Model-Based Minimum Bayes Risk Decoding for Text Generation

Yuu Jinnai [1] **Tetsuro Morimura** [1] **Ukyo Honda** [1] **Kaito Ariu** [1] **Kenshi Abe** [1]

Abstract

Minimum Bayes Risk (MBR) decoding has been shown to be a powerful alternative to beam search decoding in a variety of text generation tasks. MBR decoding selects a hypothesis from a pool of hypotheses that has the least expected risk under a probability model according to a given utility function. Since it is impractical to compute the expected risk exactly over all possible hypotheses, two approximations are commonly used in MBR. First, it integrates over a sampled set of hypotheses rather than over all possible hypotheses. Second, it estimates the probability of each hypothesis using a Monte Carlo estimator. While the first approximation is necessary to make it computationally feasible, the second is not essential since we typically have access to the model probability at inference time. We propose model-based MBR (MBMBR), a variant of MBR that uses the model probability itself as the estimate of the probability distribution instead of the Monte Carlo estimate. We show analytically and empirically that the model-based estimate is more promising than the Monte Carlo estimate in text generation tasks. Our experiments show that MBMBR outperforms MBR in several text generation tasks, both with encoder-decoder models and with language models. Our code is available at https://github.com/CyberAgentAILab/model-based-mbr.

1. Introduction

One of the key components of text generation is the decoding strategy, which is the decision rule used to generate sentences from the model. Beam search is the popular decoding strategy that has been widely used in many directed text generation tasks, including machine translation (Wu et al., 2016; Ott et al., 2019; Wolf et al., 2020), text summarization (Rush et al., 2015; Narayan et al., 2018), and image captioning (Anderson et al., 2017). However, beam search is known to have several degeneration problems. For example, Welleck et al. (2020) report that beam search can yield infinite-length outputs that the model assigns zero probability to.

Minimum Bayes Risk (MBR) decoding has recently gained attention as a decoding strategy with the potential to overcome the problems of beam search (Goodman, 1996; Kumar & Byrne, 2004; Eikema & Aziz, 2020; 2022; Freitag et al., 2022; Bertsch et al., 2023). MBR decoding consists of the following steps. First, it samples multiple sequences from the model. Then, it compares each sequence to the others according to a utility function. Finally, it selects the sequence that maximizes the expected utility over the estimated probability distribution over the sequences.

Previous work on MBR decoding uses a Monte Carlo estimate to approximate the probability distribution since it is an unbiased estimate of the true model distribution. However, because the number of possible hypotheses is enormous compared to the number of samples, the estimation error of the Monte Carlo estimate is huge. This can lead to a huge error in the expected utility estimate.

We propose **model-based MBR (MBMBR)**, a variant of MBR that uses a model-based estimate of the MBR objective instead of a Monte Carlo estimate. The model-based estimate uses the probability model itself, but with its domain limited to the set of observed hypotheses. As such, the estimate is computationally feasible and as accurate as the model for the observed sequences. MBMBR is easy to implement and requires only the model probability of the sampled sequences, which can be obtained concurrently with the sampling procedure.

We first evaluate the model-based estimate and show analytically that the Kullback-Leiber (KL) divergence from the true model probability is guaranteed to be lower than that of the Monte Carlo estimate. We also empirically evaluate it on various text generation tasks, indicating that model-based estimate is better for the use of MBR.

We then apply MBMBR to several text generation tasks: machine translation, text summarization, image captioning,

[1]CyberAgent, Tokyo, Japan. Correspondence to: Yuu Jinnai <jinnai_yu@cyberagent.co.jp>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

and data-to-text. We evaluate it in two settings: using a domain-specific conditional generation model, and using a language model (LLM) with prompting. MBMBR outperforms MBR in all settings except when the model generates low-quality sequences. The experimental results show that MBMBR is an effective decoding strategy that can replace MBR to improve text generation in a wide range of settings.

2. Background

Text generation is the task of generating an output sequence $\mathbf{y}$ given an input sequence $\mathbf{x}$. Probabilistic text generators define a probability distribution $P_{\text{model}}(\mathbf{y}|\mathbf{x})$ over an output space of hypotheses $\mathcal{Y}$. The set of complete hypotheses $\mathcal{Y}$ is:

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

where $\circ$ is a string concatenation and $\mathcal{V}^*$ is the Kleene closure of a set of vocabulary $\mathcal{V}$. The goal of decoding is to find the highest-scoring hypothesis for a given input.

One of the most common decision rules is maximum-a-posteriori (MAP) decoding. MAP decoding finds the most probable translation under the model:

$$\mathbf{h}^{\text{MAP}} = \arg\max_{\mathbf{h} \in \mathcal{Y}} P_{\text{model}}(\mathbf{h}|\mathbf{x}). \tag{2}$$

In this paper, we denote $P_{\text{model}}(\mathbf{h}|\mathbf{x})$ by $P_{\text{model}}(\mathbf{h})$ for brevity. Although it seems intuitive to solve this MAP objective, previous work has pointed out two critical problems with this strategy. First, because the size of the hypothesis set $|\mathcal{Y}|$ is extremely large, it is intractable to solve exactly. Second, the MAP objective often leads to low quality output (Stahlberg & Byrne, 2019; Holtzman et al., 2020; Meister et al., 2020). Indeed, Stahlberg & Byrne (2019) show that $\mathbf{h}^{\text{MAP}}$ is often found to be the empty sequence in their experimental setting.

As such, beam search is commonly used as a heuristic algorithm to solve a decoding problem (Graves, 2012; Sutskever et al., 2014). Beam search considers a fixed number k of options at each step. It is known to produce higher quality sequences than MAP decoding in a wide range of tasks. However, beam search is known to have degeneration problems such as repetitions and infinite length outputs (Cohen & Beck, 2019; Holtzman et al., 2020).

2.1. Minimum Bayes Risk Decoding

Unlike MAP decoding, which searches for the most probable output, MBR decoding searches for the output that maximizes expected utility, which is equivalent to minimizing risk (Goel & Byrne, 2000; Kumar & Byrne, 2002; 2004). The procedure consists of two components: a text generation model $P_{\text{model}}(\mathbf{y})$ and a utility metric $u(\mathbf{h}, \mathbf{y})$.[1]

[1]We denote $P_{\text{model}}(\mathbf{y}|\mathbf{x})$ by $P_{\text{model}}(\mathbf{y})$ for simplicity.

The utility metric $u(\mathbf{h}, \mathbf{y})$ estimates the quality of a candidate output $\mathbf{h}$ given a reference output $\mathbf{y}$. Given a set of candidate hypotheses $\mathcal{H}_{\text{cand}} \subseteq \mathcal{Y}$, MBR decoding selects the best hypothesis according to its expected utility over the distribution of human references:

$$\mathbf{h}^{\text{human}} = \arg\max_{\mathbf{h} \in \mathcal{H}_{\text{cand}}} \mathop{\mathbb{E}}_{\mathbf{y} \sim P_{\text{human}}} [u(\mathbf{h}, \mathbf{y})]. \tag{3}$$

Since P_{human} is unknown, MBR instead uses the model probability P_{model} to approximate P_{human}.

$$\begin{aligned} \mathbf{h}^{\text{model}} &= \arg\max_{\mathbf{h} \in \mathcal{H}_{\text{cand}}} \mathop{\mathbb{E}}_{\mathbf{y} \sim P_{\text{model}}} [u(\mathbf{h}, \mathbf{y})] \\ &= \arg\max_{\mathbf{h} \in \mathcal{H}_{\text{cand}}} \sum_{\mathbf{y} \in \mathcal{Y}} u(\mathbf{h}, \mathbf{y}) \cdot P_{\text{model}}(\mathbf{y}). \end{aligned} \tag{4}$$

For the rest of the paper, we will denote P_{model} as P for simplicity, unless otherwise noted. Since integration over $\mathcal{Y}$ is computationally intractable, Eq. (4) is approximated by a **Monte Carlo estimate** (Eikema & Aziz, 2022; Farinhas et al., 2023) using a collection of reference hypotheses $\mathcal{H}_{\text{ref}}$ sampled from the model P:[2]

$$\mathbf{h}^{\text{MC}} = \arg\max_{\mathbf{h} \in \mathcal{H}_{\text{cand}}} \frac{1}{|\mathcal{H}_{\text{ref}}|} \sum_{\mathbf{y} \in \mathcal{H}_{\text{ref}}} u(\mathbf{h}, \mathbf{y}). \tag{5}$$

Eq. (5) is derived by replacing the true model probability P in Eq. (4) with the **empirical distribution** $\hat{P}$, which is the number of occurrences of $\mathbf{y}$ in $\mathcal{H}_{\text{ref}}$ divided by the number of samples $|\mathcal{H}_{\text{ref}}|$:

$$\hat{P}(\mathbf{y}) = \frac{\sum_{\mathbf{y}' \in \mathcal{H}_{\text{ref}}} \mathbb{I}(\mathbf{y} = \mathbf{y}')}{|\mathcal{H}_{\text{ref}}|}. \tag{6}$$

Using $\hat{P}$, we can rewrite Eq. (5) as follows:

$$\begin{aligned} \mathbf{h}^{\text{MC}} &= \arg\max_{\mathbf{h} \in \mathcal{H}_{\text{cand}}} \mathop{\mathbb{E}}_{\mathbf{y} \sim \hat{P}} [u(\mathbf{h}, \mathbf{y})] \\ &= \arg\max_{\mathbf{h} \in \mathcal{H}_{\text{cand}}} \sum_{\mathbf{y} \in \mathcal{Y}} u(\mathbf{h}, \mathbf{y}) \cdot \hat{P}(\mathbf{y}) \\ &= \arg\max_{\mathbf{h} \in \mathcal{H}_{\text{cand}}} \sum_{\mathbf{y} \in \mathcal{H}_{\text{ref}}} u(\mathbf{h}, \mathbf{y}) \cdot \hat{P}(\mathbf{y}). \end{aligned} \tag{7}$$

Note that $\hat{P}(\mathbf{y})$ is zero for $\mathbf{y} \in \mathcal{Y} \setminus \mathcal{H}_{\text{ref}}$. Standard practice is to use the same set of hypotheses for the candidate set $\mathcal{H}_{\text{cand}}$ and the reference pool $\mathcal{H}_{\text{ref}}$ ($\mathcal{H} := \mathcal{H}_{\text{cand}} = \mathcal{H}_{\text{ref}}$).

The choice of sampling algorithm to collect $\mathcal{H}$ has been studied extensively, as it has been shown to be critical to the performance of MBR (Ohashi et al., 2024). Recent work has shown that MBR works best with a probabilistic sampling algorithm (Eikema & Aziz, 2020; Fernandes et al., 2022; Freitag et al., 2023).

[2]$\mathcal{H}_{\text{ref}}$ is a collection where it may have multiple instances of the same element.

