Bayesian Optimization for Guided Hypothesis Sampling in Minimum Bayes Risk Decoding

Batu El
University of Cambridge
be301@cam.ac.uk

Abstract

Minimum Bayes risk (MBR) decoding is an inference method used in conditional language generation tasks, such as neural machine translation. Unlike maximum a posteriori (MAP) decoding, which selects the hypothesis with the highest probability under the model's distribution, MBR identifies the hypothesis that is most representative of the model's distribution as a whole. This method has been observed to improve model performance in various tasks, especially when combined with neural-based utility functions. However, MBR decoding remains impractical for most use cases due to its quadratic computational complexity. In this paper, we explore guided hypothesis sampling with Bayesian optimization to address the computational challenges of MBR decoding. Particularly, we demonstrate that guided hypothesis performs better than random hypothesis sampling for a fixed number of calls to the utility function. Our investigation of the distribution of hypothesis translations reveals that the hypotheses are often clustered in distinct regions in the embedding space, with observable relationships between their location and MBR score. We demonstrate the merits of our approach with neural machine translation experiments on two language pairs (DE-EN & TR-EN), using chrF++ and BLEURT as evaluation metrics and MBR utility functions.[1][2]

1 Introduction

Decoding—the task of transforming the information stored in the model parameters to an output for a given input—is a key design decision in language generation tasks, such as neural machine translation (NMT). Typically, identifying the sequence with the highest likelihood under the model distribution, i.e., maximum a posteriori (MAP) decoding, is assumed as the decoding objective. Since MAP decoding is generally intractable, beam search has been widely used as an approximation (Wu et al., 2016; Ott et al., 2019).

However, recent findings in NMT have demonstrated that MAP decoding and its approximations have critical limitations. Beam search is observed to underestimate the length of the target sequences (Sountsov and Sarawagi, 2016), and better approximations of MAP objective with larger beam sizes hurt model performance (Koehn and Knowles, 2017). Moreover, in certain experimental settings, the global MAP solution is shown to be the empty sequence (Stahlberg and Byrne, 2019). Notably, beyond a certain likelihood value, the likelihood has been observed to negatively correlate with translation quality (Ott et al., 2018), and translations generated by beam search deviate from the original dataset statistics (Eikema and Aziz, 2020).

Importantly, Eikema and Aziz (2020) have demonstrated that the most likely translations account for a small probability mass under the model distribution since the set of likely translations is large and their probability distribution is relatively flat. Consequently, the goal of decoding has shifted from merely searching for the most likely translation, y^{MAP}, to identifying a translation that best represents the model's distribution holistically, y^{MBR} (Eikema and Aziz, 2020).

This objective is captured by minimum Bayes risk (MBR) decoding (Bickel and Doksum, 1977; Goel and Byrne, 2000) that aims to find the sequence with the maximum expected utility—or, equivalently, minimum expected risk—under the model distribution. Like MAP decoding, MBR is also intractable; therefore, sampling-based approximations of MBR are used in practice (Eikema and Aziz, 2020). The approximation process involves stochastically sampling a set of translations (*psuedo-references*) to serve as a proxy for the model's distribution and comparing the candidate

[1]The source code for this project and a link to our dataset can be found at https://github.com/batu-el/cam.ac.l101.

[2]Word Count: 4808.

translations (*hypotheses*) against the set of psuedo-references to identify the candidate translation that best represents the model's distribution. The evaluations are done using the MBR utility function, a textual similarity metric that measures the semantic or lexical alignment between a pair of sentences.

MBR decoding gives competitive results, outperforming beam search in language generation tasks in general (Suzgun et al., 2023) and NMT in particular (Freitag et al., 2022a). However, despite its competitive performance, MBR decoding remains impractical for most use cases due to its quadratic computational complexity. As the size of the hypothesis and pseudo-reference sets increase, the cost of sampling sequences grows linearly while the number of calls to the utility function increases quadratically. The latter makes MBR decoding prohibitively slow with larger hypothesis and pseudo-reference sets and with neural utility metrics, such as COMET (Rei et al., 2022) and BLEURT (Sellam et al., 2020), which catalyze the most significant performance improvements from MBR (Freitag et al., 2022a) but are computationally expensive. This has motivated previous explorations of efficient methods to estimate MBR decoding (Eikema and Aziz, 2022; Fernandes et al., 2022; Finkelstein et al., 2023; Cheng and Vlachos, 2023; Jinnai and Ariu, 2024).

To that end, we investigate guided hypothesis sampling with Bayesian optimization to address the computational challenges of MBR. We explore two approaches—Bayesian optimization and randomized Bayesian optimization—and demonstrate that they perform better than random hypothesis sampling for a fixed number of calls to the utility function. Our analysis of the distribution of hypothesis translations shows that the hypotheses are often clustered in the embedding space, and a hypothesis' location is linked to its MBR score, suggesting that spacial information in translation embeddings can be used for more efficient hypothesis sampling.

In our experiments, we use the multilingual translation model from Fan et al. (2021) and evaluate our approach using chrF++ (Popović, 2015) and BLEURT (Sellam et al., 2020) on two language pairs (German-English and Turkish-English) from Bojar et al. (2018b) in both directions.

2 Minimum Bayes risk decoding

2.1 Preliminaries

The goal of translation is to map an input sequence x in the source language to an output sequence $y = (y_1, y_2, \ldots, y_m)$ in the target language, where y_t represents the t^{th} token in the output sequence. Encoder-decoder models, which are the standard architecture for NMT, are typically trained to predict a probability distribution for the next token, y_t, over the vocabulary given a source sequence x and a prefix to the target sequence $y_{<t} = y = (y_1, y_2, \ldots, y_{t-1})$ (Jurafsky and Martin, 2023, Ch. 9.7 & 13).

At inference time, the target sequences are constructed token by token, autoregressively. At the first generation step, the source sequence x and a beginning of the sentence marker [3] are used to sample the first token of the target sequence. Then, at each generation step, the model estimates a probability distribution over the vocabulary for the next token y_t, conditioned on the source sequence x and the tokens sampled until that time step, $y_{<t} = y = (y_1, y_2, \ldots, y_{t-1})$. From this distribution, the next token, y_t, is sampled. The model continues this conditional generation procedure until the end of the sequence marker is generated or the sequence has reached a maximum length limit. Different sampling approaches can be employed to select the next token y_t using the probability distribution $P(y_t|x, y_{<t})$.[4]

The likelihood of a sequence under the model distribution can be calculated using the chain rule:

$$P(y|x) = P(y_1|x)P(y_2|y_1, x)\ldots P(y_m|y_1, \ldots, y_{m-1}, x)$$

MAP decoding objective aims to find the sequence that is assigned the highest probability under the model distribution:

$$y^{\text{MAP}} = \arg\max_{y \in Y} P(y|x)$$

where Y is the set of all possible translations.

