Centroid-Based Efficient Minimum Bayes Risk Decoding

Anonymous ACL submission

Abstract

Minimum Bayes risk (MBR) decoding achieved state-of-the-art translation performance by using COMET, a neural metric that has a high correlation with human evaluation. However, MBR decoding requires quadratic time since it computes the expected score between a translation hypothesis and all reference translations. We propose centroid-based MBR (CBMBR) decoding to improve the speed of MBR decoding. Our method clusters the reference translations in the feature space, and then calculates the score using the centroids of each cluster. The experimental results show that our CBMBR not only improved the decoding speed of the expected score calculation 6.9 times, but also outperformed vanilla MBR decoding in translation quality by up to 0.5 COMET% in the WMT'22 En↔Ja, En↔De, En↔Zh, and WMT'23 En↔Ja translation tasks.[1]

1 Introduction

Minimum Bayes risk (MBR) decoding achieved robust and high-quality translation by selecting the output sentence that maximizes the expected metric score computed from the set of translation hypotheses (Kumar and Byrne, 2004; Eikema and Aziz, 2020; Müller and Sennrich, 2021). Recently, neural evaluation metrics that have a high correlation with human evaluation have been proposed (Rei et al., 2020, 2022a; Sellam et al., 2020; Zhang et al., 2020), and MBR decoding using such neural metrics has achieved state-of-the-art translation performance in human evaluation compared to the conventional maximum-a-posteriori (MAP) decoding using beam search (Fernandes et al., 2022).

However, due to its formulation, the typical MBR decoding regarding the hypothesis set as a pseudo-reference set requires the computational time of $\mathcal{O}(N^2)$ when the N translation hypotheses are given. In recent work, the number of hypotheses N has exceeded 1,000 candidates (Freitag et al., 2023), making the quadratic order of computational time a challenge for MBR decoding, especially when using expensive neural metrics. To improve the decoding speed, several pruning methods have been proposed (Eikema and Aziz, 2022; Cheng and Vlachos, 2023), while these approaches require careful selection of a proxy metric (Eikema and Aziz, 2022), or it is difficult to take advantage of computational parallelism because hypotheses are iteratively pruned (Cheng and Vlachos, 2023).

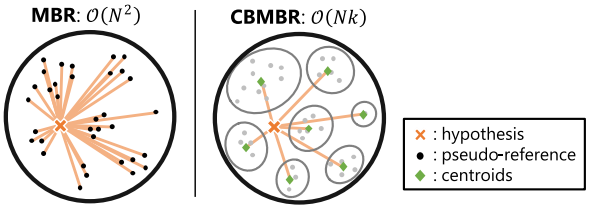

Figure 1: Overview of our CBMBR.

Given that expensive neural metrics, e.g, COMET or BLEURT, are trained to output high scores when a hypothesis sentence and a reference translation are semantically similar, we hypothesized that the distance between sentence vectors of similar sentences in the feature space of their models is close. We leverage the sentence similarity to improve the decoding speed of COMET-MBR by clustering sentence vectors of the translation candidates into $k \ll N$ clusters as shown in Figure 1. Then, we calculate the COMET scores using k centroid vectors of their clusters, instead of using N sentence vectors.

Our proposed method not only achieved a speed-up of 6.9 times in the calculation of the expected score, but also an improvement in COMET score of up to 0.5% compared with the naive MBR decoding in the WMT'22 En↔Ja, En↔De, En↔Zh, and WMT'23 En↔Ja translation tasks.

[1]We will release our source code on GitHub.

2 Background

MBR decoding MBR decoding has been demonstrated to be effective in fields such as statistical automatic speech recognition (Goel and Byrne, 2000) and statistical machine translation (Kumar and Byrne, 2004), and it has been applied to neural machine translation in recent years (Eikema and Aziz, 2020; Müller and Sennrich, 2021). Furthermore, it is more suitable for multiple translation systems than ensemble models (Ito et al., 2023).