Method		Estimator	Objective
Target Objective	$\mathbf{h}^{\text{model}}$ (Eq. 4)	P	$\sum_{\mathbf{y}\in\mathcal{Y}} u(\mathbf{h},\mathbf{y})P(\mathbf{y})$
Monte Carlo (MBR)	$\mathbf{h}^{\text{MC}}$ (Eq. 7)	$\hat{P}$	$\sum_{\mathbf{y}\in\mathcal{H}_{\text{ref}}} u(\mathbf{h},\mathbf{y})\hat{P}(\mathbf{y})$
Model-Based (MBMBR)	$\mathbf{h}^{\text{MB}}$ (Eq. 8)	$\hat{P}_{\text{MB}}$	$\sum_{\mathbf{y}\in\mathcal{R}} u(\mathbf{h},\mathbf{y})P(\mathbf{y})$

Table 1. Comparison of the surrogate objective functions of MBMBR and MBR. MBR uses Monte Carlo estimate to approximate the target MBR objective whereas MBMBR uses the model P as is.

3. Model-Based Minimum Bayes Risk (MBMBR) Decoding

The estimation error of the Monte Carlo sum comes from two approximations. First, the domain of the reference hypotheses is restricted to the collection of sampled sentences $\mathcal{H}_{\text{ref}}$ instead of $\mathcal{Y}$. This approximation is necessary because enumerating all hypotheses in $\mathcal{Y}$ is infeasible. Second, the probability of each sentence in $\mathcal{H}_{\text{ref}}$ is estimated by the Monte Carlo estimate instead of the true model probability P. While the first approximation is necessary, the second approximation is unnecessary if the model probability is accessible, which is the case for most decoding scenarios. To this end, we propose a **model-based MBR** (MBMBR) decoding, a variant of MBR that uses a **model-based estimate** instead of a Monte Carlo estimate:

$$\mathbf{h}^{\text{MB}} = \underset{\mathbf{h}\in\mathcal{H}_{\text{cand}}}{\arg\max} \sum_{\mathbf{y}\in\mathcal{R}} u(\mathbf{h},\mathbf{y}) \cdot P(\mathbf{y}), \tag{8}$$

where $\mathcal{R}$ is a set of references where the duplicates are removed from the $\mathcal{H}_{\text{ref}}$. The model-based estimate simply replaces $\hat{P}$ with P. While vanilla MBR estimates the probability distribution using only the samples, MBMBR uses both the samples and their model probability to estimate. Intuitively, MBMBR can fully exploit the inherent properties of the given model. This allows a more accurate probability density to be computed, resulting in more accurate estimates.

The idea of using model probability for MBR is not new per se. The model probability P is used when $\mathcal{H}_{\text{ref}}$ is collected by a beam search instead of probabilistic sampling algorithms (Goel & Byrne, 2000; Kumar & Byrne, 2002; 2004). The novelty of the idea is to show that we can make use of the model probability with a probabilistic sampling algorithm to further improve the MBR decoding. In the following sections, we show that (1) the divergence of the estimated distribution is guaranteed to be smaller (or equal in extremely unlikely cases) than using the Monte Carlo estimate (Section 4), (2) the divergence of the estimated distribution is reduced so that about half as many samples are needed to achieve the same divergence as with Monte Carlo estimate (Figures 1 and 2), and (3) using the model probability, MBMBR outperforms MBR in a wide range of text generation tasks (Section 5).

Mitigating the Length Bias. The problem of using P directly for MBMBR is that it imposes a bias on the hypothesis selection if the model is biased. Length bias is one of the well-known problems that text generation model tends to generate outputs that are too short (Koehn & Knowles, 2017; Murray & Chiang, 2018; Stahlberg & Byrne, 2019). This is problematic since P_{human} is unlikely to have such a bias as strong as the model probability P. Although previous work shows that vanilla MBR decoding generates outputs relatively close in length to the references (Bertsch et al., 2023), MBMBR is directly subject to the length bias since it uses the probability itself (See Section 5.3.1).

To remedy this problem, we use length normalization with a P normalized by the length of the sequence (Murray & Chiang, 2018; Bertsch et al., 2023). Suppose the model probability is biased to generate shorter sequences: $P(\mathbf{y}) \sim e^{-\ell(\mathbf{y})}P_{\text{human}}(\mathbf{y})$, where $\ell(\mathbf{y})$ is the length of $\mathbf{y}$, we compensate for this bias as follows:

$$\mathbf{h}^{\text{MB}_\text{L}} = \underset{\mathbf{h}\in\mathcal{H}_{\text{cand}}}{\arg\max} \sum_{\mathbf{y}\in\mathcal{R}} u(\mathbf{h},\mathbf{y}) \cdot e^{\ell(\mathbf{y})}P(\mathbf{y}). \tag{9}$$

In this way, we expect to mitigate the assumed length bias as:

$$e^{l(y)}P(y) \sim e^{l(y)}e^{-l(y)}P_{\text{human}}(y) = P_{\text{human}}(y). \tag{10}$$

We denote a variant of MBMBR computing $\mathbf{h}^{\text{MB}_\text{L}}$ as length-normalized MBMBR (MBMBR_L).

4. Properties of the Model-Based Estimate

In this section, we first introduce the model-based estimator $\hat{P}_{\text{MB}}$ for the sake of analysis and give an example to illustrate its difference from the Monte Carlo estimator (Table 2). We then evaluate its divergence from the true model probability analytically and empirically.

4.1. Probability Distribution of Model-Based Estimate

MBMBR can be understood as a standard MBR, but using the following probability distribution instead of the empirical distribution:

$$\hat{P}_{\text{MB}}(\mathbf{y}) = \begin{cases} \alpha P(\mathbf{y}) & \text{if } \mathbf{y}\in\mathcal{R} \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

Sampled Texts		Target	Monte Carlo Estimate	Model-Based Estimate
Text	#Occurrences	P	$\hat{P}$	$\hat{P}_{\mathrm{MB}}$
But telling the truth is not a crime.	2	0.3	0.4	0.6
However, telling the truth is not a crime.	2	0.1	0.4	0.2
But to tell the truth is not a crime.	1	0.1	0.2	0.2
(All others)	0	0.5	0	0
$D_{\mathrm{KL}}(\cdot\|\|P)$		0	0.808	**0.693**

Table 2. Illustrative example of the model-based estimate. While the Monte Carlo estimate $\hat{P}$ uniformly assigns the same weight to the sampled text without considering the P, the model-based estimate takes advantage of the fact that the P is given in the text generation task and uses P to adjust the distribution accordingly.

where $\alpha = \frac{1}{\sum_{\mathbf{y}' \in \mathcal{R}} P(\mathbf{y}')}$ is a normalization factor so that it sums to 1: $\sum_{\mathbf{y} \in \mathcal{Y}} \hat{P}_{\mathrm{MB}}(\mathbf{y}) = 1$. Note that $\hat{P}_{\mathrm{MB}}$ is introduced solely for analysis and computing it is not necessary to run MBMBR. One can follow Eq. (8) to implement MBMBR. $\hat{P}_{\mathrm{MB}}$ is a probability distribution with support restricted to $\mathcal{R}$ and its probability is proportional to P. We denote $\hat{P}_{\mathrm{MB}}(\mathbf{y})$ as a **model-based distribution**.

Using $\hat{P}_{\mathrm{MB}}$, we can rewrite the form of the MBMBR as follows:

$$\begin{aligned}\mathbf{h}^{\mathrm{MB}} &= \arg\max_{\mathbf{h} \in \mathcal{H}_{\mathrm{cand}}} \sum_{\mathbf{y} \in \mathcal{R}} u(\mathbf{h}, \mathbf{y}) \cdot P(\mathbf{y}) \\ &= \arg\max_{\mathbf{h} \in \mathcal{H}_{\mathrm{cand}}} \sum_{\mathbf{y} \in \mathcal{Y}} u(\mathbf{h}, \mathbf{y}) \cdot \hat{P}_{\mathrm{MB}}(\mathbf{y}). \qquad (12)\end{aligned}$$

Compared to $\mathbf{h}^{\mathrm{model}}$ (Eq. 4), the difference is that the probability distribution P is replaced by the model-based distribution $\hat{P}_{\mathrm{MB}}$. Therefore, as $\hat{P}_{\mathrm{MB}}$ gets closer to P, we expect $\mathbf{h}^{\mathrm{MB}}$ to get closer to $\mathbf{h}^{\mathrm{model}}$, which is the gold standard for the MBR objective (See Section 4.3).

Example. Table 2 describes an example to illustrate the property of model-based distribution. Suppose we sample five sentences $\mathcal{H}_{\mathrm{ref}} = (\mathbf{y}_0, \mathbf{y}_0, \mathbf{y}_1, \mathbf{y}_1, \mathbf{y}_2)$ and their probabilities according to the model P are $\mathbf{y}_0 = 0.3, \mathbf{y}_1 = 0.1$, and $\mathbf{y}_2 = 0.1$, respectively, and thus $\sum_{\mathbf{y} \in \mathcal{Y} \setminus \mathcal{H}_{\mathrm{ref}}} P(\mathbf{y}) = 0.5$. The empirical distribution takes into account of the number of occurrences to compute the Monte Carlo sum to get $\hat{P}(\mathbf{y}_0) = 0.4, \hat{P}(\mathbf{y}_1) = 0.4, \hat{P}(\mathbf{y}_2) = 0.2$. The model-based distribution ignores the number of occurrences and uses P to weight the sample, thus $\hat{P}_{\mathrm{MB}}(\mathbf{y}_0) = 0.6, \hat{P}_{\mathrm{MB}}(\mathbf{y}_1) = 0.2, \hat{P}_{\mathrm{MB}}(\mathbf{y}_2) = 0.2$. The Kullback–Leibler (KL) divergence (Cover & Thomas, 2006) of the model-based distribution from the model probability is $D_{\mathrm{KL}}(\hat{P}_{\mathrm{MB}}||P) = 0.693$ while that of the empirical distribution is $D_{\mathrm{KL}}(\hat{P}||P) = 0.808$. This shows that the model-based distribution is a better estimate of the true model probability than the Monte Carlo estimate.

4.2. Divergence of the Model-Based Estimate to the True Distribution

Analytical Results. The model-based estimate is guaranteed to be better than the Monte Carlo estimate with respect to the KL divergence. In fact, it is optimal over a collection of probability distributions with support restricted to $\mathcal{H}_{\mathrm{ref}}$. Let $\Delta(\mathcal{Y})$ be a collection of probability distributions over $\mathcal{Y}$, and $\Delta(\mathcal{Y}; \mathcal{H}_{\mathrm{ref}})$ be a subset of $\Delta(\mathcal{Y})$, a collection of probability distributions over $\mathcal{Y}$ with their support restricted to $\mathcal{H}_{\mathrm{ref}}$:

$$\begin{aligned}&\Delta(\mathcal{Y}; \mathcal{H}_{\mathrm{ref}}) = \\ &\qquad \{p \in \Delta(\mathcal{Y}) \mid \forall \mathbf{y} \in \mathcal{Y} \setminus \mathcal{H}_{\mathrm{ref}},\ p(\mathbf{y}) = 0\}. \qquad (13)\end{aligned}$$

Theorem 4.1. *The model-based distribution minimizes the Kullbuck-Leiber divergence over a collection of probability distributions with their support restricted to $\mathcal{H}_{\mathrm{ref}}$.*

$$\hat{P}_{\mathrm{MB}} = \arg\min_{p \in \Delta(\mathcal{Y}; \mathcal{H}_{\mathrm{ref}})} \{D_{\mathrm{KL}}(p||P)\}. \qquad (14)$$

The proof is in Appendix A. The model-based estimate is thus the information projection (Cover & Thomas, 2006) of P onto $\Delta(\mathcal{Y}; \mathcal{H}_{\mathrm{ref}})$.

