MBRS: A Library for Minimum Bayes Risk Decoding

Hiroyuki Deguchi, Yusuke Sakai, Hidetaka Kamigaito, Taro Watanabe

Nara Institute of Science and Technology (NAIST)

{deguchi.hiroyuki.db0, sakai.yusuke.sr9, kamigaito.h, taro}@is.naist.jp

Abstract

Minimum Bayes risk (MBR) decoding is a decision rule of text generation tasks that outperforms conventional maximum a posterior (MAP) decoding using beam search by selecting high-quality outputs based on a utility function rather than those with high-probability. Typically, it finds the most suitable hypothesis from the set of hypotheses under the sampled pseudo-references. MBRS is a library of MBR decoding, which can flexibly combine various metrics, alternative expectation estimations, and algorithmic variants. It is designed with a focus on speed measurement and calling count of code blocks, transparency, reproducibility, and extensibility, which are essential for researchers and developers. We published our MBRS as an MIT-licensed open-source project, and the code is available on GitHub[1,2,3,4].

1 Introduction

Text generation has now become a central topic in natural language processing owing to the success of language models. A typical text generation model generates the high-probability text using beam search and other search algorithms, which relies on maximum a posteriori (MAP) decoding; however, recent studies have demonstrated that high-probability texts are not always high-quality. This phenomenon is known as *beam search curse* and the models sometimes generate pathologically broken texts, e.g., empty sequences, n-gram repetitions, and copies of the inputs (Koehn and Knowles, 2017; Ott et al., 2018; Eikema and Aziz, 2020).

Minimum Bayes risk (MBR) decoding addresses the problems of MAP decoding by determining outputs according to the decision rule based on quality
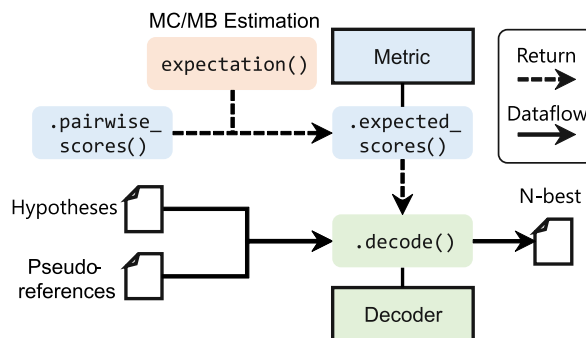

Figure 1: Workflow overview of MBRS. Decoder class gets a candidate list and outputs high-quality hypotheses. It internally calls the `expected_scores()` method implemented in `Metric` classes that calculate the expected scores for each hypothesis. The expected scores are estimated with Monte Carlo (MC) or the model-based (MB) method. In the figure, the colored boxes without border lines denote the functions or methods and the square boxes with border lines denote the abstract classes. Methods that have the same color as a class indicate that they belong to the class.

or preference rather than probability. Its effectiveness has been confirmed in statistical automatic speech recognition (Goel and Byrne, 2000) and statistical machine translation (Kumar and Byrne, 2004), and it has been applied to neural text generation, especially neural machine translation (Eikema and Aziz, 2020; Müller and Sennrich, 2021). Although many variants of MBR decoding have been proposed, there is no common library where we can try the latest algorithms and compare them systematically. It is clearly essential for both researchers and developers to establish a library for the further improvement in MBR decoding.

We introduce MBRS, a library of MBR decoding, which implements various metrics and algorithms with many useful features for comparisons and the development of new methods. Figure 1 shows the workflow overview of MBRS. In the figure, each module, i.e., "Metric" and "Decoder", is easily ex-

---

[1]GitHub: https://github.com/naist-nlp/mbrs
[2]Read the Docs: https://mbrs.readthedocs.io/en/latest/index.html
[3]YouTube: https://youtu.be/4qeHpg4PTn0
[4]HP: https://naist-nlp.github.io/mbrs-web

tensible. In addition, we carefully designed MBRS to ensure transparency and reproducibility. For instance, the profiler, which automatically measures how much time a code block consumes and how many times it is called, and the typed dataclass, which returns helpful metadata for analyses, are implemented.

Experiments on the WMT'22 En↔De general translation tasks demonstrated that our MBRS can not only systematically compare various metrics and algorithms but also present the runtime statistics and supplementary information of output texts.

2 Background

2.1 Minimum Bayes risk (MBR) decoding

The goal of MBR decoding is to find the high-quality output text from a candidate list for a given input text. We denote input and output texts as sequences of tokens $\boldsymbol{x} \in \mathcal{V}_X^*$ and $\boldsymbol{y} \in \mathcal{V}_Y^*$, respectively. Note that $\mathcal{V}_X^*$ and $\mathcal{V}_Y^*$ are the Kleene closures of input and output vocabularies, respectively. Since $\boldsymbol{y}$ is searched from the output space $\mathcal{Y} \coloneqq \{\boldsymbol{y} \mid \boldsymbol{y} \in \mathcal{V}_Y^*\}$, which is an infinite set, we usually find the optimal solution with some approximations.

The most commonly used strategy is maximum a posteriori (MAP) decoding using beam search. The solution of MAP decoding $\boldsymbol{y}^{\mathrm{MAP}_\theta} \in \mathcal{Y}$ is obtained according to its output probability which is calculated by a text generation model θ as follows:

$$\boldsymbol{y}^{\mathrm{MAP}_\theta} = \underset{\boldsymbol{y} \in \mathcal{Y}}{\operatorname{argmax}}\, p(\boldsymbol{y}|\boldsymbol{x};\theta). \quad (1)$$

Recent studies have demonstrated that this high-probability sequence $\boldsymbol{y}^{\mathrm{MAP}_\theta}$ is not always high-quality. Specifically, as the beam size increases, a sequence with higher probability is obtained; however, such sequences hurt translation performance, a phenomenon known as *beam search curse*. This often results in generating pathological sequences, e.g., empty sequences, n-gram repetitions, and copies of the input sequence (Koehn and Knowles, 2017; Ott et al., 2018; Eikema and Aziz, 2020).