Let $\mathcal{X}$ and $\mathcal{Y}$ be the spaces of possible source sentences and target sentences, respectively. MAP decoding generates the target sentence $y^*_{\text{MAP}} \in \mathcal{Y}$ by $y^*_{\text{MAP}} = \text{argmax}_{y \in \mathcal{Y}}\, p_\theta(y|x)$, where θ denotes the parameter of the translation model which calculates the likelihood of an output sentence y given an input sentence $x \in \mathcal{X}$. Since it is hard to calculate probabilities for all possible $y \in \mathcal{Y}$, usually the beam search is used to obtain the solution.

In contrast, MBR decoding determines the output sentence $y^*_{\text{MBR}} \in \mathcal{Y}$ by maximizing the expected utility as follows:

$$y^*_{\text{MBR}} = \text{argmax}_{h \in \mathcal{H}}\, \mathbb{E}_{\hat{y} \sim P(y|x)} \left[u(h, \hat{y})\right], \quad (1)$$

$$\approx \text{argmax}_{h \in \mathcal{H}}\, \mathbb{E}_{\hat{y} \in \hat{\mathcal{Y}}} \left[u(h, \hat{y})\right], \quad (2)$$

where $u\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ denotes the utility function, which represents the preference relation, and $\mathcal{H} = \{h_i\}_{i=1}^{|\mathcal{H}|} \subset \mathcal{Y}$ denotes the set of translation hypotheses. $P(y|x)$ is the true probability of being translated from a given input sentence $x \in \mathcal{X}$, and it is approximated using the sampled reference translations $\hat{\mathcal{Y}} = \{\hat{y}_i\}_{i=1}^{|\hat{\mathcal{Y}}|} \subset \mathcal{Y}$ as shown in Equation 2 since the true probability is unknown. The typical MBR decoding treats the hypothesis set itself as the pseudo-reference set, i.e., $\hat{\mathcal{Y}} := \mathcal{H}$. Note that the time complexity is $\mathcal{O}(N^2)$, where $N := |\mathcal{H}|$, which is time-consuming.

COMET-MBR COMET is an evaluation metric of translation quality that achieved a high correlation with human evaluation. The COMET model consists of the XLM-RoBERTa (XLM-R) (Conneau et al., 2020) -based sentence encoder and the output layer, and it is trained to predict direct assessment (DA) scores (Rei et al., 2020, 2022a). It first encodes the source sentence $x \in \mathcal{X}$, the hypothesis sentence $h \in \mathcal{Y}$, and the reference sentence $\hat{y} \in \mathcal{Y}$ into their D dimensional sentence vectors, independently, and then the COMET score is computed from the triplet of sentence vectors by the output layer. Let $f\colon \mathcal{X} \cup \mathcal{Y} \to \mathbb{R}^D$ be the function of sentence encoding and $s\colon \mathbb{R}^D \times \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ be the output layer, the COMET score is computed by $s(f(x), f(h), f(\hat{y}))$. MBR decoding using COMET (COMET-MBR) replaces the utility u in Equation 2 with the COMET score:

$$y^*_{\text{COMET-MBR}} = \text{argmax}_{h \in \mathcal{H}}\, \mathbb{E}_{\hat{y} \in \hat{\mathcal{Y}}} \left[s(f(x), f(h), f(\hat{y}))\right]. \quad (3)$$

3 Proposed Method

Our proposed *centroid-based MBR (CBMBR)* approximates the expected utility by using the centroids of similar sentence vectors. CBMBR decodes by computing the expected utility according to the following procedures: sentence encoding, clustering, and calculating the expected utility.

Encoding Firstly, the sentence vector of the source $f(x) \in \mathbb{R}^D$, the hypotheses $\{f(h_i)\}_{i=1}^{|\mathcal{H}|} \subset \mathbb{R}^D$, and the pseudo-references $\{f(\hat{y}_i)\}_{i=1}^{|\hat{\mathcal{Y}}|} \subset \mathbb{R}^D$ are computed.