Corollary 4.2. *The KL divergence of the model-based estimate from the true model probability is less than or equal to that of the Monte Carlo estimate:*

$$D_{\mathrm{KL}}(\hat{P}_{\mathrm{MB}}||P) \leq D_{\mathrm{KL}}(\hat{P}||P). \qquad (15)$$

The proof is immediate from the Theorem 4.1. The equality holds if and only if $p(\mathbf{y}) = cP(\mathbf{y})$ for some constant c and for $y \in \mathcal{H}_{\mathrm{ref}}$, which is an extremely unlikely condition where the ratio of the number of occurrences of each hypothesis $\mathbf{y}$ in $\mathcal{H}_{\mathrm{ref}}$ exactly matches the ratio of $P(\mathbf{y})$.

Simulation Study of the Divergence of the Model-Based Distribution. We evaluate the accuracy of model-based

distribution in a controlled experiment where the true distribution is known exactly. We use a Zipf distribution as the true distribution in this experiment (Piantadosi, 2014):

$$P(y) = \frac{y^{-a}}{\zeta(a)}. \tag{16}$$

We set $a = 2$, and limit the domain to $y \in \{0, 1, ..., 499\}$. We run this experiment 100 times and compute the mean and the standard deviation of the KL divergence. The mean ($\pm$ stddev) of the KL divergence of the Monte Carlo estimate is 0.183 (± 0.054), and that of the model-based estimate is 0.069 (± 0.016). The result shows that the model-based estimate is closer to the true probability distribution than the Monte Carlo estimate, as supported by Corollary 4.2.

Empirical Validation of Model-Based Estimates. To empirically evaluate the model-based distribution in text generation tasks, we compute the KL divergence and the Jensen-Shannon divergence on WMT'19 De-En (Barrault et al., 2019), IWSLT'17 Fr-En (Cettolo et al., 2017), XSum (Lewis et al., 2020), and MS COCO (Lin et al., 2014) datasets. We use the first 1000 inputs for each dataset. We sample 256 sentences for WMT'19 De-En and 64 for the other datasets using epsilon sampling ($\epsilon = 0.02$). Figure 1a shows the KL of the model-based and empirical distributions from P, averaged over the source sentences of WMT'19 De-En. See Appendix C for the rest of the datasets and for the Jensen-Shannon divergence. We observe that both the KL and JS divergences of the model-based distribution are significantly smaller than that of the empirical distribution in all four datasets. Approximately twice as many samples are required for the Monte Carlo estimate to achieve the same divergence as the model-based estimate. The empirical result shows that the model-based estimate is significantly more accurate than the Monte Carlo estimate in terms of KL divergence in a wide range of tasks.

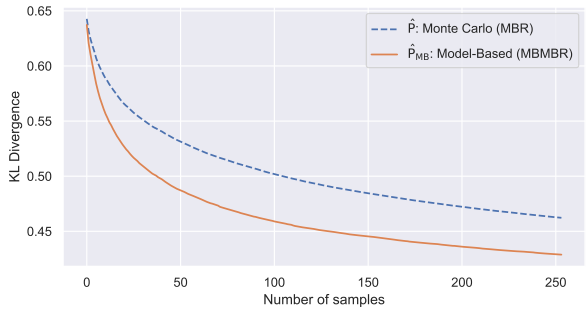
4.3. Relationship of KL Divergence on Output Quality

Analytical Results. We observe that the model-based estimate has smaller KL divergence than the Monte Carlo estimate in Section 4.2. The question is how it affects the output of the MBR decoding. We first show an upper bound of the estimation error of the MBR objective using KL divergence.

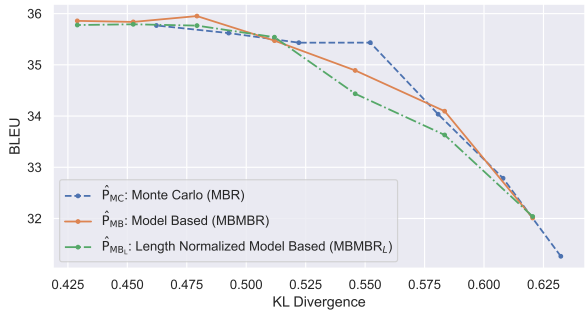
Lemma 4.3. *The estimation error of the MBR objective using a probability distribution p as an estimate is upper bounded by the KL-divergence:*

$$|\mathop{\mathbb{E}}_{\mathbf{y}\sim P}[u(\mathbf{h}, \mathbf{y})] - \mathop{\mathbb{E}}_{\mathbf{y}\sim p}[u(\mathbf{h}, \mathbf{y})]| \leq u_{\max}\sqrt{2D_{\mathrm{KL}}(p||P)}, \tag{17}$$

where $u_{\max}$ is an upper bound of $|u|$ ($u_{\max} := \max_{\mathbf{y},\mathbf{y}'\in\mathcal{Y}} |u(\mathbf{y}, \mathbf{y}')|$).



(a) KL Divergence and the number of samples $|\mathcal{H}_{\mathrm{ref}}|$



(b) BLEU score as a function of KL divergence

Figure 1. (a) Kullback-Leibler divergence of the empirical distribution and the model-based distribution from the true model probability P_{model}, averaged over the source sentences. (b) The correlation of the average KL divergence to the average BLEU score of the output. Evaluated on WMT'19 De-En.

The proof is in Appendix B. The lemma shows that a lower KL divergence corresponds to a reduced upper bound on the estimation error in the MBR objective. Therefore, by minimizing the KL divergence, we anticipate generating a better hypothesis by MBR decoding.

Empirical Validation of the KL Divergence and the BLEU Score. Although Lemma 4.3 shows that KL divergence upper bounds the estimation error of the MBR objective, its practical implication is unclear. To this end, we evaluate the correlation of the average BLEU scores and the average KL divergence of the model-based and Monte Carlo estimates using the WMT'19 De-En dataset. The experimental setup is the same as in Section 4.2. We compute the BLEU score for $|\mathcal{H}_{\mathrm{ref}}| \in \{4, 8, 16, 32, 64, 128, 256\}$. The relationship of the KL divergence and the BLEU score is present in Figure 1b. The lower the KL divergence is the BLEU score tends to be higher in both algorithms. This shows that MBR using the model-based estimates achieve higher output quality (BLEU score) by reducing the KL divergence from the true model probability.

	WMT'19 De-En							WMT'19 Ru-En						
	Epsilon Sampling ($\epsilon = 0.02$)							Epsilon Sampling ($\epsilon = 0.02$)						
$\lvert\mathcal{H}\rvert$	4	8	16	32	64	128	256	4	8	16	32	64	128	256
MBR	31.26	32.79	34.04	35.43	35.43	35.62	35.77	28.00	30.03	31.14	31.95	31.94	32.68	32.61
MBMBR	**32.01**	**34.09**	**34.89**	35.47	**35.95**	35.84	**35.86**	**29.55**	**30.77**	**31.69**	32.25	32.21	32.58	**32.90**
MBMBR_L	31.96	33.53	34.29	**35.77**	35.61	**35.85**	35.84	28.99	30.50	31.40	**32.27**	**32.27**	**33.01**	32.86
	Top-k Sampling ($k = 10$)							Top-k Sampling ($k = 10$)						
$\lvert\mathcal{H}\rvert$	4	8	16	32	64	128	256	4	8	16	32	64	128	256
MBR	29.73	31.43	32.84	33.57	34.26	34.55	34.82	27.09	29.33	30.23	30.95	31.46	31.69	31.80
MBMBR	**30.87**	**32.34**	**33.79**	**35.09**	**35.34**	**35.62**	**35.49**	27.96	**29.71**	**30.75**	**31.48**	**31.81**	31.86	31.93
MBMBR_L	30.68	31.97	33.35	34.24	34.73	34.78	35.26	**28.11**	29.54	30.39	31.26	31.59	**32.11**	**32.09**
	Nucleus Sampling ($p = 0.9$)							Nucleus Sampling ($p = 0.9$)						
$\lvert\mathcal{H}\rvert$	4	8	16	32	64	128	256	4	8	16	32	64	128	256
MBR	28.71	30.35	31.90	33.46	34.13	34.69	34.80	25.51	28.46	29.25	30.15	30.79	31.43	31.35
MBMBR	**29.63**	**31.49**	**32.43**	**33.88**	**34.14**	**34.89**	**35.49**	26.65	28.47	29.40	**30.54**	**31.24**	**31.68**	**31.71**
MBMBR_L	29.34	31.12	32.35	33.78	34.10	34.88	35.21	**26.67**	**28.61**	**29.83**	30.21	31.04	31.51	31.61
	Ancestral Sampling							Ancestral Sampling						
$\lvert\mathcal{H}\rvert$	4	8	16	32	64	128	256	4	8	16	32	64	128	256
MBR	**24.31**	**25.74**	**27.75**	29.11	**31.01**	31.85	32.77	**21.34**	**22.35**	**24.60**	**25.83**	**26.67**	27.39	27.30
MBMBR	23.84	25.48	27.43	29.04	30.70	**32.05**	32.89	20.46	21.42	24.01	25.07	25.96	26.75	27.18
MBMBR_L	23.78	25.68	27.71	**29.19**	30.81	31.92	**33.05**	20.72	21.88	24.15	25.44	26.59	**27.49**	**27.69**
	#wins MBMBR_L > MBR						24/28	#wins MBMBR_L > MBR						23/28
	#wins MBMBR > MBR						23/28	#wins MBMBR > MBR						20/28
	#wins MBMBR_L > MBMBR						7/28	#wins MBMBR_L > MBMBR						16/28

Table 3. BLEU score of MBMBR and MBMBR_L with different sampling strategies on WMT'19 De-En and Ru-En. We evaluate using the first 1000 inputs of the dataset. The best score with the same sampling algorithm and number of samples is in bold. The second best score is underlined. Among the 56 settings (2 datasets, 4 sampling algorithms, with $\lvert\mathcal{H}\rvert \in \{4, 8, 16, 32, 64, 128, 256\}$), the total #wins MBMBR_L > MBR is 47/56, #wins MBMBR > MBR is 43/56, and #wins MBMBR_L > MBMBR is 23/56.