In contrast, minimum Bayes risk (MBR) decoding, an alternative strategy, is known to be effective in avoiding the problems of MAP decoding (Eikema and Aziz, 2020). It finds the high-quality text from the given candidate list $\mathcal{H} \subseteq \mathcal{Y}$ and is formulated as follows:

$$\boldsymbol{y}^{\mathrm{MBR}_{\mathrm{true}}} = \underset{\boldsymbol{h} \in \mathcal{H}}{\operatorname{argmax}}\, \underset{\boldsymbol{y} \sim \mathrm{Pr}(\boldsymbol{y}|\boldsymbol{x})}{\mathbb{E}}[u(\boldsymbol{h}, \boldsymbol{y})], \quad (2)$$

where $\mathrm{Pr}(\boldsymbol{y}|\boldsymbol{x})$ is the true probability of $\boldsymbol{y}$ being generated from $\boldsymbol{x}$. $u \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is the utility function that measures the quality or preference of generated text under the given reference text, which is defined as $\boldsymbol{h} \succeq \boldsymbol{h}' \iff u(\boldsymbol{h}, \boldsymbol{r}) \geq u(\boldsymbol{h}', \boldsymbol{r})$ where $\succeq$ denotes the preference relation.[5] Since the true probability $\mathrm{Pr}(\boldsymbol{y}|\boldsymbol{x})$ is unknown, it is replaced with the output probability calculated by a text generation model θ:

$$\boldsymbol{y}^{\mathrm{MBR}_\theta} = \underset{\boldsymbol{h} \in \mathcal{H}}{\operatorname{argmax}}\, \underset{\boldsymbol{y} \sim p(\boldsymbol{y}|\boldsymbol{x};\theta)}{\mathbb{E}}[u(\boldsymbol{h}, \boldsymbol{y})]. \quad (3)$$

Because enumerating all possible output texts is infeasible, the expected scores calculated by the utility function (expected utility; EU) are estimated using pseudo-references that are sampled according to the output probabilities.[6] Let $\hat{\mathcal{R}}$ be a bag (a.k.a multiset) of sampled pseudo-references and $\mathcal{R}$ is the support set of $\hat{\mathcal{R}}$, i.e., a set of distinct elements of $\hat{\mathcal{R}}$, they are formulated as follows:

$$\hat{\mathcal{R}} \coloneqq \{\boldsymbol{y}_i \in \mathcal{Y} \mid \boldsymbol{y}_i \sim p(\boldsymbol{y}|\boldsymbol{x};\theta)\}_{i=1}^{|\hat{\mathcal{R}}|}, \quad (4)$$

$$\mathcal{R} \coloneqq \mathrm{Supp}(\hat{\mathcal{R}}) = \{\boldsymbol{y} \in \mathcal{Y} \mid m_{\hat{\mathcal{R}}}(\boldsymbol{y}) > 0\}, \quad (5)$$

where $m_{\hat{\mathcal{R}}} \colon \mathcal{Y} \to \mathbb{Z}_+$ is the multiplicity function[7] which returns the number of occurrences in $\hat{\mathcal{R}}$. Typically, the EU is estimated using the Monte Carlo method (Eikema and Aziz, 2022) based on the empirical distribution:

$$p_{\mathrm{MC}}(\boldsymbol{r}|\boldsymbol{x};\hat{\mathcal{R}}) \coloneqq \frac{m_{\hat{\mathcal{R}}}(\boldsymbol{r})}{|\hat{\mathcal{R}}|}, \quad (6)$$

$$\mu_{\mathrm{MC}}(\boldsymbol{h};\hat{\mathcal{R}}) \coloneqq \sum_{\boldsymbol{r} \in \mathrm{Supp}(\hat{\mathcal{R}})} p_{\mathrm{MC}}(\boldsymbol{r}|\boldsymbol{x};\hat{\mathcal{R}}) u(\boldsymbol{h}, \boldsymbol{r}),$$
$$(7)$$

$$\boldsymbol{y}^{\mathrm{MBR}_\theta^{\mathrm{MC}}} = \underset{\boldsymbol{h} \in \mathcal{H}}{\operatorname{argmax}}\, \mu_{\mathrm{MC}}(\boldsymbol{h};\hat{\mathcal{R}}), \quad (8)$$

or using the model-based method (Jinnai et al., 2024b) based on the output probability:

$$\mu_{\mathrm{MB}}(\boldsymbol{h};\mathcal{R},\theta) \coloneqq \sum_{\boldsymbol{r} \in \mathcal{R}} p_{\mathrm{MB}}(\boldsymbol{r}|\boldsymbol{x};\mathcal{R},\theta) u(\boldsymbol{h}, \boldsymbol{r}),$$
$$(9)$$

$$\boldsymbol{y}^{\mathrm{MBR}_\theta^{\mathrm{MB}}} = \underset{\boldsymbol{h} \in \mathcal{H}}{\operatorname{argmax}}\, \mu_{\mathrm{MB}}(\boldsymbol{h};\mathcal{R},\theta), \quad (10)$$

---

[5]An evaluation metric that measures the quality of output texts, e.g., BLEU (Papineni et al., 2002) or COMET (Rei et al., 2020, 2022a) in the machine translation task, is often employed for the utility function u.

[6]We call this calculation of the approximated expected score using pseudo-references "expectation estimation".

[7]$m_{\hat{\mathcal{R}}}(\boldsymbol{y}') = 0$ where $\boldsymbol{y}' \notin \mathrm{Supp}(\hat{\mathcal{R}})$.

where p_{MB} is the normalized output probability over a set of pseudo-references, as follows:

$$p_{\mathrm{MB}}(\boldsymbol{r}|\boldsymbol{x};\mathcal{R},\theta) := \frac{p(\boldsymbol{r}|\boldsymbol{x};\theta)}{\sum_{\boldsymbol{r}'\in\mathcal{R}} p(\boldsymbol{r}'|\boldsymbol{x};\theta)}. \quad (11)$$

Typically, the hypotheses themselves are regarded as the pseudo-references $\hat{\mathcal{R}}$ or $\mathcal{R}$.

