linguistics.

S. Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin and Review*, 21:1112–1130.

David M. W. Powers. 1998. Applications and explanations of Zipf's law. In *New Methods in Language Processing and Computational Natural Language Learning*.

Svetlozar Rachev, Lev Klebanov, Stoyan Stoyanov, and Frank Fabozzi. 2013. *The Methods of Distances in the Theory of Probability and Statistics*, pages 479–516. Springer.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

S.M. Ross. 1996. *Stochastic processes*. Wiley series in probability and statistics: Probability and statistics. Wiley.

Marten van Schijndel and Tal Linzen. 2018. A neural model of adaptation in reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4704–4710. Association for Computational Linguistics.

Christian Schluter. 2020. On Zipf's law and the bias of Zipf regressions. *Empirical Economics*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

N. Smirnov. 1948. Table for estimating the goodness of fit of empirical distributions. *Annals of Mathematical Statistics*, 19(2):279–281.

Ionut-Teodor Sorodoc, Kristina Gulordava, and Gemma Boleda. 2020. Probing for referential information in language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4177–4189. Association for Computational Linguistics.

Pavel Sountsov and Sunita Sarawagi. 2016. Length bias in encoder decoder models and a case for global conditioning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1516–1525. Association for Computational Linguistics.

Shuntaro Takahashi and Kumiko Tanaka-Ishii. 2017. Do neural nets learn statistical laws behind natural language? *PLOS ONE*, 12(12):1–17.

Shuntaro Takahashi and Kumiko Tanaka-Ishii. 2019. Evaluating computational language models with scaling properties of natural language. *Transactions of the Association for Computational Linguistics*, 45(3):481–513.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Constance L. Wood and Michele M. Altavela. 1978. Large-sample results for kolmogorov-smirnov statistics for discrete distributions. *Biometrika*, 65(1):235–239.

George K. Zipf. 1949. *Human Behavior and the Principle of Least Effort.* Addison-Wesley Press.