Clustering Next, we perform clustering for the sentence vectors of pseudo-references into $k \ll N$ clusters and obtain the centroid vectors of each cluster $\mathcal{C} = \{\boldsymbol{c}_i\}_{i=1}^{k} \subset \mathbb{R}^D$. Here, we employ $k$means++ (Arthur and Vassilvitskii, 2007) to prevent the centroids from being biased. $k$means++ selects the initial centroids so that the distances between each pair of centroids are farther according to the weights calculated from distances of vectors. The details of the algorithm are described in Appendix C.1. Then, the vectors $\{f(\hat{y}_i)\}_{i=1}^{|\hat{\mathcal{Y}}|}$ are clustered using the standard $k$means algorithm. Concretely, the following steps 1) and 2) are iteratively calculated: 1) assign a vector to its nearest neighbor centroid, 2) and update the centroid using vectors assigned to its cluster.

Expected utility Finally, the expected utility is calculated by replacing pseudo-reference vectors $f(\hat{y}) \in \mathbb{R}^D$ with centroids $\boldsymbol{c} \in \mathbb{R}^D$ in Equation 3:

$$y^*_{\text{CBMBR}} = \text{argmax}_{h \in \mathcal{H}}\, \mathbb{E}_{\boldsymbol{c} \in \mathcal{C}} \left[s(f(x), f(h), \boldsymbol{c})\right]. \quad (4)$$

The conventional method requires $\mathcal{O}(N^2)$ of computational time to compute the expected utility for all hypotheses, whereas our CBMBR computes in $\mathcal{O}(Nk)$. Note that k ($1 \leq k \leq N$) is a hyperparameter that balances the trade-off between the decoding speed and approximation accuracy. Especially, when $k = 1$, i.e., $\mathcal{C} = \{\boldsymbol{c}_1\}$, the centroid

Decoding	en-ja	ja-en	en-de	de-en	en-zh	zh-en	avg.
MAP	78.7	69.7	77.3	79.2	77.4	70.1	75.4
QE	86.6	76.2	82.2	82.1	82.9	76.9	81.2
MBR	**87.9**	**76.6**	**84.0**	**83.0**	**84.2**	**77.3**	**82.2**
PruneMBR	**87.9**	76.5	**84.0**	**83.0**	84.1	**77.3**	82.1
CBMBR	**87.9**	**76.6**	83.9	**83.0**	84.1	77.1	82.1
w/o $k$means++	87.8	76.4	83.8	82.9	84.0	77.2	82.0
Oracle	90.6	81.9	87.0	86.5	87.7	81.2	85.8

Table 1: Translation quality on the WMT'22 translation task with the setting of diverse translation candidates. The best scores are emphasized in bold font and second-best scores are underlined for each language direction.

Step	QE	MBR	PruneMBR	CBMBR
Encode/hypotheses	–	247.0	248.0	247.8
Encode/source	–	51.6	51.1	51.2
Rerank	450.1	–	–	–
Prune	–	–	5.5	–
$k$means++	–	–	–	36.5
Utility function; s	–	322.2	79.6	20.1
E2E	450.1	633.1	384.7	356.8

Table 2: Average processing time per sentence (msec) on the WMT'22 translation task in the diverse candidates setting. Note that "E2E" measures the end-to-end time, including miscellaneous processes.

$\boldsymbol{c}_1$ can be calculated as the average of all pseudo-reference vectors, i.e., $\boldsymbol{c}_1 = \frac{1}{|\hat{\mathcal{Y}}|}\sum_{i=1}^{|\hat{\mathcal{Y}}|} f(\hat{y}_i)$, and the time complexity of CBMBR will be $\mathcal{O}(N)$, which is equivalent to DeNero et al. (2009).