2.2 Variations in MBR decoding algorithms

While MBR decoding can generate higher-quality texts compared with MAP decoding, the EU estimation requires the time complexity of $\mathcal{O}(|\mathcal{H}||\hat{\mathcal{R}}|)$ or $\mathcal{O}(|\mathcal{H}||\mathcal{R}|)$, which is time-consuming. To address the issue, efficient variants of MBR decoding have been proposed. One is the reference aggregation, which approximates the EU using an aggregated representation of pseudo-references in the feature space (RAMBR) (DeNero et al., 2009; Vamvas and Sennrich, 2024). Similarly, Deguchi et al. (2024) cluster pseudo-references in the embedding space and use the centroid representations for the EU estimation (CBMBR). Another approach is hypothesis pruning, which prunes hypotheses that are not likely to be selected (PruneMBR) (Cheng and Vlachos, 2023). Trabelsi et al. (2024) proposed probabilistic MBR (PMBR) that reduces the number of the utility function calls. PMBR first calculates the utility scores using only sampled pairs of a hypothesis and reference instead of all pairs. Then, it approximately computes the hypothesis–pseudo-reference utility score matrix using the low-rank matrix completion (Zachariah et al., 2012).

2.3 Problems of existing implementations

Currently, MBR decoding has drawn attention from research communities, and although various methods have been proposed, there is no common library that includes many of the latest studies, making it hard to compare the quality and speed systematically. MBR-NMT[8] is the original implementation of Monte Carlo estimation (Eikema and Aziz, 2022), but it does not support other later methods or metrics. COMET (Rei et al., 2020; Fernandes et al., 2022) is known as an evaluation metric, but its framework includes a command for MBR decoding, comet-mbr. It only supports COMET as the utility function and the Monte Carlo estimation as a decoding method. Vamvas and Sennrich (2024) released MBR[9] which is highly integrated into huggingface's TRANSFORMERS (Wolf et al., 2020), but

it only comprises their work and the vanilla MBR decoding, and it cannot be combined with other frameworks like FAIRSEQ (Ott et al., 2019) and black-box large language model services.

Now, to further advance MBR decoding — a powerful technique for improving the quality of text generation — a systematic and shared library is clearly essential for both researchers and developers.

3 Our Library: MBRS

Our library MBRS is mainly implemented on Python and PYTORCH (Paszke et al., 2019). It finds the most suitable output from the given hypotheses.

3.1 Main components

Metrics Metrics are the collections of various evaluation metrics. Basically, reference-based metrics are implemented that can be used as utility functions, but reference-free metrics are also implemented for N-best list reranking. The following are the available metrics in our current repository:

- BLEU (Papineni et al., 2002)
- Translation Edit Rate (TER) (Snover et al., 2006)
- CHRF (Popović, 2015)
- COMET (Rei et al., 2020, 2022a)
- COMETKIWI (Rei et al., 2022b)
- XCOMET (Guerreiro et al., 2024)
- BLEURT (Sellam et al., 2020)

Decoders MBRS selects the most suitable output based on MBR decoding or N-best list reranking from the set of hypotheses $\mathcal{H}$. When the hypotheses are reranked using reference-free metrics like COMETKIWI, MBRS performs standard N-best list reranking; otherwise, it performs MBR decoding.

In MBR decoding, we support not only Monte Carlo estimation (Eikema and Aziz, 2022) but also model-based estimation (Jinnai et al., 2024b) by passing the log-probabilities of pseudo-references. These estimators can be combined with a variety of MBR decoding algorithms. Table 1 lists the implemented variants of MBR decoding. Note that the aggregation methods in RAMBR depend on each metric. MBRS implements the aggregation methods for the BLEU (DeNero et al., 2009), CHRF (Vamvas and Sennrich, 2024), and COMET (Vamvas and Sennrich, 2024; Deguchi et al., 2024) that has been

Decoder	Available metrics	Time complexity
MBR: Vanilla (Eikema and Aziz, 2020, 2022)	any	$\mathcal{O}(|\mathcal{H}||\hat{\mathcal{R}}|)$
PruneMBR: Confidence-based pruning (Cheng and Vlachos, 2023)	any	N/A
RAMBR: Reference aggregation	BLEU, CHRF, and COMET	$\mathcal{O}(|\mathcal{H}| + |\hat{\mathcal{R}}|)$
on BLEU: Aggregate n-gram counts and length (DeNero et al., 2009)		
on CHRF: Aggregate n-gram counts (Vamvas and Sennrich, 2024)		
on COMET: Aggregate sentence embeddings (Vamvas and Sennrich, 2024; Deguchi et al., 2024)		
CBMBR: Centroid-based reference aggregation (Deguchi et al., 2024)	COMET	$\mathcal{O}(|\mathcal{H}|k + |\hat{\mathcal{R}}|k)$
PMBR: Low-rank matrix completion (Trabelsi et al., 2024)	any	$\mathcal{O}\left(\frac{|\mathcal{H}||\hat{\mathcal{R}}|}{r}\right)$

