Hyperparameter-Free Approach for Faster Minimum Bayes Risk Decoding

Anonymous ACL submission

Abstract

Minimum Bayes-Risk (MBR) decoding is shown to be a powerful alternative to beam search decoding for a wide range of text generation tasks. However, MBR requires a huge amount of time for inference to compute the MBR objective, which makes the method infeasible in many situations where response time is critical. Confidence-based pruning (CBP) (Cheng and Vlachos, 2023) has recently been proposed to reduce the inference time in machine translation tasks. Although it is shown to significantly reduce the amount of computation, it requires hyperparameter tuning using a development set to be effective. To this end, we propose Adaptive Minimum Bayes-Risk (AMBR) decoding, a hyperparameter-free method to run MBR decoding efficiently. AMBR is derived from the observation that the problem of computing the sample-based MBR objective is the *medoid identification problem*. AMBR uses the Correlated Sequential Halving (CSH) algorithm (Baharav and Tse, 2019), the algorithm with the best performance guarantee to date for the medoid identification problem, to compute the sample-based MBR objective. We evaluate AMBR on machine translation, text summarization, and image captioning tasks. The results show that AMBR achieves on par with CBP, with CBP selecting hyperparameters through an Oracle for each given computation budget.

1 Introduction

The goal of natural language generation is to generate text representing structured information that is both fluent and contains the appropriate information. One of the key design decisions in text generation is the choice of decoding strategy. The decoding strategy is the decision rule used to generate sequences from a probabilistic language model. Beam search has been widely used in many close-ended sequence 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) reports 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 and Byrne, 2004; Eikema and Aziz, 2020, 2022; Freitag et al., 2022; Bertsch et al., 2023). Unlike beam search which seeks to find the most probable output, MBR decoding seeks to find the output that maximizes the expected utility. MBR decoding involves two steps. It first samples outputs from the probabilistic model and then computes the utility between each pair of outputs to find the hypothesis with the highest expected utility.

One of the most important shortcomings of MBR decoding is its speed. The computational complexity of MBR decoding is $O(N \cdot G + N^2 \cdot U)$, where N is the number of samples to be used, G is the time to generate a sample, and U is the time to evaluate the utility function. As the utility function is typically a time-consuming neural metric such as BLEURT and COMET (Sellam et al., 2020; Pu et al., 2021; Rei et al., 2020, 2022), $O(N^2 \cdot U)$ is the dominant factor of the computational complexity.

Confidence-based pruning (CBP) has recently proposed to reduce the number of evaluations of the utility function (Cheng and Vlachos, 2023). CBP is shown to be effective in machine translation tasks, significantly reducing the required computation using both lexical and neural utility functions with a negligible drop in the quality.

Although CBP is shown to be efficient, the performance of CBP is significantly influenced by the choice of hyperparameters. As such, CBP requires a development set for the tuning of these hyperparameters. Additionally, CBP cannot dictate the speed at which it completes tasks. The hyperparameters of CBP only offer indirect control over the number of evaluations.

To this end, we propose **Adaptive Minimum Bayes-Risk (AMBR)** decoding, a hyperparameter-free algorithm to compute the sample-based MBR objective efficiently. AMBR reformulates the MBR objective as the medoid identification problem (Rdusseeun and Kaufman, 1987) and solves it using the Correlated Sequential Halving (CSH) algorithm, the best algorithm to date to solve the medoid identification problem (Baharav and Tse, 2019). The strength of AMBR is that it is free from hyperparameters. Unlike CBP where it needs to tune the hyperparameters to empirically determine the best set of hyperparameters to achieve the desired trade-off between the speed and the quality, AMBR determines the best resource allocation automatically from the computational budget specified by the user.

We evaluate the performance of AMBR in machine translation, text summarization, and image captioning tasks. The empirical results show that AMBR is on par with CBP with Oracle hyperparameters. They are roughly 4 to 8 times faster than MBR with a marginal drop in the output quality. The result indicates that using AMBR, MBR decoding can be run efficiently for a given computation budget specified on the fly without hyperparameter tuning on a development set.

2 Background

Conditional text generation is the task of generating an output sequence $\mathbf{h}$ given an input sequence $\mathbf{x}$. Probabilistic text generators define a probability distribution $P_{\text{model}}(\mathbf{h}|\mathbf{x})$ over an output space of hypotheses $\mathcal{Y}$. In this paper, we denote $P_{\text{model}}(\mathbf{h}|\mathbf{x})$ by $P_{\text{model}}(\mathbf{h})$ for brevity. 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 output under the model:

$$\mathbf{h}^{\text{MAP}} = \arg\max_{\mathbf{h}\in\mathcal{Y}} P_{\text{model}}(\mathbf{h}). \tag{1}$$

Although it seems intuitive to solve this MAP objective, prior work has pointed out two critical problems with this strategy. First, since the size of hypotheses set $|\mathcal{Y}|$ is extremely large, solving it exactly is intractable. Second, the MAP objective often leads to low-quality outputs (Stahlberg and Byrne, 2019; Holtzman et al., 2020; Meister et al., 2020). In fact, Stahlberg and Byrne (2019) shows that $\mathbf{h}^{\text{MAP}}$ is often the empty sequence in their experiment setting.

As such, beam search is commonly used as a heuristic algorithm to solve decoding problems (Graves, 2012; Sutskever et al., 2014). Beam search is known to generate higher-quality sequences than MAP decoding in a wide range of tasks. Still, prior work has reported the degeneration issues of beam search such as repetitions and infinite-length outputs (Cohen and Beck, 2019; Holtzman et al., 2020).

2.1 Minimum Bayes-Risk (MBR) Decoding

Unlike MAP decoding which searches for the most probable output, MBR decoding seeks to find the output that maximizes the expected utility, thus minimizing the risk equivalently (Kumar and Byrne, 2002, 2004). The procedure is made of two components: a machine translation model and a utility metric. The model $P_{\text{model}}(\mathbf{y}|\mathbf{x})$ estimates the probability of an output $\mathbf{y}$ given an input sentence $\mathbf{x}$. The utility metric $u(\mathbf{y}, \mathbf{y}')$ estimates the quality of a candidate translation $\mathbf{y}$ given a reference translation $\mathbf{y}'$. Given a set of candidate hypotheses $\mathcal{H} \subseteq \mathcal{Y}$, we select the best hypothesis according to its expected utility with respect to the distribution of human references P_{human}.

$$\mathbf{h}^{\text{human}} = \arg\max_{\mathbf{h}\in\mathcal{H}} \mathop{\mathbb{E}}_{\mathbf{y}\sim P_{\text{human}}} [u(\mathbf{h}, \mathbf{y})]. \tag{2}$$

Because P_{human} is unknown, MBR instead uses the model probability P_{model} to approximate P_{human}:

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

For the rest of the paper, we denote P_{model} as P for simplicity if not confusing. As integration over $\mathcal{Y}$ is computationally intractable, Eq. (3) is approximated by a **Monte Carlo estimate** (Eikema and Aziz, 2022; Farinhas et al., 2023) using a pool of references $\mathcal{R}$ sampled from P:

$$\mathbf{h}^{\text{MC}} = \arg\max_{\mathbf{h}\in\mathcal{H}} \frac{1}{|\mathcal{R}|} \sum_{\mathbf{y}\in\mathcal{R}} u(\mathbf{h}, \mathbf{y}). \tag{4}$$

In this paper, we investigate algorithms to compute $\mathbf{h}^{\text{MC}}$ efficiently.

2.2 Computational Complexity of MBR Decoding

The shortcoming of the MBR is that it requires a huge amount of computation at inference time. The computational complexity of MBR is $O(|\mathcal{H} \cup \mathcal{R}| \cdot G + |\mathcal{H}||\mathcal{R}| \cdot U)$ where G is the upper bound of the time to generate a hypothesis, and U is the upper bound of the time to evaluate the utility function for a pair of hypotheses (Eikema and Aziz, 2022). Sample-based MBR typically uses the same set of hypotheses for the candidate set $\mathcal{H}$ and the reference pool $\mathcal{R}$ ($\mathcal{H} = \mathcal{R}$). In this way the computational complexity is $O(N \cdot G + N^2 \cdot U)$, where $N = |\mathcal{H}| = |\mathcal{R}|$. Thus, The bottleneck of the computation is typically the evaluation of the utility function.

Several approaches have been proposed to improve the efficiency of MBR decoding before confidence-based pruning (Eikema and Aziz, 2022; Freitag et al., 2022). **N-by-S (NbyS)** seeks to reduce the total number of evaluations by reducing the reference pool (Eikema and Aziz, 2022). Eikema and Aziz (2022) provides empirical evidence showing that increasing the number of candidates is more effective than increasing the number of references. The computational complexity of N-by-S with $S'(< N)$ references is $O(N \cdot G + NS' \cdot U)$. **Coarse-to-Fine (C2F)** reduces the size of the candidate and reference hypotheses using a coarse utility function (Eikema and Aziz, 2022). It first runs coarse evaluation using a faster utility function (e.g. non-neural lexical scoring function). It then selects the top-scoring hypotheses as a pruned candidate set and reference set. Finally, it runs the MBR decoding with the finer utility function using the pruned candidate and reference set to output the best hypothesis. In this way, the total computation required by C2F is $O(N \cdot G + N^2 \cdot U' + N'S' \cdot U)$ where U' is the computational cost of the coarse utility function, $N', S'(\leq N)$ are the size of the pruned candidate and reference set.

Reference Aggregation (RA) computes the MBR score against aggregated reference representations to reduce the computational complexity to $O(N \cdot G + N \cdot U^A)$, where U^A is the upper bound on the complexity of evaluating the aggregated utility function (Vamvas and Sennrich, 2024). The shortcoming of RA is that it is not applicable to non-aggregatable utility functions. For example, MetricX-23 (Juraska et al., 2023) is a transformer-based metric where the input is a sequence of embeddings of the tokens instead of the embedding of the whole sentence, making it non-aggregatable. Another example is where the utility function involves a reward function. See Appendix A for details.

