Generating Diverse Translation with Perturbed kNN-MT

Yuto Nishida[1] **Makoto Morishita**[2] **Hidetaka Kamigaito**[1] **Taro Watanabe**[1]
[1]Nara Institute of Science and Technology
[2]NTT Communication Science Laboratories, NTT Corporation
{nishida.yuto.nu8, kamigaito.h, taro}@is.naist.jp
makoto.morishita@ntt.com

Abstract

Generating multiple translation candidates would enable users to choose the one that satisfies their needs. Although there has been work on diversified generation, there exists room for improving the diversity mainly because the previous methods do not address the overcorrection problem—the model underestimates a prediction that is largely different from the training data, even if that prediction is likely. This paper proposes methods that generate more diverse translations by introducing perturbed k-nearest neighbor machine translation (kNN-MT). Our methods expand the search space of kNN-MT and help incorporate diverse words into candidates by addressing the overcorrection problem. Our experiments show that the proposed methods drastically improve candidate diversity and control the degree of diversity by tuning the perturbation's magnitude.

1 Introduction

In natural language, there are multiple lexically distinct translations given an input sentence. Therefore, machine translation systems should offer multiple translation candidates to users so that the final choice should be made by them considering their demands, e.g., styles or domains. However, standard neural machine translation (NMT) models suffer from a low diversity problem in which the generated translation candidates are almost identical. One reason lies in beam search, which is a standard inference algorithm, where the search space is expanded in a left-to-right fashion while keeping only the top-N candidates in every decoding step and just preserving slightly different translations (Gimpel et al., 2013; Vijayakumar et al., 2018; Freitag and Al-Onaizan, 2017). The other reason is the overcorrection problem (Zhang et al., 2019), which is caused by a model trained with cross-entropy loss that underestimates a prediction that is largely different from the training data, even if it is likely. This phenomenon discourages the model from generating synonymous expressions and leans toward gold standards, reducing the diversity in the candidates.

To encourage the model to generate more diverse candidates, Vijayakumar et al. (2018), Holtzman et al. (2020), and Freitag and Al-Onaizan (2017) proposed variants of beam search algorithms in which diverse candidates are retained in the search space. However, their methods do not directly address the overcorrection problem, limiting their effect in generating diverse translations.

To alleviate this issue, we propose kNN diversified decoding that combines diversified beam search and k-nearest neighbor machine translation (kNN-MT; Khandelwal et al., 2021), which addresses the overcorrection problem by retrieving alternative target tokens from the training data during decoding (Yang et al., 2022). To further diversify the search space, we also propose two methods, i.e., stochastic and deterministic methods. The stochastic method expands the search space by perturbation so that the model can generate more likely tokens that are less focused. We proposed two types of perturbations, *noised-kNN* (Figure 1 (1)), which adds a noise vector to the query of the kNN search, and *randomized-kNN* (Figure 1 (2)), which arbitrarily selects k neighbors from a more extensive search space. The deterministic method, *uniquify-kNN* (Figure 1 (3)), removes duplicates from the retrieved kNN tokens so that no token can be dominant and thus more diverse candidates remain.

Our experiments showed that our proposed methods alleviate the overcorrection problem that leads to the generation of more diverse candidates, and maintain fluency and oracle translation quality in multiple domains and language pairs. We also show that the degree of diversity can be controlled by changing the perturbation's magnitude, which benefits end-applications, e.g., human post-editing.

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 9–31

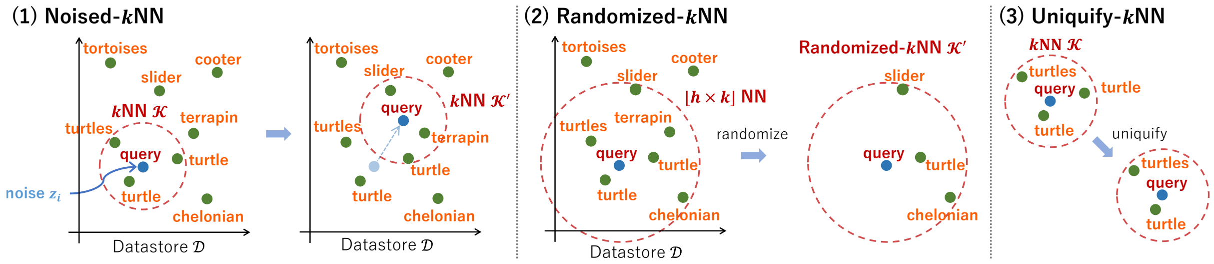


Figure 1: Overview of our proposed method: See §3 for details. Green points represent target tokens in datastore. Blue points represent query vectors, and surrounding circles denote retrieved neighbors. (1) Noised-kNN adds a noise vector to the query, changing the retrieved tokens. (2) Randomized-kNN initially retrieves more neighbors and randomly selects k-neighbors. (3) Uniquify-kNN only considers unique target tokens from retrieved neighbors. In this figure, number of neighbors k per query is set to 3, and hyperparameter h of (2) Randomized-kNN is set to 2.0.

