An Information Divergence Measure between Neural Text and Human Text

Anonymous Author(s)
Affiliation
Address
email

Abstract

As major progress is made in open-ended text generation, measuring how close machine-generated text is to human language remains a critical open problem. We propose Mauve, a comparison measure for open-ended text generation, which directly compares a generation model's distribution to that of human-written text. Mauve measures the mean area under a divergence curve for the two distributions, exploring the trade-off between two types of errors: those arising from parts of the human distribution that the model distribution approximates well, and those it does not. Mauve extends a family of information divergence metrics, introducing a tractable approximation based on computing the KL divergence in a quantized embedding space. This yields an efficient implementation that scales up to modern text generation models. Through an extensive empirical study on three open-ended generation tasks, we find that Mauve identifies known properties of generated text, scales naturally with model size, and correlates with human judgments, with fewer restrictions than existing distributional evaluation metrics.

1 Introduction

Recent large-scale text generation models show an ability to produce human-like text of remarkable quality and coherence in open-ended generation [31, 46, 6]. In this setting, a text generation model forms a distribution over natural language sequences, induced by an autoregressive neural sequence model (e.g., GPT-3 [6]) paired with a decoding algorithm (e.g., nucleus sampling [18]). Generating text amounts to sampling from this distribution, with the goal of obtaining samples that resemble those from a true distribution of text.

Evaluating the similarity between a generation model's distribution and that of actual text requires considering two types of errors – (I) where the model assigns high probability to sequences which do *not* resemble human-written text, and, (II) where the model distribution does not cover the human distribution, i.e., it fails to yield diverse samples. However, quantifying these aspects with a principled measure that is tractable to compute is challenging, as the distributions encountered in text generation are high-dimensional and discrete, accessed only through samples or expensive model evaluations, and contain pathological features [18, 43, 47].

In this paper, we develop Mauve, a measure of the divergence between two sequence distributions that is efficient, interpretable, and practical for evaluating modern text generation models. Mauve measures the area of a *divergence curve* which captures notions of both types of errors (Figure 1), building on a family of *information-divergence frontier* methods [35, 21, 10] so far underexplored in natural language processing. Our key idea for making Mauve tractable, yet effective, is to reduce its measurement to computing KL-divergence in a quantized, low-dimensional space after embedding samples from each distribution with an external language model. From an end-user's perspective, Mauve has a simple interface: given two sets of text, it provides a scalar measure of their divergence.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

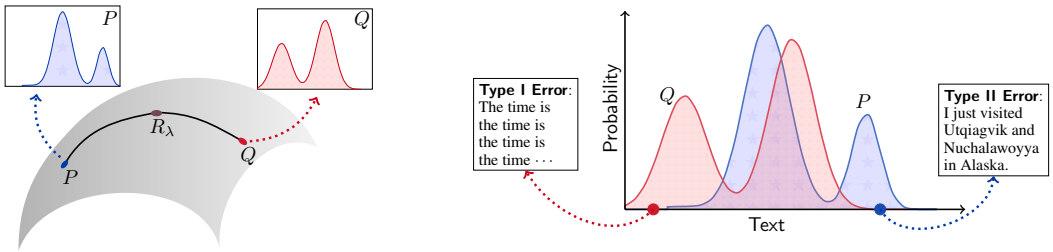

Figure 1: **Left**: Mauve compares the machine text distribution Q to that of human text P by using the family of mixtures $R_\lambda = \lambda P + (1-\lambda)Q$ for $\lambda \in (0,1)$. **Right**: Example Type I error where Q produces degenerate, repetitive text which is unlikely under P and Type II error where Q cannot produce plausible human text due to truncation heuristics [18]. Mauve measures these errors softly, by attributing a λ-fraction of error in the middle region as Type I and the rest as Type II. Varying λ in $(0,1)$ gives a divergence curve and captures a spectrum of Type I and Type II errors. Mauve summarizes the divergence curve in a single scalar as the area under the curve.

We summarize our contributions below. First, we introduce Mauve, an information divergence-based comparison measure between neural text and human text. Second, we empirically show that Mauve is able to quantify known properties of generated text with respect to text length, model size, and decoding more correctly and with fewer restrictions than existing distributional evaluation metrics. Third, we find through a human evaluation that Mauve better correlates with human quality judgements of text. Finally, we find that Mauve can be highly robust to the choice of quantization, embeddings, and scaling. We will open-source a Python package to compute Mauve.[1]

2 A Divergence Measure for Neural Text Generation

In this section we discuss the open-ended text generation setting and introduce Mauve for measuring the divergence between machine generated text and human text.

Open-ended text generation. A language model is an estimate $\hat{P}(\mathbf{x})$ of the probability distribution over sequences of text $\mathbf{x} = (x_1, \dots, x_{|\mathbf{x}|})$, consisting of tokens x_t belonging to some fixed vocabulary (e.g. characters, or words). Prevailing neural autoregressive language models estimate the joint distribution $\hat{P}(\mathbf{x})$ by modeling the conditional distribution $P(x_{t+1}|\mathbf{x}_{1:t})$ over the next token in a sequence. The open-ended text generation task asks us to output text $\hat{\mathbf{x}}_{t+1:|\mathbf{x}|}$ in continuation to a context $\mathbf{x}_{1:t}$. Unlike targeted generation tasks like translation or summarization, there is no "correct" output; the main criteria for open-ended text generation are coherence, creativity, and fluency.