2.2 MBR Decoding

In contrast to MAP decoding, minimum Bayes risk (MBR) decoding aims to find the sequence with maximum expected similarity, or utility, under

[3] This is often accompanied by source and target language markers in multilingual translation models (Ott et al., 2019)

[4] See Section 5.6.

the model distribution (Bickel and Doksum, 1977; Goel and Byrne, 2000):

$$y^{\text{MBR}*} = \arg\max_{y \in Y} \sum_{y' \in Y} U(y, y') P(y'|x) \quad (1)$$

where $P(y'|x)$ is the probability the model assigns to translation y' for the given source sentence x, and the utility function U is a semantic or lexical similarity metric. Hence, y^{MBR} is the sequence that has the highest total similarity to all other possible translations in Y, where the similarity between y^{MBR} and each $y' \in Y$ is weighted by y''s probability under the model. Since evaluating the utility function for every element of the set of all possible translations, Y, is not feasible, sampling-based approximations of MBR are used in practice.

Eikema and Aziz (2020) proposed obtaining estimates of expected utility via Monte Carlo sampling. This can be achieved by firstly sampling[5] two sets of translations, hypotheses H and psuedo-references R, from the model. The set H contains the potential candidates for the output of the model, which are evaluated against the sequences in R. Hence, Equation 1 is approximated as

$$y^{\text{MBR}*} \approx y^{\text{MBR}} = \arg\max_{h \in H} \sum_{r \in R} U(h, r) \quad (2)$$

The term $P(y'|x)$ from Equation 1 is subsumed under r in Equation 2, because the elements constituting the set R are sampled from the probability distribution $P(.|x)$ defined by the model. Consequently, the frequency and composition of the sequences in R reflect the model's probability distribution.

2.3 Computational Complexity of MBR

The exact solution for Equation 2 requires $|H| \times |R|$ calls to the utility function, i.e., $|R|$ calls to the utility function to calculate the total utility for a given hypothesis from H with respect to all references. Additionally, $|H| + |R|$ calls to the model are needed to sample hypotheses and pseudo-references. Frequently, to reduce the computational cost of sampling sequences, the same set of translations is used as both references and hypotheses ($H = R$). However, $|H| \times |R|$ calls to the utility function make MBR decoding prohibitively expensive with larger hypothesis and/or pseudo-reference sets, especially with computationally expensive neural utility metrics, such as COMET (Rei et al., 2022) and BLEURT (Sellam et al., 2020).

2.4 Connection to Voting Theory

MBR decoding can also be seen as an application of soft majority voting (Suzgun et al., 2023). In this conceptualized political process, each hypothesis translation acts as a candidate that is being voted for, and each pseudo-reference acts as a voter who votes for the candidate translations. In a soft majority vote, each voter gives their preference score for each candidate instead of voting for just one candidate. The pseudo-references express their preference scores for the hypotheses in H via the utility function, preferring candidates with which they share similar words/characters (lexical similarity) or similar meanings (semantic similarity), depending on the choice of the utility function. The preferred candidates are assigned higher utilities. After each pseudo-reference casts their votes for each hypothesis, the hypothesis that achieves the highest total utility score wins the soft majority vote and is elected as the most representative candidate under the model distribution.[6]

3 Methods

3.1 MBR Decoding as Optimization

Equation 2 can be reframed as a maximization problem of the objective function f^{MBR} that maps each hypothesis $h \in H$ to a real number, which we call the MBR score. That is, $f^{\text{MBR}} : h \rightarrow \mathbb{R}$.

$$f^{\text{MBR}}(h) = \sum_{r \in R} U(h, r) \quad (3)$$

In this equation, $f^{\text{MBR}}(h)$ denotes the total utility score that a hypothesis achieves from its pairwise evaluations against all the pseudo references in R. This number is equal to the row sums in the utility matrix in Equation 4, where each row represents a hypothesis, and each column corresponds to a pseudo-reference.

[5] Eikema and Aziz (2020) suggested using unbiased sampling, where at each generation step, the next token is selected based on its probability in the model's distribution.

[6] Additionally, other generation techniques, including self-consistency (Wang et al., 2023), range voting (Borgeaud and Emerson, 2020), output ensembling (Martínez Lorenzo et al., 2023), can be formulated as instances of MBR, which provides theoretical justifications for their performance (Bertsch et al., 2023).

$$\begin{bmatrix} u(h_1, r_1) & u(h_1, r_2) & \dots & u(h_1, r_{|R|}) \\ u(h_2, r_1) & u(h_2, r_2) & \dots & u(h_2, r_{|R|}) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ u(h_{|H|}, r_1) & u(h_{|H|}, r_2) & \dots & u(h_{|H|}, r_{|R|}) \end{bmatrix} \quad (4)$$

As described in Section 2.3, MBR decoding requires $|H| \times |R|$ calls to the utility function. Equivalently, evaluating f^{MBR} costs $|R|$ calls to the utility function, and $|H|$ calls to f^{MBR} is necessary to identify the MBR winner.

We explore the possibility of improving the performance of MBR for a fixed number of calls to f^{MBR} by sequentially sampling hypotheses from H to be evaluated by f^{MBR}. In this process, we use the information from the previously sampled hypotheses to inform which hypothesis to sample next. Thereby, we aim to select $H' \subset H$, such that the MBR winner over H' performs similarly to the MBR winner over H and better than a randomly sampled subset H'' with $|H'| = |H''|$. In our experiments we set $|H| = 128$ and $|H'| = 10$. In particular, we test if we can select the elements in H' in an informed manner so that the MBR winner of H' performs better than the MBR winner in H''.

3.2 Bayesian optimization

Bayesian optimization is a sequential sampling strategy often used to optimize objective functions that are costly to evaluate (Mockus, 2012; Frazier, 2018). It is particularly useful in cases when accessing the derivatives of the function is not feasible, which makes Bayesian optimization an appealing approach for finding the hypothesis for which f^{MBR} takes its maximum value,

$$y^{\text{MBR}} = \arg\max_{h \in H} f^{\text{MBR}}(h) \quad (5)$$

in a minimal number of calls to the function f^{MBR}.

3.2.1 Hypothesis Embeddings

Bayesian optimization requires the input to be represented as points in $\mathbb{R}^n$, where $n \leq 20$ (Frazier, 2018). Therefore, we map every element in $h \in H$ to a k dimensional sentence embedding as described in Section 5.8.[7] We refer to the embedding for a hypothesis h_i as η_i and use $f^{\text{MBR}}(\eta_i)$ to denote $f^{\text{MBR}}(h_i)$.