Table 1: Implementation list of MBR decoding variants. The time complexity is based on the Monte Carlo estimation. Since PruneMBR dynamically decides the termination condition based on confidence that depends on the input data, we are unable to show the time complexity analytically.

```
1  from mbrs.metrics import MetricCOMET
2  from mbrs.decoders import DecoderMBR
3
4  SOURCE = "ありがとう"
5  HYPOTHESES = ["Thanks", "Thank you", "Thank
       you so much", "Thank you.", "thank you"]
6
7  metric_cfg = MetricCOMET.Config(
8      model="Unbabel/wmt22-comet-da",
9  )
10 metric = MetricCOMET(metric_cfg)
11
12 decoder_cfg = DecoderMBR.Config()
13 decoder = DecoderMBR(decoder_cfg, metric)
14
15 output = decoder.decode(HYPOTHESES,
       references=HYPOTHESES, source=SOURCE,
       nbest=1)
16
17 print(f"Selected index: {output.idx}")
18 print(f"Output sentence: {output.sentence}")
19 print(f"Expected score: {output.score}")
```

Listing 1: An example code of MBR decoding using the COMET metric on our MBRS.

```
1  >>> from mbrs import timer
2  >>> from mbrs.metrics import MetricBLEU
3  >>> HYPOTHESES = ["Thank you so much."] * 100
4  >>> metric = MetricBLEU(MetricBLEU.Config())
5  >>> scores = []
6  >>> for h in HYPOTHESES:
7  ...     for r in HYPOTHESES:
8  ...         with timer.measure("score"):
9  ...             scores.append(metric.score(h, r))
10 >>> timer.aggregate().result(nsentences=1)
11 [{'name': 'score', 'acctime':
       0.6932655478303786, 'acccalls': 10000,
       'ms/call': 0.06932655478303787,
       'ms/sentence': 693.2655478303786,
       'calls/sentence': 10000.0}]
```

Listing 2: An example usage of our code block profiler.

3.3 Reproducibility

Fixing random number seed Some algorithms use random numbers. To ensure the reproducibility of experiments, MBRS generates all random numbers from `torch.Generator` with a manual seed.

Dataclass/YAML-based configuration All metrics and decoders are configured using `dataclass`, making it more robust by using typed variables. In CLI, the command-line arguments are automatically generated and parsed using the configuration dataclasses. Furthermore, instead of specifying arguments directly, the YAML configuration file can also be passed via `--config_path` option in CLI.

3.4 Transparency

Code block profiler One of the key features of MBRS is the code block profiler as shown in Listing 2. It can measure the elapsed time and number of calls within the code block using a context manager, i.e., `with` statement of Python. After running the program, it automatically aggregates all profilers and reports the statistics. This feature is helpful

proposed so far[10]. The listed decoders can be combined with either the Monte Carlo or model-based estimations.

3.2 Interfaces

MBRS has two interfaces: Python application programming interface (API) and command-line interface (CLI). Listing 1 is an example code via Python API. The detailed references of Python API and CLI are available on Read the Docs[11].

---

[10] Other metrics, e.g., TER, XCOMET, and BLEURT, cannot aggregate the references due to their calculation (DeNero et al., 2009; Vamvas and Sennrich, 2024; Deguchi et al., 2024).

[11] Read the Docs: https://mbrs.readthedocs.io/en/latest/index.html

for identifying the bottleneck of the codes and designing new algorithms.

Metadata analysis In addition to the configuration, the outputs of the decoders are also carried by `dataclass` as shown in L17–19 of Listing 1. At least, all decoders always return not only the output texts but also expected scores and their indices in the list of hypotheses. This allows us to analyze where the output text came from when the hypothesis set is constructed from the multiple generation systems. In addition, the decoders can return the reranked N-best lists; thus, it can show the N-best output texts and their expected scores.

3.5 Extensibility

Metrics and decoders are easily customized using the abstract classes. By inheriting these classes, a new class can leverage the predefined common methods and needs only to implement specific methods minimally, e.g., `.score()` for metrics. If the required methods are not implemented, an exception error will be raised. The necessary methods can be implemented according to the type annotations and docstrings of the parent class.

4 Experiments

4.1 Setup

Using our MBRS, we conducted translation experiments on the WMT'22 general translation task (Kocmi et al., 2022) in En–De and De–En, and evaluated the translation quality and speed. We compared various MBR decoding methods: vanilla (Eikema and Aziz, 2020, 2022), reference aggregation (DeNero et al., 2009; Vamvas and Sennrich, 2024), centroid-based aggregation (Deguchi et al., 2024), confidence-based pruning (Cheng and Vlachos, 2023), and probabilistic MBR (Trabelsi et al., 2024). In addition to MBR decoding, we compared other decoding methods, MAP decoding (MAP) and N-best reranking using a quality estimation model (QE). We also measured the upper bound of the translation quality (Oracle) by selecting translations that maximize the evaluation metric using the reference translations.

Translation candidates were generated using M2M100[12] (Fan et al., 2021). For QE reranking, MBR decoding, and oracle evaluation, we sampled 1,024 translations using epsilon sampling with

$\epsilon = 0.02$. We used the same collection for hypotheses and pseudo-references. In MAP decoding, we compared the same hypotheses as MBR decoding (MAP_ϵ) and the higher-probability hypotheses generated by beam search with a beam size of 256 (MAP_{beam}).

We evaluated the translation quality on BLEU (Papineni et al., 2002), CHRF (Popović, 2015), COMET (Rei et al., 2022a), BLEURT (Sellam et al., 2020), and COMETKIWI (KIWI) (Rei et al., 2022b). We employed KIWI for QE reranking, and other metrics for the utility functions of MBR decoding and the oracle. The detailed setup is described in Appendix C.

4.2 Results

Quality of decoded texts Table 2 shows the experimental results of the translation quality. In the table, "Est." indicates the estimation of expected utility. "MC" and "MB" denote the Monte Carlo and model-based estimations, respectively. Due to our computational limitations, the jobs exceeding 100 hours are noted as "OOT". The results show that our MBRS works in various combinations of metrics and decoding methods, and improve the quality of texts compared with both MAP_ϵ and MAP_{beam}. We confirmed that several approximated algorithms also work well, e.g., RAMBR, CBMBR, PruneMBR, and PMBR improved the COMET scores when the COMET was used as the utility function.

To summarize, MBRS can compare various metrics and decoding methods systematically.