5. Experiments

We evaluate MBR, MBMBR, and MBMBR_L on machine translation, text summarization, and image captioning tasks. We compare them using various probabilistic sampling algorithms which we describe in Section 5.1. We use BERTScore (Zhang et al., 2020) as the utility function of MBR for all experiments. We use Huggingface's Transformers library to run all experiments (Wolf et al., 2020). For reproducibility, all experiments are performed using publicly available pretrained models and datasets. Due to computational resource constraints, we evaluate the first 1000 inputs of each dataset. For SAMSum dataset (Section 5.3.1), we use the entire test dataset (819 inputs).

5.1. Sampling Algorithms for MBR

The choice of sampling algorithm to collect the hypotheses has been studied extensively, as it has been shown to be critical to the performance of MBR. While most of the classical work on MBR relies on beam search to generate samples, Eikema & Aziz (2020) propose to use unbiased sampling through ancestral sampling (Robert & Casella, 1999). Fernandes et al. (2022) have found that MBR with top-k sampling (Fan et al., 2018) and nucleus sampling (Holtzman et al., 2020) generates much higher quality sequences compared to ancestral sampling. Top-k sampling is a simple modification of ancestral sampling that truncates all tokens except the k most probable tokens. Similar to top-k sampling, nucleus sampling also truncates the lower tail of the probability distribution. Nucleus sampling truncates all tokens except those in the *nucleus*, the smallest possible set of tokens that covers a fraction p of the model probability. Freitag et al. (2023) show that epsilon sampling (Hewitt et al., 2022) further improves these sampling algorithms on machine translation tasks. Epsilon sampling is also a variant of ancestral sampling that truncates tokens whose probability is less than a fixed threshold ϵ. While the empirical distributions of top-k sampling, nucleus sampling, and epsilon sampling are biased and inconsistent estimators of the model probability P, MBR decoding using these sampling algorithms has been empirically shown to produce higher quality text than using ancestral sampling, which is

unbiased and consistent.

5.2. Machine Translation

We evaluate machine translation in two scenarios: (1) decoding a domain-specific machine translation model and (2) decoding an instruction-tuned language model with a prompt to trigger its translation capability.

5.2.1. Machine Translation Model

We evaluate the performance of machine translation models using the WMT'19 dataset (Barrault et al., 2019). WMT'19 dataset examines translation between English and other languages in the news domain. We run experiments on four language pairs: English ↔ German (En↔De) and English ↔ Russian (En↔Ru) using the pretrained models of each language pair provided by fairseq (Ng et al., 2019).

We sample 256 sentences per source sentence for De and Ru→En. We use epsilon sampling (Hewitt et al., 2022; Freitag et al., 2023), top-k sampling (Fan et al., 2018), nucleus sampling (Holtzman et al., 2020), and ancestral sampling. The parameters for the sampling methods are set according to the work of Freitag et al. (2023). For epsilon sampling, $\epsilon = 0.02$. k is set to $k = 10$ for top-k sampling. For nucleus sampling, $p = 0.9$ is set. The temperature is set to 1.0 for all algorithms. We evaluate the BLEU score (Papineni et al., 2002) using the sacreBLEU library (Post, 2018).

Table 3 shows the BLEU score from German and Russian to English. Overall, we observe that MBMBR and MBMBR_L outperform MBR, except for ancestral sampling. As a reference, the BLEU score of a beam search with reranking is 40.8 for De-En and 40.0 for Ru-En (Ng et al., 2019). Table 4 shows the BLEU score of En-De and En-Ru with epsilon sampling. We observe that the proposed method outperforms the baseline in all settings.

We additionally evaluate the performance of MBMBR and MBR using utility functions trained for machine translation which we describe in Appendix D.

5.2.2. Language Model with Prompting

For the experiments on a language model, we use IWSLT 2017 French → English dataset as a benchmark (Cettolo et al., 2017). We use BLOOMZ and mT0 model (7.1B) loaded in 8-bit precision to reduce memory consumption (Muennighoff et al., 2023). 64 sentences are sampled using epsilon sampling with $\epsilon = 0.02$. The prompt is described in Appendix G. The BLEU score of IWSLT'17 Fr-En is reported in Table 5. We observe that MBMBR outperforms MBR, especially when the number of samples is small. We additionally evaluate MBMBR using LLM-based machine translation models which we describe in Appendix E.

	WMT'19 En-De				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	29.03	30.73	32.10	33.18	34.04
MBMBR	**30.00**	**31.45**	**32.32**	33.06	34.00
MBMBR_L	29.94	31.17	32.26	**33.35**	**34.41**
	WMT'19 En-Ru				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	24.68	26.26	27.25	27.80	27.97
MBMBR	**25.52**	**26.72**	27.19	**28.14**	**28.53**
MBMBR_L	25.15	26.60	**27.34**	28.08	28.19

Table 4. BLEU scores of MBMBR and MBR on WMT'19 En-De and En-Ru using machine translation models (Ng et al., 2019). We use epsilon sampling with $\epsilon = 0.02$. BERTScore is used as the utility function of MBR. Due to computational constraints, we evaluate using the first 1000 inputs of the dataset.

	IWSLT'17 Fr-En				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	25.33	26.89	27.94	28.48	29.43
MBMBR	**26.58**	**27.77**	**28.52**	**29.51**	29.25
MBMBR_L	26.26	27.16	28.38	29.15	**29.51**

Table 5. BLEU scores of MBMBR and MBR on IWSLT'17 using BLOOMZ and mT0 model (Muennighoff et al., 2023). We use epsilon sampling with $\epsilon = 0.02$. BERTScore is used as the utility function of MBR. Due to computational constraints, we evaluate using the first 1000 inputs of the dataset.

	XSum				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	32.01	33.96	34.80	35.41	35.90
MBMBR	31.69	33.09	34.00	34.72	35.51
MBMBR_L	**32.65**	**34.01**	**35.00**	**35.63**	**36.18**
	SAMSum				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	30.55	31.16	31.98	32.51	33.03
MBMBR	**31.68**	**32.44**	**33.14**	**33.86**	**34.50**
MBMBR_L	31.17	32.06	32.62	32.99	33.14

Table 6. ROUGE-L scores of MBMBR and MBR on XSum and SAMSum using text summarization models (Lewis et al., 2020). We use epsilon sampling with $\epsilon = 0.02$. BERTScore is used as the utility function of MBR. Due to computational constraints, we evaluate using the first 1000 inputs of the dataset.

	CNN/DM				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	16.72	16.62	16.94	17.55	17.26
MBMBR	**17.37**	**17.16**	**17.63**	**18.07**	**17.73**
MBMBR_L	16.75	16.75	16.98	17.63	17.38

Table 7. ROUGE-L scores of MBMBR and MBR on CNN/DM using Mistral-7B-Instruct-v0.1 (Jiang et al., 2023). We use epsilon sampling with $\epsilon = 0.02$. BERTScore is used as the utility function of MBR. Due to computational constraints, we evaluate using the first 1000 inputs of the dataset.

5.3. Text Summarization

We evaluate text summarization using (1) a BART model specifically fine-tuned for each text summarization dataset and (2) a general-purpose LLM with prompting.

5.3.1. Text Summarization Model

We use XSum (Narayan et al., 2018) and SAMSum (Gliwa et al., 2019) datasets to evaluate the performance of MBMBR on the text summarization task. XSum dataset is a benchmark for an abstractive single-document summarization. The documents are collected from BBC articles. We use a BART model pretrained on XSum dataset (Lewis et al., 2020). SAMSum corpus is a collection of messenger-like conversations with summaries. We use a BART model pretrained on SAMSum dataset. We evaluate ROUGE-L (Lin, 2004) using HuggingFace's evaluate library.

Although the performance of sampling algorithms has been extensively compared for machine translation tasks (Freitag et al., 2023), little has been evaluated for other text generation tasks. As a preliminary experiment, we evaluate the performance of MBR using epsilon sampling ($\epsilon = 0.02$), top-k sampling ($k = 10$), nucleus sampling ($p = 0.9$), and ancestral sampling on text summarization tasks (XSum and SAMSum) and image captioning tasks (MS COCO and nocaps; see Section 5.4). The results show that epsilon sampling outperforms others on all datasets (Appendix F). Based on this observation, we will continue the evaluation of MBMBR and MBR using epsilon sampling with $\epsilon = 0.02$.

The result using epsilon sampling is shown in Table 6. Overall, MBMBR_L outperforms MBR in both datasets. However, we observe that MBMBR drops in the ROUGE-L score in XSum dataset. To investigate the effect of length normalization, we compute the relative length of the selected sentence compared to the reference text ($\ell(\mathbf{y})/\ell_{\text{ref}}$) by the three methods. The average (standard deviation) of the relative length over the dataset of MBR, MBMBR, and MBMBR_L are 0.870 (± 0.268), 0.777 (± 0.234), and 0.851 (± 0.270), respectively. MBMBR has decreased the relative length by about 10% compared to MBR, which we speculate is the reason for the decrease in the ROUGE-L score. MBMBR_L also becomes shorter but only by about 2%. For the SAMSum dataset, the decrease in relative length of MBMBR is about 2% which may be the reason why MBMBR successfully achieves a higher ROUGE-L score.

	MSCOCO				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	27.12	28.46	30.42	31.51	33.24
MBMBR	**28.75**	**30.56**	**33.12**	**34.25**	**34.96**
MBMBR_L	28.65	29.29	31.14	32.54	33.99
	nocaps				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	25.43	27.67	28.73	30.93	31.42
MBMBR	23.31	24.73	25.87	26.91	27.27
MBMBR_L	**28.86**	**29.78**	**30.47**	**32.10**	**32.68**

Table 8. BLEU scores of MBMBR and MBR on MSCOCO and nocaps using BLIP-2 (Li et al., 2023) with Flan T5-xl (Chung et al., 2022). We use epsilon sampling with $\epsilon = 0.02$. BERTScore is used as the utility function of MBR. Due to computational constraints, we evaluate using the first 1000 inputs of the dataset.

	E2E NLG				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	**20.49**	20.77	20.42	**21.04**	**22.30**
MBMBR	17.79	17.01	15.80	16.26	16.07
MBMBR_L	19.93	**21.28**	**20.88**	20.99	22.03

Table 9. BLEU scores of MBMBR and MBR on E2E NLG using Mistral-7B-Instruct-v0.1 (Jiang et al., 2023). We use epsilon sampling with $\epsilon = 0.02$. BERTScore is used as the utility function of MBR. Due to computational constraints, we evaluate using the first 1000 inputs of the dataset.

5.3.2. Language Model with Prompting

CNN/DM is a collection of news articles and their summaries by journalists from CNN and the Daily Mail (Hermann et al., 2015). We run experiments on CNN/DM with Mistral-7B-Instruct-v0.1 (Jiang et al., 2023) loaded with 4-bit precision. See Appendix G for the prompt we use for CNN/DM. The result is shown in Table 7. We see that MBMBR has a higher ROUGE-L score compared to MBR.