3 Confidence-Based Pruning (CBP)

Confidence-based pruning (CBP) is recently proposed by Cheng and Vlachos (2023) to significantly reduce the number of evaluations of the utility function. The idea is to iteratively evaluate the hypotheses with a subset of the reference set to prune the hypotheses not promising enough.

CBP keeps a current candidate set $\mathcal{H}_i$ and a current reference set $\mathcal{R}_i$ during the run. The candidate set starts from the whole candidates ($\mathcal{H}_0 = \mathcal{H}$) and the reference set starts empty ($\mathcal{R}_0 = \emptyset$). At every iteration i, it draws samples and adds them to the reference set until the size of the reference set reaches the limit r_i, where $\{r_i\}$ are hyperparameters. Then it computes the incumbent best solution $\mathbf{h}_i^*$ at i-th iteration:

$$\mathbf{h}_i^* = \underset{\mathbf{h} \in \mathcal{H}_i}{\arg\max} \frac{1}{|\mathcal{R}_i|} \sum_{\mathbf{y} \in \mathcal{R}_i} u(\mathbf{h}, \mathbf{y}). \quad (5)$$

Then it generates a series of bootstrap reference sets $\hat{\mathcal{R}}_i^b$ which is a with-replacement size-$|\mathcal{R}|$ resample of $\mathcal{R}_i$. Using a series of bootstrap reference sets, it computes the estimated win ratio of each hypothesis against $\mathbf{h}_i^*$ in $\mathcal{H}_i$:

$$w(\mathbf{h}) = \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}[\sum_{\mathbf{y} \in \hat{\mathcal{R}}_i^b} u(\mathbf{h}, \mathbf{y}) \geq \sum_{\mathbf{y} \in \hat{\mathcal{R}}_i^b} u(\mathbf{h}_i^*, \mathbf{y})], \quad (6)$$

where B is the number of bootstrap reference sets. Then, it prunes all candidates from the candidate set with the win ratio lower than $1 - \alpha$, where α is a hyperparameter. It repeats the process until the size of the candidate set reaches 1 or the sample size scheduler terminates.

Although CBP is shown to be significantly more efficient than the standard MBR, there are several shortcomings. First, it requires a hyperparameter tuning using the development set. The sample size scheduler r_i and the confidence threshold α need to be tuned to optimize the performance. The number of bootstrap reference sets B is also a hyperparameter that needs to be tuned according to the quality and the speed trade-off. Note that the optimal set

Algorithm 1: Adaptive MBR (AMBR) (Correlated Sequential Halving for MBR)

Input: a set of candidates $\mathcal{H}$, references $\mathcal{R}$, and a budget T
Output: a hypothesis $\mathbf{h}^{\mathrm{AMBR}}$

$\mathcal{H}_0 \leftarrow \mathcal{H}$
$\mathcal{R}_0 \leftarrow \emptyset$
$N = \max(|\mathcal{H}|, |\mathcal{R}|)$
for $i = 0$ **to** $\lceil \log N \rceil - 1$ **do**
  $t_i = \min(\max(\lfloor \frac{T}{|\mathcal{H}_i|\lceil \log n \rceil} \rfloor, 1), n)$
  Let J_i be a set of $t_i - |\mathcal{R}_i|$ references sampled from $\mathcal{R} \setminus \mathcal{R}_i$ without replacement
  $\mathcal{R}_{i+1} = J_i \cup \mathcal{R}_i$
  for $\mathbf{h} \in \mathcal{H}_i$ **do**
    $\hat{U}(\mathbf{h}) \leftarrow \frac{1}{|\mathcal{R}_{i+1}|} \sum_{\mathbf{y} \in \mathcal{R}_{i+1}} u(\mathbf{h}, \mathbf{y})$
  end for
  if $t_i = n$ **then**
    return $\arg\max_{\mathbf{h} \in \mathcal{H}_i} \hat{U}(\mathbf{h})$
  else
    Let $\mathcal{H}_{i+1}$ be the set of $\lceil |\mathcal{H}_i|/2 \rceil$ candidates in $\mathcal{H}_i$ with the largest $\hat{U}(\mathbf{h})$
  end if
end for
return $\arg\max_{\mathbf{h} \in \mathcal{H}^i} \hat{U}(\mathbf{h})$

of hyperparameters is influenced by the desired speed-up. If one wants to choose 2x speed-up and 4x speed-up according to the situation, one needs to search for two sets of hyperparameters for each budget constraint. Additionally, CBP cannot give a budget constraint and optimize under that. Because the hyperparameters of CBP only indirectly control the number of evaluations it needs to finish, a user has no direct control over the desired speed-up.

4 Adaptive Minimum Bayes Risk (AMBR) Decoding

We propose **Adaptive Minimum Bayes-Risk (AMBR)** decoding, a variant of MBR that can efficiently compute the MBR objective under a budget on the maximum number of evaluations that a user can specify. The advantages of AMBR over CBP are twofold. First, AMBR has no hyperparameter. The schedules of the number of references and the candidates are automatically determined by the algorithm. Second, a user can enforce the upper bound of the computation budget to AMBR. AMBR enforces the budget constraint and the algorithm automatically schedules how to use the limited resource accordingly.

AMBR is derived from the observation that MBR decoding is the *medoid identification problem* (Kaufman and Rousseeuw, 1990): the problem of computing $\mathbf{h}^{\mathrm{MC}}$ (Eq 4) is tantamount to determining the medoid of $\mathcal{H}$. The medoid, denoted as $\mathbf{y}^*$, is defined as the point in a dataset Y that minimizes the sum of distances to all other points:[1]

$$\mathbf{x} = \arg\min_{\mathbf{x} \in X} \sum_{\mathbf{y} \in Y} d(\mathbf{x}, \mathbf{y}). \tag{7}$$

Let $d = -u$, $X = \mathcal{H}$, and $Y = \mathcal{R}$. Then, the problem can be translated into the following:

$$\mathbf{y}^* = \arg\max_{\mathbf{y} \in \mathcal{H}} \sum_{\mathbf{y}' \in \mathcal{R}} u(\mathbf{y}, \mathbf{y}'). \tag{8}$$

This is exactly the objective defined in Eq. (4).

Our approach is to use the best algorithm proposed so far for solving the medoid identification problem and repurpose it for MBR decoding. The algorithm with the best performance guarantee to date for solving the medoid identification is the **Correlated Sequential Halving (CSH)** algorithm (Baharav and Tse, 2019). We describe the procedure of AMBR in Algorithm 1. AMBR keeps a current candidate set $\mathcal{H}_i$ which starts with $\mathcal{H}$ and a current reference set $\mathcal{R}_i$ which starts as an empty set. First, it picks t_i hypotheses from $\mathcal{R}$ and adds them to the current reference set $\mathcal{R}_{i+1}$ where t_i is automatically determined by the number of candidates and the budget. Then it computes $u(\mathbf{h}, \mathbf{y})$ for all $\mathbf{h}$ in the current candidate set $\mathcal{H}_i$ and for all $\mathbf{y}$ in the current reference set $\mathcal{R}_{i+1}$. The average utility of $\mathbf{h} \in \mathcal{H}_i$ over the current reference set is stored in $\hat{U}(\mathbf{h})$. Then, it runs the halving operation, pruning the lower half of the candidates according to the current estimate $\hat{U}$. Ties are broken arbitrarily. It repeats this process for up to $\lceil \log N \rceil - 1$ times and returns the candidate with the best estimate in $\mathcal{H}_i$ at that point.

The procedure of Algorithm 1 is identical to the procedure of CSH with modification to the notations to place it in the context of the decoding problem. Our contribution is the reinvention of CSH which is proposed as a solution to the medoid identification problem as a tool to compute the MBR objective adaptively by converting the sum of distances to the expected utility.

[1] The formulation of Eq. (7) represents the same class of problem as the standard formulation of medoid identification problem where it assumes $X = Y$. See Appendix B for the details.

4.1 Analytical Result

CSH has a theoretical guarantee of the probability of choosing the hypothesis with the highest utility in its original form (Baharav and Tse, 2019). The original form of CSH is recovered by replacing Line 7 of Algorithm 1 with the following equation:

$$\mathcal{R}_{i+1} = J_i. \tag{9}$$

In this way, AMBR inherits the theoretical guarantee of CSH:

Lemma 1. *Assuming $T \geq N \log N$, AMBR replacing Line 7 with Eq. (9) correctly identifies $\mathbf{h}^{\mathrm{MC}}$ with probability at least $1 - \log N \exp(-\frac{T}{\log N} C)$ where C is an instance dependent variable determined by u and $\mathcal{H}$.*

See Theorem 2.1. of Baharav and Tse (2019) for proof and a detailed description of the instance-dependent variable C.

AMBR reuses the reference set from the previous iteration so that all the available references are used to estimate the expected utility. Therefore, it is expected to perform better in practice than the CSH of its original form, as noted in Remark 1 in Baharav and Tse (2019).

5 Experiments

We evaluate the performance of the efficient MBR decoding algorithms on machine translation, text summarization, and image captioning tasks. We evaluate the performance of the MBR decoding algorithms under a budget constraint on the number of evaluations. We evaluate with a budget size of $1/32, 1/16, 1/8, 1/4, 1/2$ of $N(N-1)$, the number of evaluations of the standard MBR with N samples.[2] We use epsilon sampling with $\epsilon = 0.02$ as a sampling algorithm (Hewitt et al., 2022; Freitag et al., 2023). Temperature is fixed to 1.0. We use the same set of samples for all the algorithms.