[7]We use $k = 10$.

3.2.2 Normalizing MBR Scores

As a pre-processing step, we normalize $f^{\text{MBR}}(\eta)$ by dividing its output by $|R|$ to calculate the average MBR score for a candidate h instead of the total score (see Equation 3). This results in values of $f^{\text{MBR}}(\eta)$ that are centered around the mean of our utility function, which we subtract from all outputs to center $f^{\text{MBR}}(\eta)$ around 0.[8]

3.2.3 Gaussian Process Regression

We use Gaussian process regression (GPR) as a surrogate function to model the objective function $f^{\text{MBR}}(\eta)$. GPR is a kernel-based regression model that provides uncertainty estimates for all the points in the space along with predictions (Rasmussen and Williams, 2005, Chapter 2). These uncertainty estimates are particularly useful for our sequential function evaluations when selecting which points to evaluate next (See Section 3.2.4).

Rasmussen and Williams (2005, Definition 2.1) define a Gaussian process as a collection of random variables, any finite number of which have a joint Gaussian distribution. For our problem, these random variables are the values of the function $f^{\text{MBR}}(\eta)$.

Consider a dataset $(\mathcal{H}, \Upsilon)$ where $\mathcal{H}$ is a matrix that contains translation embeddings for a subset of H, and Υ is a vector that contains corresponding MBR scores for the hypotheses in $\mathcal{H}$. Hence, $(\mathcal{H}, \Upsilon)$ contains pairs (η_i, υ_i), where $\upsilon_i = f^{\text{MBR}}(\eta_i) = f^{\text{MBR}*}(\eta_i) + \epsilon_i$ with noise $\epsilon_i \sim N(0, \sigma_\epsilon^2)$. In this expression, $f^{\text{MBR}*}$ is the true MBR score function

$$f^{\text{MBR}*}(\eta_i) = \sum_{y' \in Y} U(h_i, y') P(y'|x) \quad (6)$$

where η_i is the translation embedding for h_i.[9]

GPR assumes that υ are drawn from a multivariate normal distribution characterized by a mean function and a covariance (kernel) function, $\mathbf{K}(.,.)$ (Rasmussen and Williams, 2005, Ch. 2.2).

[8]A reasonable estimate for the mean is 0.50 for BLEURT-20 scores (the values range roughly between 0 and 1) and 50 for chrF++ scores (the values range between 0 and 100) in most examples. We use these values for normalization in all examples.

[9]Theoretically, our observations $\upsilon = f^{\text{MBR}}(\eta)$ are noisy samples from the distribution of true MBR scores, which justifies the term ϵ_i. This is because our MBR calculation via the sampling-based approximation (Equation 2) scores every hypothesis using a randomly sampled subset R of Y under the model's probability distribution instead of using all elements of Y weighted by the corresponding probabilities.

For a new data point η_t, the joint distribution of the observed values of υ, Υ, and $f^{\text{MBR}}(\eta_t) = \upsilon_t$ is

$$\begin{bmatrix} \Upsilon \\ \upsilon_t \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K}(\mathcal{H},\mathcal{H}) + \sigma_\epsilon^2 I & \mathbf{K}(\mathcal{H},\eta_t) \\ \mathbf{K}(\mathcal{H},\eta_t)^T & \mathbf{K}(\eta_t,\eta_t) \end{bmatrix}\right) \quad (7)$$

(Wang et al., 2022). Therefore, the mean and the variance for υ_t can be estimated using the multivariate normal conditional distribution $P(\upsilon_t|\eta_t, \mathcal{H}, \Upsilon) \sim \mathcal{N}(\mu(\eta_t), \sigma(\eta_t))$ (Rasmussen and Williams, 2005, Ch. 2.2). For more compact notation, we let $\mathbf{K}_\mathcal{H} = \mathbf{K}(\mathcal{H},\mathcal{H}) + \sigma_\epsilon^2 I$.

$$\mu(\eta_t) = \mathbf{K}(\mathcal{H},\eta_t)^T \mathbf{K}_\mathcal{H}^{-1} \Upsilon \quad (8)$$

And $\sigma(\eta_t) =$

$$\mathbf{K}(\eta_t,\eta_t) - \mathbf{K}(\mathcal{H},\eta_t)^T \mathbf{K}_\mathcal{H}^{-1} \mathbf{K}(\mathcal{H},\eta_t) \quad (9)$$

For all our reported results, we use the squared exponential kernel.[10] For $\mathbf{K}(\mathcal{H},\mathcal{H})_{ij} = \kappa(\eta_i, \eta_j)$

$$\kappa(\eta_i,\eta_j) = \sigma_f \times \exp\left(-\frac{d(\eta_i,\eta_j)^2}{l^2}\right) \quad (10)$$

where $d(.,.)$ denotes Euclidean distance, and l is the length scale that controls how the similarity between the function values $f^{\text{MBR}}(\eta_i)$ and $f^{\text{MBR}}(\eta_j)$ changes as the distance between points η_i and η_j increases, and σ_f is a scalar that scales all elements of the matrix. This kernel encodes the assumption that the translations that are closer in the embedding space have more similar f^{MBR} function values. We come back to this point in Section 7.

During the training phase of our GPR model, optimum length scale l[11] and constant σ_f[12] parameters are selected as a part of the optimization process.[13] Our model is trained by maximizing[14] log marginal likelihood, $\log p(y|X,\theta) =$

$$-\frac{1}{2}\Upsilon^T \mathbf{K}_\mathcal{H}^{-1}\Upsilon - \frac{1}{2}\log|\mathbf{K}_\mathcal{H}| - \frac{n}{2}\log 2\pi \quad (11)$$

using LBFGS-B (Byrd et al., 1995; Morales and Nocedal, 2011) optimizer (Wang et al., 2022; Rasmussen and Williams, 2005, Algorithm 2.1). In our experiments, we select the hyperparameter σ_ϵ with grid search.[15]

3.2.4 Sequential Hypothesis Sampling

Our guided hypothesis sampling approach uses a sequential sampling algorithm that works as follows. Firstly, we randomly sample a subset H' of size n_{init}[16] from H. Let $\mathcal{H}$ be the set that contains the translation embeddings for the hypotheses in H', and $(\mathcal{H}, \Upsilon)$ contains translation embedding and MBR score pairs, where MBR scores are the normalized row sums of the utility matrix from 4. We fit our GPR model to the points in $\mathcal{H}$ to predict MBR scores in Υ, as explained in Section 3.2.3.