Given a neural autoregressive language model $\hat{P}$, we can generate open-ended text in a serial, left-to-right fashion: sample $\hat{x}_{t+1} \sim \hat{P}(\cdot|\mathbf{x}_{1:t})$, $\hat{x}_{t+2} \sim \hat{P}(\cdot|\mathbf{x}_{1:t}, \hat{x}_{t+1})$, etc. In practice, this simple decoding algorithm is often modified by adjusting the conditional distribution $\hat{P}(\cdot|\mathbf{x}_{1:t})$ to promote more conservative outputs. The decoding algorithm and the language model taken together define a distribution Q over text, which we call the *model distribution*. Common decoding algorithms include temperature rescaling [1] and truncation [11, 18]. Note that truncation methods in particular create sparsity in Q, which leads to degeneracy of some measures including test-set perplexity.

Sources of error in text generation. Our goal in this work is to measure the divergence between the model distribution Q and the target distribution P of human text. As highlighted in Fig. 1, this divergence arises from two sources of error:

(Type I) Q places high mass on text which is unlikely under P,
(Type II) Q cannot generate text which is plausible under P.

The Type I errors are false positives, including the common failure case where a model generates text with semantic repetitions [9, 18, 44] that are highly unlikely to be written by humans.[2] The Type II errors are false negatives, which can occur, for instance, because some pieces of plausible human text cannot be generated by truncation-based decoding algorithms such as nucleus sampling [18].

[1] See Appendix B for an example of the `mauve` library in action.

[2] Let text $\mathbf{x}$ with $P(\mathbf{x}) \gg 0$ be the positive class and $P(\mathbf{x}) \approx 0$ be the negative class. If $Q(\mathbf{x}) \gg 0$ for some negative $\mathbf{x}$, then the model incorrectly considers it a positive, so it is a *false* positive.

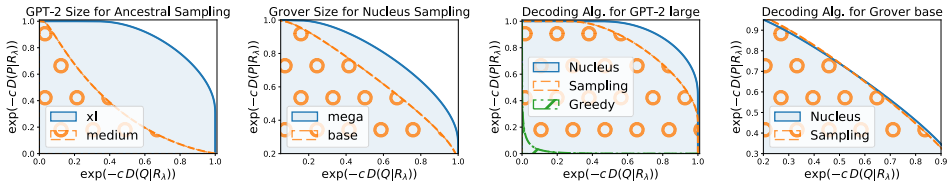

Figure 2: Divergence curves for different models (GPT-2 [31], Grover [46]) and decoding algorithms (ancestral sampling, nucleus sampling and greedy decoding). Mauve is computed as the area of the shaded region, and larger values of Mauve indicate that Q is closer to P. In general, Mauve indicates that generations from larger models and nucleus sampling are closer to human text. **Rightmost**: Nucleus sampling has a slightly higher Type II error than ancestral sampling but a smaller Type I error, indicating that ancestral sampling with Grover base produces more degenerate text while nucleus sampling does not effectively cover the human text distribution.

When P and Q place different non-zero amounts of probability mass on some $\mathbf{x}$, we cannot attribute the discrepancy between $P(\mathbf{x})$ and $Q(\mathbf{x})$ uniquely as a false positive or a false negative. Following recent work on information-divergence frontiers [35, 10], we instead measure these errors *softly* by considering a family of Type I and II error values for different $\lambda \in (0, 1)$, where a λ-fraction of this error is considered as Type I and the remaining as Type II.

Summarizing errors with a divergence curve. To quantify the error at each level $\lambda \in (0, 1)$, we measure how far Q and P are from a mixture distribution $R_\lambda = \lambda P + (1 - \lambda)Q$ in terms of the Kullback-Leibler (KL) divergence, denoted $\mathrm{KL}(\cdot|\cdot)$. In particular, $\mathrm{KL}(Q|R_\lambda)$ measures the Type I error, penalizing Q if there exists text $\mathbf{x}$ such that $Q(\mathbf{x})$ is large but $R_\lambda(\mathbf{x})$ is small. Similarly, $\mathrm{KL}(P|R_\lambda)$ measures the Type II error. Varying $\lambda \in (0, 1)$ yields a *divergence curve*,

$$\mathcal{C}(P, Q) = \Big\{ \big(\exp(-c\,\mathrm{KL}(Q|R_\lambda)), \exp(-c\,\mathrm{KL}(P|R_\lambda)) \big) \;:\; R_\lambda = \lambda P + (1 - \lambda)Q,\ \lambda \in (0, 1) \Big\},$$

where $c > 0$ is a hyperparameter for scaling. The divergence curve formalizes and encodes information about all mixtures of Type I and II errors.[3]

Our proposed divergence measure, **Mauve**(P, Q), is the area under the divergence curve $\mathcal{C}(P, Q)$. Mauve provides a scalar summary of all mixtures of Type I and Type II errors. Mauve(P, Q) lies in $(0, 1]$, with a larger value meaning that Q is closer to P. Further, Mauve$(P, Q) = 1$ if and only if $Q = P$.

Connections to common divergences. The divergence curve encodes more information than the KL divergence $\mathrm{KL}(P|Q)$, which can be obtained from the second coordinate of the curve $\mathcal{C}(P, Q)$ as $\lambda \to 0$, and the reverse KL divergence $\mathrm{KL}(Q|P)$ which can be obtained from the first coordinate of the curve $\mathcal{C}(P, Q)$ as $\lambda \to 1$. Note that one or both of $\mathrm{KL}(Q|P)$ and $\mathrm{KL}(P|Q)$ could be infinite. The midpoint of the divergence curve ($\lambda = 1/2$) corresponds to the Jensen-Shannon (JS) divergence. Further, the Jensen-Shannon (JS) divergence $\mathrm{JS}(P, Q) = \big(\mathrm{KL}(P|R_{1/2}) + \mathrm{KL}(Q|R_{1/2})\big)/2$, can be obtained from the two coordinates of $\mathcal{C}(P, Q)$ at $\lambda = 1/2$. Mauve summarizes *all* of the divergence curve $\mathcal{C}(P, Q)$.