4 Experiments

Setup We conducted translation experiments with two settings: one uses diversified translation candidates, and the other simulates a more realistic scenario, multi-system translation. We evaluated the translation quality using COMET score, which is the same as the utility function of MBR decoding used in our experiments. For comparison, we also performed translation candidate reranking using a quality estimation model COMETKIWI (Rei et al., 2022b) (QE), confidence-based pruning MBR decoding (PruneMBR) (Cheng and Vlachos, 2023), and the quality upper bound (Oracle), which selects the hypothesis with the best score according to COMET using reference translations. We also compared CBMBR without $k$means++, where initial centroids were randomly selected from the sample set (w/o $k$means++). We used COMET-22 (Rei et al., 2022a) for the evaluation metric and utility function. In MBR decoding, we treat the hypothesis set as the pseudo-reference set, i.e., $\hat{\mathcal{Y}} := \mathcal{H}$. We set the number of centroids to $k = 64$. Details of our setup are shown in Appendix D.

Diverse translation candidates In this setting, we evaluated the translation quality in six language directions: En↔Ja, En↔De, and En↔Zh in the WMT'22 translation task (Kocmi et al., 2022). We generated translation candidates using the pre-trained multilingual translation model, M2M100. We employed beam search with a beam size of 256 for MAP decoding, and generated 1,024 translations using epsilon sampling with $\epsilon = 0.02$ (Freitag et al., 2023) for MBR decoding.

Table 1 shows the translation quality of each decoding method. From the average scores in the table, when compared with MAP decoding, both MBR decoding and the proposed CBMBR decoding improve the COMET score by +6.8 and +6.7%, respectively, and the gap between Oracle has been narrowed. The results also show that the difference between CBMBR and MBR is within 0.1% by using $k$means++ initialization.

Next, we compared the decoding time of each method as shown in Table 2. From the table, the overall decoding time (E2E) shows that CBMBR is 1.8 times faster than vanilla MBR. Specifically, in the computation of the expected utility which required quadratic time, the speed was increased by 6.9 times when including $k$means++, and by 16.0 times when comparing only the utility computation. Compared to PruneMBR, we confirmed that the speed of the expected utility calculation improved by 1.5 times. One reason for this improvement is that, unlike PruneMBR, CBMBR computes the expected utility at once, making it easier to leverage parallel computation capabilities of GPU.

In summary, CBMBR maintains comparable translation quality to the naive MBR decoding while accelerating the computational time by 6.9 times, including clustering.

Multi-system translation We also evaluated the effectiveness of our CBMBR in the setting where translation candidates are generated from multiple translation systems. In particular, we follow the practice in Deguchi et al. (2023), in which 18 candidate sets with each set comprising 50-best translations are generated from nine different models and two decoding methods: beam search and top-p sampling ($p = 0.7$) with a beam size of 50. We evaluated the translation quality in two language

Decoding	WMT'22 en-ja	WMT'22 ja-en	WMT'23 en-ja	WMT'23 ja-en	avg.
MAP	86.4	80.9	83.5	80.4	82.8
QE	89.8	82.6	87.6	82.3	85.6
MBR	90.5	**84.1**	88.7	83.7	86.7
PruneMBR	88.9	82.8	86.6	82.2	85.1
CBMBR	**90.9**	**84.1**	**89.2**	**83.8**	**87.0**
w/o $k$means++	90.5	**84.1**	88.8	83.7	86.8
Oracle	93.4	89.4	91.9	88.5	90.8

Table 3: The translation quality in the multi-system translation setting.

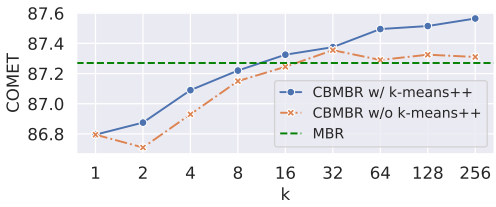

Figure 2: Translation quality of various k in the multi-system translation setting. The scores are averaged COMET on the WMT'22 En-Ja and Ja-En.

directions: En↔Ja in the WMT'22 and WMT'23 translation tasks (Kocmi et al., 2022, 2023).

Table 3 shows the results. Unlike the diverse candidates setting, CBMBR improved the translation quality by up to 0.5%, compared with MBR. Naive MBR calculates the expected utility using all samples equally, which is prone to translation bias when candidates have multimodal distribution. In contrast, CBMBR estimates the expected utility using only centroids; therefore, it decodes robustly even if the distribution is multimodal. The detailed analysis is shown in Appendix F.