We compare the performance of (Standard) MBR, N-by-S (NbyS), Coarse-to-fine (C2F), confidence-based pruning (CBP), and AMBR. Standard MBR refers to the implementation of MBR which uses the same set of samples for the candidate and reference set. We run standard MBR with the number of samples $N' \in \{1...N\}$. The number of evaluations for standard MBR is $N'(N'-1)$. We implement N-by-S in a way that uses all the samples $\mathcal{H}$ as the candidate set and reduces the size of references according to the budget. That is, it randomly subsamples S' hypotheses from $\mathcal{H}$ to be the reference set so that S' is the smallest integer such that $(N-1)S' \geq T$. For C2F, we set $S' = N$ and N' to be the smallest integer such that $N'(N-1) \geq T$. We run a hyperparameter sweep for CBP to find the best hyperparameters. We search over $r_0 \in \{1, 2, 4, 8\}$ and $\alpha \in \{0.8, 0.9, 0.99\}$. Following Cheng and Vlachos (2023), we set the schedule of the size of the references r_i to double each step: $r_i = 2^i r_0$. The number of bootstrap reference sets is 500. We enforce the budget constraint to CBP by terminating the iteration once the number of evaluations reaches T. We run CBP with each set of hyperparameters on the test set to find the best hyperparameters. The result of the hyperparameter search is described in Appendix C. We observe that the best set of hyperparameters of CBP is dependent on the size of the budget. As such, we report the Oracle score, the best score over all combinations of hyperparameters for each budget. AMBR is implemented as in Algorithm 1 without using Eq. (9). Thus, Lemma 1 does not apply to the algorithm we evaluate in this section. We run NbyS, CBP, and AMBR five times for each budget size and report the average, minimum, and maximum scores over the runs.

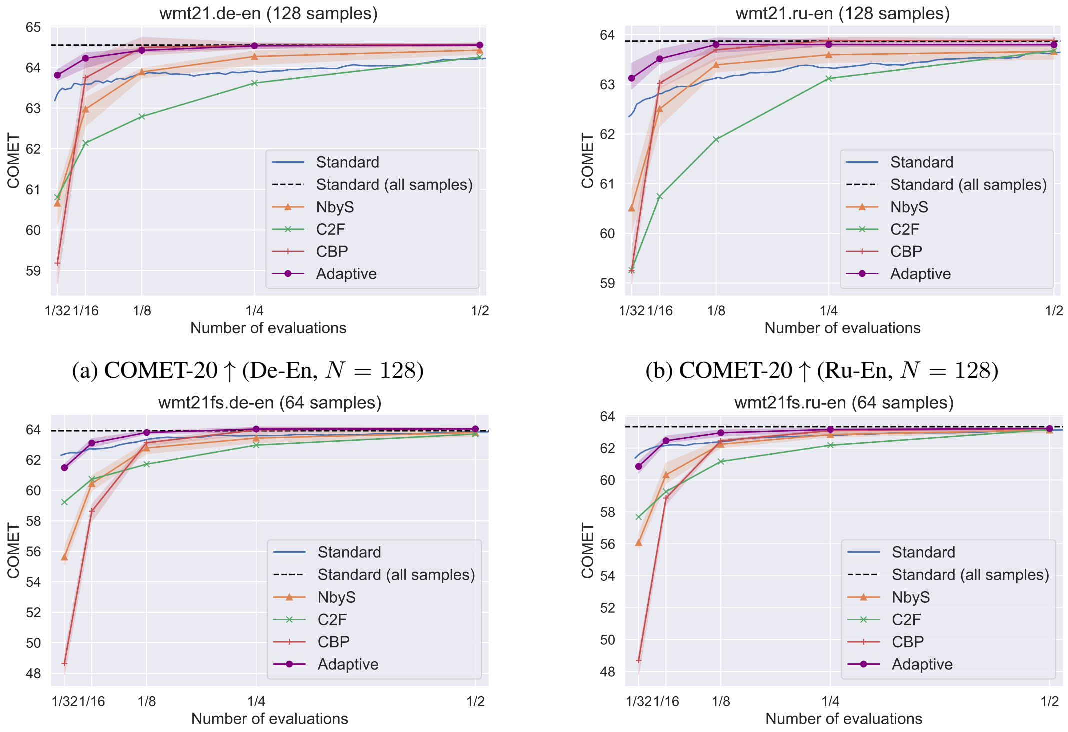
We use Huggingface's Transformers library for running all the experiments (Wolf et al., 2020). All the experiments are conducted using publicly available pretrained models and datasets for reproducibility. Due to limitations in computational resources, we evaluate the first 1000 entries of each dataset.

5.1 Machine Translation

We evaluate the performance on machine translation tasks using WMT'21 test dataset. We use German-English (De-En) and Russian-English (Ru-En) language pairs. We use the WMT 21 X-En model and M2M100 418M model to sample sequences for both language pairs (Tran et al., 2021; Fan et al., 2021). We load the WMT 21 X-En model in 4-bit precision to reduce the GPU memory consumption. We use COMET-20 as the utility function and the evaluation metric (Rei et al., 2020). We use the BLEU score as a coarse utility function of C2F.

AMBR is on par with Oracle CBP Figure 1 shows the results with varying evaluation budgets with a fixed number of samples ($N = 64, 128$) using the WMT 21 X-En model. We observe that

[2] We assume $u(\mathbf{h}, \mathbf{h})$ is a constant for all $\mathbf{h} \in \mathcal{H}$.



(a) COMET-20 ↑ (De-En, $N = 128$)

(b) COMET-20 ↑ (Ru-En, $N = 128$)

(c) COMET-20 ↑ (De-En, $N = 64$)

(d) COMET-20 ↑ (Ru-En, $N = 64$)

Figure 1: COMET-20 score on WMT'21 De-En and Ru-En using the WMT 21 X-En model. The shaded regions show the minimum and the maximum values over five runs. The horizontal axis shows the reduction in the number of evaluations compared to the standard MBR with all samples.

AMBR achieves the best COMET score and the error rate compared to the others. The error rate is the ratio of selecting a hypothesis different from the standard MBR using all 128 (64) samples. It achieves almost the same score as the standard MBR with all samples within $1/4 - 1/8$ number of evaluations, resulting in $4 - 8$ times speed up compared to standard MBR. We observe qualitatively the same result on the M2M100 model (see Appendix D.3). Additional evaluations on WMT'21 En-De and En-Ru are described in Appendix D.2.

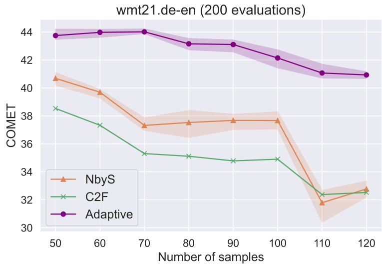
AMBR scales with the number of samples given enough budget To evaluate the scalability of AMBR on the number of samples, we evaluate the COMET scores with varying numbers of samples with a fixed amount of evaluation budgets using the M2M100 418M model. Figure 2 shows the COMET scores with varying numbers of samples with a fixed amount of evaluation budgets on De-En. The COMET score of AMBR scales with the number of samples if and only if the number of evaluations is large enough. This is to be expected as Lemma 1 only holds when the budget is large enough. The same trend is observed on Ru-En (Appendix D.4).

5.2 Text Summarization

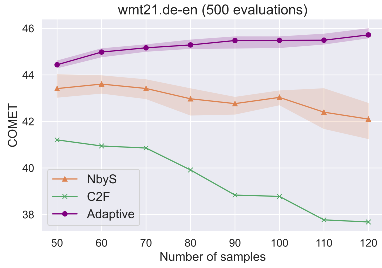
We evaluate the performance of AMBR on text summarization tasks using SAMSum (Gliwa et al., 2019) and XSum dataset (Narayan et al., 2018). We use BART models fine-tuned on each dataset (Lewis et al., 2020). We use InfoLM (Colombo et al., 2022) with the Fisher-Rao distance (Rao, 1987) as a utility function as it is shown to have a high correlation with human judgment on text summarization tasks. We generate $N = 64$ samples as a candidate set for each input. Following Eikema and Aziz (2022), we use the F1 score of the unigram as a coarse utility function of C2F.

The results are summarized in Figure 3. Despite AMBR reduces the error rate significantly (Figure 3c and 3f), it only slightly improves upon standard MBR with respect to InfoLM and ROUGE-L score (Figure 3a, 3b, 3d, and 3e). We speculate that this is because many of the top-scoring samples are similar in quality measured by InfoLM and ROUGE-L.

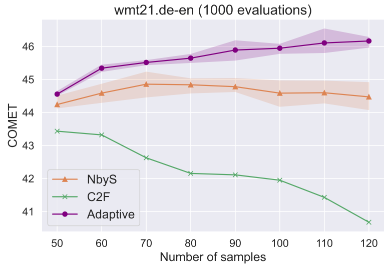
C2F can surpass the score of standard MBR under conditions Interestingly, we observe that C2F surpasses the performance of standard MBR



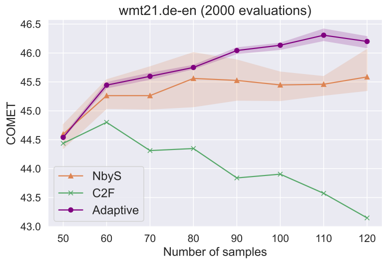
(a) COMET-20 ↑ (De-En, 200 evaluations)



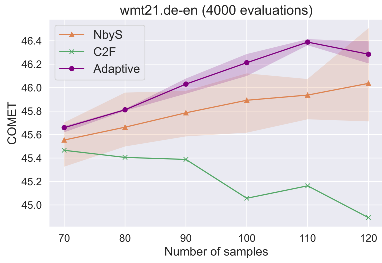
(b) COMET-20 ↑ (De-En, 500 evaluations)



(c) COMET-20 ↑ (De-En, 1000 evaluations)



(d) COMET-20 ↑ (De-En, 2000 evaluations)



(e) COMET-20 ↑ (De-En, 4000 evaluations)

Figure 2: Evaluation of AMBR with a varying number of samples with a fixed evaluation budget on WMT'21 De-En with COMET-20 score using the M2M100 418M model. The shaded regions show the minimum and the maximum values over five runs.

with all samples on ROUGE-L for XSum dataset. We speculate that C2F may improve upon MBR because it effectively ensembles two utility functions. Because the F1 score of the unigram may be more aligned to ROUGE-L score than InfoLM is, it can pick sentences favored by ROUGE-L metric. As such, C2F can not only speed up the computation of the MBR objective but also improve the alignment to the target metric.