2 Related Work

2.1 Diverse Text Generation

Given the importance of generating diverse translations, many of the proposed search algorithm variations can be categorized into either deterministic or stochastic types.

For the former method, Vijayakumar et al. (2018) proposed diverse beam search (DBS) in which beams are divided into several groups, and a modified score function penalizes the overlapped tokens among the groups. Freitag and Al-Onaizan (2017) proposed a method that determines the maximum number of candidates that share the same partial hypothesis.

As for the latter approach, top-k sampling (Fan et al., 2018) randomly samples the output tokens from the top k-tokens with the highest likelihood at each time step. Similarly, nucleus sampling (Holtzman et al., 2020) randomly samples from the smallest subset of candidates whose total likelihood exceeds p at each time step. Noisy parallel approximate decoding (Cho, 2016) explores multiple modes by injecting noise into the model's hidden states. Wu et al. (2020) proposed a method that samples different models derived by applying concrete dropout. MixDiversity (Li et al., 2021) leverages the hidden representations of the randomly sampled sentence pairs from the subset of the training corpus.

Although all of the above methods diversify the output text, they do not explicitly address the overcorrection problem, which is the root cause of the limited diversity (§2.3).

2.2 kNN-MT

Khandelwal et al. (2021) proposed k-nearest neighbor machine translation (kNN-MT), which uses kNN search for retrieving similar examples during inference by reflecting the retrieval results in the score function. The translation quality is improved by allowing the model to directly access large-scale cached translation examples. kNN-MT consists of two steps, datastore creation and generation.

Datastore creation Before inferences with kNN-MT, we need to create a datastore, i.e., key-value pairs of high-dimensional representations and tokens. We feed all the training data into the NMT model and save each target token as a value and its decoder hidden state vector as a key representation. Formally, let $f(\boldsymbol{x}, \boldsymbol{y}_{<i})$ be the hidden state vector at time step i for source sentence $\boldsymbol{x} \in \mathcal{S}$ and target sentence $\boldsymbol{y} \in \mathcal{T}$ of training data $(\mathcal{S}, \mathcal{T})$, and then datastore $\mathcal{D}$ can be represented:

$$\mathcal{D} = \{(f(\boldsymbol{x}, \boldsymbol{y}_{<i}),\ y_i),\ \forall y_i \in \boldsymbol{y} \mid (\boldsymbol{x}, \boldsymbol{y}) \in (\mathcal{S}, \mathcal{T})\}. \quad (1)$$

Generation To generate a sentence from a given input sentence $\boldsymbol{x}$, we extract k-nearest neighbors $\mathcal{K} \subset \mathcal{D}$ from the datastore using decoder hidden state $\boldsymbol{q}_i$ at time step i as a query corresponding to output token y_i. k-nearest neighbor probability $p_{k\mathrm{NN}}$ is calculated from the distances between query $\boldsymbol{q}_i$ and the k-nearest neighbors:

$$p_{k\mathrm{NN}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i}) \propto \sum_{(\boldsymbol{k}_j, v_j) \in \mathcal{K}} \mathbb{1}_{y_i = v_j} \exp\left(\frac{-\mathrm{dist}(\boldsymbol{k}_j, \boldsymbol{q}_i)}{\tau}\right), \quad (2)$$

where $\mathrm{dist}(\cdot, \cdot)$ is a distance function and τ is a softmax temperature parameter. The word probability of y_i is calculated by the linear interpolation of kNN probability $p_{k\mathrm{NN}}$ and output probability

p_{MT} of the NMT model:

$$p(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i}) = \lambda p_{k\mathrm{NN}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i}) + (1-\lambda) p_{\mathrm{MT}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i}), \quad (3)$$

where λ is a hyperparameter that determines the weight of the kNN probability.

kNN-MT substantially improves the translation performance without additional model training, and several variants have been proposed. Jiang et al. (2021); Zheng et al. (2021); Jiang et al. (2022) further improved the translation performance by dynamically changing the number of neighbors and the interpolation weight. Wang et al. (2022); Meng et al. (2022); Deguchi et al. (2023) proposed methods for faster inference by reducing search space. However, no research uses kNN-MT for improving generation diversity.