In summary, we found that CBMBR not only improved decoding speed, but also improved translation quality compared to the vanilla MBR when the translation is determined from the candidate sets generated from multiple translation systems.

5 Discussion

5.1 The number of centroids k

We evaluated the COMET scores of various $k \in \{2^i\}_{i=0}^{8}$ in the multi-system translation setting. Figure 2 shows the results. When $k = 1$, while the time complexity is linear time $\mathcal{O}(N)$, the COMET score of CBMBR was degraded by 0.5% compared with vanilla MBR. The figure shows that translation quality improves as k is increased, and CBMBR outperformed vanilla MBR when $k \geq 16$. In addition, the translation quality was better when we use $k$means++ compared to the standard $k$means.

Model	dev	test
RoBERTa$_{\text{large}}$ (Liu et al., 2019)	53.7	43.0
fastText (Joulin et al., 2017)	65.3	53.6
XLM-R$_{\text{large}}$ (Conneau et al., 2020)	39.1	31.6
LaBSE (Feng et al., 2022)	72.9	72.7
COMET (Rei et al., 2022a)	78.2	73.6

Table 4: Pearson $r \times 100$ in the STS-B task using the encoder of COMET model.

5.2 Distance between similar sentence vectors

As a prerequisite for the proposed method, it is necessary that sentence similarity be represented as the distances between sentence vectors. To verify this assumption, we investigated the distances between sentence vectors of similar sentences using the Semantic Textual Similarity Benchmark (STS-B) task (Cer et al., 2017). We evaluated the Pearson correlation coefficient r with the ground truth similarity score. Table 4 shows the experimental results. Despite sentence vectors are not explicitly trained like contrastive learning, COMET demonstrates a strong correlation of 73.6. Moreover, the result shows that it has implicitly learned sentence similarity through the training of score prediction, as evidenced by its significantly better correlation of 73.6 compared to the pre-trained XLM-R score of 31.6. Furthermore, we confirmed that COMET outperformed LaBSE (Feng et al., 2022) trained by contrastive learning.

To summarize, the sentence vectors of COMET demonstrate a strong correlation with gold scores in the STS-B task although the sentence representations are not explicitly trained. Also, we confirmed that the encoder of COMET implicitly learned sentence similarity through the score prediction.

6 Conclusion

In this paper, we proposed CBMBR, which improves the speed of MBR decoding by clustering the sentence vectors of similar sentences and computing the score with the centroid representations of each cluster. Our CBMBR achieved a 6.9 times speed-up in the expected score calculation and an improvement in COMET of up to 0.5% compared with vanilla MBR decoding in the WMT'22 En↔Ja, En↔De, En↔Zh, and WMT'23 En↔Ja translation tasks. For future work, we would like to apply our method to other evaluation metrics including both neural and non-neural metrics.

Limitations

This study focuses only on improving the speed of MBR decoding, especially the neural evaluation metric, COMET. For non-neural metrics, it is necessary to apply the appropriate clustering method for each metric.

In COMET-MBR, there are two bottlenecks of computational time: the calculation of the expected utility and sentence encoding. However, this study only improves the computation speed of the expected utility, which took quadratic time. Although the sentence encoding can be computed in a linear time, the sentences are encoded using the expensive XLM-R encoder, which is time-consuming.

Our method can only be applied to metrics for which we can compute the representation independently for each sentence. This limitation is the same as that of the method of DeNero et al. (2009) and is also explained in their paper as well.

The decoding times reported in this paper are measured on a single computer and only a single run; the amount of speed improvement may differ when different computer architectures are used.

Ethical Consideration

Both vanilla MBR decoding and CBMBR decoding select output sentences from a set of translation candidates generated by translation systems, so if the systems generate toxic text, it may be selected.