Computing Mauve for open-ended text generation. Each point on the divergence curve $\mathcal{C}(P, Q)$ consists of a coordinate

$$\mathrm{KL}(P, R_\lambda) = \sum_{\mathbf{x} \in \mathcal{X}} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{R_\lambda(\mathbf{x})}, \tag{1}$$

and a similarly defined coordinate $\mathrm{KL}(Q, R_\lambda)$. We cannot compute the summation as written in Equation 1: we do not know the ground-truth probabilities $P(\mathbf{x})$, and the size of the sequence space $\mathcal{X}$ – and hence the support of a typical model distribution – is prohibitively large. As a result, Mauve cannot be tractably computed in closed form.

We overcome these difficulties using a Monte Carlo estimator, computed in a quantized space that is sensitive to important features of text. First, we obtain embedded samples $H_P =$

[3] More generally, the divergence curve $\mathcal{C}(P, Q)$ encodes the **Pareto frontier** of $\big(\mathrm{KL}(P|R), \mathrm{KL}(Q|R)\big)$ for all distributions R, not just mixtures of the form R_λ. We prove this in Appendix A.

Type	Metric	Measures	Approximates
Statistics	Zipf Coefficient [18]	Unigram rank-frequency statistics	–
	Self-BLEU [50]	N-gram diversity	–
	Generation Perplexity [11]	Generation quality via external model R	$\lvert\mathbb{E}_Q[\log R(\mathbf{x})]-\mathbb{E}_P[\log R(\mathbf{x})]\rvert$ (a single point inside $\mathcal{C}(P,Q)$)
Language Modeling	Perplexity	Test-set perplexity	$\mathbb{E}_P[\log Q(\mathbf{x})]$
	ε-perplexity [27]	Perplexity w/ Laplace smoothing	$\mathbb{E}_P[\tilde{Q}(\mathbf{x})]$
	Sparsemax Score [27]	LM quality (sparsemax loss [26])	$\mathbb{E}_P[\tilde{Q}(\mathbf{x})]$
	Token JS-Div. [27]	LM quality (JS divergence)	$\mathbb{E}_P[\tilde{Q}(\mathbf{x})]$
Divergence Curve	Mauve (this work)	Quality & diversity via the divergence curve	$\mathcal{C}(P,Q)$ at all λ

Table 1: Summary of automatic distributional metrics for evaluating open-ended text generation. Mauve provides a summary of all points along the divergence curve, rather than a single point. The summary is based on comparisons in a joint embedding space, rather than a statistic computed independently on each distribution. $\tilde{Q}$ informally refers to a quantity related to Q.

$\{M(\mathbf{x}_1),\dots,M(\mathbf{x}_N)\}$ using samples $\mathbf{x}_i \sim P$, and $H_Q = \{M(\mathbf{x}_1),\dots,M(\mathbf{x}_M)\}$ with $\mathbf{x}_i \sim Q$. Each embedding $M(\mathbf{x}) \in \mathbb{R}^d$ is computed with an external neural language model, e.g. GPT-2 [31].

Next, we jointly quantize the embedded samples H_P and H_Q into k elements using a quantization algorithm $\mathcal{A}(H_P, H_Q) \to (C_P, C_Q)$, e.g. k-means [24]. Here $C_P = \{c_1,\dots,c_N\}$ gives an identifier in $\{1,2,\dots,k\}$, e.g. k-means cluster id, of each embedded sample from P (and similarly for C_Q).

This yields an approximation of each distribution as a discrete, multinomial distribution on k elements,

$$P(\mathbf{x}) \approx \tilde{P}(c(\mathbf{x})) = \frac{1}{|C_P|}|\{c_i \in C_P|\ c_i = c(\mathbf{x})\}|, \tag{2}$$

where $c(\mathbf{x}) \in \{1,\dots,k\}$ returns the identifier of $\mathbf{x}$ (e.g. nearest cluster), and similarly for $Q \approx \tilde{Q}$.

Computing the divergence curve is now tractable, as each coordinate is reduced to a KL divergence involving the k-element multinomial distributions $\tilde{P}$ and $\tilde{Q}$. To recap, our proposed measure **Mauve**(P,Q) is the area under this divergence curve, providing a summary of all Type I and Type II errors through an efficient approximation designed for text generation. Next, we discuss how Mauve compares to prior comparison measures for text (§3), then empirically study Mauve (§4).

3 Related Work

Divergence Measures for Text. Prior measures of similarity/divergence between machine text and human text come in three broad categories: (a) reference-based, (b) statistics-based, and (c) language modeling. Table 1 summarizes the latter two categories, and contrasts them with Mauve.

Reference-based measures evaluate generated text with respect to a (small set of) reference text sample(s), rather than comparing full sequence distributions. These include classical metrics [30, 22, 2] and recent model-based ones [39, 48, 49, 37]. While this paradigm is useful for conditional generation tasks such as translation and summarization where correctness is paramount, it is unsuitable for open-ended generation as there typically are several plausible continuations for each context.

Statistics-based measures compare the model distribution Q with respect to the human distribution P on the basis of some statistic $T(P)$ and $T(Q)$. Property-specific statistics such as the amount of repetition [18, 44], verifiability [28], or termination [43] are orthogonal to Mauve, which provides a summary metric rather than focusing on an individual property. Another statistic is the generation perplexity [11, 18], which compares the perplexity of the model text $\mathbf{x} \sim Q$ with that of human text $\mathbf{x}' \sim P$ under an external model R. By virtue of $T(\cdot)$ being a scalar, generation perplexity cannot trade-off the Type I and Type II errors like Mauve. In fact, we show in Appendix A that the generation perplexity can be derived from *a single point* enclosed between the divergence curve and the axes.