5.3 Image Captioning

We evaluate the performance of AMBR on image captioning task using MS COCO dataset (Lin et al., 2014). We use BLIP-2 (Li et al., 2023a) with Flan T5-xl (Chung et al., 2022) fine-tuned for MS COCO loaded in 4-bit precision. We use a cosine similarity of the textual CLIP embeddings as the utility function (Radford et al., 2021; Hessel et al., 2021). We use RefCLIPScore and BLEU as an evaluation metric (Hessel et al., 2021; Papineni et al., 2002). We generate $N = 64$ samples for each image. We use the F1 score of the unigram as a coarse utility function of C2F.

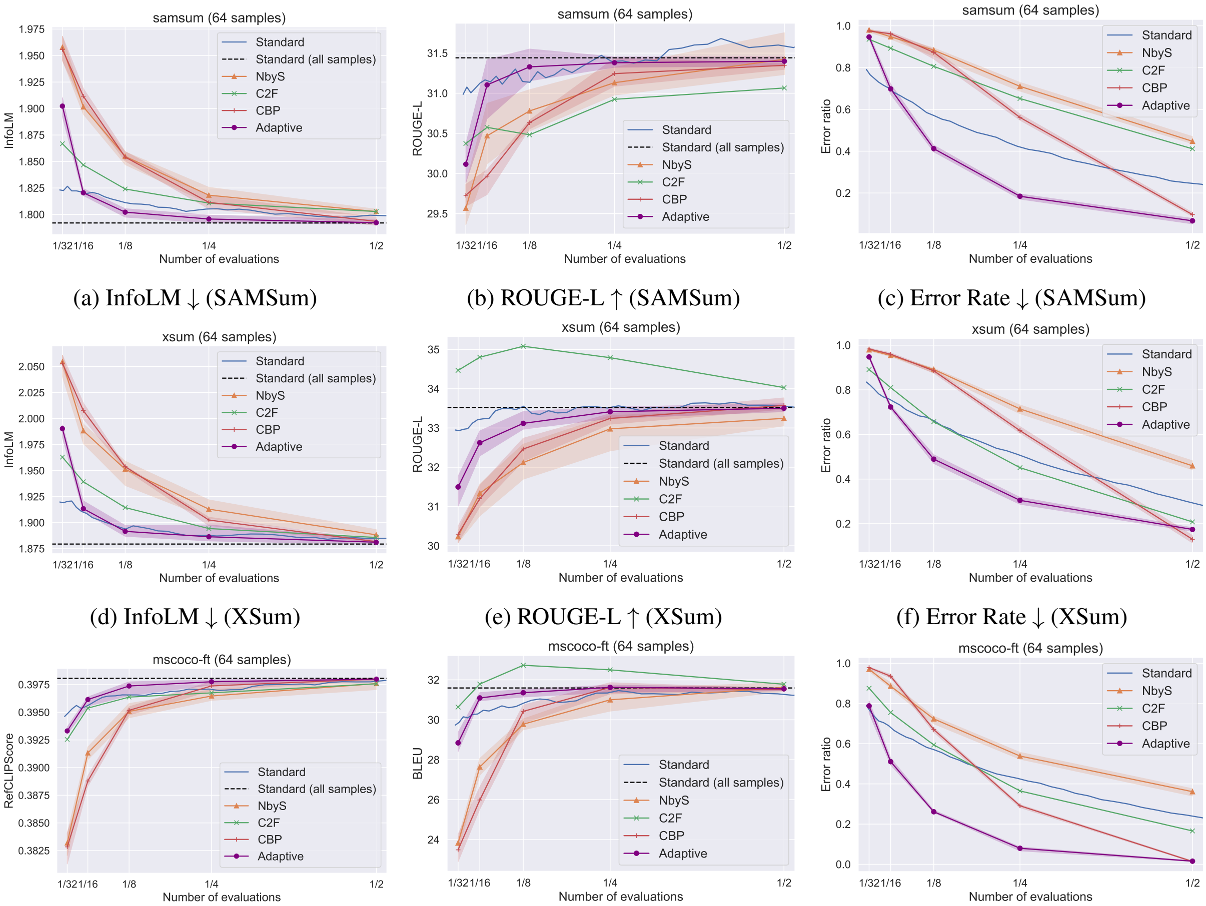
The empirical result is shown in Figure 3. AMBR achieves roughly 4 to 8 times speed-up compared to MBR with a marginal drop in RefCLIPScore and BLEU score (Figure 3g and 3h). We observe C2F to improve upon standard MBR with respect to BLEU score (Figure 3h). As in text summarization (Section 5.2), We speculate that this is because the F1 score has a better alignment with the BLEU score than the CLIP embeddings so that the coarse utility function is effectively serving as another utility function.

6 Related Work

MBR has been investigated in many NLP tasks including parsing (Goodman, 1996), speech recognition (Goel and Byrne, 2000), bilingual word alignment (Kumar and Byrne, 2002), and machine translation (Kumar and Byrne, 2004). MBR has recently gained attention in machine translation as a decision rule as a method to overcome some of the biases of MAP decoding in NMT (Eikema and Aziz, 2020; Müller and Sennrich, 2021; Eikema and 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) rather than lexical overlap metrics (e.g. BLEU) further improves MBR.

CSH (Baharav and Tse, 2019) is not the only algorithm proposed to solve the medoid identification problem. There are several other algorithms to solve the medoid identification (Eppstein and



(a) InfoLM ↓ (SAMSum) (b) ROUGE-L ↑ (SAMSum) (c) Error Rate ↓ (SAMSum)

(d) InfoLM ↓ (XSum) (e) ROUGE-L ↑ (XSum) (f) Error Rate ↓ (XSum)

(g) RefCLIPScore ↑ (MS COCO) (h) BLEU ↑ (MS COCO) (i) Error Rate ↓ (MS COCO)

Figure 3: (a) InfoLM score, (b) ROUGE-L score, and (c) error rate on SAMSum dataset. (d) InfoLM score, (e) ROUGE-L score, and (f) error rate on XSum dataset. (g) RefCLIPScore, (h) BLEU score, and (i) error rate on MS COCO dataset. The shaded regions show the minimum and the maximum values over five runs. The error rate is defined as the ratio of selecting a hypothesis different from the one selected by standard MBR using all the samples ($N = 64$).

Wang, 2006; Okamoto et al., 2008; Bagaria et al., 2018). We pick to use CSH as it has the best theoretical performance.

Algorithms to solve the problem of identifying the best option out of the candidates with a budget constraint (fixed-budget best-arm identification problems) are known to be highly sensitive to the choice of the hyperparameters if they have ones (Carpentier and Locatelli, 2016; Kaufmann et al., 2016). In fact, we observe that the effectiveness of CBP hinges on the appropriate selection of hyperparameters, given each budget constraint.

7 Conclusions

We propose Adaptive Minimum Bayes-Risk (AMBR) decoding, a hyperparameter-free algorithm for efficient MBR decoding. AMBR considers the problem of computing the MBR objective as the medoid identification problem and uses the known best algorithm to solve it. The strength of the AMBR is that it doesn't need a development set to tune the set of hyperparameters. AMBR automatically computes the strategy on the fly given the budget specified by the user.

Experimental result shows that the performance of AMBR is on par with CBP with hyperparameters picked by an Oracle on machine translation tasks. AMBR outperforms CBP on text summarization and image captioning tasks, using the same set of hyperparameters as in machine translation tasks for CBP. We speculate that CBP requires a different set of hyperparameters for each task to perform on par with AMBR.

We believe that AMBR will be a practical choice for future MBR decoding because of its applicability and significant performance improvements.

8 Limitations

Even with the improvement, AMBR is still many times more costly to run than beam search.

Using Eq. (9), the computational complexity of the evaluation of the utility function of AMBR is $O(N \log N \cdot U)$ to achieve the theoretical guarantee. This is still larger than the complexity of generation which is $O(N \cdot G)$. Therefore, the evaluation procedure is still the bottleneck of MBR to scale with the number of samples.

Although our focus is on reducing the computation of the utility function of MBR decoding, it is not the only way to speed up the text generation. Finkelstein and Freitag (2023) shows that by self-training a machine translation model by its own MBR-decoded output, it can improve the performance of more efficient decoding methods such as beam search. Yang et al. (2023) proposes the use of Direct Preference Optimization (Rafailov et al., 2023) to train the model to learn the ranking of the sequences according to the MBR objective. Foks (2023) shows that by training a model to predict the Monte Carlo estimate of the Bayes risk, we can directly estimate the Bayes risk using the trained model without running Monte Carlo estimation, resulting in $O(N \cdot G + N \cdot U')$ where U' is the inference time of the trained model.

We measure the number of evaluations of the utility function as a metric of efficiency. Practically, the computation of the utility function is not linear to the number of calls. One can optimize the implementation by batching and caching the computation effectively. For example, the sentence embeddings of embedding-based utility functions such as COMET can be cached to significantly speed up the computation of the utility (Amrhein and Sennrich, 2022; Cheng and Vlachos, 2023).

The paper focuses on how to effectively use the given budget and lacks a discussion on what to set the budget to. Baharav and Tse (2019) suggests the doubling trick (Besson and Kaufmann, 2018) to find the appropriate budget size. That is, we run the algorithm with a certain budget T, and then double the budget to $2T$ and rerun the algorithm. If the two answers are the same, then we output it. Because the probability of selecting the same incorrect answer twice in a row is very low, it is likely to be the best hypothesis. Empirical evaluation of the strategies to decide the budget size is future work.

The other question is on what to set the number of samples to. Figure 2 shows that having too many samples is not necessarily beneficial when the evaluation budget is too small. Finding the optimal number of samples given a budget on computation is an open question.

We consider the Monte Carlo estimate $\mathbf{h}^{\mathrm{MC}}$ as the target objective function to compute. Evaluation of AMBR using other objective functions such as model-based estimate (Jinnai et al., 2023) is future work.