2.3 Overcorrection

The standard NMT models trained with cross-entropy loss suffer from the overcorrection problem (Zhang et al., 2019) in which the model underestimates a prediction that is largely different from the training data, even if it is likely. We hypothesize that this problem decreases the diversity of candidates due to the low probabilities for alternative tokens assigned by the underlying model.

Zhang et al. (2019) alleviated overcorrection by mitigating the discrepancy between training and inference. Yang et al. (2022) argued that the kNN-MT's improvement is derived from alleviating the overcorrection problem by a kNN search. However, the relationship between overcorrection and generation diversity remains unclear. In this study, we propose expanding the search space of kNN-MT to alleviate overcorrection. We also conduct a quantitative analysis of overcorrection and diversity (§5.2).

3 kNN Diversified Decoding

We propose to employ kNN-MT to alleviate the overcorrection problem and thus encourage diverse generation model-wise. It is further combined with diversified decoding together with our proposed stochastic and deterministic methods for the more controlled expansion of the search space in kNN-MT.[1]

[1]We combine these methods by calculating word probability with vanilla or perturbed kNN-MT and generate candidates by using diversified decoding methods as a search strategy.

In kNN-MT, kNN search is expected to improve the output probability of alternative tokens that are not normally included in the top-N of the output probability. Furthermore, the search space is extensively explored using a diversified decoding method to generate diverse and likely translation candidates. Although kNN search is limited by k, more space is explored by stochastically expanding it by adding perturbations from noising (§3.1) and randomizing (§3.2). In addition, deterministically considering only unique tokens in neighbors further allows the model to explore alternative candidates (§3.3).

3.1 Noised-kNN

As a simple way to perturb the kNN distribution, we propose noised-kNN, a method that adds a noise vector to the query for a kNN search (Figure 1 (1)). This method diversifies the candidates by stochastically extending the range of the kNN search. In this method, we perform a kNN search with query $\boldsymbol{q}_i + \boldsymbol{z}_i$, where $\boldsymbol{q}_i$ is the hidden decoder states, and $\boldsymbol{z}_i$ is the noise vector for output token y_i to obtain k-nearest neighbors $\mathcal{K}'$. We then compute the kNN probability from $\mathcal{K}'$ in Eq. 2. Noise vector $\boldsymbol{z}_i$ is generated independently at each time and for each beam as the white Gaussian noise of norm $|a|$ where $a \sim \mathcal{N}(m, s^2)$ with mean m and variance s^2. We propose the following two methods to set m and s.

Static noise We introduce static noise by setting $m = h_m, s = h_s$ using hyperparameters h_m, h_s. Hyperparameters h_m and h_s should be set to appropriate values based on the statistics of the datastore. In this study, we computed the mean and variance of the distance to the nearest neighbors on the validation data in advance.

Adaptive noise As an alternative to static noise, we introduce adaptive noise in which the magnitude of the noises is computed on the fly for each query at each decoding step. Specifically, a usual kNN search is performed to obtain maximum d_{max} and standard deviation d_{std} of the distances to the neighbors. Then an actual noisy kNN search is performed by setting $m = h'_m \times d_{\mathrm{max}}$ and $s = h'_s \times d_{\mathrm{std}}$ using hyperparameters h'_m, h'_s. This method's benefit is that the magnitude of the noise is determined on the fly and eliminates the need for the prior computation of the datastore distributions at the cost of an additional kNN search at each decoding step.

3.2 Randomized-kNN

Randomized-kNN, as described in Figure 1 (2), stochastically samples a portion of the expanded neighbors to alleviate the drawbacks of two noising approaches (§3.1) that demand prior computation of parameters m and s. $\lfloor h \times k \rfloor$ neighbors are retrieved where h is a hyperparameter satisfying $h > 1$, and randomized k-nearest neighbors $\mathcal{K}'$ are obtained by uniformly randomly sampling k from the $\lfloor h \times k \rfloor$ neighbors. This method is expected to diversify the candidates because it includes more neighbors in the search space. We do not need to collect any statistics of the distribution of the distances from the query to the k-nearest neighbors in advance because we do not perturb the query itself. In addition, since this method requires only one kNN search at each time step, it is identical to the vanilla kNN-MT.