Then, we infer the mean and the standard deviation for all other hypotheses η in our hypothesis set H using equations 8 and 9. To select which hypothesis to sample next, we use expected improvement (Močkus, 1975; Jones et al., 1998) as our acquisition function, which calculates the expected improvement from the maximum observed so far under the model distribution (Frazier, 2018; Wang et al., 2022).

$$EI(\eta) = \mathbb{E}[max(0, \upsilon_t^* - f^{\text{MBR}}(\eta_i))] \quad (12)$$

With the addition of a ξ parameter that encourages exploration for $\xi > 0$, expected improvement can be calculated as

$$(\upsilon_t^* - \xi - \mu(\eta_i))\,\Phi\left(\frac{\upsilon_t^* - \xi - \mu(\eta_i)}{\sigma(\eta_i)}\right)$$
$$+\sigma(x)\phi\left(\frac{\upsilon_t^* - \xi - \mu(x)}{\sigma(\eta_i)}\right)$$

where Φ denotes the normal cumulative distribution function (CDF) and ϕ denotes the normal probability density function (PDF) (Wang et al., 2022).[17]

Bayesian Optimization. In our standard Bayesian optimization approach, at each sampling step, we select the next hypothesis to be the one that has the highest expected improvement. After sampling a new hypothesis, we train our Gaussian process regression model again, including the new hypothesis in our training set, so that the most recently sampled point also informs the selection of the next hypothesis. We continue this sequential

[10] We use the scikit-learn implementation of Gaussian process regression and the squared exponential kernel. This kernel is also known as the Radial Basis Function (RBF) kernel. In our experiments with other kernels, we have not observed any improvements in the performance of our model. Hence, we only report our results with the RBF Kernel.

[11] A length scale is selected between the lower bound 10^-5 and upper bound 10^5.

[12] A scalar is selected between the lower bound 10^-5 and upper bound 10^5.

[13] In scikit-learn implementation, this is achieved by multiplying an RBF kernel with a constant kernel.

[14] in fact, minimizing the negative of

[15] We identify $\sigma_\epsilon = 10^{-5}$ to be the optimum hyperparameter in all experiments. Our search space is $\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$.

[16] We set $n_{\text{init}} = 5$ in our reported experiments.

[17] We use $\xi = 0.01$ for all our reported results.

sampling process until we reach a pre-defined maximum size for $\mathcal{H}$, n_{final}.[18] Finally, we report the MBR winner from $\mathcal{H}$.

Randomized Bayesian Optimization. Acknowledging the success of randomized approaches in language models, we also experiment with another variant of Bayesian optimization, where, instead of choosing the point that takes the highest acquisition value at each sampling step, we prune m hypotheses with the lowest acquisition values, and randomly select a hypothesis from the remaining ones.[19] The rest of the steps are identical to the Bayesian optimization approach.

4 Related Work

There have been various attempts at reducing the complexity of MBR decoding in the recent literature (Eikema and Aziz, 2022; Fernandes et al., 2022; Finkelstein et al., 2023; Cheng and Vlachos, 2023; Jinnai and Ariu, 2024). Eikema and Aziz (2022) have proposed *coarse-to-fine MBR decoding*, where they used proxy utility functions, which correlate with the MBR utility functions but are cheaper to compute, to filter the hypothesis space before calculating the MBR scores for the remaining candidates. They demonstrated that pruning the hypothesis set with a proxy utility function can reduce the cost of MBR without sacrificing translation quality. Similarly, Fernandes et al. (2022) proposed *quality aware decoding*, where they used reference-free quality estimation metrics to select a higher-quality subset of candidates, on which MBR is done. Notably, Finkelstein et al. (2023) have demonstrated that the quality gains of MBR decoding can be distilled at training time by fine-tuning the NMT models on translations selected by MBR decoding.

Recently, Cheng and Vlachos (2023) proposed *confidence-based pruning*, an algorithm that gradually increases the size of the pseudo-reference set while shrinking the hypothesis set. This algorithm significantly reduces the runtime of MBR without any significant impact on performance. Finally, a concurrent work with our paper, Jinnai and Ariu (2024) proposed *approximate minimum Bayes risk*, which uses an efficient approximation algorithm for the medoid identification problem to compute the sample-based MBR objective.

[18] We set $n_{\text{final}} = 10$ in our reported experiments.
[19] We set $m = 100$ in our reported experiments.

5 Experimental Setup

5.1 Data

We use the German-English and German-Turkish test sets from Bojar et al. (2018a) for our experiments. From this parallel corpora, we randomly sample 200 examples from the German-English test set for our preliminary experiments. For our main experiments on both language pairs, we use 1000 randomly sampled examples for each language pair.

5.2 Candidate Translations

In all our experiments, we use the same set of candidate translations as both hypotheses and pseudo-references for sampling efficiency, i.e., $H = R$. We set $|H| = |R| = 128$. Our methods use the full set R in every call to the function f^{MBR}.

5.3 Evaluation Metrics and MBR Utility Functions

BLEURT. Freitag et al. (2022a) have demonstrated that MBR decoding with neural reference-based utility metric BLEURT (Sellam et al., 2020) significantly improves the model performance in human evaluations. Our experiments employ BLEURT as both the evaluation metric and MBR utility function. For our preliminary experiments on 200 examples, where the target language is English, we use the monolingual "bleurt-tiny-128" checkpoint to save computational resources. In our main experiments on 1000 examples, we use the distilled version of BLEURT-20, which is a more accurate multilingual model (Pu et al., 2021). This model is based on RemBERT (Chung et al., 2020), which was trained on a dataset covering 110 languages, including English, German, and Turkish (Xue et al., 2021; Chung et al., 2020, p. 15). We use a distilled model from the "BLEURT-20-D3" checkpoint due to limited computational resources.

chrF++. We also use chrF++ (Popović, 2015) as a lexical evaluation metric and MBR utility function that is faster and simpler, following (Cheng and Vlachos, 2023). We compute chrF++ using SacreBLEU (Post, 2018) with the default settings. We avoid using BLEU (Papineni et al., 2002) following the recommendation of (Freitag et al., 2022b).

5.4 Model

In all our experiments, we use the multilingual encoder-decoder transformer model with 418M parameters from Fan et al. (2020). The encoder of

this model transforms a sequence of tokens and a source language marker into a sequence of embeddings with the same length. The decoder takes this sequence of embeddings and a target language marker and produces the target sentence token by token, as described in Section 2.1.[20] This model is trained on many-to-many parallel corpora for 100 languages (Schwenk et al., 2019; El-Kishky et al., 2019) with label smoothing 0.1 (Szegedy et al., 2015; Pereyra et al., 2017).