Language modeling metrics calculate how (un)likely human text $\mathbf{x} \sim P$ is under the model distribution Q, for instance, using the probability $Q(\mathbf{x})$. These metrics are related to a single point on the

Task domain	Model	Finetuning	Dataset	Prompt length	Max. gen. length	Number of generations
Web text	GPT-2 (all sizes)	Pretrained	Webtext	35 tokens	1024 tokens	5000
News	Grover (all sizes)	Pretrained	RealNews	varying	1024 tokens	5000
Stories	GPT-2 medium	Finetuned	WritingPrompts	50 tokens	512 tokens	5000

Table 2: Dataset and task summary. Note that 1024 tokens correspond to ~ 750 words on average.

divergence curve, rather than a full summary. Examples include the perplexity of the test set (which is a sample from P) under the model Q and its generalizations to handle sparse distributions [27]. Unlike Mauve, these metrics never see model text samples $\mathbf{x}' \sim Q$, so they cannot account for how likely the model text is under the human distribution P. Moreover, they cannot be used for decoding algorithms such as beam search which do not define a token-level distribution.

Automatic metrics for specific domains such as dialog [41] or story [13] generation capture task-specific properties; see the surveys [7, 34]. In contrast, Mauve compares machine and human text in a domain-agnostic manner.

Non-automatic metrics. Due to the costs of human evaluation, we consider metrics requiring human evaluation, such as single-pair evaluation or HUSE [16] as complementary to automatic comparison measures such as Mauve. As a separate technical caveat, it is unclear how to use HUSE for sparse Q that assigns zero probability to a subset of text, which is the case with state-of-the-art decoding algorithms [18, 27].

Evaluation of Generative Models. Evaluation of generative models is an active area of research in computer vision, where generative adversarial networks [12] are commonly used. However, metrics such as Inception Score [36] are designed for supervised classification settings, and thus inappropriate for text generation. The Fréchet Distance [17, 38] and its unbiased counterpart, the Kernel Inception Distance [5] are both used for evaluating generative models, but unlike Mauve, do not take into account a trade-off between different kinds of errors between the learned and a reference distribution. Sajjadi et al. [35] and Kynkäänniemi et al. [21] both proposed metrics based on precision-recall curves, and Djolonga et al. [10] proposed a unified framework which encompassed both these works as special cases. Mauve extends the above line of work, and is operationalized for open-ended text generation, applicable for data generated by large-scale neural language models.

4 Experiments

We perform three sets of experiments to validate Mauve, compare it with existing text generation metrics, and study its approximation. In the first set (§4.1) we examine how known properties of generated text with respect to generation length, decoding algorithm, and model size can be identified and quantified by Mauve. Next, in §4.2 we demonstrate that Mauve's approximation is robust under various embedding models, quantization algorithms, and hyperparameter settings. Finally, in §4.3 we find that Mauve correlates with human judgments.

Tasks. We consider open-ended text generation using a text completion task [18, 44] in three domains: web text, news and stories. Each domain consists of a sequence dataset split into (context, continuation) pairs. Given a context $\mathbf{x}_{1:k}$, the task is to generate a continuation $\hat{\mathbf{x}}_{k+1:T} \sim Q(\cdot \mid p_\theta, \mathbf{x}_{1:k})$, forming a completion. Each ground-truth completion $\mathbf{x}_{1:T}$ is considered a sample from the true distribution P, while the completion $(\mathbf{x}_{1:k}, \hat{\mathbf{x}}_{k+1:T})$ is considered a sample from Q. The datasets, context and completion lengths, and number of completions used for each domain are shown in Table 2.

Models. As the language model $p_\theta(\mathbf{x})$ we use GPT-2, a large-scale transformer [42] pretrained on the web text dataset (see Radford et al. [31]), that is representative of state-of-the-art autoregressive language models. As the embedding model $M(\mathbf{x})$ we use GPT-2 Large, and compare others in §4.2.

Decoding Algorithms. We consider three common decoding algorithms: *ancestral sampling* which samples directly from the language model's per-step distributions, $x_t \sim p_\theta(x_t \mid \mathbf{x}_{<t})$, *nucleus sampling* [18] which samples from truncated per-step distributions, $x_t \sim q(x_t \mid p_\theta, \mathbf{x}_{<t})$, and *greedy decoding* which selects the most likely next-token, $x_t = \arg\max_{x \in \mathcal{V}} p_\theta(x \mid \mathbf{x}_{<t})$.

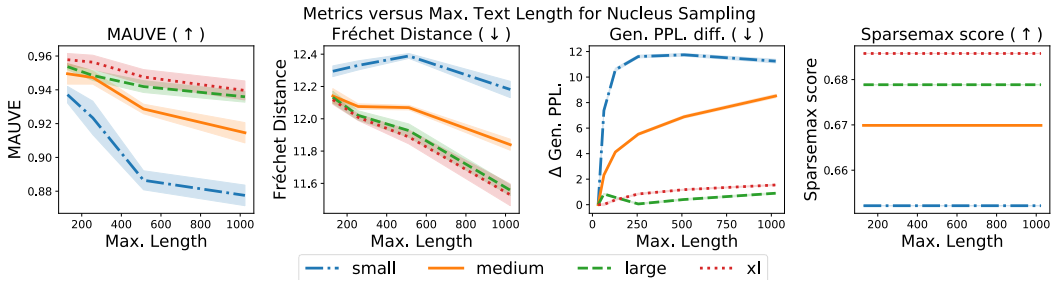

Figure 3: Generation quality versus maximum generation length according to Mauve and three alternative measures (web text, GPT-2). Mauve is the only comparison measure which identifies that generation quality decreases monotonically with increasing text length. The shaded area shows one standard deviation over generations from 5 random seeds.