3.3 Uniquify-kNN

The perturbations in §3.1 and §3.2 may have a limited effect on increasing diversity when duplicated tokens are retrieved from the nearest neighbors. We alleviate this problem by introducing uniquify-kNN in which duplicated tokens are explicitly removed from the neighbors (Figure 1 (3)).

Since the datastore accumulates all the tokens on target-side of the training data, the k-nearest neighbors retrieved from the datastore can contain duplicated tokens. As seen in Eq. 2, their distance scores are accumulated for duplicated tokens, creating a spuriously dominant probability mass in the neighbor distribution. Biased probabilities can negatively impact diversity. Since a larger datastore implies more potential for overlapped tokens, it would further degrade the diversity.

To address this issue, after retrieving the k-nearest neighbors, we propose uniquify-kNN, a method that eliminates the duplicate tokens from the neighbors, leaving only unique tokens that are closest to the query. Our new method is formally defined as follows:

$$p_{k\mathrm{NN}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i}) \propto \max_{(\boldsymbol{k}_j, v_j)\in\mathcal{K}} \mathbb{1}_{y_i=v_j} \exp\left(\frac{-\mathrm{dist}(\boldsymbol{k}_j, \boldsymbol{q}_i)}{\tau}\right). \quad (4)$$

This operation prevents the kNN probability from becoming peaky and decreasing in diversity.

4 Experiments

We experimentally confirmed whether our method can generate diverse translation candidates.

4.1 Experimental Settings

4.1.1 Dataset

The experiments are divided into a domain adaptation setting and a general-domain setting. In the domain adaptation setting, we used German-English (De-En) and Japanese-English (Ja-En) language pairs. For De-En, we used five domain data (Koehn and Knowles, 2017; Aharoni and Goldberg, 2020): Koran, IT, Medical, Law, and Subtitles. For Ja-En, we used four domain data: the Asian Scientific Paper Excerpt Corpus (ASPEC; Nakazawa et al., 2016), the Kyoto Free Translation Task (KFTT; Neubig, 2011), TED talks (Cettolo et al., 2012), and the Business Scene Dialogue corpus (BSD; Rikters et al., 2019). We used the designated test set for each domain.

In the general-domain setting, we used three language pairs: WMT'19 news task data (Barrault et al., 2019) for German-English (De-En) and WMT'22 general task data (Kocmi et al., 2022) for Japanese-English (Ja-En) and Ukrainian-Czech (Uk-Cs). For the general-domain test set, we used newstest2019 for De-En and generaltest2022 for Ja-En and Uk-Cs. The statistics of the dataset for both settings are in Appendix A.1.

4.1.2 Models

Baseline All our experiments were carried out with Transformer models (Vaswani et al., 2017). In the domain adaptation and general domain for De-En, we used the WMT'19 De-En pre-trained model (Ng et al., 2019) available for the `fairseq` toolkit (Ott et al., 2019). In the domain adaptation for Ja-En, we used the Transformer Big model trained on JParaCrawl v3.0 (Morishita et al., 2022) as a base model.[2] In the general domain for Ja-En and Uk-Cs, we used Transformer Big models trained on WMT'22 data as a base model for each language pair. These models were used for datastore creation and as baseline models. In all the experiments, the beam size was set to 20.

kNN-MT We used `FAISS` (Johnson et al., 2019) for datastore creation and kNN search. The detailed settings are described in Appendix A.2.

[2]We did not use WMT'22 data for the domain adaptation settings for fair comparisons since it includes KFTT, which is one target domain.

Diversified decoding We used DBS and nucleus sampling (Nucleus) as the diversified decoding method; the number of DBS groups was set to 20, the diversity strength was set to 0.5, and hyperparameter p of Nucleus was tuned with the validation data. For our proposed methods, we combined them with DBS and Nucleus.[3] The hyperparameters of the proposed methods were tuned with the validation data. The detailed settings are in Appendix A.2.

4.1.3 Evaluation Metrics

We used the following metrics to confirm how correctly our model translates and how diverse its candidates are.[4]

BLEU@N is a variant of corpus-wise BLEU (Papineni et al., 2002) computed by the largest sentence-level BLEU score (Chen and Cherry, 2014) for each N-best candidate, also known as oracle BLEU. It corresponds to the upper bound of performance through N-best reranking. We report BLEU@1 and BLEU@20 in our experiment. Note that BLEU@1 is a standard BLEU.