5.5 MBR with Label Smoothing

Eikema and Aziz (2020, 2022) have observed that label smoothing compromises the performance of MBR. Similarly, Fernandes et al. (2022) noted that disabling label smoothing improves model performance when selecting the MBR candidates using unbiased sampling. On the other hand, the impact of label smoothing can be offset by appropriate sampling methods. Notably, Fernandes et al. (2022) observed that the degradation in MBR performance can be countered by using nucleus sampling. Similarly, Yan et al. (2023) have shown that this effect can be mitigated by using smaller temperatures in sampling, i.e., $\tau < 1$. Hence, we carefully selected our sampling method with a preliminary experiment.

5.6 Sampling Method

Recently, Freitag et al. (2023) observed that MBR decoding based on epsilon-sampling (Hewitt et al., 2022) outperforms MBR decoding with all other sampling methods they tested, including top-k sampling, nucleus sampling, and ancestral (or unbiased) sampling. Following their observation, we use epsilon sampling to generate our candidate translations.

Our sampling scheme involves two steps: (i) pruning tokens with a probability smaller than a specified threshold ϵ under the model, and (ii) adjusting the probability distribution of the remaining tokens using a temperature parameter τ (Hewitt et al., 2022; Freitag et al., 2023). In their investigation of the impact of ϵ and τ hyper-parameters on the performance of MBR decoding, Freitag et al. (2023) found that for candidate sets of size 128, $\epsilon = 0.02$ and $t = 2.0$ are the optimum hyper-parameters (Freitag et al., 2023, Figure 8). However, their model was trained using maximum likelihood estimation without label smoothing (Freitag et al., 2023, Section 3.2).

In contrast, for our model, the label-smoothing regularization that prevents the model from getting too confident can lead to degenerate translations when combined with a high-temperature scaling ($t = 2.0$) that further flattens the distribution. We observe that this is indeed true in a preliminary experiment.[21] Our findings are described in Table 1. In this experiment, we identified optimum sampling hyper-parameters $\epsilon = 0.02$ and $\tau = 1.0$ for our model and verified that MBR decoding with the optimum hyper-parameters outperforms beam search.

Method	Parameters	CharF++	BLEURT
Beam Search	*beam size* $= 5$	57.11	17.70
MBR_{BLEURT}	$\tau = 0.5, \epsilon = 0.01$	57.17	24.50
	$\tau = 1.0, \epsilon = 0.01$	55.48	28.97
	$\tau = 2.0, \epsilon = 0.01$	44.13	16.54
	$\tau = 0.5, \epsilon = 0.02$	57.16	24.05
	$\tau = 1.0, \epsilon = 0.02$	54.82	**29.90**
	$\tau = 2.0, \epsilon = 0.02$	44.26	16.16
MBR_{chrF++}	$\tau = 0.5, \epsilon = 0.01$	57.57	17.76
	$\tau = 1.0, \epsilon = 0.01$	57.32	14.82
	$\tau = 2.0, \epsilon = 0.01$	50.35	-13.07
	$\tau = 0.5, \epsilon = 0.02$	57.38	17.03
	$\tau = 1.0, \epsilon = 0.02$	**58.35**	16.44
	$\tau = 2.0, \epsilon = 0.02$	50.00	-15.78

Table 1: Epsilon Sampling Optimal Sampling Parameters. Results are calculated on 200 randomly sampled examples from the German-English test set and translated sentences from German to English. Note that the values for the "bleurt-tiny-128" checkpoint of BLEURT, which we used in this preliminary experiment, range between -1 and 1. We scale the BLEURT scores by multiplying by 100 for better readability.

5.7 Generating Translations

To obtain the dataset we use in our main experiment, we translate 1000 sentences for both language pairs in both directions (DE $\rightarrow$ EN, EN $\rightarrow$ DE, EN $\rightarrow$ TR, TR $\rightarrow$ EN), which in total amounts to 4000 source sentences that are translated to the target language. For each of the 4000 sentences, we generate 128 candidate translations.[22] In total, we make $512,000$ calls to the translation model to obtain all candidate translations.

[20] We access the pre-trained transformer via the HuggingFace library from the checkpoint "facebook/m2m100_418M."

[21] More details on this experiment is included in Appendix B.

[22] All 128 translations may not be unique since they are generated via sampling. Appendix B explores this further.

5.8 Translation Embeddings

We compute translation embeddings with the RemBERT model (Chung et al., 2020)[23], which was pre-trained on 110 languages, including English, German, and Turkish (Chung et al., 2020, p. 15), using a masked language modeling objective. Our decision to use RemBERT to generate embeddings is in line with our selection of neural utility metric, BLEURT-20, which is based on RemBERT.

For each sentence, we extract the embeddings from the final hidden state of the model RemBERT model, which assigns a 1152 dimensional vector to each token in the sentence. The tokenization in RemBERT is based on the SentencePiece tokenizer (Kudo and Richardson, 2018). We take the average of these vectors to normalize the representations for different translation lengths and obtain a 1152 dimensional translation embedding for each $h \in H$. Then, we apply principal component analysis (Shlens, 2014) to obtain the first 10 principal components, which we use as 10 dimensional embeddings for the hypothesis translations (unless stated otherwise, $k = 10$ for all our experiments). [24] Note that the principal components are identified independently for each example, i.e., principal component analysis is conducted on 128 elements from H that correspond to the candidate translations for the same source sentence x.

6 Results

Tables 2 and 3 demonstrate our findings for BLEURT and chrF++, respectively. We compare the average MBR utility of the MBR winner ($f^{\text{MBR}}(h)/|R|$), the rank of the MBR winner in H in terms of average MBR utility,[25] and the evaluation score of the MBR winner against the gold reference. Additionally, for each example, we analyze whether our models successfully select an MBR winner with a higher overall rank than the MBR winner identified by the random baseline. We report the percentage of times the models perform better or worse than the random baseline. All the

[23] We use the RemBERT model from HuggingFace with the default configurations. https://huggingface.co/docs/transformers/model_doc/rembert

[24] We observe that the first 10 principal components account for 90% of the variability in the hypotheses embeddings on average.

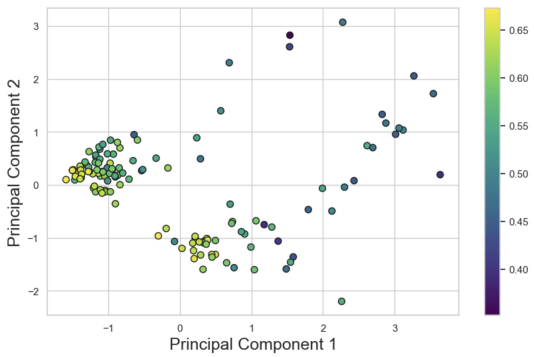
[25] Ranks are represented in ascending order, i.e., rank 128 represents the candidate with highest average MBR utility. Note that full MBR may not achieve 128 rank in all cases due to the presence of duplicates in the set of candidate translations.