We also consider an adversarial sampling procedure, designed to generate low-quality text that nevertheless matches the perplexity of human text. Adversarial perplexity sampling proceeds in two phases: (1) we generate the first 15% of tokens in a sequence uniformly at random from the vocabulary, and (2) we generate the remaining tokens greedily to make the running perplexity of the generated sequence as close as possible to the perplexity of human text.

4.1 Quantifying Properties of Generated Text

To study Mauve's effectiveness as a measure for comparing text distributions, we first examine how Mauve quantifies known properties of generated text: a good measure should meet expected behavior that is known from existing research on each property. Specifically, we investigate how Mauve behaves under changes in generation length, decoding algorithm, and model size.

Mauve quantifies quality differences due to generation length. Although large transformer-based models can generate remarkably fluent text, it has been observed that the quality of generation deteriorates with text length: as the generation gets longer, the model starts to wander, switching to unrelated topics and becoming incoherent [32]. As a result, an effective measure should indicate lower quality (e.g. lower Mauve) as generation length increases.

Figure 3 shows Mauve as the generation length increases, along with three alternative metrics: generation perplexity, sparsemax score, and Fréchet distance [17, 38]. Mauve reflects the desired behavior, showing a decrease in quality (lower Mauve) as generation length grows, with the trend consistent across model sizes. The other three metrics, however, show less favorable trends. Fréchet distance indicates *improving* quality as the length increases, while generation perplexity shows non-monotonic quality trends for the small and large models. Finally, language modeling metrics such as the sparsemax score [27] remain constant, since they do not depend on the samples generated.

Mauve identifies quality differences between decoding algorithms. Recent work has identified two clear trends in open-ended text generation with standard autoregressive models: (1) using greedy decoding results in repetitive, degenerate text [18, 44, 43]; (2) nucleus sampling (and related truncated sampling methods) yields higher quality text than ancestral sampling [11, 18].[4] An effective measure should thus indicate the quality relationship greedy $\prec$ ancestral $\prec$ nucleus.

Table 3 summarizes Mauve's quality measures of greedy decoding, ancestral sampling, and nucleus sampling, along with alternative automated metrics and a human quality score. Mauve correctly identifies the expected quality relationship, assigning the lowest quality to greedy decoding (.016) followed by ancestral sampling (.882), and the highest quality to nucleus sampling (.940). Other commonly-used metrics fail to identify this relationship: generation perplexity rates rates the highly degenerate greedy-decoded text as better than ancestral sampling (11.324 vs. 19.284), while the language-modeling metrics (SP, JS, ε-PPL) rate nucleus-decoded text as equal to or worse than greedy decoding or ancestral sampling. Finally, generation perplexity falls victim to the adversarial decoder (Adv.), while Mauve remains robust.[5]

[4] In general this relationship depends on the nucleus hyperparameter and task; however, [18] used the same setting that we consider here, and we additionally include a human-evaluated measure of quality.

[5] The results are consistent across model sizes and random seeds (see Appendix D).

	Adv.	Greedy	Sampling	Nucleus
Gen. PPL(↓)	**0.05**	11.3	19.3	1.54
Zipf(↓)	0.03	0.02	0.02	**0.01**
Self-BLEU(↓)	0.07	0.03	**0.02**	0.03
SP(↑)	–	0.50	**0.69**	0.69
JS(↓)	–	**0.35**	0.37	0.36
ε-PPL(↓)	–	497	**11.4**	13.7
Mauve (↑)	0.06	0.02	0.88	**0.94**
Human(↑)	–	–	9.0	**15.7**

Table 3: Quality differences between **decoding algorithms** (web text, GPT-2 xl). Mauve correctly captures the relationship greedy $\prec$ ancestral $\prec$ nucleus, and rates the adversarial decoder's text as low quality. Results are consistent across model sizes and random seeds. Boldfaced/highlighted entries denote the greatest similarity between the model and human distributions per each measure.

	Small	Medium	Large	XL
Gen. PPL(↓)	11.2	8.5	**0.9**	1.5
Zipf(↓)	0.06	**0.00**	0.02	0.01
Self-BLEU(↓)	0.05	**0.02**	0.03	0.03
SP(↑)	0.65	0.67	0.68	**0.69**
JS(↓)	0.41	0.39	0.37	**0.36**
ε-PPL(↓)	25.9	18.8	14.9	**13.7**
Mauve (↑)	0.878	0.915	0.936	**0.940**
Human(↑)	−15.9	−3.4	12.6	**15.7**

Table 4: Quality differences between **model sizes** (web text, nucleus sampling). Mauve captures the relationship between model size and generation quality, agreeing with human-evaluated quality. Results are consistent across random seeds and decoding algorithms. Boldfaced/highlighted entries denote the greatest similarity between the model and human distributions per each measure.

Mauve quantifies quality differences due to model size. Scaling the model size has been a key driver of recent advances in NLP, with larger models leading to better language modeling and higher quality generations in open-ended settings [31, 6]. An effective metric should capture the relationship between model size and generation quality, which we verify with human quality scores.