MergedBLEU@N is a variant of BLEU@N computed on the merged outputs from two systems. We employ MergedBLEU@40, which merges 20 candidates from the baseline and a diversified method. The higher MergedBLEU@40 than BLEU@20 of the baseline implies that the diversified method helps generate the better translations.

Diversity The BLEU-based discrepancy metric (DP; Shu et al., 2019) is a measure of the diversity. DP captures how many unique n-grams are included in each candidate sentence, where a higher DP indicates the candidates are diverse.[5]

Diversity and translation quality The diversity enhancement per quality (DEQ; Sun et al., 2020) measures the quality-diversity trade-off. We adapt the DEQ for our experimental settings by using kNN-MT as our base:

$$\mathrm{DEQ} = -\frac{\mathrm{DP}_{\mathrm{base}} - \mathrm{DP}_{\mathrm{sys}}}{\mathrm{RefBLEU}_{\mathrm{base}} - \mathrm{RefBLEU}_{\mathrm{sys}}} \quad (5)$$

where $\mathrm{DP}_{\mathrm{sys}}$ and $\mathrm{DP}_{\mathrm{base}}$ are DP of the evaluated system and kNN-MT, respectively, and $\mathrm{RefBLEU}_{\mathrm{sys}}$ and $\mathrm{RefBLEU}_{\mathrm{base}}$ refer to reference BLEU (RefBLEU; Sun et al., 2020), the average corpus-wise BLEU across all translation candidates, of the evaluated system and kNN-MT, respectively. The DEQ will be higher if the evaluated system achieves a better quality-diversity trade-off.

Fluency The pseudo-log-likelihood score (PLL; Salazar et al., 2020) is a metric of fluency using the MLM model.[6] We defined a variant of the PLL for the entire output translations, named SPLL, using statistical function stat:

$$\mathrm{SPLL}(\mathbb{W}) = \frac{1}{|\mathbb{W}|} \sum_{\mathbf{B} \in \mathbb{W}} \underset{\hat{\boldsymbol{y}} \in \mathbf{B}}{\mathrm{stat}} \left(\frac{1}{|\hat{\boldsymbol{y}}|} \mathrm{PLL}(\hat{\boldsymbol{y}}) \right), \quad (6)$$

where $\mathbb{W} = \{\mathbf{B}_1, \ldots, \mathbf{B}_M\}$ is system output, $\mathbf{B}_k = \{\hat{\boldsymbol{y}}_k^1, \ldots, \hat{\boldsymbol{y}}_k^N\}$ is the set of N-best hypotheses for a source sentence $\boldsymbol{x}_k \in \mathcal{X}$, and $\mathcal{X}$ is a test set with M sentences.

In the experiment, to investigate the variances in fluency, we use MaxPLL, MinPLL, and MeanPLL, which use $\max$, $\min$, and mean functions for the stat of SPLL. We also compute the reference's MeanPLL to check how practical the translations' fluency are. If the generated texts are not as fluent as the reference, the MeanPLL will be lower than the reference.

4.2 Experimental Results

4.2.1 Domain Adaptation

Summaries of the De-En and Ja-En results are shown in Table 1 and Table 2, respectively, by averaging the metrics across the domains. Detailed results are shown in Appendix B.

In De-En, our proposed DBS+kNN-MT and Nucleus+kNN-MT outperformed both of DP and oracle BLEU of DBS and Nucleus. These methods also decrease BLEU@20 more than kNN-MT, although the drop in performance is comparable to that observed between Baseline and DBS or Nucleus. Our perturbation methods, i.e., +Adaptive, +Static, and +Randomize, drastically improved DP while maintaining comparable performance to DBS+ and Nucleus+kNN-MT under BLEU@20. Nor did the PLL of the proposed methods suffer substantial drops when compared to kNN-MT; the differences are marginal compared to the PLL of

[3] From preliminary experiments, we describe the uniquify-kNN results in the general-domain setting.

[4] The detailed settings are described in Appendix A.3. We also used COMET and BERTScore, but since these scores tend to be similar to BLEU, we show the details and results for these metrics in Appendix B.

[5] As an additional diversity metric, we discuss the number of differences in n-gram type in §5.3.

[6] We used a multilingual BERT (Devlin et al., 2019) as the MLM model.