References

David Arthur and Sergei Vassilvitskii. 2007. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA. Society for Industrial and Applied Mathematics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

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.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Hiroyuki Deguchi, Kenji Imamura, Yuto Nishida, Yusuke Sakai, Justin Vasselli, and Taro Watanabe. 2023. NAIST-NICT WMT'23 general MT task submission. In *Proceedings of the Eighth Conference on Machine Translation*, pages 110–118, Singapore. 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.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

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, Behrooz Ghorbani, and Patrick Fernandes. 2023. Epsilon sampling rocks: Investigating sampling strategies for minimum Bayes risk decoding for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9198–9209, Singapore. Association for Computational Linguistics.

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

Ikumi Ito, Takumi Ito, Jun Suzuki, and Kentaro Inui. 2023. Investigating the effectiveness of multiple expert models collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14393–14404, Singapore. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.

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.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

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.

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.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Licenses

In our experiments, we used COMET-22 model licensed under the Apache-2.0 license and COMETKIWI model licensed under the CC BY-NC-SA 4.0 license. We evaluated our method on the test set of WMT'21, WMT'22, and WMT'23 translation tasks, which are described as "The data released for the WMT General MT task can be freely used for research purposes".

B Details of Dataset

Table 5 shows the number of sentences for each dataset we used in our experiments.

Dataset	en-ja	ja-en	en-de	de-en	en-zh	zh-en
WMT'21	1,000	1,005	1,002	1,000	1,002	1,948
WMT'22	2,037	2,008	2,037	1,984	2,037	1,875
WMT'23	2,074	1,992	–	–	–	–

Table 5: Number of sentences for each dataset we used.

C Details of Algorithms and Models

C.1 $k$means++

We describe the algorithm of initial centroids selection of $k$means++:

1. Pick up the first centroid from the set $\hat{\mathcal{Y}}$ and add it into $\mathcal{C}$.

2. Calculate the squared Euclidean distance between a vector $f(\hat{y}_i)$ and its nearest centroid $d^2(\hat{y}_i) = \min_{\boldsymbol{c} \in \mathcal{C}} \|f(\hat{y}_i) - \boldsymbol{c}\|_2^2$.

3. Sample the vector $f(\hat{y}_i)$ from multinominal distribution according to the weights $\frac{d^2(\hat{y}_i)}{\sum_{j=1}^{|\hat{\mathcal{Y}}|} d^2(\hat{y}_j)}$ and add it to the set $\mathcal{C}$.

4. Repeat steps 2 and 3 until k centroids are selected.

C.2 COMET Model

Figure 3 shows the overview of the COMET model. A triplet of sentences are independently encoded into their sentence vectors, and then the COMET score is calculated from the vectors.

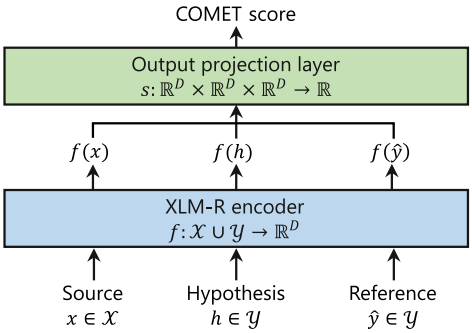

Figure 3: Overview of COMET model.

D Details of Experimental Setup

Table 6 shows the details of our experimental setup. Note that we implemented vanilla MBR, PruneMBR, and CBMBR using PyTorch. We will release our implementation.

Model	
COMET	`Unbabel/wmt22-comet-da`[2]
QE	`Unbabel/wmt22-cometkiwi-da`[3]
GPU	NVIDIA A100 $\times 1$
Batch size	256 sentences
(sentence encoding)	
Diverse translation candidates setting	
Translation model	M2M100 (418M parameters)[4]
MAP decoding	
Generation	beam search
Beam size	256
MBR decoding	
Candidate generation	
# of candidates	1,024 translations
Generation	epsilon sampling ($\epsilon = 0.02$) (Freitag et al., 2023)
CBMBR	
# of centroids k	64
# of iterations	1
Multi-system translation setting	
Translation model	9 various Transformer models (Deguchi et al., 2023)
MAP decoding	
Generation	beam search using the ensemble model
Beam size	50
MBR decoding	
Candidate generation	
# of candidates	900 translations[5]
Generation	beam search and top-p sampling ($p = 0.7$) (Deguchi et al., 2023)
CBMBR	
# of centroids k	64
# of iterations	1

Table 6: The details of our experimental setup.