5.4. Image Captioning with BLIP-2

We evaluate our method for image captioning tasks. Although the input for the image captioning task is an image rather than a text, MBR is applicable as is. That is, we generate a set of hypotheses conditional on the input image and

then select the best hypothesis using the MBR objective.

We use BLIP-2 (Li et al., 2023) with Flan T5-xl (Chung et al., 2022) loaded in 8-bit to evaluate on two datasets: MS COCO dataset (Lin et al., 2014) and nocaps dataset (Agrawal et al., 2019). We use a fine-tuned model for MS COCO and a base model for nocaps. Overall, MBMBR_L outperforms MBR in both datasets (Table 8).

5.5. Data-to-Text with Few-Shot Learning

We evaluate the proposed method on a data-to-text generation task using E2E NLG dataset (Novikova et al., 2017). E2E NLG is a data-driven natural language generation task in the restaurant domain. Given a set of key-value pairs about a restaurant, the task is to provide a short English description of the restaurant in a few sentences. We use a Mistral-7B-Instruct-v0.1 (Jiang et al., 2023) loaded with 4-bit with a prompt provided by Suzgun et al. (2023). The result is shown in Table 9. Overall, we observe no improvement over the baseline in E2E NLG dataset. We speculate that this is because the quality of the sentences generated is not high enough for the MBR to work properly.

6. Related Work

MBR decoding has been studied in many NLP tasks including parsing (Goodman, 1996), speech recognition (Goel & Byrne, 2000), bilingual word alignment (Kumar & Byrne, 2002), and machine translation (Kumar & Byrne, 2004). MBR decoding has recently gained attention in machine translation to overcome some of the biases of MAP decoding in neural machine translation (Eikema & Aziz, 2020; Müller & Sennrich, 2021; Eikema & Aziz, 2022).

Freitag et al. (2022) and Fernandes et al. (2022) show that using neural-based utility functions such as BLEURT (Sellam et al., 2020; Pu et al., 2021) and COMET (Rei et al., 2020; 2022) instead of lexical overlap metrics (e.g., BLEU) further improves the output quality of the MBR decoding. The improvement in utility function is orthogonal to our approach as our improvement is on the estimation of the distribution.

The main drawback of MBR decoding is that it is computationally intensive. Cheng & Vlachos (2023) shows that the number of calls to the utility functions can be significantly reduced by iteratively reducing the number of candidates. Finkelstein & Freitag (2024) and Yang et al. (2023) show that self-training a machine translation model using its own MBR-decoded output can improve the performance of more efficient decoding methods such as beam search.

7. Conclusion

We propose model-based MBR (MBMBR) decoding, a variant of MBR that uses a model-based distribution as an estimator of the model probability. We evaluate the model-based distribution analytically and empirically to show that it is closer to the true model probability than the Monte Carlo estimate with respect to KL divergence. The result suggests that the model-based distribution is likely to be a better estimator for the purpose of MBR decoding. We perform MBMBR decoding on a variety of text generation tasks, including machine translation, text summarization, image captioning, and data-to-text, using both domain-specific sequence-to-sequence models and domain-independent large language models. The empirical results show that MBMBR outperforms MBR in most cases. We believe that MBMBR will be a practical choice for future MBR decoding because of its applicability and significant performance improvements.

Acknowledgments

We thank all the reviewers for their constructive comments throughout the manuscript review cycles. We thank William Byrne for the insightful feedback and for improving the proof of Theorem 4.1. Kaito Ariu's research is supported by JSPS KAKENHI Grant No. 23K19986.

Impact Statement

While language generation can be used for malicious purposes, we do not foresee any specific ethical concerns or societal implications with the analysis in this paper beyond those discussed by Bender et al. (2021).

References

Agrawal, H., Anderson, P., Desai, K., Wang, Y., Chen, X., Jain, R., Johnson, M., Batra, D., Parikh, D., and Lee, S. nocaps: novel object captioning at scale. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South)*, pp. 8947–8956. IEEE, 2019. doi: 10.1109/ICCV.2019.00904. URL https://doi.org/10.1109/ICCV.2019.00904.

Alves, D. M., Pombal, J., Guerreiro, N. M., Martins, P. H., Alves, J., Farajian, A., Peters, B., Rei, R., Fernandes, P., Agrawal, S., Colombo, P., de Souza, J. G. C., and Martins, A. F. T. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*, 2024.

Anderson, P., Fernando, B., Johnson, M., and Gould, S. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Con-*

ference on Empirical Methods in Natural Language Processing, pp. 936–945, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1098. URL https://aclanthology.org/D17-1098.

Barrault, L., Bojar, O., Costa-jussà, M. R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P., Malmasi, S., Monz, C., Müller, M., Pal, S., Post, M., and Zampieri, M. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 1–61, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5301. URL https://aclanthology.org/W19-5301.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pp. 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

Bertsch, A., Xie, A., Neubig, G., and Gormley, M. It's MBR all the way down: Modern generation techniques through the lens of minimum Bayes risk. In Elazar, Y., Ettinger, A., Kassner, N., Ruder, S., and A. Smith, N. (eds.), *Proceedings of the Big Picture Workshop*, pp. 108–122, Singapore, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.bigpicture-1.9.

Cettolo, M., Federico, M., Bentivogli, L., Niehues, J., Stüker, S., Sudoh, K., Yoshino, K., and Federmann, C. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pp. 2–14, Tokyo, Japan, December 14-15 2017. International Workshop on Spoken Language Translation. URL https://aclanthology.org/2017.iwslt-1.1.

Cheng, J. and Vlachos, A. Faster minimum Bayes risk decoding with confidence-based pruning. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12473–12480, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.emnlp-main.767.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Cohen, E. and Beck, C. Empirical analysis of beam search performance degradation in neural sequence models. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1290–1299. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/cohen19a.html.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

Csiszár, I. and Körner, J. *Information Theory - Coding Theorems for Discrete Memoryless Systems, Second Edition*. Cambridge University Press, 2011. ISBN 978-0-51192188-9. doi: 10.1017/CBO9780511921889. URL https://doi.org/10.1017/CBO9780511921889.

Eikema, B. and Aziz, W. 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*, pp. 4506–4520, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.398. URL https://aclanthology.org/2020.coling-main.398.

Eikema, B. and Aziz, W. Sampling-based approximations to minimum Bayes risk decoding for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 10978–10993, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.754. URL https://aclanthology.org/2022.emnlp-main.754.

Fan, A., Lewis, M., and Dauphin, Y. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL https://aclanthology.org/P18-1082.

Farinhas, A., de Souza, J., and Martins, A. An empirical study of translation hypothesis ensembling with large language models. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 11956–11970, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.733. URL https://aclanthology.org/2023.emnlp-main.733.

Fernandes, P., Farinhas, A., Rei, R., C. de Souza, J. G., Ogayo, P., Neubig, G., and Martins, A. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1396–1412, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.100. URL https://aclanthology.org/2022.naacl-main.100.

Finkelstein, M. and Freitag, M. MBR and QE finetuning: Training-time distillation of the best and most expensive decoding methods. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=bkNx3O0sND.

Freitag, M., Grangier, D., Tan, Q., and Liang, B. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825, 2022. doi: 10.1162/tacl_a_00491. URL https://aclanthology.org/2022.tacl-1.47.

Freitag, M., Ghorbani, B., and Fernandes, P. Epsilon sampling rocks: Investigating sampling strategies for minimum Bayes risk decoding for machine translation. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9198–9209, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.findings-emnlp.617.

Gliwa, B., Mochol, I., Biesek, M., and Wawer, A. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5409. URL https://aclanthology.org/D19-5409.

Goel, V. and Byrne, W. J. Minimum bayes-risk automatic speech recognition. *Computer Speech & Language*, 14 (2):115–135, 2000.

Goodman, J. Parsing algorithms and metrics. In *34th Annual Meeting of the Association for Computational Linguistics*, pp. 177–183, Santa Cruz, California, USA, June 1996. Association for Computational Linguistics. doi: 10.3115/981863.981887. URL https://aclanthology.org/P96-1024.

Graves, A. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

Hermann, K. M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. Teaching machines to read and comprehend. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 1693–1701, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/afdec7005cc9f14302cd0474fd0f3c96-Abstract.html.

Hewitt, J., Manning, C., and Liang, P. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3414–3427, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.249. URL https://aclanthology.org/2022.findings-emnlp.249.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b. *arXiv preprint arxiv:2310.06825*, 2023.

Koehn, P. and Knowles, R. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pp. 28–39, Vancouver, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3204. URL https://aclanthology.org/W17-3204.

Kumar, S. and Byrne, W. Minimum Bayes-risk word alignments of bilingual texts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pp. 140–147. Association for Computational Linguistics, July 2002. doi: 10.3115/1118693.1118712. URL https://aclanthology.org/W02-1019.

Kumar, S. and Byrne, W. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pp. 169–176, Boston, Massachusetts, USA, May 2 - May 7 2004. Association

for Computational Linguistics. URL https://aclanthology.org/N04-1022.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Li, J., Li, D., Savarese, S., and Hoi, S. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 19730–19742. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/li23q.html.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Meister, C., Cotterell, R., and Vieira, T. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2173–2185, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.170. URL https://aclanthology.org/2020.emnlp-main.170.

Muennighoff, N., Wang, T., Sutawika, L., Roberts, A., Biderman, S., Le Scao, T., Bari, M. S., Shen, S., Yong, Z. X., Schoelkopf, H., Tang, X., Radev, D., Aji, A. F., Almubarak, K., Albanie, S., Alyafeai, Z., Webson, A., Raff, E., and Raffel, C. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15991–16111, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.891. URL https://aclanthology.org/2023.acl-long.891.

Müller, M. and Sennrich, R. Understanding the properties of minimum Bayes risk decoding in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 259–272, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.22. URL https://aclanthology.org/2021.acl-long.22.

Murray, K. and Chiang, D. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 212–223, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6322. URL https://aclanthology.org/W18-6322.

Narayan, S., Cohen, S. B., and Lapata, M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206.

Ng, N., Yee, K., Baevski, A., Ott, M., Auli, M., and Edunov, S. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 314–319, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5333. URL https://aclanthology.org/W19-5333.

Novikova, J., Dušek, O., and Rieser, V. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 201–206, Saarbrücken, Germany, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5525. URL https://aclanthology.org/W17-5525.

Ohashi, A., Honda, U., Morimura, T., and Jinnai, Y. On the true distribution approximation of minimum bayes-risk decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Mexico City, Mexico, July 2024. Association for Computational Linguistics.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceed-*

ings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pp. 48–53, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL https://aclanthology.org/N19-4009.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Piantadosi, S. T. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21:1112–1130, 2014.

Post, M. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319.

Pu, A., Chung, H. W., Parikh, A., Gehrmann, S., and Sellam, T. Learning compact metrics for MT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 751–762, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.58. URL https://aclanthology.org/2021.emnlp-main.58.