Method	Diversity DP ↑	Translation Quality (BLEU ↑) @1	@20	Merged@40	Ref	Both DEQ ↑	Fluency (PLL ↑) Max	Min	Mean
Reference	-	-	-	-	-	-	-	-	-3.35
Baseline	31.4	34.1	42.6	42.6	30.9	-0.12	-2.26	-4.55	$\mathbf{-3.28}$
DBS	35.9	33.6	40.0	43.8	30.3	0.44	-2.23	-4.63	$\mathbf{-3.28}$
Nucleus	48.0	33.4	42.1	44.6	30.0	1.88	-2.31	$\mathbf{-4.42}$	-3.29
kNN-MT	32.3	**43.2**	**51.8**	**53.5**	**38.4**	-	-2.23	-4.74	-3.32
DBS+kNN-MT	42.0	42.0	48.6	51.8	36.5	5.28	-2.18	-4.90	-3.35
+Static	55.2	40.4	49.0	52.0	33.5	4.68	$\mathbf{-2.02}$	-5.23	-3.37
+Adaptive	53.7	41.0	49.0	52.1	34.2	5.10	-2.04	-5.21	-3.38
+Randomize	54.4	39.5	48.4	51.5	32.6	3.81	-2.08	-5.16	-3.38
Nucleus+kNN-MT	51.6	42.1	50.4	52.8	37.0	**14.5**	-2.37	-4.50	-3.33
+Static	55.0	42.7	49.9	52.5	34.9	6.47	-2.29	-4.87	-3.36
+Adaptive	55.6	42.6	49.8	52.4	34.7	6.32	-2.27	-4.92	-3.36
+Randomize	**59.4**	42.3	49.2	52.0	33.1	5.09	-2.24	-5.10	-3.41

Table 1: Domain adaptation in German-English: We report averages of five domains.

Method	Diversity DP ↑	Translation Quality (BLEU ↑) @1	@20	Merged@40	Ref	Both DEQ ↑	Fluency (PLL ↑) Max	Min	Mean
Reference	-	-	-	-	-	-	-	-	-2.75
Baseline	38.0	18.1	26.0	26.1	16.5	0.25	-1.75	-3.67	-2.55
DBS	54.9	17.2	24.8	28.2	14.4	3.88	-1.66	-3.92	-2.63
Nucleus	63.9	17.6	26.5	28.9	14.8	6.43	-1.69	-3.72	-2.60
kNN-MT	37.4	**20.9**	**29.7**	31.5	**18.9**	-	-1.65	$\mathbf{-3.57}$	$\mathbf{-2.43}$
DBS+kNN-MT	60.7	19.7	27.9	30.9	15.6	7.04	-1.44	-4.02	-2.52
+Static	66.5	19.5	28.3	31.3	14.9	7.23	-1.36	-4.12	-2.51
+Adaptive	66.8	19.5	28.4	31.3	14.8	7.18	-1.36	-4.11	-2.51
+Randomize	65.9	19.2	27.8	30.9	14.6	6.61	$\mathbf{-1.35}$	-4.13	-2.51
Nucleus+kNN-MT	66.6	20.3	29.0	**31.7**	16.6	12.9	-1.60	-3.61	-2.49
+Static	64.0	20.5	28.9	31.5	16.8	12.9	-1.56	-3.65	-2.46
+Adaptive	64.0	20.6	28.8	31.5	16.9	**13.2**	-1.57	-3.65	-2.46
+Randomize	**74.8**	20.3	28.7	31.4	14.6	8.71	-1.50	-4.16	-2.59

Table 2: Domain adaptation in Japanese-English: We report averages of four domains.

Reference. In Ja-En, the proposed methods improved DP like in De-En without any Merged-BLEU@40 loss.[7] The PLL of the proposed methods is also comparable to Baseline.

We observed almost no substantial differences for the perturbation types. +Static requires prior estimation on distance metrics, and +Adaptive needs an additional kNN search for each time step for the inferences. Therefore, +Randomize is the best choice since it overcomes both drawbacks.

These results indicate that the proposed methods improved the diversity without lowering the fluency and maintained oracle translation quality on some domains.

Trade-off between quality and diversity We observed our proposed methods suffered from a quality-diversity trade-off (Ippolito et al., 2019; Zhang et al., 2021), i.e., our methods improve diversity (DP) but decrease average translation quality (RefBLEU). However, all of our proposed DBS- and Nucleus-based methods outperformed the DEQ of DBS and Nucleus.[8] Thus, our methods achieved better quality-diversity trade-offs than the existing methods.

4.2.2 General domain

Table 3 summarizes the general-domain results obtained by averaging the metrics across the language pairs. Detailed results are shown in Appendix B.