Although AMBR is based on the best algorithm known to solve the medoid identification problem, it does not use any task-dependent knowledge to speed up the algorithm. One may exploit the domain knowledge of the task to further improve upon it (e.g. reference aggregation; Vamvas and Sennrich, 2024).

References

Chantal Amrhein and Rico Sennrich. 2022. Identifying weaknesses in machine translation metrics through minimum Bayes risk decoding: A case study for COMET. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1125–1141, Online only. Association for Computational Linguistics.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Vivek Bagaria, Govinda Kamath, Vasilis Ntranos, Martin Zhang, and David Tse. 2018. Medoids in almost-linear time via multi-armed bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 500–509. PMLR.

Tavor Baharav and David Tse. 2019. Ultra fast medoid identification via correlated sequential halving. *Advances in Neural Information Processing Systems*, 32.

Amanda Bertsch, Alex Xie, Graham Neubig, and Matthew R Gormley. 2023. It's mbr all the way down: Modern generation techniques through the lens of minimum bayes risk. *arXiv preprint arXiv:2310.01387*.

Lilian Besson and Emilie Kaufmann. 2018. What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint arXiv:1803.06971*.

Alexandra Carpentier and Andrea Locatelli. 2016. Tight (lower) bounds for the fixed budget best arm identification bandit problem. In *29th Annual Conference on*

Learning Theory, volume 49 of *Proceedings of Machine Learning Research*, pages 590–604, Columbia University, New York, New York, USA. PMLR.

Julius Cheng and Andreas Vlachos. 2023. Faster minimum Bayes risk decoding with confidence-based pruning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12473–12480, Singapore. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Eldan Cohen and Christopher Beck. 2019. Empirical analysis of beam search performance degradation in neural sequence models. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1290–1299. PMLR.

Pierre Jean A Colombo, Chloé Clavel, and Pablo Piantanida. 2022. Infolm: A new metric to evaluate summarization & data2text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10554–10562.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

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

Bryan Eikema and Wilker Aziz. 2022. 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*, pages 10978–10993, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

David Eppstein and Joseph Wang. 2006. Fast approximation of centrality. *Graph algorithms and applications*, 5(5):39.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(1).

António Farinhas, José G. C. de Souza, and André F. T. Martins. 2023. An empirical study of translation hypothesis ensembling with large language models. *arXiv*.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. 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*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Mara Finkelstein and Markus Freitag. 2023. Mbr and qe finetuning: Training-time distillation of the best and most expensive decoding methods. *arXiv preprint arXiv:2309.10966*.

Gerson Foks. 2023. Towards efficient minimum bayes risk decoding. Master's thesis, University of Amsterdam.

Markus Freitag, Behrooz Ghorbani, and Patrick Fernandes. 2023. Epsilon sampling rocks: Investigating sampling strategies for minimum bayes risk decoding for machine translation. *arXiv preprint arXiv:2305.09860*.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

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

Joshua Goodman. 1996. Parsing algorithms and metrics. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 177–183, Santa Cruz, California, USA. Association for Computational Linguistics.

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

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Hewitt, Christopher Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv*.

Yuu Jinnai, Tetsuro Morimura, Ukyo Honda, Kaito Ariu, and Kenshi Abe. 2023. Model-based minimum bayes risk decoding. *arXiv preprint arXiv:2311.05263*.

Juraj Juraska, Mara Finkelstein, Daniel Deutsch, Aditya Siddhant, Mehdi Mirzazadeh, and Markus Freitag. 2023. MetricX-23: The Google submission to the WMT 2023 metrics shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 756–767, Singapore. Association for Computational Linguistics.

Leonard Kaufman and Peter J. Rousseeuw. 1990. *Partitioning Around Medoids (Program PAM)*, chapter 2. John Wiley & Sons, Ltd.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. 2016. On the complexity of best-arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17(1):1–42.

Shankar Kumar and William Byrne. 2002. Minimum Bayes-risk word alignments of bilingual texts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 140–147. Association for Computational Linguistics.

Shankar Kumar and William Byrne. 2004. 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*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations – democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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*, pages 7871–7880, Online. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024. More agents is all you need. *arXiv preprint arXiv:2402.05120*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023b. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Clara Meister, Ryan Cotterell, and Tim Vieira. 2020. If beam search is the answer, what was the question? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2185, Online. Association for Computational Linguistics.

Mathias Müller and Rico Sennrich. 2021. 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)*, pages 259–272, Online. Association for Computational Linguistics.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. 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*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Kazuya Okamoto, Wei Chen, and Xiang-Yang Li. 2008. Ranking of closeness centrality for large-scale social networks. In *International workshop on frontiers in algorithmics*, pages 186–195. Springer.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for MT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 751–762, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

C Radakrishna Rao. 1987. Differential metrics in probability spaces. *Differential geometry in statistical inference*, 10:217–240.

LKPJ Rdusseeun and P Kaufman. 1987. Clustering by means of medoids. In *Proceedings of the statistical data analysis based on the L1 norm conference, neuchatel, switzerland*, volume 31.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Unbabel's participation in the WMT20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 911–920, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Felix Stahlberg and Bill Byrne. 2019. 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)*, pages 3356–3362, Hong Kong, China. Association for Computational Linguistics.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook AI's WMT21 news translation task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 205–215, Online. Association for Computational Linguistics.

Jannis Vamvas and Rico Sennrich. 2024. Linear-time minimum bayes risk decoding with reference aggregation. *arXiv preprint arXiv:2402.04251*.

Sean Welleck, Ilia Kulikov, Jaedeok Kim, Richard Yuanzhe Pang, and Kyunghyun Cho. 2020. 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)*, pages 5553–5568, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Guangyu Yang, Jinghong Chen, Weizhe Lin, and Bill Byrne. 2023. Direct preference optimization for neural machine translation with minimum bayes risk decoding. *arXiv preprint arXiv:2311.08380*.

A Minimum Bayes Risk Decoding with Reward Model for Alignment

Reference aggregation is not applicable using an utility function that is not aggregatable. In the following experiment, we show an instance of an utility function that is practically useful but not aggregatable.

We evaluate the performance of MBR and its variants on Alpaca Eval dataset (Li et al., 2023b). The task is to generate a response to a human query that follows the human preference. One of the popular decoding strategy for LLMs is best-of-n strategy (Stiennon et al., 2020; Nakano et al., 2022). Best-of-n generates multiple outputs and simply picks the output with the highest reward value according to a reward function R that is trained to predict the human preference:

$$\mathbf{h}^{\text{bon}} = \operatorname*{arg\,max}_{\mathbf{h} \in \mathcal{H}} R(\mathbf{y}). \tag{10}$$

MBR decoding is also shown to be an efficient strategy on text generation tasks using large language models (LLMs) (Li et al., 2024).[3] We compare the performance of epsilon sampling, best-of-n, MBR without using a reward function (Li et al., 2024), MBR with reference aggregation without using a reward function (RA-MBR) (Vamvas and Sennrich, 2024), MBR using a reward function, and AMBR using a reward function. We implement MBR using a reward function as follows:

$$\mathbf{h}^{\text{reward}} = \operatorname*{arg\,max}_{\mathbf{h} \in \mathcal{H}} \frac{1}{|\mathcal{R}|} \sum_{\mathbf{y} \in \mathcal{R}} u(\mathbf{h}, \mathbf{y}) \cdot R(\mathbf{y}). \tag{11}$$

Note that $\mathbf{h}^{\text{reward}}$ is not immediately aggregatable as most of the state-of-the-art reward functions are based on transformer architecture where the input is a sequence of token embeddings instead of a sentence embedding. Thus, RA-MBR is not directly applicable when combined with a reward function. We use Mistral-7B-Instruct-v0.1 (Jiang et al., 2023) as the text generation model. We generate 128 samples with epsilon sampling with $\epsilon = 0.01$ for best-of-n and MBRs. We use sentence BERT (Reimers

[3] MBR decoding is called Sampling-and-voting (Algorithm 1) in (Li et al., 2024).

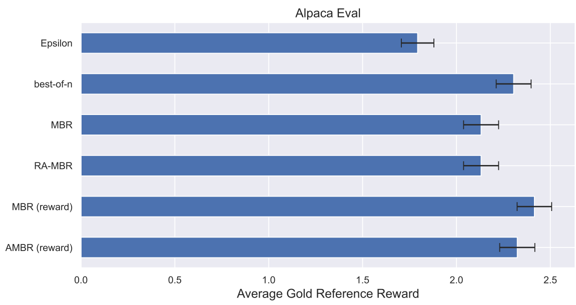


Figure 4: Average reward according to OASST (gold reference reward) on Alpaca Eval dataset.

and Gurevych, 2019) as the utility function u. We compute the embedding of each output using the sentence BERT and compute the cosine similarity of each pair of outputs. We use ALL-MPNET-BASE-V2 model as it has shown to be one of the most effective in sentence embedding tasks.[4] We use SteamSHP-Large as a reward function (Ethayarajh et al., 2022). The budget of AMBR is set to 10000. The output is evaluated using an OASST reward model as the gold reference (Köpf et al., 2023).[5] We use OASST as it is shown to be one of the most accurate reward model in prior work (Touvron et al., 2023; Cui et al., 2023).

Figure 4 is the summary of the reward scores. While MBR and RA-MBR without using a reward model has lower score than best-of-n, MBR with a reward function has higher score than best-of-n. RA-MBR achieves mostly the same score as MBR as the utility function is a cosine similarity of the sentence embedding itself. Thus, linear aggregation of the references results in exactly the mean of the references in the embedding space. Still, because it does not use the reward function, its score is lower than best-of-n and MBRs with reward functions.

The analysis shows that in this setting, non-aggregatable utility function has a potential to achieve higher performance than aggregatable one, and thus reference aggregation is not applicable but AMBR is.