Method	DEEN	ENDE	ENTR	TREN
Random Selection - 10 Candidates (Baseline)				
Avg. MBR Utility	71.43	63.98	63.808	67.23
Rank	116.88	117.08	116.997	117.01
BLEURT	65.92	62.089	53.98	61.09
Full MBR - 128 Candidates (Upper Bound)				
Avg. MBR Utility	72.84	66.07	66.24	69.08
Rank	126.9	127.4	127.5	127.4
BLEURT	67.28	64.20	56.14	62.83
Bayesian Optimization - 10 Candidates				
Avg. MBR Utility	71.55	64.19	64.12	67.45
Rank	117.51	118.22	118.55	118.20
BLEURT	66.05	62.20	54.21	61.29
Baseline is Better	40.8%	41.1%	40.1%	39.9%
Model is Better	47.9%	50.8%	**51.9%**	51.5%
Tie	11.2%	8.1%	7.9%	8.5%
Randomized Bayesian Optimization - 10 Candidates				
Avg. MBR Utility	71.64	64.18	64.14	67.50
Rank	**118.68**	**118.51**	**118.96**	**118.86**
BLEURT	66.10	62.28	54.16	61.34
Baseline is Better	37.2%	40.6%	39.4%	39.1%
Model is Better	**49.9%**	**51.0%**	51.7%	**51.8%**
Tie	12.8%	8.5%	8.9%	9.1%

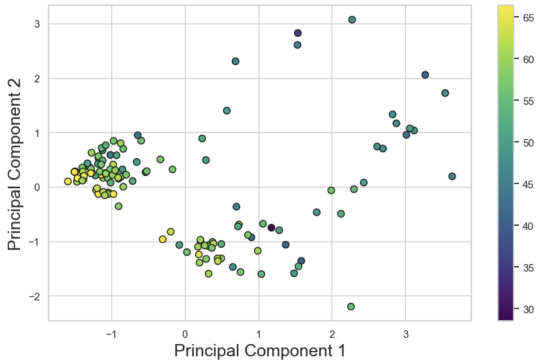
Table 2: Results (BLEURT). BLEURT is used as an evaluation metric and MBR utility function for the results reported in this table. We use the BLEURT-20 checkpoint in the reported results, which takes values roughly between 0 and 1. We scale the BLEURT scores, multiplying by 100 for better readability.

Method	DEEN	ENDE	ENTR	TREN
Random Selection - 10 Candidates (Baseline)				
Avg. MBR Utility	65.33	58.12	56.18	56.8
Rank	116.77	116.92	117.18	117.16
CHRF	56.78	54.56	44.6	48.18
Full MBR - 128 Candidates (Upper Bound)				
Avg. MBR Utility	67.44	60.52	58.56	59.23
Rank	126.6	127.2	127.3	127.2
CHRF	58.29	56.85	46.34	49.96
Bayesian Optimization - 10 Candidates				
Avg. MBR Utility	65.60	58.49	56.48	57.11
Rank	118.03	118.7	118.5	118.40
CHRF	56.89	54.84	44.85	48.48
Baseline is Better	37.7%	38.8%	39.5%	38.9%
Model is Better	50.6%	51.9%	51.5%	**52.2%**
Tie	11.7%	9.3%	8.9%	8.9%
Randomized Bayesian Optimization - 10 Candidates				
Avg. MBR Utility	65.72	58.54	56.59	57.13
Rank	**118.94**	**119.20**	**119.27**	**118.74**
CHRF	57.02	55.00	44.93	48.46
Baseline is Better	35.2%	37.5%	37.9%	38.9%
Model is Better	**50.4%**	**52.4%**	**51.8%**	51.4%
Tie	14.4%	10.1%	10.3%	9.7%

Table 3: Results (chrF++). chrF++ is used as an evaluation metric and MBR utility function for the results reported in this table.



Avg. MBR (BLEURT) Utility



Avg. MBR (CHRF) Utility

Figure 1: Relationship between location in translation embedding space and average MBR utility. DE → EN translations for the source sentence: "Seit dem gescheiterten Coup hat Staatschef Erdogan 169 der 326 Generäle und Admiräle gefeuert." 128 (127 unique) hypotheses are represented as points in the embedding space. The first 2 dimensions of the 10 dimensional translation embeddings are used in the figure. Coloring is done based on normalized f^{MBR} function values. Among the 128 hypotheses, 68 are members of the larger cluster (including one duplicate), which contains hypotheses with PC1 < 0 and PC2 > -0.5.

reported numbers show averaged results from 10 experiments.

Our random selection baseline selects 10 hypotheses randomly from H and evaluates them against all pseudo-references in R to determine the MBR winner.[26] The full MBR method evaluates the complete set of hypotheses in H against all pseudo-references in R to determine the MBR winner. The performance of this method constitutes an upper bound for our approach. Our models, Bayesian optimization and randomized Bayesian optimization, start by randomly selecting 5 candidates from H, then sequentially sample 5 more candidates as described in 3.2.4.

We find that both of our Bayesian optimization approaches perform better than randomly sampling candidates in terms of the average MBR utility and the rank of MBR winner, as well as the evaluation score of the MBR winner with respect to the gold reference. This is true for both utility metrics in all experiments. However, this is a modest performance difference: Our models identify translations with ranks better than or equal to random baseline only in 60-65% of the time.

Moreover, we find that our randomized Bayesian optimization approach further improves the performance over the Bayesian optimization approach in 6 out of 8 experiments. Nevertheless, estimation of MBR scores using only 10 out of 128 hypotheses is not competitive against the full MBR approach, even when the hypotheses are sequentially selected with Bayesian optimization.

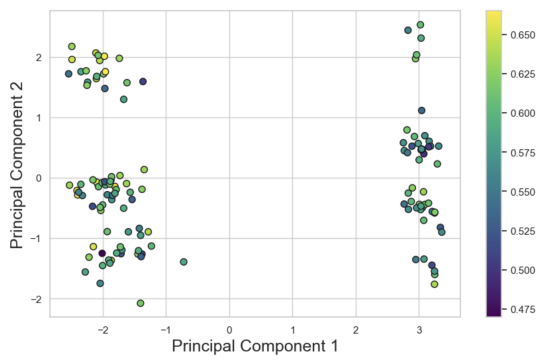
7 Discussion

Our investigation of the distribution of hypothesis translations reveals that the hypotheses are often clustered in distinct regions in the embedding space, with observable relationships between their location and MBR score, verifying that spacial information from translation embeddings can be used for more efficient sampling.