Table 4 shows Mauve's quality measures as the model size increases, along with alternatives and human quality scores. Mauve increases as model size increases, agreeing with the human quality measure and the expectation that larger models should have higher quality generations. The widely-used generation perplexity, however, incorrectly rates the large model's text as the best. Although the language modeling metrics (SP, JS, and ε-PPL) capture the size-quality relationship, they are constant with respect to length (Figure 3), and did not correctly quantify decoding algorithm quality (Table 3).

Table 6 in Appendix D shows additional results with ancestral sampling. In this case, human evaluators rated generations from the small model as better than those from the medium model. Interestingly, Mauve also identified this relationship, agreeing with the human ratings, in contrast to the other automatic metrics we surveyed.

Summary. Mauve identifies properties of generated text that a good measure should capture, related to length, decoding algorithm, and model size. In contrast, commonly used language modeling and statistical measures did not capture all of these properties. Unlike these alternatives, which capture a single statistic or relate to a single point on the divergence curve, Mauve's summary measure incorporates type I errors that quantify the degenerate text produced by greedy decoding (recall Figure 1), while capturing distribution-level information that describes quality changes from generation length, model size, and the nuanced distinction between ancestral and nucleus sampling.

4.2 Approximations in Mauve

Mauve summarizes the divergence between two text distributions with an approximation that relies on two components: an embedding model $M(\mathbf{x})$ and a quantization algorithm $\mathcal{A}$ (§3, Eq. (2)). We study the effects of these two components.

Mauve works with alternative embedding models. Figure 4 (left) shows that Mauve with features from RoBERTa large [23] gives qualitatively similar trends across model size and decoding as Mauve with features from GPT-2 large. Quantitatively, the Spearman rank correlation between them across all model and decoders is 0.993. We observe that RoBERTa penalizes smaller models more than GPT-2 but rates greedy decoding higher. We leave further study of inductive biases in the different embedding models to future work.

Mauve is robust to quantization. We compare different three different quantization algorithms:

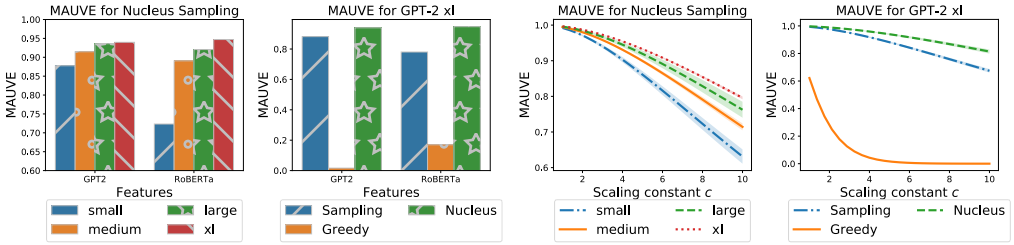

Figure 4: **Left**: Mauve computed using feature encodings from GPT-2 large (default) and RoBERTa large across model size and decoding. The Spearman rank correlation between the two is **0.993** across model sizes and decoding algorithms. Full numbers are given in Table 11. **Right**: Effect of the scaling constant c on Mauve. Choice of c does not affect the relative order of the curves but only the numerical value. We use $c = 5$ because it is is neither too small (where the nucleus sampling is pushed to 1) nor too large (where the greedy decoding is pushed to 0).

(a) k-Means: We cluster the hidden representations using k-means, and represent them by their cluster membership to get a discrete distribution with size equal to the number of clusters.

(b) Deep Residual Mixture Models (DRMM): As a generalization of k-means, we train a deep generative model known as DRMM [14]. We convert the soft clustering returned by DRMM into a hard clustering by assigning each point to its most likely cluster, and quantize the data using the cluster membership. We use DRMM with 3 layers and 10 components per layer for a total of 10^3 clusters, and train it for 20 epochs.

(c) Lattice Quantization: We learn a 4-dimensional feature representation of the vectors $M(\mathbf{x})$ using a deep network which maintains the neighborhood structure of the data while encouraging the features to be uniformly distributed on the unit sphere [33]. We quantize the data on a uniform lattice into 744 bins.

We compare different choices of the quantization to k-means with $k = 500$, which is our default. The Spearman rank correlation between Mauve computed with k-means for k ranging from 100 to 5000 correlates nearly perfectly with that of $k = 500$. In particular, the Spearman correlation is exactly 0.99 or 1.00. Likewise, Mauve computed with DRMM or lattice quantization has a near-perfect Spearman correlation of at least 0.99 with k-means. While the actual numerical value of Mauve could vary with the quantization algorithm, these results show that the *rankings induced by various variants of Mauve are nearly identical*.

Practical recommendation for scaling parameter. Figure 4 (right) shows the effects of adjusting the scaling parameter c, which does not affect the relative order of the divergence curve, but adjusts the numerical value returned by Mauve. As a practical recommendation, we found $c = 5$ to yield interpretable values.

4.3 Correlation with Human Judgments

Next, we evaluate how Mauve's quality judgments correlate with human quality judgments. In our study, a quality judgment means choosing a particular (model, decoder) setting over another based on their generations. An effective metric should yield judgments that correlate highly with human judgments, assuming that human evaluators represent a gold-standard. First, we give an overview of our evaluation protocol; for full details and additional results, see Appendix E.

Evaluation protocol. To obtain human judgments, we employ a pairwise setup: at each round, an annotator receives a context and continuations from two different (model, decoder) settings, and selects the continuation they found more natural using a 5-point Likert scale. Our interface for collecting annotations is shown in Figure 8 of Appendix E.

