

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:** Illustration of *Type I errors*, where Q produces degenerate, repetitive text which is unlikely under P , and, *Type II errors*, where Q cannot produce plausible human text due to truncation heuristics [26]. MAUVE measures these errors softly, by using the mixture distribution R_λ . Varying λ in $(0, 1)$ gives a divergence curve and captures a spectrum of soft Type I and Type II errors. MAUVE summarizes the entire divergence curve in a single scalar as the area under this curve.

We summarize our contributions. First, we introduce MAUVE, a 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 open-source a pip-installable Python package to compute MAUVE.¹

2 MAUVE

We begin by discussing the basics of open-ended text generation, and then introduce MAUVE for measuring the divergence between machine generated text and human text.

Open-ended Text Generation. A language model is an estimate $\hat{P}(x)$ of the probability distribution over sequences of text $x = (x_1, \dots, x_{|x|})$, consisting of tokens x_t belonging to a fixed vocabulary (e.g. characters, or words). Prevailing neural autoregressive language models estimate the joint distribution $\hat{P}(x)$ by modeling the conditional distribution $\hat{P}(x_{t+1}|x_{1:t})$ over the next token in a sequence. The open-ended text generation task asks us to output text $\hat{x}_{t+1:|x|}$ in continuation of a given context $x_{1:t}$. 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, by sampling $\hat{x}_{t+1} \sim \hat{P}(\cdot|x_{1:t})$, $\hat{x}_{t+2} \sim \hat{P}(\cdot|x_{1:t}, \hat{x}_{t+1})$, etc. In practice, this simple decoding algorithm is often modified by adjusting the conditional distribution $\hat{P}(\cdot|x_{1:t})$ 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 [18, 26]. 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 gap between the model distribution Q and the target distribution P of human text. As highlighted in Figure 1, this gap 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 [15, 26, 59] that are highly unlikely to be written by humans.² The Type II

¹Available from <https://github.com/krishnap25/mauve>. See Appendix B for an example of the mauve package in action.

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

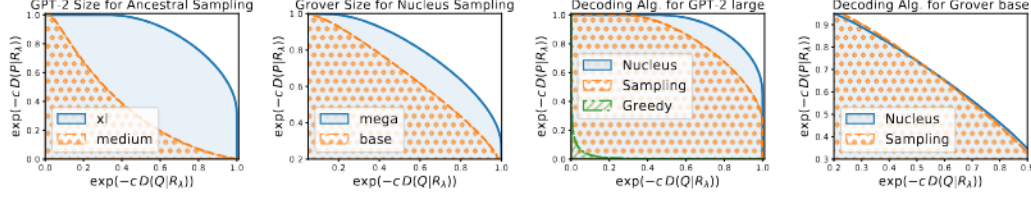


Figure 2: Divergence curves for different models (GPT-2 [45], Grover [61]) and decoding algorithms (greedy decoding, ancestral and nucleus sampling). MAUVE is computed as the area of the shaded region, and larger values of MAUVE indicate that Q is closer to P . In general, MAUVE indicates that generations from larger models and nucleus sampling are closer to human text. **Rightmost:** Nucleus sampling has a slightly smaller Type I error than ancestral sampling but a higher Type II error, indicating that ancestral sampling with Grover base produces more degenerate text while nucleus sampling does not effectively cover the human text distribution.

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 [26]. The gap between P and Q is small only if both of these errors are small.

Quantifying the Errors. We formalize the Type I and II errors with the Kullback-Leibler (KL) divergences $\text{KL}(Q|P)$ and $\text{KL}(P|Q)$, respectively. The divergence $\text{KL}(Q|P)$ penalizes Q if there exists text x such that $Q(x)$ is large but $P(x)$ is small, so it quantifies the Type I error. Likewise, $\text{KL}(P|Q)$ quantifies the Type II error.

Unfortunately, one or both of the KL divergences $\text{KL}(P|Q)$ and $\text{KL}(Q|P)$ are infinite if the supports of P and Q are not identical, which is often the case in open-ended generation. This makes the KL divergence itself unsuitable as an evaluation metric. We overcome this issue by *softly* measuring the two errors using the mixture distribution $R_\lambda = \lambda P + (1 - \lambda)Q$ for some $\lambda \in (0, 1)$. In particular, we define the (soft) Type I error at level λ as $\text{KL}(Q|R_\lambda)$ and the (soft) Type II error as $\text{KL}(P|R_\lambda)$.

Summarizing the Errors with a Divergence Curve. Since the mixture weight λ was arbitrary, we consider a family of Type I and II error values by varying λ between 0 and 1, in the same spirit as information divergence frontiers [49, 16]. This yields a *divergence curve*,

$$\mathcal{C}(P, Q) = \left\{ \left(\exp(-c \text{KL}(Q|R_\lambda)), \exp(-c \text{KL}(P|R_\lambda)) \right) : R_\lambda = \lambda P + (1 - \lambda)Q, \lambda \in (0, 1) \right\}, \quad (1)$$

where $c > 0$ is a hyperparameter for scaling. The divergence curve formalizes and encodes information about the trade-off between Type I and II errors.³ Figure 2 illustrates the divergence curves for different models and decoding algorithms.