Execution time Table 3 and 4 show the execution time in milliseconds (msec) per sentence when we used BLEU and COMET as the utility functions, respectively, in the WMT'22 En–De translation. "QE" in Table 4 shows the reference time of QE reranking with KIWI. In the tables, "Step" indicates each step of decoding, and "E2E" reports the end-to-end total time including miscellaneous processes. Both tables demonstrate that alternative algorithms of the vanilla MBR decoding improved the speed. The additional results with other utility functions are shown in Appendix D.

MBRS can measure the time spent for each profiling code block as shown in the tables, it is helpful for identifying bottlenecks and developing new algorithms.

Distribution of the expected utility As described in Section 3.4, MBRS can present the ad-

---

Decoding	Utility	Est.	En–De					De–En				
			BLEU	CHRF	COMET	BLEURT	KIWI	BLEU	CHRF	COMET	BLEURT	KIWI
MAP_ϵ	–	–	23.8	50.1	75.6	54.9	74.2	26.0	50.8	78.4	62.0	75.4
MAP_{beam}	–	–	25.1	52.8	77.3	56.5	76.0	27.3	52.5	79.2	62.7	76.2
QE	KIWI	–	22.5	51.8	82.1	58.4	83.6	23.8	50.7	82.2	63.3	81.9
MBR	BLEU	MC	25.6	52.9	75.2	54.8	73.7	27.2	52.1	78.5	62.0	75.3
		MB	24.2	50.5	75.5	54.8	74.1	26.4	51.1	78.5	61.9	75.3
	CHRF	MC	24.4	54.2	76.3	56.4	75.1	27.5	53.4	79.0	62.8	76.0
		MB	24.4	51.2	76.0	55.7	74.8	26.8	51.9	78.8	62.5	76.0
	COMET	MC	24.1	52.8	83.9	58.7	79.5	25.7	51.9	83.0	63.7	78.5
		MB	24.1	51.1	81.0	57.1	77.5	26.2	51.4	81.6	63.2	77.6
	BLEURT	MC	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT
		MB	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT
RAMBR	BLEU	MC	25.6	52.9	75.2	54.9	73.8	27.3	52.1	78.6	62.1	75.4
		MB	24.2	50.5	75.5	54.8	74.1	26.4	51.1	78.5	62.0	75.4
	CHRF	MC	24.3	54.2	76.1	56.3	75.0	27.4	53.4	79.0	62.9	76.0
		MB	24.4	51.1	75.8	55.6	74.7	26.9	51.9	78.9	62.6	76.0
	COMET	MC	23.9	52.9	83.0	58.1	79.1	26.0	51.9	82.3	63.5	78.0
		MB	24.3	51.0	80.1	56.6	77.0	26.2	51.4	81.0	63.1	77.2
CBMBR	COMET	MC	23.2	52.4	83.8	58.3	79.2	25.1	51.4	83.0	63.3	78.3
		MB	22.6	51.6	82.2	57.4	77.8	24.2	50.4	81.9	62.6	77.4
Pruning	BLEU	MC	25.7	52.9	75.0	54.6	73.6	27.1	52.0	78.5	61.9	75.3
		MB	24.3	52.0	75.5	54.9	74.2	26.2	51.4	78.6	61.9	75.4
	CHRF	MC	24.5	54.3	76.2	56.3	75.1	27.4	53.5	78.9	62.8	75.9
		MB	24.6	52.8	76.1	55.9	75.0	26.8	52.3	78.8	62.4	75.9
	COMET	MC	24.0	52.8	83.9	58.7	79.5	25.8	51.9	83.0	63.7	78.4
		MB	24.4	52.5	82.1	57.9	78.4	26.3	51.9	81.6	63.3	77.6
	BLEURT	MC	22.7	51.7	77.6	63.4	75.6	24.9	51.3	79.4	65.1	76.4
		MB	24.1	52.2	77.3	59.4	75.9	26.0	51.5	79.1	63.5	76.1
PMBR	BLEU	MC	25.0	52.2	74.2	54.4	72.8	26.4	51.4	78.2	61.8	75.0
		MB	24.9	52.0	73.7	54.2	72.3	26.4	51.5	78.0	61.6	74.7
	CHRF	MC	23.0	52.9	74.3	55.0	73.1	26.0	52.2	78.3	62.1	75.1
		MB	22.9	52.7	74.1	54.8	72.8	25.7	52.1	78.1	61.9	75.0
	COMET	MC	23.9	52.7	83.8	58.7	79.5	25.9	51.9	82.8	63.6	78.3
		MB	23.6	51.8	82.6	57.7	78.6	25.7	51.5	82.1	63.5	77.9
	BLEURT	MC	22.7	51.8	77.7	63.3	75.7	25.0	51.4	79.4	65.1	76.4
		MB	23.1	51.5	77.2	60.8	75.8	25.5	51.4	79.2	64.1	76.3
Oracle	BLEU	–	44.4	62.8	76.1	59.5	71.6	46.4	63.4	80.9	66.6	74.3
	CHRF	–	39.5	66.6	77.7	62.0	72.9	43.5	66.0	81.3	67.4	74.8
	COMET	–	30.2	58.0	87.0	64.1	80.4	34.4	58.3	86.4	68.5	78.9
	BLEURT	–	31.0	58.2	79.7	71.0	75.0	35.7	59.0	82.5	72.6	76.3

Table 2: Experimental results on the WMT'22 general MT task in En–De and De–En.

Step	MBR	PruneMBR	RAMBR	PMBR
Prune	–	950.7	–	–
Aggregate	–	–	494.3	–
ALS	–	–	–	159.3
Utility	15789.8	12158.0	104.3	2268.6
E2E	16076.0	11206.8	605.6	2704.2

Table 3: The execution time in the WMT'22 En–De translation task when BLEU is employed as the utility function.

ditional information of the output texts. We visualized the distribution of expected utility scores in the hypothesis set by using them. Figure 2 compares the distribution of the expected utility scores

between the test cases that have the maximum and minimum variance in the test set of the WMT'22 En–De translation task. We observed that there are cases in which the expected utility varies significantly depending on each hypothesis and cases in which it does not vary much. The latter result suggests that the N-best outputs of MBR decoding are sometimes similar to each other.