E Other Experimental Results

E.1 Translation quality on the development set in the diverse translation candidates setting

Table 7 shows the experimental results of the diverse translation candidates setting on the development set. In the table, "niter" denotes the number of iterations of $k$means clustering. We chose niter=1 from the results.

[2] https://huggingface.co/Unbabel/wmt22-comet-da

[3] https://huggingface.co/Unbabel/wmt22-cometkiwi-da

[4] https://huggingface.co/facebook/m2m100_418M

[5] We will release translation candidates we created.

Decoding	en-ja	ja-en	en-de	de-en	en-zh	zh-en	avg.
MAP	78.8	62.6	74.5	80.4	73.0	68.6	73.0
QE	86.7	71.7	80.3	83.6	80.1	77.0	79.9
MBR	88.2	72.6	82.1	84.3	81.6	77.7	81.1
PruneMBR	88.2	72.6	82.0	84.3	81.6	77.7	81.1
CBMBR	88.2	72.3	81.9	84.4	81.5	77.5	81.0
w/o $k$means++	88.1	72.4	81.8	84.3	81.4	77.5	80.9
CBMBR with various numbers of $k$means++ iterations							
niter=1	88.2	72.3	81.9	84.4	81.5	77.5	81.0
niter=2	88.2	72.2	81.9	84.4	81.6	77.5	81.0
niter=3	88.2	72.3	81.9	84.4	81.5	77.5	81.0
niter=4	88.2	72.3	81.8	84.4	81.5	77.5	81.0
niter=5	88.2	72.3	81.9	84.4	81.6	77.5	81.0
CBMBR with various numbers of $k$means iterations							
niter=1	88.1	72.4	81.8	84.3	81.4	77.5	80.9
niter=2	88.1	72.4	81.8	84.3	81.4	77.5	80.9
niter=3	88.1	72.4	81.8	84.3	81.4	77.5	80.9
niter=4	88.1	72.4	81.8	84.3	81.5	77.5	80.9
niter=5	88.1	72.4	81.8	84.3	81.5	77.5	80.9
oracle	89.9	76.3	84.0	87.3	84.2	80.2	83.7

Table 7: The translation quality (COMET%) in the diverse translation candidates setting on the WMT'21 translation task. "niter" denotes the number of iterations of $k$means clustering.

E.2 Decoding speed in the multi-system translation setting

Table 8 shows the decoding speed in the multi-system translation setting measure on the WMT'22 and WMT'23 En↔Ja translation tasks. As shown in the table, our CBMBR improved the speed of the expected score calculation by 5.0 times compared to vanilla MBR and 1.5 times compared to PruneMBR in the multi-system setting.

Step	QE	MBR	PruneMBR	CBMBR
Encode				
hypotheses; $\mathcal{H}$	–	198.1	198.7	199.1
source; x	–	22.0	22.0	21.9
Rerank	313.0	–	–	–
Prune	–	–	5.4	–
$k$means++	–	–	–	36.1
Utility function; s	–	281.1	79.5	20.1
E2E	336.0	511.9	306.0	278.4

Table 8: Average processing time per sentence (msec) in the multi-system translation setting measured on the WMT'22 and WMT'23 En↔Ja translation tasks. Note that "E2E" measures the end-to-end time, including miscellaneous processes.

F Multimodality of Translation Candidates

In the multi-system translation setting, CBMBR outperformed vanilla MBR also in terms of translation quality as shown in Table 3 and Figure 2. In this section, we discuss the multimodal nature of translation, two approximations of MBR decoding, and why our CBMBR outperformed vanilla MBR in the multi-system translation setting.