Our proposed measure, $\text{MAUVE}(P, Q)$, is the area under the divergence curve $\mathcal{C}(P, Q)$. It provides a scalar summary of the trade-off between Type I and Type II errors. $\text{MAUVE}(P, Q)$ lies in $(0, 1]$, with a larger value meaning that Q is closer to P . Further, $\text{MAUVE}(P, Q) = 1$ if and only if $Q = P$. The area under the curve is a common summary of trade-off curves in machine learning [13, 11, 12, 19].

Connections to Common Divergences. The divergence curve encodes more information than the KL divergence $\text{KL}(P|Q)$, which can be obtained from the second coordinate of the curve $\mathcal{C}(P, Q)$ as $\lambda \rightarrow 0$, and the reverse KL divergence $\text{KL}(Q|P)$ which can be obtained from the first coordinate of the curve $\mathcal{C}(P, Q)$ as $\lambda \rightarrow 1$. Further, the Jensen-Shannon (JS) divergence $\text{JS}(P, Q) = (\text{KL}(P|R_{1/2}) + \text{KL}(Q|R_{1/2}))/2$, 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

$$\text{KL}(P|R_\lambda) = \sum_x P(x) \log \frac{P(x)}{R_\lambda(x)}, \quad (2)$$

and a similarly defined coordinate $\text{KL}(Q|R_\lambda)$. We cannot compute the summation as written in Eq. (2), as we do not know the ground-truth probabilities $P(\cdot)$ and the support of a typical model

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

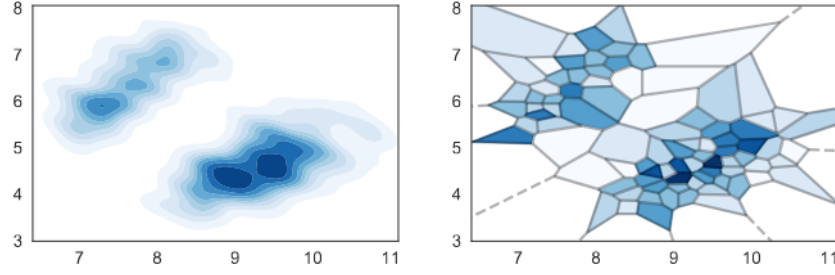


Figure 3: Illustration of the quantization. **Left:** A continuous two-dimensional distribution P . **Right:** A partitioning of the Euclidean plane $\mathbb{R}^2$ and the corresponding quantized distribution $\tilde{P}$.

distribution is prohibitively large, since it is the space of all sequences of tokens. As a result of these two issues, MAUVE cannot be tractably computed in closed form.

We employ a Monte Carlo estimator using samples $\mathbf{x}_i \sim P$ and $\mathbf{x}'_i \sim Q$ to overcome the fact that ground-truth probabilities $P(\cdot)$ are unknown. We circumvent the intractable support size by computing MAUVE in a quantized embedding space that is sensitive to important features of text.

The overall estimation procedure is as follows. First, we sample human text $\mathbf{x}_i \sim P$ and machine text $\mathbf{x}'_i \sim Q$. We then embed each text sequence using an external language model M (e.g., GPT-2 [45]) to obtain embeddings $\{M(\mathbf{x}_i)\}_{i=1}^N$ and $\{M(\mathbf{x}'_i)\}_{i=1}^{N'}$. Each embedding is now a vector $M(\mathbf{x}) \in \mathbb{R}^d$. Next, we jointly quantize the embedded samples (e.g. with k -means [36]), and count the cluster assignments to form histograms, giving low-dimensional discrete distributions that approximate each high-dimensional text distribution. In particular, the distribution P of human text is approximated by the discrete distribution $\tilde{P}$ of support size k , which is defined as,

$$\tilde{P}(j) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\phi(\mathbf{x}_i) = j), \quad (3)$$

where $\phi(\mathbf{x}) \in \{1, \dots, k\}$ returns the cluster id of $\mathbf{x}$. The model distribution Q is approximated as $\tilde{Q}$ similarly. Here, $\tilde{P}$ and $\tilde{Q}$ can be interpreted as piecewise constant approximations of P and Q , similar to a histogram; see Figure 3 for an illustration. Computing the divergence curve is now tractable, as each coordinate is a KL divergence between the k -element discrete distributions.

To recap, our proposed measure $\text{MAUVE}(P, Q)$ 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 present empirical results with 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 for n -gram matching [44, 32, 2], which are designed to capture similarities in the surface form of the generated text and the human references, making them fundamentally ill-suited for open-ended generation. Moreover, it has been recently shown in [42] show that these classical metrics only weakly agree with human judgments.

More recent reference-based metrics are capable of comparisons in a high dimensional space [53, 63, 51, 9], thereby capturing distributional semantics beyond superficial n -gram statistics. For instance, Moverscore [64] relies on the Word Mover’s distance [30], and is an instance of an optimal transportation distance [57]. It computes the minimum cost of transforming the generated text to the reference text, taking into account Euclidean distance between vector representations of n -grams,