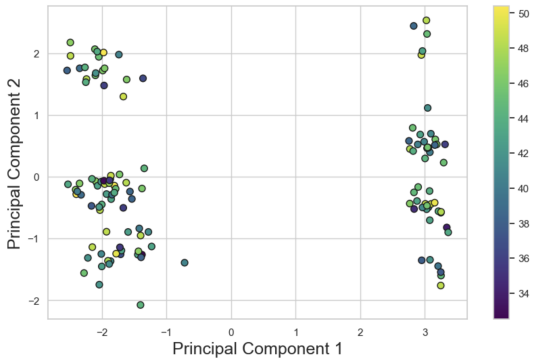
Our choice of kernel encodes the assumption that the translations that are closer in the embedding space have more similar f^{MBR} function values. Through manual inspection of our data, we observe that, indeed, the translations that are closer in the embedding space have more similar values for f^{MBR}, verifying this assumption (see Figures 1,2, and 3). In some cases, we note that the information in only the first two principal components, i.e., the first two dimensions of our translation embeddings, is predictive of the MBR winner. The most clear patterns are observed in DE → EN translations with BLEURT as the utility function. For instance, in the example from Figure 1, the MBR (BLEURT) winner is the hypothesis with the lowest value for the first principal component.[27] Furthermore, we observe that in the cases where strong relations between location in embedding space and MBR score

[26] The analytical solution for the expected rank of our random baseline can also be calculated using order statistics.

[27] MBR(chrF) winner was different for this example, but manual inspection revealed that it was lower quality: "Since the failure of the coup, President Erdogan has *shouted* 169 of the 326 generals and admirals."

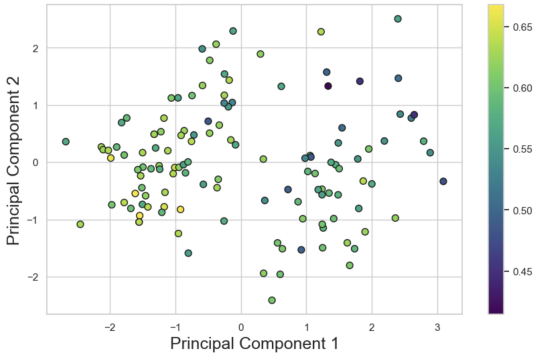


Avg. MBR (BLEURT) Utility

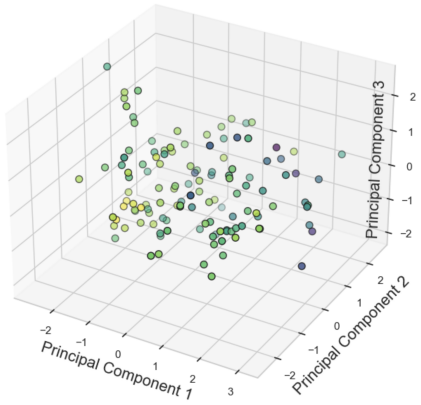


Avg. MBR (CHRF) Utility

Figure 2: Clustering in translation embedding space. DE → EN translations for the source sentence:"In seiner Hast, das Gesetz zu erlassen", habe der Kongress ein paar "verfassungswidrige Vorschriften" hineinverpackt, ätzte er."



Avg. MBR (BLEURT) Utility



Avg. MBR (BLEURT) Utility

Figure 3: Spread out patterns in translation embedding space. TR → EN translations for the source sentence:"Gözünü yüzde 80-90 kaybetme olasığı var."

exist, Bayesian optimization outperforms random baseline by a significant margin, often identifying one of the top-5 candidates (Rank > 123).

Moreover, we note that the hypotheses are often clustered in the embedding space in certain locations. Figure 2 demonstrates this with an example from the DE-EN translation dataset, where two distinct clusters are observable. We note that, in general, hypothesis sets for longer translations form distinct clusters in the embedding space, whereas the embeddings for shorter translations appear to be more dispersed. This phenomenon can also be observed in Figures 1 and 3.

Notably, we observe that while the translations that are closer in the embedding space have more similar f^{MBR} function values in general, this pattern does not hold for all examples in our dataset when embeddings are projected to lower dimensions. We inspect embeddings in 2-dimensional and 3-dimensional space and note that in some cases, the MBR scores for nearby points can vary significantly. This can impact the performance of Bayesian optimization if the same is true in our 10 dimensional embeddings. It is also possible that the randomized Bayesian optimization approach partially addresses this issue, which can justify the resulting performance improvement.

8 Conclusion

In this paper, we proposed two guided hypothesis sampling approaches with Bayesian optimization for MBR decoding. Our results demonstrate that these methods outperform random hypothesis sampling for a fixed number of calls to the utility function. Through our investigation of the distribution of hypothesis translations, we revealed that the hypotheses are often clustered in distinct regions in the embedding space, with observable relationships between their location and MBR score, verifying that spacial information from translation embed-

dings can be used for more efficient hypothesis sampling.

Using spatial information from translation embeddings to inform hypothesis sampling is a promising avenue for more efficient minimum Bayes risk decoding. We believe that further refinements to our methods can enhance the efficiency of sampling-based approximations of MBR decoding.

9 Limitations

We have demonstrated that guided hypothesis sampling can improve performance for a fixed number of calls to the utility function. However, to achieve faster MBR decoding without sacrificing translation quality, one needs to demonstrate that a fixed level of performance can be achieved with fewer calls to the utility function. While these two objectives are aligned, they are not equivalent.

In some cases, we observed examples that are close together in the reduced embedding space can have very different MBR scores. The noise in Gaussian process regression (see the paragraph above Equation 5) can model this phenomenon to a certain extent. Nonetheless, our noise hyperparameter, $\sigma_\epsilon = 10^{-5}$, which is fixed across all experiments, likely does not suffice. We also tried modeling noise for individual hypotheses by using a white kernel; however, our model struggled to converge with more complex kernels. Our implementation with squared exponential kernel also faced convergence issues when dealing with translations to target languages other than English.

Using PCA as a dimensionality reduction technique is far from perfect for our choice of GPR kernel. MBR scores vary the most with respect to the first principal component, which accounts for most of the variability in the translation embeddings. In contrast, the distance metric in our squared exponential kernel treats all dimensions equally when calculating the covariances (See Equation 10), i.e., Euclidean distance. To account for this mismatch, we also experimented with an anisotropic kernel (Rasmussen and Williams, 2005, Section 4.2) that selects different length scales for each of the 10 dimensions. However, this model also faced convergence challenges and has not resulted in any performance improvements.

When selecting the hyperparameter σ_ϵ, we conducted a grid search separately for each language pair and identified the noise level of 10^{-5} to be the optimum hyperparameter. However, we did not conduct an exhaustive grid search for different combinations of kernels and hyperparameters. Our model can potentially benefit from an exhaustive grid search approach.