B Formulation of Medoid Identification Problem

We show that the Eq. (7) represents the same class of problem as the standard formulation of the medoid identification problem where $X = Y$ is assumed. Let (d, X, Y) be an instance of generalized medoid identification problem (Eq. 7):

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in X} \sum_{\mathbf{y} \in Y} d(\mathbf{x}, \mathbf{y}).$$

Let $X' = X \cup Y$ and d' as follows:

$$d'(\mathbf{x}, \mathbf{y}) = \begin{cases} \infty & \text{if } \mathbf{x} \notin X \\ 0 & \text{if } \mathbf{x} \in X \wedge \mathbf{y} \notin Y \\ d(\mathbf{x}, \mathbf{y}) & \text{otherwise.} \end{cases} \tag{12}$$

Then, (d', X', X') returns the same solution as (d, X, Y). Thus, Eq. (7) represents the same class of problem as the standard formulation of the medoid identification problem where $X = Y$ is assumed.

C Hyperparameters for Confidence-Based Pruning

The performance of CBP with varying hyperparameters is present in Table 1 (machine translation), Table 2 (text summarization), and Table 3 (image captioning). The average score over five runs is reported. Smaller r_0 and α tend to achieve higher COMET scores when the budget is small and larger r_0 and α achieve higher scores when the budget is large enough.

D Additional Evaluations

We describe additional experiments to evaluate the performance of the MBR decoding algorithms.

D.1 Error Rate on WMT'21 De-En and Ru-En

Figure 5 shows the error ratio for WMT'21 De-En and Ru-En. Interestingly, although AMBR achieves a higher or equivalent COMET score to CBP (Oracle), the error ratio is higher than CBP (Oracle). This suggests that when AMBR fails to find the best hypothesis, it tends to find a hypothesis close to the best hypothesis in quality.

D.2 Evaluation on WMT'21 En-De and En-Ru

To evaluate the performance of AMBR in generating non-English languages, we run experiments on WMT'21 En-De and En-Ru datasets. We use the WMT 21 En-X model for generating the samples (Tran et al., 2021). For CBP, we search over $r_0 \in \{1, 2, 4\}$ and $\alpha \in \{0.8, 0.9, 0.99\}$. Other experimental details are the same as in Section 5.1.

[4] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[5] OpenAssistant/reward-model-deberta-v3-large-v2

#Evaluations		Mean		1/32		1/16		1/8		1/4		1/2	
r_0	α	COMET	Rank	COMET	Rank	COMET	Rank	COMET	Rank	COMET	Rank	COMET	Rank
WMT'21 De-En ($N = 128$)													
1	0.80	64.00	2	62.14	2	64.21	2	64.56	1	64.56	2	64.55	9
1	0.90	63.76	4	61.43	4	63.80	5	64.50	5	64.53	11	64.55	8
1	0.99	63.33	8	60.36	7	63.02	10	64.22	10	64.47	13	64.58	1
2	0.80	63.85	3	61.60	3	64.08	3	64.50	3	64.54	6	64.55	13
2	0.90	63.72	5	61.23	5	63.78	6	64.43	7	64.59	1	64.57	3
2	0.99	63.37	6	60.50	6	63.18	8	64.11	12	64.52	12	64.56	5
4	0.80	63.35	7	59.33	10	63.83	4	64.48	6	64.55	5	64.55	11
4	0.90	63.31	9	59.19	13	63.75	7	64.50	4	64.54	7	64.56	4
4	0.99	63.17	10	59.47	8	63.07	9	64.20	11	64.54	8	64.55	10
8	0.80	63.12	11	59.33	9	62.67	13	64.50	2	64.56	3	64.55	12
8	0.90	63.10	12	59.27	11	62.80	12	64.32	9	64.55	4	64.56	6
8	0.99	63.06	13	59.22	12	62.89	11	64.08	13	64.54	9	64.58	2
AMBR		64.31	1	63.81	1	64.23	1	64.42	8	64.53	10	64.55	7
WMT'21 Ru-En ($N = 128$)													
1	0.80	63.33	2	61.73	2	63.39	3	63.82	3	63.86	5	63.87	10
1	0.90	63.20	4	61.20	4	63.20	5	63.84	2	63.88	3	63.88	4
1	0.99	62.83	6	60.17	7	62.79	8	63.45	13	63.85	6	63.90	1
2	0.80	63.23	3	61.20	3	63.42	2	63.84	1	63.83	8	63.87	11
2	0.90	63.14	5	60.93	5	63.27	4	63.80	5	63.83	9	63.87	8
2	0.99	62.80	7	60.27	6	62.66	10	63.45	12	63.72	13	63.87	9
4	0.80	62.74	9	58.98	12	63.17	6	63.77	6	63.92	1	63.87	7
4	0.90	62.75	8	59.25	10	63.02	7	63.70	9	63.88	2	63.89	3
4	0.99	62.65	10	59.27	9	62.68	9	63.58	10	63.81	10	63.90	2
8	0.80	62.54	13	58.95	13	62.31	12	63.75	8	63.81	11	63.87	12
8	0.90	62.57	12	59.04	11	62.30	13	63.76	7	63.85	7	63.88	6
8	0.99	62.63	11	59.39	8	62.46	11	63.57	11	63.86	4	63.88	5
AMBR		63.61	1	63.12	1	63.51	1	63.80	4	63.80	12	63.80	13
WMT'21 De-En ($N = 64$)													
1	0.80	61.06	2	52.54	2	61.35	2	63.34	3	64.01	2	64.04	11
1	0.90	60.68	3	51.76	3	60.52	5	63.17	5	63.90	8	64.07	2
1	0.99	59.57	9	47.93	12	59.48	6	62.67	9	63.70	12	64.07	3
2	0.80	60.18	4	48.26	7	61.05	3	63.55	2	63.97	3	64.07	4
2	0.90	59.97	5	48.00	11	60.53	4	63.28	4	63.94	6	64.08	1
2	0.99	59.59	8	48.26	8	59.13	7	62.79	8	63.72	11	64.04	10
4	0.80	59.67	7	48.43	5	58.79	9	63.16	6	63.94	5	64.05	7
4	0.90	59.69	6	48.65	4	58.64	13	63.13	7	63.96	4	64.06	5
4	0.99	59.40	11	48.11	9	58.65	11	62.43	10	63.77	10	64.03	12
8	0.80	59.43	10	48.28	6	58.69	10	62.26	11	63.90	9	64.04	9
8	0.90	59.35	12	48.02	10	58.64	12	62.10	13	63.90	7	64.06	6
8	0.99	59.25	13	47.24	13	59.06	8	62.24	12	63.65	13	64.05	8
AMBR		63.29	1	61.49	1	63.10	1	63.80	1	64.02	1	64.03	13
WMT'21 Ru-En ($N = 64$)													
1	0.80	60.63	2	53.30	2	60.65	3	62.79	3	63.18	4	63.23	5
1	0.90	60.44	3	52.63	3	60.50	4	62.70	5	63.13	7	63.24	2
1	0.99	59.13	10	48.31	11	58.99	7	62.12	9	62.99	13	63.21	11
2	0.80	59.68	4	48.00	13	60.93	2	63.02	1	63.19	3	63.24	3
2	0.90	59.58	5	48.58	7	60.18	5	62.68	6	63.23	1	63.22	9
2	0.99	59.20	8	48.40	9	59.31	6	62.07	10	63.01	12	63.22	10
4	0.80	59.26	7	48.32	10	58.82	9	62.72	4	63.20	2	63.23	4
4	0.90	59.27	6	48.70	4	58.87	8	62.45	7	63.09	10	63.23	6
4	0.99	59.17	9	48.65	5	58.71	11	62.15	8	63.10	9	63.22	8
8	0.80	59.03	13	48.19	12	58.65	13	61.92	11	63.14	6	63.25	1
8	0.90	59.11	11	48.61	6	58.78	10	61.82	13	63.11	8	63.21	12
8	0.99	59.06	12	48.52	8	58.70	12	61.87	12	63.02	11	63.20	13
AMBR		62.53	1	60.85	1	62.46	1	62.96	2	63.17	5	63.22	7

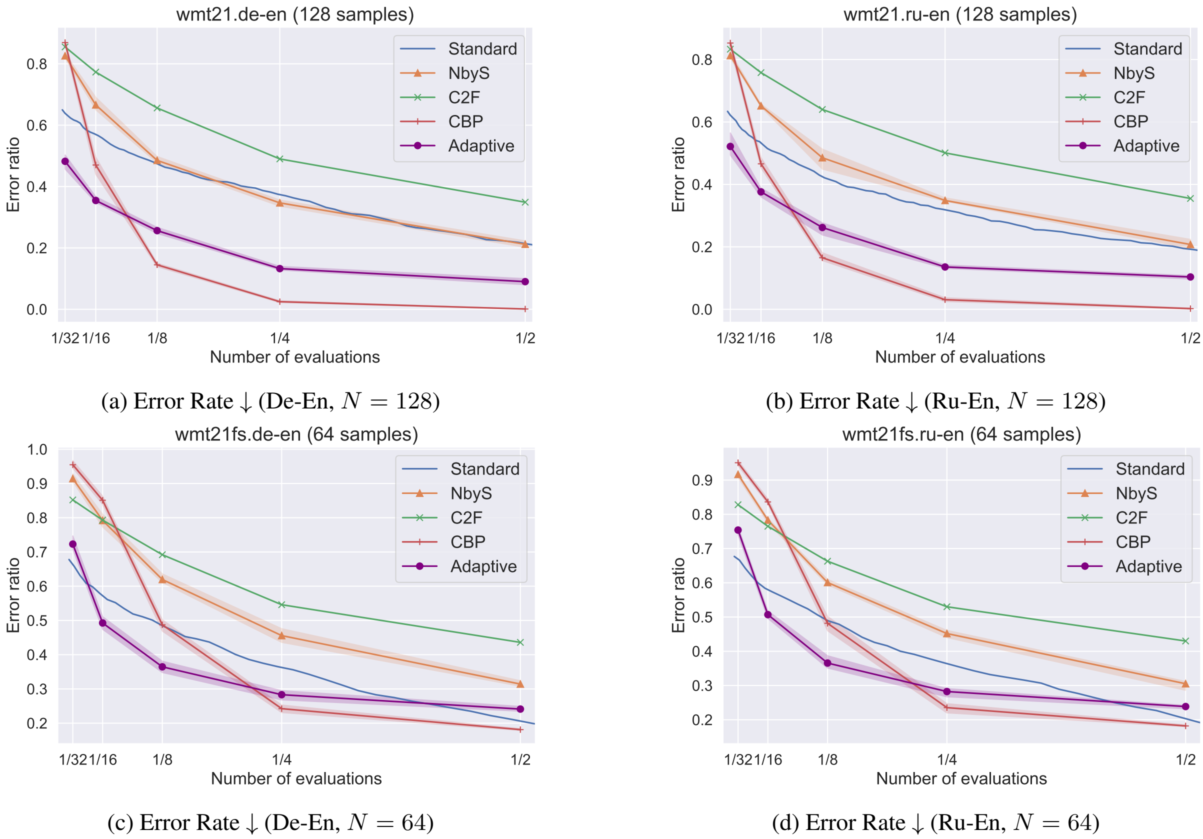
Table 1: Evaluation of confidence-based pruning (CBP) with varying hyperparameters. r_0 is the number of references at the first iteration. α is the threshold of the win rate on pruning. The average COMET-20 score over five runs is reported. Rank denotes the rank of the average COMET-20 score over a set of runs of CBP and AMBR. Mean column reports the average COMET score over 1/32, 1/16, 1/8, 1/4, 1/2.