Rei, R., Stewart, C., Farinha, A. C., and Lavie, A. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pp. 911–920, Online, November 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.wmt-1.101.

Rei, R., C. de Souza, J. G., Alves, D., Zerva, C., Farinha, A. C., Glushkova, T., Lavie, A., Coheur, L., and Martins, A. F. T. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pp. 578–585, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.wmt-1.52.

Robert, C. P. and Casella, G. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

Rush, A. M., Chopra, S., and Weston, J. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1044. URL https://aclanthology.org/D15-1044.

Sellam, T., Das, D., and Parikh, A. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7881–7892, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.704. URL https://aclanthology.org/2020.acl-main.704.

Stahlberg, F. and Byrne, B. On NMT search errors and model errors: Cat got your tongue? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3356–3362, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1331. URL https://aclanthology.org/D19-1331.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

Suzgun, M., Melas-Kyriazi, L., and Jurafsky, D. Follow the wisdom of the crowd: Effective text generation via minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4265–4293, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.262. URL https://aclanthology.org/2023.findings-acl.262.

Welleck, S., Kulikov, I., Kim, J., Pang, R. Y., and Cho, K. Consistency of a recurrent language model with respect to incomplete decoding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5553–5568, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.448. URL https://aclanthology.org/2020.emnlp-main.448.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Łukasz Kaiser, Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016.

Xu, H., Kim, Y. J., Sharaf, A., and Awadalla, H. H. A paradigm shift in machine translation: Boosting translation performance of large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=farT6XXntP.

Yang, G., Chen, J., Lin, W., and Byrne, B. Direct preference optimization for neural machine translation with minimum bayes risk decoding. *arXiv preprint arXiv:2311.08380*, 2023.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. BERTScore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

A. Proof of Theorem 4.1

From the convention $0 \log 0 = 0$ and the log sum inequality (LSI) (Theorem 2.7.1 in Cover & Thomas (2006)), we have for any $p \in \Delta(\mathcal{Y}; \mathcal{H}_{\text{ref}})$:

$$\begin{aligned} &D_{\text{KL}}(p||P) \\ &\overset{0 \log 0 = 0}{=} \sum_{\mathbf{y} \in \mathcal{H}_{\text{ref}}} p(\mathbf{y}) \log \frac{p(\mathbf{y})}{P(\mathbf{y})} \\ &\overset{LSI}{\geq} (\sum_{\mathbf{y} \in \mathcal{H}_{\text{ref}}} p(\mathbf{y})) \log \frac{\sum_{\mathbf{y} \in \mathcal{H}_{\text{ref}}} p(\mathbf{y})}{\sum_{\mathbf{y} \in \mathcal{H}_{\text{ref}}} P(\mathbf{y})} \\ &= -\log \sum_{\mathbf{y} \in \mathcal{H}_{\text{ref}}} P(\mathbf{y}) \\ &= \log \alpha, \end{aligned} \tag{18}$$

where the third equality follows from the fact that $\sum_{\mathbf{y} \in \mathcal{H}_{\text{ref}}} p(\mathbf{y}) = 1$, and the last equality follows from the definition of $\alpha = \frac{1}{\sum_{\mathbf{y} \in \mathcal{H}_{\text{ref}}} P(\mathbf{y})}$.

On the other hand, from (11), we get:

$$\begin{aligned} D_{\text{KL}}(\hat{P}_{\text{MB}}||P) &= \sum_{\mathbf{y} \in \mathcal{H}_{\text{ref}}} \hat{P}_{\text{MB}}(\mathbf{y}) \log \frac{p(\mathbf{y})}{P(\mathbf{y})} \\ &= \sum_{\mathbf{y} \in \mathcal{H}_{\text{ref}}} \alpha P(\mathbf{y}) \log \alpha \\ &= (\alpha \log \alpha) \frac{1}{\alpha} = \log \alpha, \end{aligned}$$

which matches the lower bound in (18). Therefore, we have $D_{\text{KL}}(p||P) \geq D_{\text{KL}}(\hat{P}_{\text{MB}}||P)$ for any $p \in \Delta(\mathcal{Y}; \mathcal{H}_{\text{ref}})$.

B. Proof of Lemma 4.3

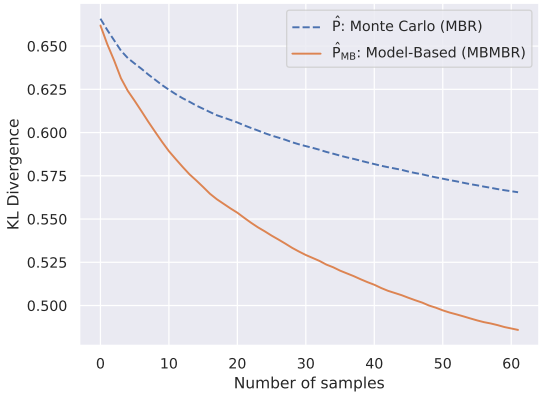
Lemma 4.3 is derived by Pinsker's inequality (Csiszár & Körner, 2011):

$$\begin{aligned} &|\mathop{\mathbb{E}}_{\mathbf{y} \sim P}[u(\mathbf{h}, \mathbf{y})] - \mathop{\mathbb{E}}_{\mathbf{y} \sim p}[u(\mathbf{h}, \mathbf{y})]| \\ &= |\sum_{\mathbf{y} \in \mathcal{Y}} u(\mathbf{h}, \mathbf{y})(P(\mathbf{y}) - p(\mathbf{y}))| \\ &\leq u_{\max} \sum_{\mathbf{y} \in \mathcal{Y}} |P(\mathbf{y}) - p(\mathbf{y})| \\ &\leq u_{\max} \sqrt{2 D_{\text{KL}}(p||P)}. \end{aligned}$$

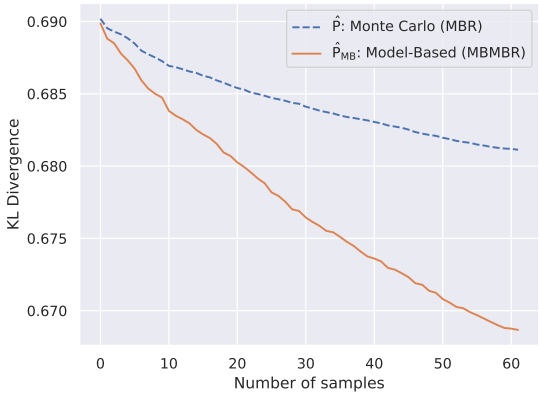
C. Empirical Evaluation of the Divergence

The result of the KL divergence on IWSLT'17 Fr-En, XSum, and MS COCO is shown in Figure 2. See Section 4 for the experimental setup. The model-based estimate significantly reduces the KL divergence compared to the Monte Carlo estimate in all domains.

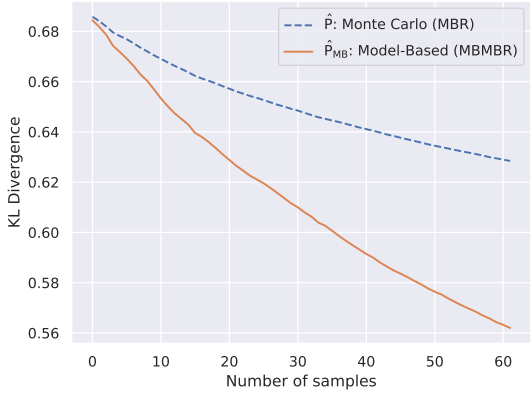
As an additional measure of the divergence, we compute the Jensen-Shannon Divergence (JSD). Note that the JSD of



(a) IWSLT'17 Fr-En



(b) XSum



(c) MS COCO

Figure 2. Kullback-Leibler Divergence of the Monte Carlo estimate and the model-based estimate to the P_{model}, averaged over the source sentences.

a distribution p and P is computable without enumerating all the hypotheses in $\mathcal{Y}$ if the support of p is restricted to $\mathcal{H}_{\text{ref}}$. Let $M = \frac{1}{2}(p + P)$ be a mixture distribution of p and P. Using the convention $0 \ln \frac{0}{q(x)} = 0$ (Cover & Thomas, 2006):

$$
\begin{aligned}
&JSD(p||P) \\
=&\frac{1}{2}(D_{\text{KL}}(p||M) + D_{\text{KL}}(P||M)) \\
=&\frac{1}{2}(\sum_{\mathbf{y}\in\mathcal{Y}} p(\mathbf{y}) \ln(\frac{p(\mathbf{y})}{M(\mathbf{y})}) \\
&+ \sum_{\mathbf{y}\in\mathcal{Y}} P(\mathbf{y}) \ln(\frac{P(\mathbf{y})}{M(\mathbf{y})})) \\
=&\frac{1}{2}(\sum_{\mathbf{y}\in\mathcal{H}_{\text{ref}}} p(\mathbf{y}) \ln(\frac{p(\mathbf{y})}{M(\mathbf{y})}) \\
&+ \sum_{\mathbf{y}\in\mathcal{Y}\setminus\mathcal{H}_{\text{ref}}} p(\mathbf{y}) \ln(\frac{p(\mathbf{y})}{M(\mathbf{y})}) \\
&+ \sum_{\mathbf{y}\in\mathcal{H}_{\text{ref}}} P(\mathbf{y}) \ln(\frac{P(\mathbf{y})}{M(\mathbf{y})})) \\
&+ \sum_{\mathbf{y}\in\mathcal{Y}\setminus\mathcal{H}_{\text{ref}}} P(\mathbf{y}) \ln(\frac{P(\mathbf{y})}{M(\mathbf{y})})) \\
=&\frac{1}{2}(\sum_{\mathbf{y}\in\mathcal{H}_{\text{ref}}} p(\mathbf{y}) \ln(\frac{p(\mathbf{y})}{M(\mathbf{y})}) \\
&+ 0 \\
&+ \sum_{\mathbf{y}\in\mathcal{H}_{\text{ref}}} P(\mathbf{y}) \ln(\frac{P(\mathbf{y})}{M(\mathbf{y})})) \\
&+ (1 - \sum_{\mathbf{y}\in\mathcal{H}_{\text{ref}}} P(\mathbf{y})) \ln(2)).
\end{aligned}
$$

Therefore, $JSD(p||P)$ is computable without enumerating all the hypotheses in $\mathcal{Y}$. Figure 3 shows the JSD of the model-based and the Monte Carlo estimator. For all datasets, the JSD of the model-based distribution is significantly smaller than that of the empirical distribution. Compared to the Monte Carlo estimate, the model-based estimate requires less than half of the samples to achieve the same divergence.

D. Evaluation using COMET and BLEURT

The state-of-the-art evaluation function for neural machine translation is neural-based models (Freitag et al., 2022; Fernandes et al., 2022). We evaluate using COMET-22 (Rei et al., 2020; 2022) and BLEURT-20 (Sellam et al., 2020; Pu et al., 2021) as a utility function for WMT'19 De-En and Ru-En using 64 samples with epsilon sampling ($\epsilon = 0.02$). Other experimental settings are the same as in Section 5.2.1. The results are in Tables 10 and 11. MBMBR improves upon MBR using COMET-22 but decreases performance in De-En using BLEURT-20.