Finally, our average pooling method (described in Section 5.8) to construct translation embeddings is a simple approach. Works on sentence embeddings, such as SentenceBERT (Reimers and Gurevych, 2019), can inform the construction of better translation embeddings.

References

Amanda Bertsch, Alex Xie, Graham Neubig, and Matthew Gormley. 2023. It's MBR all the way down: Modern generation techniques through the lens of minimum Bayes risk. In *Proceedings of the Big Picture Workshop*, pages 108–122, Singapore, Singapore. Association for Computational Linguistics.

P.J. Bickel and K.A. Doksum. 1977. *Mathematical Statistics: Basic Ideas and Selected Topics*. Prentice Hall.

Ond rej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018a. Findings of the 2018 conference on machine translation (wmt18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 272–307, Belgium, Brussels. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018b. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Sebastian Borgeaud and Guy Emerson. 2020. Leveraging sentence similarity in natural language generation: Improving beam search using range voting. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 97–109, Online. Association for Computational Linguistics.

Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208.

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, Thibault Févry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2020. Rethinking embedding coupling in pre-trained language models.

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.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzman, and Philipp Koehn. 2019. A massive collection of cross-lingual web-document pairs. *arXiv preprint arXiv:1911.06154*.

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. 2020. Beyond english-centric multilingual machine translation.

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).

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, Subhajit Naskar, Mehdi Mirzazadeh, Apurva Shah, and Markus Freitag. 2023. Mbr and qe finetuning: Training-time distillation of the best and most expensive decoding methods.

Peter I. Frazier. 2018. A tutorial on bayesian optimization.

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

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

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022b. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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

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.

Yuu Jinnai and Kaito Ariu. 2024. Hyperparameter-free approach for faster minimum bayes risk decoding.

Donald Jones, Matthias Schonlau, and William Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492.

Dan Jurafsky and James H. Martin. 2023. *Speech and Language Processing (3rd ed. draft)*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

Abelardo Carlos Martínez Lorenzo, Pere Lluís Huguet Cabot, and Roberto Navigli. 2023. AMRs assemble! learning to ensemble with autoregressive models for AMR parsing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1595–1605, Toronto, Canada. Association for Computational Linguistics.

J. Močkus. 1975. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*, pages 400–404, Berlin, Heidelberg. Springer Berlin Heidelberg.

J. Mockus. 2012. *Bayesian Approach to Global Optimization: Theory and Applications*. Mathematics and its Applications. Springer Netherlands.

José Luis Morales and Jorge Nocedal. 2011. Remark on "algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound constrained optimization". *ACM Trans. Math. Softw.*, 38(1).

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation.

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 NAACL-HLT 2019: Demonstrations*.

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.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. 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.

C.E. Rasmussen and C.K.I. Williams. 2005. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning series. MIT Press.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation.

Jonathon Shlens. 2014. A tutorial on principal component analysis.

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

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2023. Follow the wisdom of the crowd: Effective text generation via minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4265–4293, Toronto, Canada. Association for Computational Linguistics.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision.

Xilu Wang, Yaochu Jin, Sebastian Schmitt, and Markus Olhofer. 2022. Recent advances in bayesian optimization.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.

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.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer.

Jianhao Yan, Jin Xu, Fandong Meng, Jie Zhou, and Yue Zhang. 2023. Dc-mbr: Distributional cooling for minimum bayesian risk decoding.

A Hardware

Our experiments with the translation model and the neural evaluation metric are conducted using one NVIDIA A100 40GB GPU, which we accessed through the cloud computing services offered by Google Colab. We ran multiple experiments over the course of 4 weeks.

B Preliminary Experiments

The results in Table 1 are calculated on 200 randomly selected examples from the DE-EN test set from Bojar et al. (2018a). The sentences are translated from German to English (DE $\rightarrow$ EN). 128 candidate translations are generated for each source sentence. We observe that $\tau = 1.0$, $\epsilon = 0.02$ achieves the best performance with respect to BLEURT when BLEURT is used as the MBR utility function. The same is true for chrF++ when it is used as both the evaluation metric and the MBR utility function. Interestingly, $\tau = 0.5$ tends to perform better when the MBR utility metric is different from the evaluation metric. We select $\tau = 1.0$ and $\epsilon = 0.02$ to be the best hyper-parameters for this setting and continue our experiments with translations sampled using these parameters.

Figure 4 shows the frequency of the number of unique candidates in a set of 128 candidates for different temperature parameters. The x-axis represents the number of unique candidates among the 128 sampled candidates, and the y-axis shows the frequency of that number of unique candidates in 200 examples. All three plots are generated using $\epsilon = 0.02$. From top to bottom, the temperature values are $\tau = 0.5, \tau = 1.0, \tau = 2.0$. The plots demonstrate that $\tau = 0.5$ undermines the diversity of the candidates set, while $\tau = 1$ still generates a reasonably diverse pool of candidates.

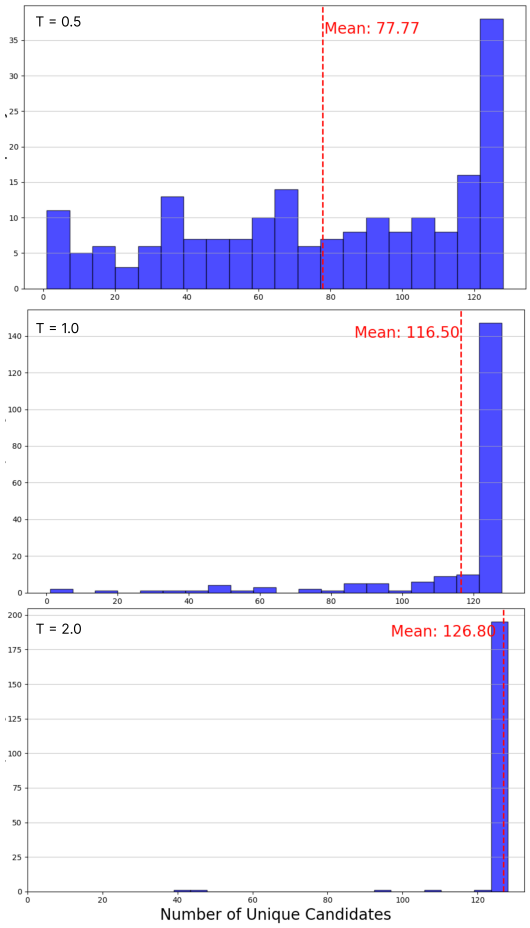


Figure 4: Number of Unique Candidates for Different Temperatures.