We collect these annotations for web text generation with 8 different (model, decoder) settings plus a ninth setting for human-written continuations. Each setting is a GPT-2 model size paired with either ancestral or nucleus sampling. This gives us a total of 36 pairs of settings. Given the known difficulties with human evaluation of longer texts [20], we use a maximum completion length of 256 tokens. We obtain 90 preference ratings for each pair of settings, coming from a total of 214

Metric	Task	Gen. PPL	Zipf Coef.	REP	Distinct-4	Self-BLEU	Mauve
Human-like/BT	Web text	0.810	0.833	−0.167	0.738	0.595	**0.952**
Interesting/BT	Web text	0.643	0.524	−0.143	0.524	0.405	**0.810**
Sensible/BT	Web text	0.738	0.690	−0.071	0.595	0.524	**0.857**
% Disc. Acc.	News	0.468	0.595	0.792	0.653	0.516	**0.956**
% Disc. Acc.	Stories	0.643	0.643	0.250	0.750	0.857	**0.893**

Table 5: Correlation of various similarity measures with human judgments when available, and the accuracy of a trained discriminator otherwise. "BT" denotes the Bradley-Terry score for a pairwise human evaluation (§ 4.3). Boldfaced/highlighted numbers indicate highest correlation in each row. We observe that Mauve has the highest correlation with human evaluation and discriminator accuracy.

crowd-workers from the Amazon Mechanical Turk platform. The evaluators were paid USD 0.40 per evaluation based on an estimated wage of USD 16 per hour.

We convert these pairwise preferences to a ranking by fitting a Bradley-Terry model [25], a parametric model used to predict the outcome of a head-to-head comparison. In particular, we obtain a score w_i for each setting i so that the log odds of humans preferring setting i to setting j in a head-to-head comparison is given by the difference $w_i - w_j$. For a given comparison measure, we compute the Spearman rank correlation between the comparison measure and the fitted Bradley-Terry coefficients w_i for each of the (model, decoder) settings. The end result is a correlation score in $[-1, 1]$, with higher values meaning that quality judgments using the comparison measure correlate with quality judgments made by human evaluators.

Mauve correlates with human judgments. Table 5 shows the correlation between human judgments and five automatic evaluation metrics obtained using our evaluation protocol on the web text domain. Mauve correlates highly with human judgments of how human-like (0.952), interesting (0.810), and sensible (0.857) the machine text is. Mauve's correlations with human judgments are substantially higher than those for the other automated measures; for instance, the commonly-used generation perplexity has correlations that are 0.12 to 0.17 lower than Mauve's. The results suggest that Mauve may act as an effective, automatic surrogate for costly human judgments.

Mauve correlates with learned discriminators. We also measure the quality of generations by how well a trained model (a discriminator) can distinguish between real and generated text. We report the test accuracy of a binary classifier trained to discriminate between machine and human text; a lower discrimination accuracy implies that the generation is harder to distinguish from human text. We report the accuracy of Grover mega as the discriminator for the news generations as it produced the highest discrimination accuracy [46] while we use GPT-2 large for the story domain. As seen in Table 5, Mauve correlates the highest with the discrimination accuracy (0.96 for news and 0.89 for stories) among all comparison measures. Computing the discrimination accuracy for each (model, decoder) pair requires fine-tuning a separate model, which is particularly expensive for large models such as Grover mega. Mauve, on the other hand, does not require any training.

5 Conclusion

We presented Mauve, an automatic comparison metric for open-ended text generation. Mauve measures the area under a divergence curve, formalizing and summarizing a spectrum of errors that capture phenomena present in machine and human-generated text. Mauve correlated with human judgments and identified quality differences due to generation length, decoding algorithm, and model size, which existing metrics struggled to capture. Automated metrics have driven advances in computer vision and many other machine learning domains. Mauve's principled foundation and strong empirical performance offers a similar path forward for open-ended text generation systems.

Broader Impacts Statement Mauve rewards model text which resembles human-authored text. However, we acknowledge the risks of rewarding systems that try to mimic humans [4], which is the ultimate goal of open-ended text generation. While our research is important for developing better language generators, we also encourage the community to pay attention to the development of technology that can reliably distinguish between human and machine text. We leave the extension of our method towards building such systems to future work.

References

[1] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

[2] S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, 2005.

[3] A. Belz, S. Mille, and D. M. Howcroft. Disentangling the Properties of Human Evaluation Methods: A Classification System to Support Comparability, Meta-Evaluation and Reproducibility Testing. In *INLG*, pages 183–194, 2020.

[4] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proc. of FAccT*, 2021.

[5] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton. Demystifying MMD GANs. In *Proc. of ICLR*, 2018.

[6] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language Models are Few-Shot Learners. In *Proc. of NeurIPS*, 2020.

[7] A. Celikyilmaz, E. Clark, and J. Gao. Evaluation of Text Generation: A Survey. *arXiv Preprint*, 2020.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*, pages 4171–4186, 2019.

[9] E. Dinan, V. Logacheva, V. Malykh, A. Miller, K. Shuster, J. Urbanek, D. Kiela, A. Szlam, I. Serban, R. Lowe, S. Prabhumoye, A. W. Black, A. Rudnicky, J. Williams, J. Pineau, M. Burtsev, and J. Weston. The Second Conversational Intelligence Challenge (ConvAI2), 2019.

[10] J. Djolonga, M. Lucic, M. Cuturi, O. Bachem, O. Bousquet, and S. Gelly. Precision-Recall Curves Using Information Divergence Frontiers. In *Proc. of AISTATS*, pages 2550–2559, 2020.

[11] A. Fan, M. Lewis, and Y. N. Dauphin. Hierarchical Neural Story Generation. In *Proc. of ACL*, pages 889–898, 2018.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. In *Proc. of NeurIPS*, 2014.

[13] J. Guan and M. Huang. UNION: An Unreferenced Metric for Evaluating Open-ended Story Generation. In *Proc. of EMNLP*, pages 9157–9166, 2020.