#Evaluations		Mean		1/32		1/16		1/8		1/4		1/2	
r_0	α	InfoLM	Rank	InfoLM	Rank	InfoLM	Rank	InfoLM	Rank	InfoLM	Rank	InfoLM	Rank
SAMSum ($N = 64$)													
1	0.80	1.864	4	1.982	13	1.896	2	1.850	4	1.802	2	1.792	3
1	0.90	1.868	8	1.976	12	1.902	4	1.856	7	1.812	7	1.794	8
1	0.99	1.870	10	1.960	8	1.909	5	1.863	11	1.821	11	1.798	11
2	0.80	1.860	2	1.955	2	1.902	3	1.845	2	1.803	3	1.793	6
2	0.90	1.866	6	1.958	5	1.909	6	1.856	6	1.813	8	1.793	5
2	0.99	1.871	12	1.960	9	1.916	11	1.858	8	1.821	12	1.797	10
4	0.80	1.864	3	1.961	10	1.915	10	1.847	3	1.806	4	1.793	4
4	0.90	1.865	5	1.956	3	1.912	7	1.854	5	1.811	6	1.793	7
4	0.99	1.872	13	1.957	4	1.922	13	1.860	10	1.822	13	1.801	13
8	0.80	1.867	7	1.961	11	1.916	12	1.859	9	1.809	5	1.792	1
8	0.90	1.869	9	1.958	6	1.913	9	1.864	13	1.816	9	1.795	9
8	0.99	1.870	11	1.959	7	1.912	8	1.864	12	1.818	10	1.799	12
AMBR		1.823	1	1.902	1	1.820	1	1.802	1	1.796	1	1.792	2
XSum ($N = 64$)													
1	0.80	1.954	3	2.069	13	1.997	3	1.935	2	1.889	2	1.880	4
1	0.90	1.961	8	2.066	12	1.998	4	1.952	7	1.904	7	1.882	8
1	0.99	1.964	12	2.057	10	2.006	9	1.961	13	1.910	9	1.888	13
2	0.80	1.952	2	2.056	9	1.993	2	1.943	3	1.891	3	1.878	1
2	0.90	1.959	6	2.052	3	2.003	5	1.950	5	1.909	8	1.881	6
2	0.99	1.963	11	2.053	8	2.008	12	1.955	10	1.912	10	1.886	11
4	0.80	1.956	4	2.057	11	2.005	8	1.943	4	1.893	4	1.880	3
4	0.90	1.960	7	2.053	7	2.008	11	1.954	8	1.903	6	1.882	7
4	0.99	1.963	10	2.052	5	2.006	10	1.951	6	1.920	13	1.885	10
8	0.80	1.957	5	2.052	2	2.004	6	1.954	9	1.896	5	1.879	2
8	0.90	1.962	9	2.052	4	2.004	7	1.956	11	1.914	11	1.883	9
8	0.99	1.965	13	2.053	6	2.012	13	1.960	12	1.915	12	1.886	12
AMBR		1.913	1	1.990	1	1.913	1	1.892	1	1.886	1	1.881	5

Table 2: Evaluation of confidence-based pruning (CBP) with varying hyperparameters on SAMSum and XSum. r_0 is the number of references at the first iteration. α is the threshold of the win rate on pruning. The average InfoLM over five runs is reported. Rank denotes the rank of the average score over a set of runs of CBP and AMBR. Mean column reports the average InfoLM score over 1/32, 1/16, 1/8, 1/4, 1/2.

#Evaluations		Mean		1/32		1/16		1/8		1/4		1/2	
r_0	α	RCLIP	Rank	RCLIP	Rank	RCLIP	Rank	RCLIP	Rank	RCLIP	Rank	RCLIP	Rank
MS COCO ($N = 64$)													
1	0.80	39.27	3	38.09	13	39.10	2	39.60	2	39.76	2	39.81	4
1	0.90	39.22	7	38.10	12	38.99	5	39.49	7	39.72	8	39.81	2
1	0.99	39.21	10	38.28	5	38.86	10	39.44	11	39.69	11	39.78	12
2	0.80	39.29	2	38.23	8	39.08	3	39.58	3	39.75	4	39.81	3
2	0.90	39.26	4	38.23	10	39.01	4	39.56	5	39.73	7	39.80	6
2	0.99	39.22	8	38.33	2	38.89	6	39.42	13	39.67	12	39.78	11
4	0.80	39.26	5	38.29	3	38.88	7	39.56	4	39.75	3	39.81	1
4	0.90	39.24	6	38.28	4	38.88	8	39.52	6	39.74	6	39.80	7
4	0.99	39.19	12	38.23	9	38.81	13	39.46	8	39.69	10	39.76	13
8	0.80	39.21	9	38.21	11	38.86	9	39.45	9	39.75	5	39.80	9
8	0.90	39.20	11	38.24	7	38.84	11	39.44	10	39.69	9	39.81	5
8	0.99	39.19	13	38.25	6	38.82	12	39.42	12	39.65	13	39.79	10
AMBR		39.65	1	39.33	1	39.61	1	39.74	1	39.78	1	39.80	8

Table 3: Evaluation of confidence-based pruning (CBP) with varying hyperparameters on MS COCO. r_0 is the number of references at the first iteration. α is the threshold of the win rate on pruning. The average RefCLIPScore (RCLIP) over five runs is reported. Rank denotes the rank of the average score over a set of runs of CBP and AMBR. Mean column reports the average RCLIP score over 1/32, 1/16, 1/8, 1/4, 1/2.



(a) Error Rate ↓ (De-En, $N = 128$)

(b) Error Rate ↓ (Ru-En, $N = 128$)

(c) Error Rate ↓ (De-En, $N = 64$)

(d) Error Rate ↓ (Ru-En, $N = 64$)

Figure 5: The error rate on WMT'21 De-En and Ru-En using the WMT 21 X-En model. The shaded regions show the minimum and the maximum values over five runs. The error rate is the ratio of selecting a hypothesis different from the standard MBR using all samples. The horizontal axis shows the reduction in the number of evaluations compared to the standard MBR with all samples.

Figure 6 reports the COMET scores. AMBR and CBP significantly reduce the number of evaluations compared to standard MBR with a marginal drop in the COMET score. NbyS and C2F are less efficient than AMBR and CBP. The performance of AMBR is roughly on par with CBP. The result of the hyperparameter search for CBP is described in Table 4. The best set of hyperparameters is dependent to the size of the budget.

D.3 Evaluation on M2M100 418M Model

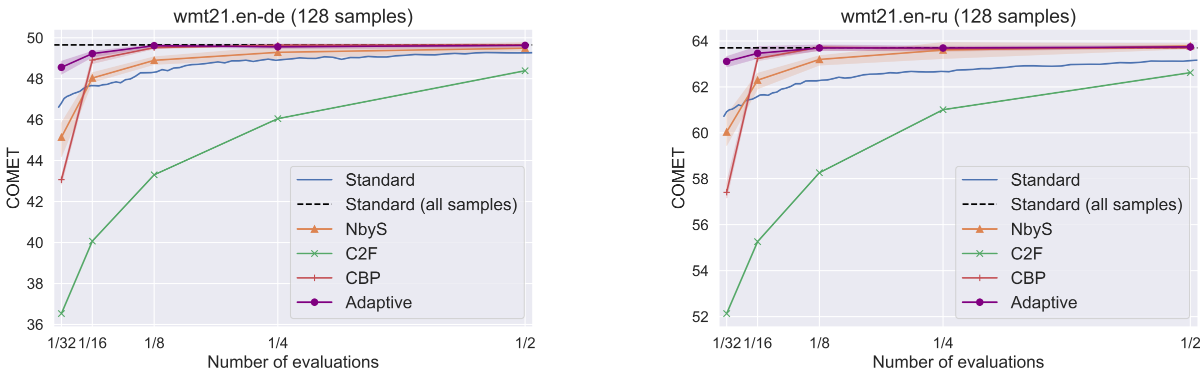
To compare the performance of the methods on a smaller translation model, we evaluate using the M2M100 418M model. Figure 7 shows the results. Overall, we observe qualitatively the same results as using the WMT 21 En-X model (4.7B). AMBR and CBP significantly reduce the number of evaluations compared to standard MBR with a marginal drop in the COMET score. NbyS and C2F are less efficient than AMBR and CBP in WMT'21 tasks. The performance of AMBR is on par with CBP with hyperparameters set by Oracle.

D.4 Scaling with the Number of Samples on Ru-En

Figure 8 shows the result on WMT'21 Ru-En with varying sample sizes with a fixed evaluation budget on the M2M100 418M model. We observe the same trends as in WMT'21 De-En (Figure 2). AMBR scales with the number of samples if there is enough evaluation budget.