The proposed DBS+ and Nucleus+kNN-MT slightly improved DP, and the MergedBLEU@40 and fluency are comparable to DBS and Nucleus. The effect of stochastic perturbations for DP was limited, especially on Nucleus-based, but +Uniquify substantially improved DP, and Merged-BLEU@40 and fluency preserved comparable re-

[7] We also evaluate the oracle BLEU for 40 candidates (BLEU@40) to compare to MergedBLEU@40, and the results are in Appendix C.1.

[8] kNN-MT is not comparable to our methods because the base of the DEQ is kNN-MT.

Method	Diversity DP ↑	Translation Quality (BLEU ↑) @1	@20	Merged@40	Ref	Both DEQ ↑	Fluency (PLL ↑) Max	Min	Mean
Reference	-	-	-	-	-	-	-	-	−2.83
Baseline	37.5	30.2	41.2	41.2	27.6	1.24	−1.94	−3.85	−2.77
DBS	51.3	28.9	37.9	43.1	24.3	3.97	−1.78	−4.04	−2.78
Nucleus	62.8	29.2	40.4	44.2	23.7	6.16	−1.78	−3.99	−2.76
kNN-MT	37.3	**30.5**	**41.4**	42.6	**27.8**	-	−1.92	−3.85	−2.76
DBS+kNN-MT	52.6	29.1	38.2	43.4	24.2	4.22	−1.70	−4.09	−2.76
+Static	54.8	29.0	38.3	43.4	23.8	4.38	−1.66	−4.16	−2.75
+Adaptive	54.3	29.0	38.3	43.4	23.9	4.34	−1.67	−4.15	−2.75
+Randomize	53.9	29.0	38.2	43.4	23.9	4.29	−1.66	−4.14	−2.75
+Uniquify	54.9	28.8	37.8	43.2	23.5	4.08	−1.69	−4.18	−2.78
+Static	55.9	28.8	37.7	43.1	23.3	4.09	−1.65	−4.24	−2.77
+Adaptive	55.7	28.8	37.8	43.2	23.3	4.10	−1.65	−4.22	−2.78
+Randomize	55.8	28.8	37.7	43.2	23.3	4.09	**−1.64**	−4.25	−2.78
Nucleus+kNN-MT	64.5	29.1	40.5	**44.3**	23.5	6.22	−1.73	−4.02	−2.75
+Static	52.4	30.2	38.9	43.4	25.8	**7.25**	−1.85	**−3.80**	**−2.73**
+Adaptive	52.8	30.0	38.9	43.4	25.5	6.75	−1.86	−3.83	−2.74
+Randomize	62.3	29.8	38.9	43.5	23.4	5.64	−1.76	−4.13	−2.77
+Uniquify	**70.8**	28.0	39.3	44.0	21.0	4.88	−1.70	−4.16	−2.78
+Static	55.4	29.8	38.7	43.3	24.9	6.09	−1.84	−3.88	−2.75
+Adaptive	55.7	29.9	38.7	43.4	24.8	6.05	−1.84	−3.92	−2.76
+Randomize	67.7	29.5	38.4	43.3	21.6	4.86	−1.75	−4.44	−2.84

Table 3: General domain: We report averages of three language pairs.

sults. As in the domain-adaptation setting, the DEQ of our methods outperformed existing methods.

These experiments show that the proposed methods achieve better quality-diversity trade-offs without any fluency loss.

5 Analysis

5.1 Tuning kNN Diversified Decoding

We investigated how the hyperparameters of our proposed method affect its performance. Figure 2 shows the relationship between DP and BLEU@20 in the De-En IT domain.[9] The results show that +Randomize outperformed the diversity of DBS+kNN-MT while maintaining oracle translation quality with some hyperparameters, indicating that our proposed method can adjust DP and BLEU by varying the magnitude of the perturbation.[10]

5.2 Overcorrection Analysis

We hypothesized that the overcorrection problem discourages the generation of diverse candidates that is alleviated by our proposed methods. To verify the hypothesis, we evaluated how well our methods mitigate the overcorrection problem and clarified the relationship between overcorrection and diversity.