MBRS enables us this analysis by returning the metadata of the output texts.

Visualize the decision of MBR decoding MBRS also returns which indices in the hypotheses are selected by MBR decoding. Figure 3 shows the empirical cumulative distribution of ranks selected

Step	QE	MBR	Prune-MBR	RA-MBR	CB-MBR	PMBR
Rerank	304.0	–	–	–	–	–
Encode	–	316.2	310.4	325.0	324.0	319.2
Prune	–	–	5.6	–	–	–
Aggregate	–	–	–	0.0	–	–
Clustering	–	–	–	–	21.2	–
ALS	–	–	–	–	–	5.9
Utility	–	277.5	45.3	0.3	11.4	21.4
E2E	522.3	618.2	361.6	325.9	363.9	567.3

Table 4: The execution time in the WMT'22 En–De translation task when COMET is employed as the utility function.

Figure 2: The distribution of expected utility scores in the set of hypotheses. The left and right ones show the examples that have the maximum and minimum variance of the expected utility in the test set, respectively.

by MBR decoding from descending rankings of output probabilities in the WMT'22 En–De translation task. In the figure, the x-axis indicates the 0-indexed rank of output probabilities in the hypothesis set, i.e., 0 is the highest probability in the hypotheses set, and the y-axis indicates the ratio of the number of selected by MBR decoding when using COMET as the utility function.

From the results, by using MBRS, we could observe that MBR decoding does not always select high-probability texts, i.e., high-quality texts and high-probability texts are different, as mentioned by Freitag et al. (2022).

5 Conclusion

This paper describes MBRS, a library for MBR decoding, which implements various metrics and decoding methods. It is designed to ensure transparency, reproducibility, and extensibility for researchers and developers. We will continue to implement the latest metrics and decoding methods[13]. Furthermore, we will support evaluation metrics

---

[13]This paper describes MBRS up to the version v0.1.2. The latest version, v0.1.3, now implements diverse MBR (Jinnai et al., 2024a).
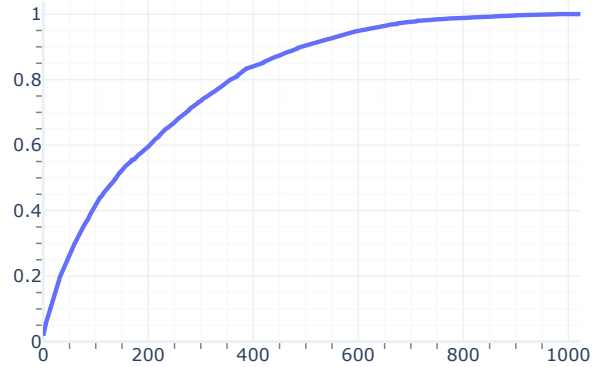

Figure 3: The empirical cumulative distribution of ranks selected by MBR decoding from descending rankings of output probabilities.

other than those used in the machine translation task. We hope our MBRS will be used as a tool to further improve the quality of text generation models.

Limitations

We measured the execution times only in a single run; thus, the speed may differ when different computer architectures are used.

Currently, our repository mainly implements metrics for translation tasks. Implementing evaluation metrics other than the translation task is a future work.

Ethics Statement and Broader Impact

MBR decoding selects output texts from a set of a candidate list generated by text generation models; hence, if the systems generate toxic text, it may be selected depending on the utility functions. In addition, if a utility function is biased and not aligned with human preferences, MBR decoding is more likely to generate biased texts. Since MBR decoding reflects human preferences in the output texts through the utility function, it must be carefully designed to avoid generating harmful texts.

Acknowledgments

References

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.

Hiroyuki Deguchi, Yusuke Sakai, Hidetaka Kamigaito, Taro Watanabe, Hideki Tanaka, and Masao Utiyama. 2024. Centroid-based efficient minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11009–11018, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

John DeNero, David Chiang, and Kevin Knight. 2009. Fast consensus decoding over translation forests. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 567–575, Suntec, Singapore. Association for Computational Linguistics.

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.

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.

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.

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

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins.

2024. xCOMET: Transparent Machine Translation Evaluation through Fine-grained Error Detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Yuu Jinnai, Ukyo Honda, Tetsuro Morimura, and Peinan Zhang. 2024a. Generating diverse and high-quality texts by minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 8494–8525, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Yuu Jinnai, Tetsuro Morimura, Ukyo Honda, Kaito Ariu, and Kenshi Abe. 2024b. Model-based minimum bayes risk decoding for text generation. In *Forty-first International Conference on Machine Learning*.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. 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.

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.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*.

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)*,

358

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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.

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. 2022a. 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. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022b. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). 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.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Firas Trabelsi, David Vilar, Mara Finkelstein, and Markus Freitag. 2024. Efficient minimum bayes risk decoding using low-rank matrix completion algorithms. *Preprint*, arXiv:2406.02832.

Jannis Vamvas and Rico Sennrich. 2024. Linear-time minimum Bayes risk decoding with reference aggregation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 790–801, Bangkok, Thailand. 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.

Dave Zachariah, Martin Sundin, Magnus Jansson, and Saikat Chatterjee. 2012. Alternating least-squares for low-rank matrix reconstruction. *IEEE Signal Processing Letters*, 19(4):231–234.

Metric	Model
COMET	Unbabel/wmt22-comet-da
KIWI	Unbabel/wmt22-cometkiwi-da
BLEURT	BLEURT-20-D12

Table 5: Model list of evaluation metrics that we used in the experiments.