	De-En	Ru-En
MBR	85.75	85.37
MBMBR	**85.84**	85.44
MBMBR_{L}	85.82	**85.47**

Table 10. Evaluation on WMT'19 using COMET-22 as the utility function and the evaluation metric.

	De-En	Ru-En
MBR	**28.31**	31.14
MBMBR	26.92	29.42
MBMBR_{L}	28.03	**31.27**

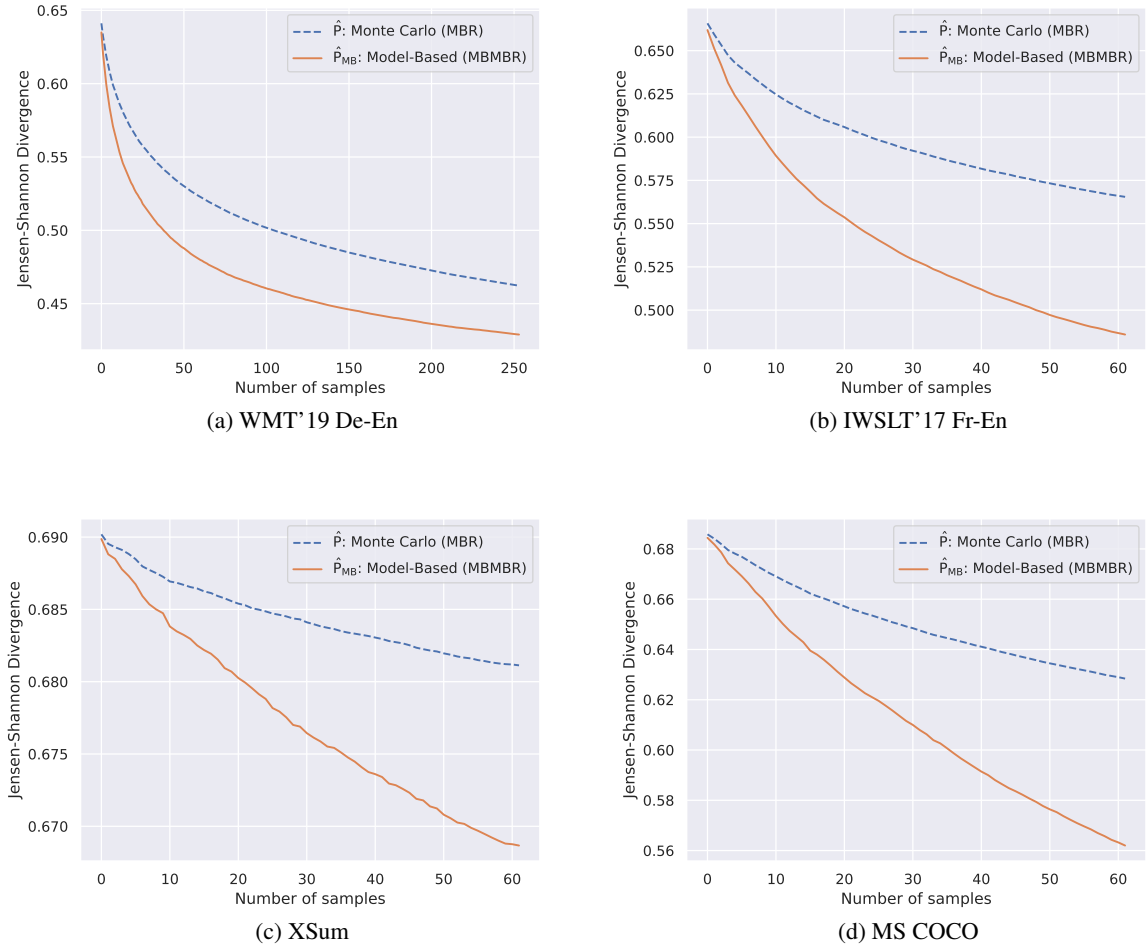
Table 11. Evaluation on WMT'19 using BLEURT-20 as the utility function and the evaluation metric.

E. Evaluation with LLM-based Machine Translation Models

Recent work shows that fine-tuning LLMs for tasks related to machine translation achieves state-of-the-art performance in machine translation. We evaluate the performance of the MBMBR using two state-of-the-art LLM-based translation models, TowerInstruct-7B-v0.1 (Alves et al., 2024) and ALMA-7B-R (Xu et al., 2024). We sample 64 outputs with epsilon sampling ($\epsilon = 0.02$). The prompt for the models is described in Appendix G. Other experimental settings are the same as in Section 5.2.2. Table 12 shows the COMET-22 and BLEU scores of the two models. MBMBR outperforms MBR using TowerInstruct-7B-v0.1 but slightly drops in performance using ALMA-7B-R.

	TowerInstruct-7B-v0.1	
	COMET-22	BLEU
MBR	85.47	36.11
MBMBR	**86.55**	36.10
MBMBR_{L}	**86.55**	**36.35**
	ALMA-7B-R	
	COMET-22	BLEU
MBR	**84.65**	**31.68**
MBMBR	84.52	31.63
MBMBR_{L}	84.39	31.46

Table 12. Evaluation of LLM-based translation models on IWSLT'17 Fr-En. BERTScore is used as a utility function.



(a) WMT'19 De-En

(b) IWSLT'17 Fr-En

(c) XSum

(d) MS COCO

Figure 3. Jensen-Shannon Divergence of the Monte Carlo estimate and the model-based estimate to the P_{model}, averaged over the source sentences.

F. Evaluation of Sampling Algorithms for MBR Decoding

We evaluate the performance of MBR with epsilon sampling ($\epsilon = 0.02$), top-k sampling ($k = 10$), nucleus sampling ($p = 0.9$), and ancestral sampling on text summarization and image captioning. Other experimental settings are the same as in Section 5. The results are shown in Tables 13, 14, 15, and 16. We observe that epsilon sampling outperforms the others on all four datasets, and its results are further improved with MBMBR and MBMBR_{L}.

G. Prompts for Large Language Models

For IWSLT'17 in Section 5.2.2, We use the following prompt as a guide to trigger its translation capability for BLOOMZ and mT0:

> Translate the following sentence from French to English.
> Q: [[QUESTION]]
> A:

We use the following prompt for TowerInstruct-7B-v0.1 and ALMA-7B-R in Appendix E:

> Translate the following text from French into English.
> French: [[QUESTION]]
> English:

For CNN/DM, we use the following prompt:

> Given a BBC article, write a short summary of the article in one sentence.
> Article: [[QUESTION]]

XSum					
	Epsilon Sampling ($\epsilon = 0.02$)				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	32.01	33.96	34.80	35.41	35.90
MBMBR	31.69	33.09	34.00	34.72	35.51
MBMBR_L	32.65	34.01	35.00	35.63	36.18
	Top-k Sampling ($k = 10$)				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	31.89	33.60	34.15	35.03	35.42
MBMBR	31.71	32.83	33.76	34.28	34.90
MBMBR_L	32.28	33.76	34.38	35.21	35.83
	Nucleus Sampling ($p = 0.9$)				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	30.73	32.03	33.15	33.92	34.60
MBMBR	30.49	31.41	32.55	33.25	33.72
MBMBR_L	30.69	31.79	32.96	33.96	34.68
	Ancestral Sampling				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	27.48	28.27	29.41	30.60	31.82
MBMBR	24.72	25.88	26.97	28.63	30.16
MBMBR_L	24.75	25.97	27.15	29.09	31.05

Table 13. ROUGE-L scores on XSum dataset. We evaluate using the first 1000 inputs of the dataset. BERTScore is used as the utility function.

SAMSum					
	Epsilon Sampling ($\epsilon = 0.02$)				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	30.55	31.16	31.98	32.51	33.03
MBMBR	31.68	32.44	33.14	33.86	34.50
MBMBR_L	31.17	32.06	32.62	32.99	33.14
	Top-k Sampling ($k = 10$)				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	28.04	28.21	28.17	28.33	31.81
MBMBR	28.32	28.91	29.84	30.76	31.34
MBMBR_L	28.26	28.90	29.74	30.47	31.47
	Nucleus Sampling ($p = 0.9$)				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	27.76	27.84	28.13	28.24	30.95
MBMBR	27.87	28.21	28.32	28.68	29.17
MBMBR_L	27.87	28.26	28.35	28.69	29.27
	Ancestral Sampling				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	25.94	26.33	26.43	26.60	29.83
MBMBR	25.81	25.86	25.90	26.07	26.24
MBMBR_L	25.81	25.86	25.90	26.07	26.26

Table 14. ROUGE-L scores on SAMSum dataset. BERTScore is used as the utility function.

MS COCO					
	Epsilon Sampling ($\epsilon = 0.02$)				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	27.12	28.46	30.42	31.51	33.24
MBMBR	28.75	30.56	33.12	34.25	34.96
MBMBR_L	28.65	29.29	31.14	32.54	33.99
	Top-k Sampling ($k = 10$)				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	26.13	28.08	30.07	31.89	32.40
MBMBR	21.36	23.80	26.22	28.81	30.93
MBMBR_L	21.36	23.51	26.46	28.76	30.85
	Nucleus Sampling ($p = 0.9$)				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	26.40	29.12	31.12	31.88	32.98
MBMBR	21.68	24.36	26.84	29.08	31.00
MBMBR_L	21.75	24.31	27.07	29.08	30.95
	Ancestral Sampling				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	22.70	25.87	28.08	29.84	31.55
MBMBR	18.63	20.88	22.93	25.26	27.59
MBMBR_L	18.59	20.48	22.65	25.90	29.52

Table 15. BLEU scores on MS COCO dataset. We evaluate using the first 1000 inputs of the dataset. BERTScore is used as the utility function.

nocaps					
	Epsilon Sampling ($\epsilon = 0.02$)				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	25.43	27.67	28.73	30.93	31.42
MBMBR	23.31	24.73	25.87	26.91	27.27
MBMBR_L	28.86	29.78	30.47	32.10	32.68
	Top-k Sampling ($k = 10$)				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	23.69	25.91	27.41	29.33	30.67
MBMBR	21.39	23.01	24.05	25.18	25.66
MBMBR_L	26.64	28.63	29.15	31.53	31.58
	Nucleus Sampling ($p = 0.9$)				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	21.58	25.76	27.20	28.55	29.24
MBMBR	20.89	22.91	25.29	26.70	27.14
MBMBR_L	25.48	28.72	29.72	30.56	31.43
	Ancestral Sampling				
$\lvert\mathcal{H}\rvert$	4	8	16	32	64
MBR	18.48	21.54	23.93	24.89	25.61
MBMBR	17.42	19.80	22.35	23.40	23.66
MBMBR_L	21.40	25.10	27.57	28.47	29.36

Table 16. BLEU scores on nocaps dataset. We evaluate using the first 1000 inputs of the dataset. BERTScore is used as the utility function.