	WMT'22		WMT'23		
Decoding	en-ja	ja-en	en-ja	ja-en	avg.
MAP	86.4	80.9	83.5	80.4	82.8
QE	89.8	82.6	87.6	82.3	85.6
MBR	90.5	**84.1**	88.7	83.7	86.7
PruneMBR	88.9	82.8	86.6	82.2	85.1
CBMBR	**90.9**	**84.1**	**89.2**	**83.8**	**87.0**
w/o $k$means++	90.5	**84.1**	88.8	83.7	86.8
$\text{CBMBR}_{\text{cnt}}$	90.4	83.9	88.6	83.5	86.6
w/o $k$means++	90.4	84.0	88.6	83.6	86.6
Oracle	93.4	89.4	91.9	88.5	90.8

Table 9: Results of the multi-system translation setting with weighting by the numbers of samples.

The n-best translations generated by beam search are often similar to each other (Vijayakumar et al., 2018). To diversify the candidates while maximizing translation quality, Deguchi et al. (2023) generated the 50-best translation sets from each translation system, resulting in the candidates that exhibit multimodality. Vanilla MBR decoding calculates the expected score by treating all samples equally, which means it is prone to being affected by the number of similar translation samples in the candidates with such a multimodal distribution.

Now, there are two approximation variants of the MBR decoding in our CBMBR. One is our proposed method, which calculates the expected score using centroid representations:

$$y^*_{\text{CBMBR}} = \text{argmax}_{h \in \mathcal{H}} \, \mathbb{E}_{\boldsymbol{c} \in \mathcal{C}} \left[s(f(x), f(h), \boldsymbol{c}) \right]. \tag{5}$$

The other $\text{CBMBR}_{\text{cnt}}$ multiplies each centroid-based score by the weight according to the number of samples in each cluster, as follows:

$$y^*_{\text{CBMBR}_{\text{cnt}}} = \text{argmax}_{h \in \mathcal{H}} \, \mathbb{E}_{\boldsymbol{c} \in \mathcal{C}} \left[s(f(x), f(h), \boldsymbol{c}) \times w(\boldsymbol{c}) \right], \tag{6}$$

where $w \colon \mathbb{R}^D \to [0, 1]$ returns the weight of the

given centroid, as follows:

$$w(\boldsymbol{c}) = \frac{\mathrm{count}(\boldsymbol{c})}{\sum_{i=1}^{k} \mathrm{count}(\boldsymbol{c}_i)}, \tag{7}$$

$$\mathrm{count}(\boldsymbol{c}) = \left| \left\{ \hat{y} \in \hat{\mathcal{Y}} : \boldsymbol{c} = \mathrm{NN}\left(f(\hat{y}), \mathcal{C}\right) \right\} \right|, \tag{8}$$

$$\mathrm{NN}(\boldsymbol{q}, \mathcal{C}) = \mathrm{argmin}_{\boldsymbol{c} \in \mathcal{C}} \|\boldsymbol{q} - \boldsymbol{c}\|_2, \tag{9}$$

where $\mathrm{NN}\colon \mathbb{R}^D \times \mathcal{C} \to \mathbb{R}^D$ finds the nearest neighbor centroid of a vector $\boldsymbol{q} \in \mathbb{R}^D$ from the given set of centroids $\mathcal{C}$, and $\mathrm{count}\colon \mathbb{R}^D \to \mathbb{N} \cup \{0\}$ counts the number of samples in the cluster of the given centroid. CBMBR$_{\text{cnt}}$, which uses the number of samples in a cluster, can be regarded to more accurately approximate vanilla MBR compared to our CBMBR, which ignores the number of samples in a cluster.

We compared the translation quality of our CBMBR and CBMBR$_{\text{cnt}}$ with the multi-system translation setting. Table 9 shows the results. From the results, the difference between vanilla MBR and CBMBR$_{\text{cnt}}$ is narrowed to 0.1%, and degraded by 0.6% compared to CBMBR.

In other words, CBMBR$_{\text{cnt}}$, which more accurately approximates vanilla MBR, is worse than our proposed CBMBR. We attribute this observation to the biased distribution caused by the beam search or sampling. With CBMBR, we can robustly decode against the bias.