A License

MBRS is published under the MIT license. Our implementation calls the COMET and BLEURT models, licensed under the Apache-2.0 license, and the COMETKIWI and xCOMET models, licensed under the CC BY-NC-SA 4.0 license. We used the test sets of WMT'22 general translation tasks published under the following policy: "The data released for the WMT General MT task can be freely used for research purposes".

B Links to Our Project

The following are links to our project pages:

- GitHub: https://github.com/naist-nlp/mbrs
- HP: https://naist-nlp.github.io/mbrs-web
- Docs: https://mbrs.readthedocs.io/en/latest
- YouTube: https://youtu.be/4qeHpg4PTn0

C Detailed Experimental Setup

In CBMBR, we set the number of centroids to 64. The reduction factor of PMBR was set to 8 and we set $\alpha = 0.99$ in PruneMBR.

We measured the speeds on an NVIDIA® RTX™ 6000 Ada GPU with a batch size of 256 and the half-precision computation for COMET, KIWI, and BLEURT metrics, and on the 32 core Intel® Xeon® Platinum 8160 CPUs for BLEU and CHRF metrics.

Table 5 lists the model names that we used for the evaluation metrics.

D Additional Experiments

D.1 Execution time on the other metrics

Table 6 and 7 show the execution times of MBR decoding in the WMT'22 En–De translation task when CHRF and BLEURT are used as the utility functions, respectively.

D.2 Additional translation experiments on other language directions

Table 8 and 9 show the comparisons of the translation quality in the WMT'22 En↔Zh and En↔Ja

Step	MBR	PruneMBR	RAMBR	PMBR
Prune	–	958.4	–	–
Aggregate	–	–	1090.8	–
ALS	–	–	–	96.2
Utility	36540.7	10792.9	539.7	4694.3
E2E	36695.3	11751.7	1642.5	5019.6

Table 6: The execution time when CHRF is employed as the utility function.

Step	MBR	PruneMBR	PMBR
Prune	–	322.0	–
Aggregate	–	–	–
ALS	–	–	5.5
Utility	OOT	3026.8	38647.1
E2E	OOT	3349.0	38797.8

Table 7: The execution time when BLEURT is employed as the utility function.

general translation tasks, respectively. The experimental setup is the same as the experiments of the WMT'22 En↔De translation tasks described in Section 4.1.

Decoding	Utility	Est.	En–Zh					Zh–En				
			BLEU	CHRF	COMET	BLEURT	KIWI	BLEU	CHRF	COMET	BLEURT	KIWI
MAP_ϵ	–	–	25.1	25.0	76.4	53.0	73.3	12.1	35.0	67.2	49.5	65.4
MAP_{beam}	–	–	28.2	27.0	77.4	53.6	74.2	16.1	43.0	70.1	51.7	68.4
QE	KIWI		25.8	25.1	83.0	55.7	82.8	15.4	46.6	76.9	55.0	77.7
MBR	BLEU	MC	29.8	27.8	77.3	53.8	74.2	17.2	46.8	71.9	51.8	69.4
		MB	25.6	25.5	76.5	53.2	73.4	12.4	35.4	67.4	49.7	65.7
	CHRF	MC	29.5	28.5	77.6	53.8	74.4	17.3	48.9	72.8	53.1	70.4
		MB	25.4	25.8	76.8	53.4	73.7	12.3	35.9	67.9	50.5	66.2
		MC	28.1	26.5	84.1	55.4	78.7	15.8	46.7	77.3	54.1	73.0
		MB	25.5	25.7	81.4	54.5	76.6	12.3	35.6	69.9	50.8	67.4
	BLEURT	MC	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT
		MB	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT
RAMBR	BLEU	MC	29.8	27.7	77.3	53.7	74.2	17.2	46.8	72.1	52.0	69.6
		MB	25.4	25.4	76.4	53.2	73.3	12.0	35.3	67.3	49.8	65.6
	CHRF	MC	29.4	28.6	77.5	53.5	74.4	17.1	48.7	72.8	53.1	70.4
		MB	25.1	25.7	76.6	53.0	73.6	12.2	35.8	67.8	50.5	66.2
	COMET	MC	28.7	26.9	83.2	55.4	78.0	16.0	46.9	76.6	53.9	72.6
		MB	25.6	25.7	80.5	54.3	76.0	12.3	35.7	69.6	50.7	67.2
CBMBR	COMET	MC	27.6	26.0	84.0	55.2	78.5	15.7	46.5	77.2	53.7	72.8
		MB	26.9	25.6	82.8	54.4	77.5	14.5	44.5	75.0	52.6	71.1
Pruning	BLEU	MC	29.8	27.8	77.3	53.8	74.2	17.2	46.8	72.0	51.8	69.4
		MB	28.5	27.1	77.3	53.7	74.1	15.5	44.1	71.2	51.4	68.9
	CHRF	MC	29.5	28.5	77.6	53.8	74.5	17.2	48.8	72.8	53.2	70.5
		MB	28.8	27.4	77.6	53.9	74.4	15.9	45.2	71.7	52.2	69.4
	COMET	MC	28.1	26.5	84.1	55.5	78.6	15.9	46.8	77.3	54.1	72.9
		MB	28.9	27.3	82.3	55.1	77.4	15.2	44.3	74.4	52.7	70.9
	BLEURT	MC	27.3	25.9	78.0	58.7	75.0	15.7	46.5	73.6	56.7	71.3
		MB	28.3	26.9	77.7	55.7	74.6	15.1	44.2	72.0	53.7	69.7
PMBR	BLEU	MC	29.0	27.1	77.0	53.7	73.9	16.9	46.5	71.8	51.7	69.2
		MB	28.9	27.0	76.7	53.2	73.7	16.5	45.8	71.4	51.4	68.8
	CHRF	MC	29.0	27.9	77.3	53.7	74.2	16.0	47.6	72.2	52.5	69.8
		MB	28.7	27.8	76.9	53.1	73.8	15.9	47.4	72.0	52.4	69.6
	COMET	MC	28.3	26.6	83.9	55.5	78.5	15.9	46.7	77.1	54.0	72.9
		MB	27.0	26.1	82.5	54.9	77.4	14.1	41.5	73.5	52.3	70.0
	BLEURT	MC	27.6	26.2	78.0	58.6	74.9	15.5	46.4	73.5	56.6	71.2
		MB	27.7	26.3	77.6	56.9	74.5	14.7	44.1	72.0	54.4	69.6
Oracle	BLEU	–	45.5	41.9	80.9	61.4	74.2	31.9	54.9	74.4	56.3	69.0
	CHRF	–	45.6	42.8	81.0	61.3	74.1	28.2	58.2	75.5	57.8	70.0
	COMET	–	35.4	33.4	87.6	61.8	79.3	20.5	50.8	81.2	58.9	73.9
	BLEURT	–	36.3	34.2	81.9	68.6	75.6	21.3	50.9	76.6	63.8	71.6