E Pretrained Models used in the Experiments

We list the pretrained models we used in the experiments in Table 5.



(a) COMET-20 ↑ (En-De, $N = 128$)

(b) COMET-20 ↑ (En-Ru, $N = 128$)

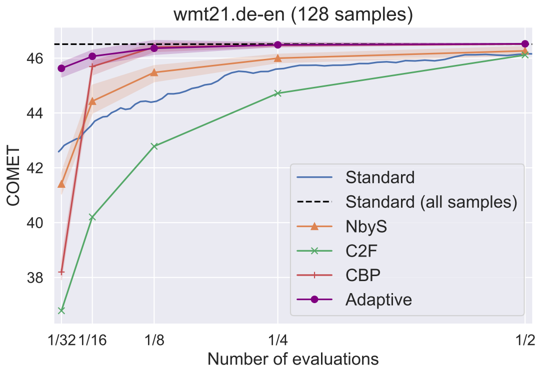
Figure 6: COMET-20 score and error rate on WMT'21 En-De and En-Ru using WMT 21 En-X model (4.7B). The shaded regions show the minimum and the maximum values over five runs. The error rate is the ratio of selecting a hypothesis different from the standard MBR using all 128 samples. The horizontal axis shows the reduction in the number of evaluations compared to the standard MBR with all 128 samples.

#Evaluations		Mean		1/32		1/16		1/8		1/4		1/2	
r_0	α	COMET	Rank	COMET	Rank	COMET	Rank	COMET	Rank	COMET	Rank	COMET	Rank
WMT'21 En-De ($N = 128$)													
1	0.80	49.12	2	47.33	2	49.45	1	49.57	6	49.59	7	49.65	8
1	0.90	48.88	4	46.44	4	49.09	5	49.58	4	49.65	3	49.65	7
1	0.99	48.41	6	45.17	6	48.45	8	49.22	9	49.55	10	49.65	1
2	0.80	49.01	3	46.88	3	49.26	2	49.58	5	49.66	1	49.65	6
2	0.90	48.83	5	46.29	5	48.98	6	49.61	3	49.62	5	49.65	2
2	0.99	48.36	7	44.90	7	48.27	10	49.38	8	49.58	8	49.65	9
4	0.80	48.23	8	42.98	10	49.23	3	49.63	1	49.65	2	49.65	6
4	0.90	48.16	9	43.06	9	48.91	7	49.52	7	49.64	4	49.65	4
4	0.99	48.04	10	43.32	8	48.42	9	49.22	10	49.60	6	49.65	3
AMBR		49.32	1	48.56	1	49.23	4	49.61	2	49.56	9	49.63	10
WMT'21 En-Ru ($N = 128$)													
1	0.80	63.26	2	61.80	2	63.48	4	63.69	9	63.66	9	63.71	5
1	0.90	63.07	5	60.84	5	63.41	6	63.73	3	63.66	8	63.70	7
1	0.99	62.65	6	59.35	6	62.70	10	63.75	1	63.76	1	63.70	9
2	0.80	63.18	3	61.25	3	63.50	3	63.72	5	63.72	2	63.71	3
2	0.90	63.10	4	60.86	4	63.54	1	63.70	7	63.70	4	63.72	2
2	0.99	62.63	7	59.33	7	62.74	8	63.74	2	63.61	10	63.71	4
4	0.80	62.47	8	57.76	8	63.51	2	63.71	6	63.68	6	63.71	6
4	0.90	62.35	9	57.42	10	63.23	7	63.73	4	63.66	7	63.70	8
4	0.99	62.27	10	57.66	9	62.71	9	63.58	10	63.72	3	63.69	10
AMBR		63.54	1	63.11	1	63.46	5	63.70	8	63.69	5	63.74	1

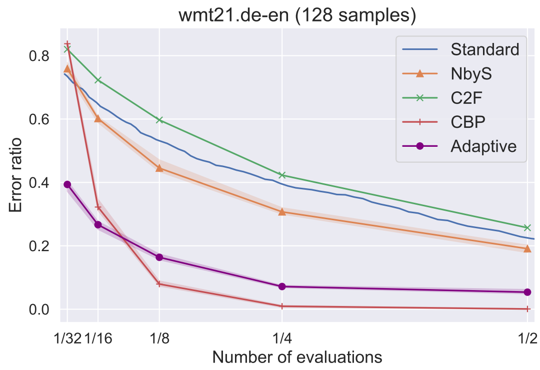
Table 4: Evaluation of confidence-based pruning (CBP) with varying hyperparameters on WMT'21 En-De and En-Ru. r_0 is the number of references at the first iteration. α is the threshold of the win rate on pruning. The average COMET-20 score over five runs is reported. Rank denotes the rank of the average score over a set of runs of CBP and AMBR. Mean column reports the average COMET score over 1/32, 1/16, 1/8, 1/4, 1/2.

WMT'21 (Section 5.1)	Tran et al. (2021) https://huggingface.co/facebook/wmt21-dense-24-wide-x-en
WMT'21 (Section 5.1)	Fan et al. (2021) https://huggingface.co/facebook/m2m100_418M
WMT'21 (Section D.2)	Tran et al. (2021) https://huggingface.co/facebook/wmt21-dense-24-wide-en-x
SAMSum (Section 5.2)	https://huggingface.co/philschmid/bart-large-cnn-samsum
XSum (Section 5.2)	Lewis et al. (2020) https://huggingface.co/facebook/bart-large-xsum
MS COCO (Section 5.3)	Li et al. (2023a) https://huggingface.co/Salesforce/blip2-flan-t5-xl-coco
MS COCO (Section 5.3)	Hessel et al. (2021) (CLIPScore) https://huggingface.co/openai/clip-vit-large-patch1

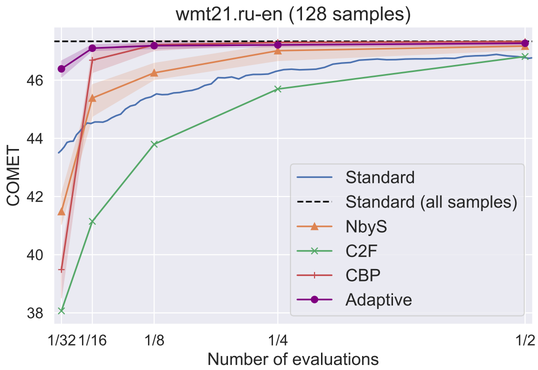
Table 5: List of pretrained models we used in the experiments.



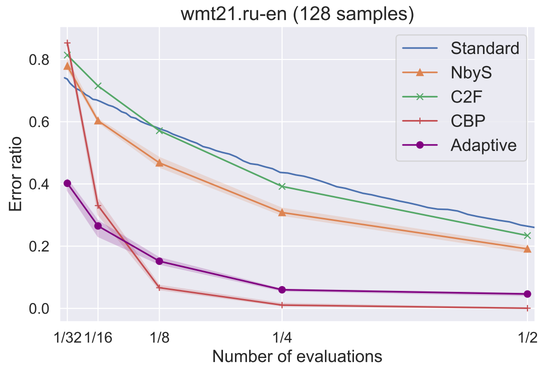
(a) COMET-20 ↑ (De-En, $N = 128$)



(b) Error Rate ↓ (De-En, $N = 128$)

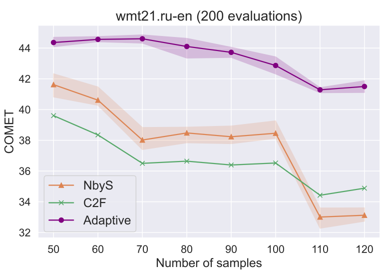


(c) COMET-20 ↑ (Ru-En, $N = 128$)

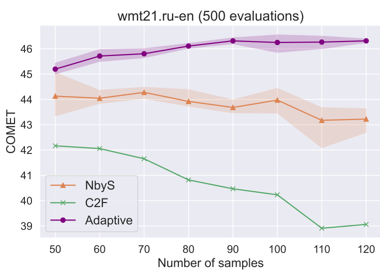


(d) Error Rate ↓ (Ru-En, $N = 128$)

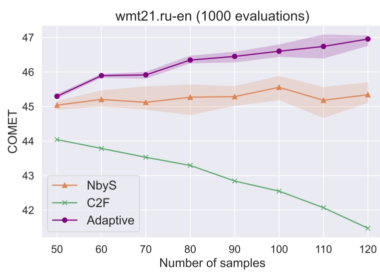
Figure 7: COMET-20 score and error rate on WMT'21 De-En and Ru-En using the M2M100 418M model. The shaded regions show the minimum and the maximum values over five runs. The error rate is the ratio of selecting a hypothesis different from the standard MBR using all 128 samples. The horizontal axis shows the reduction in the number of evaluations compared to the standard MBR with all 128 samples.



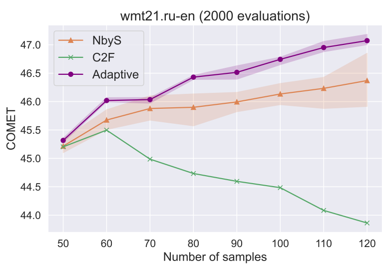
(a) COMET-20 ↑ (Ru-En, 200 evaluations)



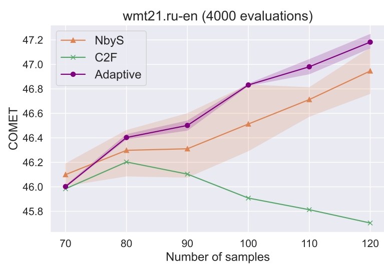
(b) COMET-20 ↑ (Ru-En, 500 evaluations)



(c) COMET-20 ↑ (Ru-En, 1000 evaluations)



(d) COMET-20 ↑ (Ru-En, 2000 evaluations)



(e) COMET-20 ↑ (Ru-En, 4000 evaluations)

Figure 8: COMET-20 score on WMT'21 Ru-En with varying number of samples using M2M100 418M model. The shaded regions show the minimum and the maximum values over five runs.