Q: Summarize the above article briefly in one sentence.
A:

For E2E NLG, we use the few-shot learning prompt provided by Suzgun et al. (2023).[3]

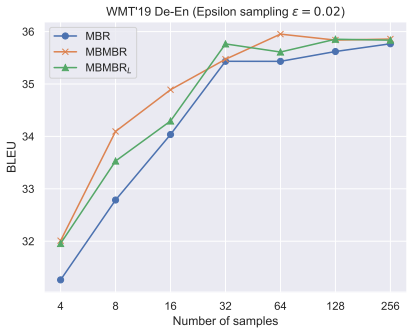
H. Additional Figures

Figures 4, 5, 6, 7, 8, 9, 10 show the performance of the methods as a function of the number of samples.

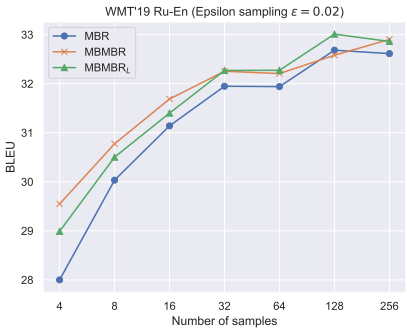
I. Pretrained Models used in the Experiments

We list the pretrained models we used in the experiments in Table 17.

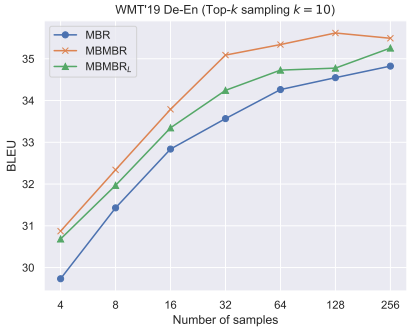
[3] https://github.com/suzgunmirac/crowd-sampling/blob/3f7fc674925d5e43691b3e23def0bd3b6b0ff799/prompts/e2e_nlg_clean_fs.txt



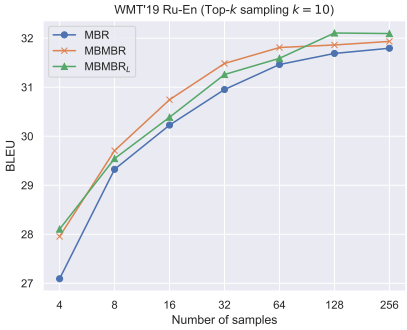
(a) WMT'19 De-En (Epsilon)



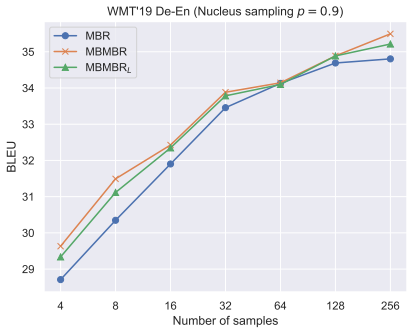
(b) XWMT'19 Ru-En (Epsilon)



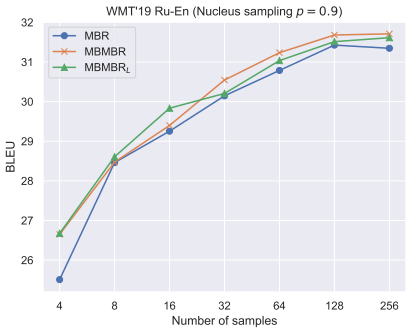
(c) WMT'19 De-En (Top-k)



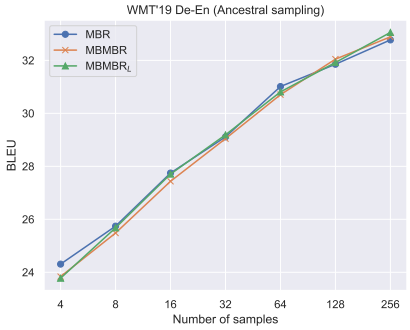
(d) XWMT'19 Ru-En (Top-k)



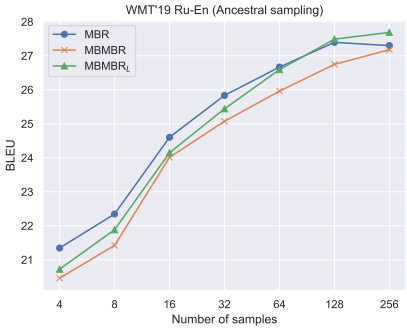
(e) WMT'19 De-En (Nucleus)



(f) XWMT'19 Ru-En (Nucleus)

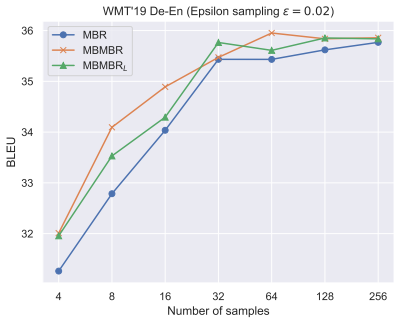


(g) WMT'19 De-En (Ancestral)

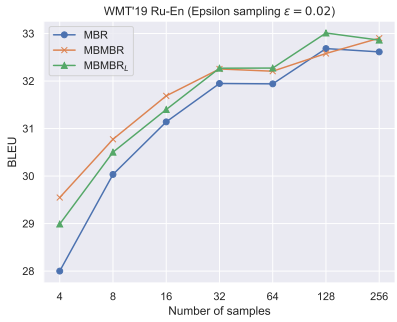


(h) XWMT'19 Ru-En (Ancestral)

Figure 4. BLEU score as a function of the number of samples on WMT'19 De-En and Ru-En (Section 5.2.1).



(a) WMT'19 En-De (Epsilon)



(b) XWMT'19 En-Ru (Epsilon)

Figure 5. BLEU score as a function of the number of samples on WMT'19 En-De and En-Ru (Section 5.2.1).

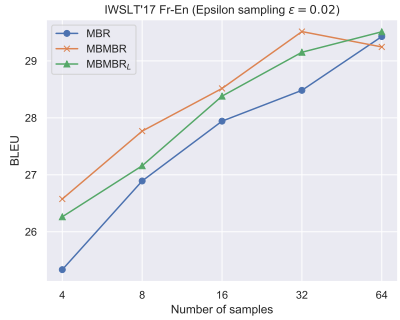
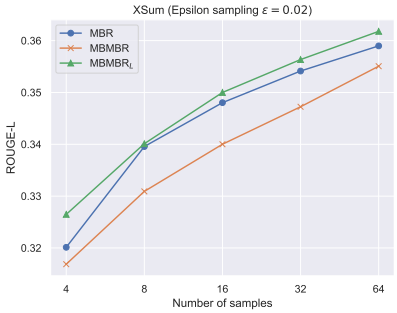
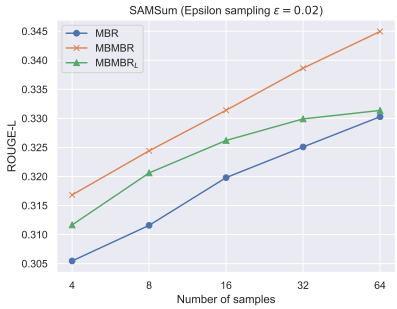


Figure 6. BLEU score as a function of the number of samples on IWSLT'17 Fr-En (Section 5.2.2).



(a) XSum (Epsilon)



(b) SAMSum (Epsilon)

Figure 7. ROUGE-L score as a function of the number of samples on XSum and SAMSum (Section 5.3.1).

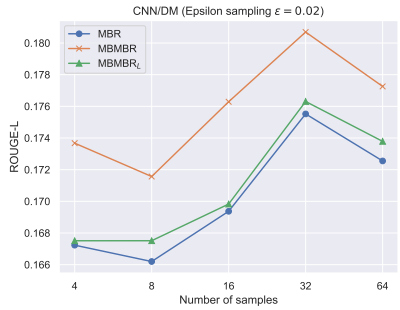
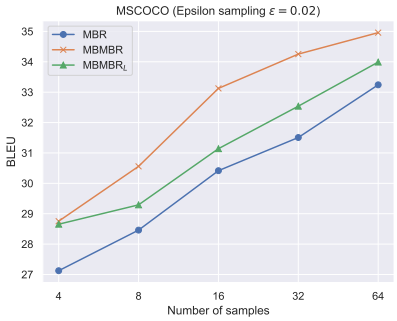
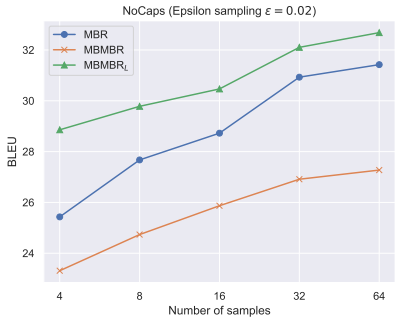


Figure 8. ROUGE-L score as a function of the number of samples on CNN/DM (Section 5.3.2).



(a) MSCOCO (Epsilon)



(b) nocaps (Epsilon)

Figure 9. BLEU score as a function of the number of samples on MSCOCO and nocaps (Section 5.4).

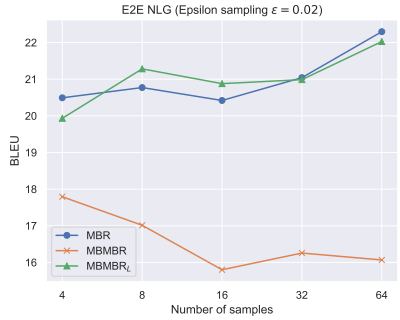


Figure 10. BLEU score as a function of the number of samples on E2E NLG (Section 5.5).

Task	Model
WMT'19 (Section 5.2.1)	Ng et al. (2019) https://github.com/facebookresearch/fairseq/blob/main/examples/wmt19/README.md
IWSLT 2017 (Section 5.2.2)	Muennighoff et al. (2023) https://huggingface.co/bigscience/bloomz-7b1-mt
XSum (Section 5.3.1)	Lewis et al. (2020) https://huggingface.co/facebook/bart-large-xsum
SAMSum (Section 5.3.1)	https://huggingface.co/philschmid/bart-large-cnn-samsum
CNN/DM (Section 5.3.2)	Jiang et al. (2023) https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1
MS COCO (Section 5.4)	Li et al. (2023) https://huggingface.co/Salesforce/blip2-flan-t5-xl-coco
nocaps (Section 5.4)	Li et al. (2023) https://huggingface.co/Salesforce/blip2-flan-t5-xl
E2E NLG (Section 5.5)	Jiang et al. (2023) https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1
IWSLT 2017 (Appendix E)	Alves et al. (2024) https://huggingface.co/Unbabel/TowerInstruct-7B-v0.1 Xu et al. (2024) https://huggingface.co/haoranxu/ALMA-7B-R

Table 17. List of pretrained models we used in the experiments.