Table 8: Comparisons of the translation quality in the WMT'22 En↔Zh general translation tasks.

			En–Ja					Ja–En				
Decoding	Utility	Est.	BLEU	CHRF	COMET	BLEURT	KIWI	BLEU	CHRF	COMET	BLEURT	KIWI
MAP$_\epsilon$	–	–	15.1	22.6	79.2	46.1	77.4	9.0	29.2	68.0	46.5	68.5
MAP$_{beam}$	–	–	14.6	22.3	78.6	46.0	77.4	10.9	33.9	69.7	48.5	70.5
QE	KIWI		16.3	26.1	86.4	50.8	86.7	10.8	37.0	76.3	52.3	80.1
MBR	BLEU	MC	17.9	26.2	81.5	48.2	80.0	11.7	35.5	70.1	47.4	69.9
		MB	15.6	23.1	79.9	46.8	78.0	9.4	29.6	68.1	46.3	68.5
	CHRF	MC	16.8	26.7	81.9	49.1	80.9	12.0	38.4	70.9	49.5	71.1
		MB	15.6	23.4	80.1	47.3	78.6	9.8	31.0	68.8	47.7	69.5
	COMET	MC	16.6	26.0	87.9	50.5	83.5	10.5	36.0	76.7	50.4	74.0
		MB	15.7	23.6	84.0	48.7	80.5	9.6	30.4	72.2	48.4	71.2
	BLEURT	MC	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT
		MB	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT	OOT
RAMBR	BLEU	MC	17.7	26.0	81.5	48.0	80.0	11.8	35.6	70.1	47.8	70.0
		MB	15.5	23.0	79.7	46.6	77.9	9.4	29.9	68.1	46.7	68.7
	CHRF	MC	16.8	26.8	81.8	49.1	80.8	11.8	38.4	71.0	49.5	71.2
		MB	15.3	23.4	80.0	47.3	78.6	9.8	31.1	68.8	47.9	69.5
	COMET	MC	16.9	26.2	87.3	50.5	83.4	10.8	36.2	75.8	50.1	73.4
		MB	15.9	23.6	83.3	48.3	80.1	9.4	30.3	71.5	48.1	70.6
CBMBR	COMET	MC	16.5	26.0	87.8	50.1	83.3	10.0	35.4	76.6	50.1	73.5
		MB	15.7	24.9	86.4	49.0	82.3	9.3	33.8	74.5	48.8	72.1
Pruning	BLEU	MC	17.9	26.2	81.5	48.2	80.0	11.7	35.4	70.0	47.4	69.8
		MB	16.5	25.1	81.3	47.5	79.4	10.1	33.1	69.4	46.9	69.3
	CHRF	MC	16.9	26.8	81.8	49.1	80.9	11.9	38.3	70.8	49.5	71.0
		MB	16.6	25.4	81.5	48.0	79.9	10.8	34.9	70.2	48.4	70.3
	COMET	MC	16.5	26.0	87.9	50.4	83.6	10.4	35.9	76.6	50.3	73.9
		MB	17.0	25.7	86.1	49.7	82.3	10.0	34.1	73.9	49.2	72.2
	BLEURT	MC	15.8	25.5	82.5	53.9	80.6	10.2	35.2	71.3	53.2	71.4
		MB	16.2	25.0	81.9	50.0	80.0	9.9	33.2	70.4	50.0	70.4
PMBR	BLEU	MC	17.4	25.7	81.1	48.1	79.7	11.2	34.9	69.9	47.4	69.8
		MB	17.3	25.7	81.2	47.8	79.6	11.1	34.6	69.4	47.1	69.2
	CHRF	MC	16.4	26.1	81.3	48.5	80.2	11.1	37.2	70.4	49.0	70.6
		MB	16.3	26.0	81.1	48.2	79.9	11.2	37.2	70.1	48.7	70.2
	COMET	MC	16.7	26.0	87.7	50.3	83.4	10.5	35.8	76.5	50.3	73.8
		MB	16.0	24.7	85.9	49.4	82.1	10.0	32.6	74.0	49.1	72.1
	BLEURT	MC	15.8	25.5	82.5	53.7	80.6	10.1	35.2	71.5	53.2	71.6
		MB	16.0	24.7	82.0	51.2	79.9	9.6	33.5	70.5	51.1	70.7
Oracle	BLEU	–	34.7	39.0	83.4	54.4	78.7	28.0	46.2	73.2	53.1	69.5
	CHRF	–	33.2	41.4	84.2	56.3	79.2	24.0	50.4	74.5	55.2	70.8
	COMET	–	22.4	31.6	90.6	56.8	83.6	15.9	41.2	81.9	56.3	75.5
	BLEURT	–	23.5	33.0	85.3	64.3	80.4	16.3	41.6	75.9	62.2	72.8

Table 9: Comparisons of the translation quality in the WMT'22 En↔Ja general translation tasks.