[14] P. Hämäläinen and A. Solin. Deep Residual Mixture Models. *arXiv preprint*, 2020.

[15] T. S. Han and K. Kobayashi. *Mathematics of Information and Coding*, volume 203. American Mathematical Soc., 2007.

[16] T. Hashimoto, H. Zhang, and P. Liang. Unifying human and statistical evaluation for natural language generation. In *Proc. of NAACL*, pages 1689–1701, 2019.

[17] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Proc. of NeurIPS*, page 6629–6640, 2017.

[18] A. Holtzman, J. Buys, M. Forbes, and Y. Choi. The Curious Case of Neural Text Degeneration. In *Proc. of ICLR*, 2020.

[19] D. R. Hunter. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1):384–406, 2004.

[20] D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck. Automatic Detection of Generated Text is Easiest when Humans are Fooled. In *Proc. of ACL*, pages 1808–1822, July 2020.

[21] T. Kynkäänniemi, T. Karras, S. Laine, J. Lehtinen, and T. Aila. Improved Precision and Recall Metric for Assessing Generative Models. In *Proc. of NeurIPS*, 2019.

[22] C.-Y. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, 2004.

[23] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv Preprint*, 2019.

[24] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 2001. ISBN 978-0-262-13360-9.

[25] J. I. Marden. *Analyzing and modeling rank data*, volume 64 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London, 1995. ISBN 0-412-99521-2.

[26] A. Martins and R. Astudillo. From Softmax to Sparsemax: A Sparse model of Attention and Multi-label Classification. In *Proc. of ICML*, pages 1614–1623. PMLR, 2016.

[27] P. H. Martins, Z. Marinho, and A. F. T. Martins. Sparse Text Generation. In *Proc. EMNLP*, pages 4252–4273, 2020.

[28] L. Massarelli, F. Petroni, A. Piktus, M. Ott, T. Rocktäschel, V. Plachouras, F. Silvestri, and S. Riedel. How Decoding Strategies Affect the Verifiability of Generated Text. *arXiv preprint arXiv:1911.03587*, 2019.

[29] K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12. Springer Science & Business Media, 2012.

[30] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL*, pages 311–318, 2002.

[31] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9, 2019.

[32] H. Rashkin, A. Celikyilmaz, Y. Choi, and J. Gao. PlotTMachines: Outline-Conditioned Generation with Dynamic Plot State Tracking. *arXiv Preprint*, 2020.

[33] A. Sablayrolles, M. Douze, C. Schmid, and H. Jégou. Spreading vectors for similarity search. In *Proc. of ICLR*, 2019.

[34] A. B. Sai, A. K. Mohankumar, and M. M. Khapra. A Survey of Evaluation Metrics Used for NLG Systems. *arXiv Preprint*, 2020.

[35] M. S. M. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly. Assessing generative models via precision and recall. In *Proc. of NeurIPS*, 2018.

[36] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved Techniques for Training GANs, 2016.

[37] T. Sellam, D. Das, and A. P. Parikh. BLEURT: Learning Robust Metrics for Text Generation. In D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, editors, *Proc. of ACL*, pages 7881–7892, 2020.

[38] S. Semeniuta, A. Severyn, and S. Gelly. On Accurate Evaluation of GANs for Language Generation. *arXiv Preprint*, 2018.

[39] H. Shimanaka, T. Kajiwara, and M. Komachi. RUSE: Regressor Using Sentence Embeddingsfor Automatic Machine Translation Evaluation. In *Proc. of Conference on Machine Translation*, pages 751–758, 2018.

[40] A. Shimorina and A. Belz. The Human Evaluation Datasheet 1.0: A Template for Recording Details of Human Evaluation Experiments in NLP. *arXiv Preprint*, 2021.

[41] C. Tao, L. Mou, D. Zhao, and R. Yan. RUBER: An Unsupervised Method for Automatic Evaluation of Open-Domain Dialog Systems. In *Proc. of AAAI*, volume 32, 2018.

[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is All you Need. In *Proc. of NeurIPS*, pages 5998–6008, 2017.

[43] S. Welleck, I. Kulikov, J. Kim, R. Y. Pang, and K. Cho. Consistency of a Recurrent Language Model With Respect to Incomplete Decoding. In *Proc. of EMNLP*, pages 5553–5568, 2020.

[44] S. Welleck, I. Kulikov, S. Roller, E. Dinan, K. Cho, and J. Weston. Neural Text Generation With Unlikelihood Training. In *Proc. of ICLR*, 2020.

[45] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. Transformers: State-of-the-Art Natural Language Processing. In *Proc. of EMNLP*, pages 38–45, 10 2020.

[46] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi. Defending Against Neural Fake News. In *Proc. of NeurIPS*, 2019.

[47] H. Zhang, D. Duckworth, D. Ippolito, and A. Neelakantan. Trading off diversity and quality in natural language generation. In *Proc. of HumEval*, pages 25–33, 2021.

[48] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. BERTScore: Evaluating text generation with BERT. In *Proc. of ICLR*, 2020.

[49] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger. MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. In *Proc. of EMNLP*, 2019.

[50] Y. Zhu, S. Lu, L. Zheng, J. Guo, W. Zhang, J. Wang, and Y. Yu. Texygen: A Benchmarking Platform for Text Generation Models, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See §4.2
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See "Broader Impacts" on page 9.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] The only theoretical resutls are in Appendix A
   (b) Did you include complete proofs of all theoretical results? [Yes] Appendix A
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] We will release a library `mauve` under the GPLv3 license; see Appendix B
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] See Appendix E
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] See Appendix E