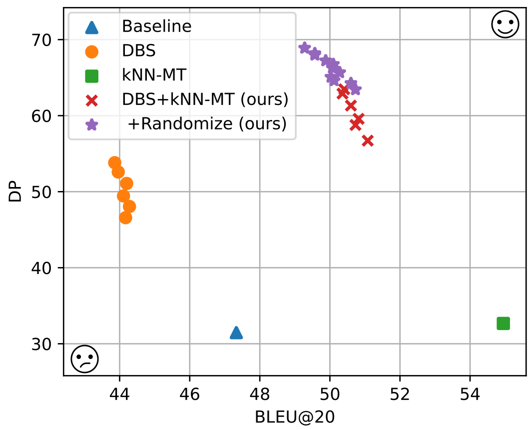


Figure 2: Relationship between translation quality (BLEU@20) and diversity (DP) in De-En IT domain: Top-right is most desirable.

Overcorrection is a phenomenon in which the likelihoods of valid translations are underestimated by a model. Therefore, a model that suffers less from the issue will assign a similar likelihood to valid translations that only have small differences. Thus, we propose a mean of the absolute difference in the log-likelihoods (MADLL) of two reference translations as a metric that quantifies the degree of overcorrection, in which a lower MADLL value implies a decreased likely of suffering from

[9]For DBS and DBS+kNN-MT, we varied DBS's diversity strength by 0.1 in the range of $[1.5, 2.0]$. For +Randomize, we used 1.5 for diversity strength and varied perturbation's magnitude h by 0.1 in the range of $[1.5, 2.5]$.

[10]Further analysis of the relationship between the hyperparameters and DP/BLEU of our methods is in Appendix C.2.

	MADLL ↓	DP ↑	BLEU@1 ↑	BLEU@20 ↑
Method	WMT'21 (newstest2021)			
Baseline	0.695	41.0	29.5 / 36.0	38.2 / 45.4
kNN-MT	0.683	41.1	**30.1 / 36.8**	**38.7 / 45.8**
+Uniq	**0.657**	43.6	29.6 / 36.2	38.0 / 45.5
+Rand	0.660	**43.9**	29.6 / 36.3	37.8 / 44.7
Method	WMT'22 (generaltest2022)			
Baseline	0.712	42.2	30.3 / 34.5	38.3 / 43.1
kNN-MT	0.714	42.3	**30.6 / 35.0**	**38.6 / 43.5**
+Uniq	**0.696**	44.9	30.5 / 34.7	38.2 / 42.9
+Rand	0.702	**45.2**	30.5 / 34.7	38.0 / 42.5

Table 4: Overcorrection analysis on newstest2021 and generaltest2022 in De-En: MADLL is an indicator where a lower score denotes less likely to suffer from overcorrection. DP and BLEUs are scores when DBS is used as the decoding method. BLEU is written in the form of scores for refA/refB. Uniq and Rand are abbreviations for Uniquify and Randomize, respectively.

overcorrection issue.[11] We evaluated the proposed methods on the test data of WMT'21 De-En (newstest2021) and WMT'22 De-En (generaltest2022) in the De-En general-domain setting. These test data have two reference translations (refA/refB) for one source sentence, and we report the MADLL between refA and refB by forced decoding.

Table 4 shows the relationship between overcorrection, diversity, and translation quality. The proposed methods have lower MADLL and higher DP scores than Baseline and kNN-MT for both WMT'21 and WMT'22. We also found that BLEUs of the Baseline and kNN-MT are almost comparable to the proposed methods. This implies that the proposed methods managed to resolve overcorrection and improved diversity while almost maintaining the translation quality.

5.3 Counting Distinct n-grams

In §4, we used DP as a diversity metric. DP captures how many unique n-grams are included in each candidate. In order to evaluate the diversity of translation candidates of our proposed methods from a different perspective, we employed another metric: the number of distinct n-grams, which measures the richness of vocabulary and phrases across the entire N-best list. We calculated the ratio of the number of distinct n-grams to the total number of n-grams for $n \in \{1, 2, 3, 4\}$.

[11] We report MADLL along with BLEU because it is easy to improve only MADLL but hard to improve both (if the model assigns the same likelihood to all sentences, MADLL will be zero, but BLEU will be substantially affected).

Method	Ratio of distinct n-grams (%) ↑ $n=1$	$n=2$	$n=3$	$n=4$
Baseline	1.6	7.8	14.1	18.5
DBS	1.6	8.8	16.9	22.2
DBS+kNN-MT	1.7	9.7	18.6	24.8
+Randomize	2.1	12.4	24.7	33.4
Nucleus	2.0	11.1	23.1	32.8
Nucleus+kNN-MT	1.9	11.9	25.9	37.1
+Randomize	**2.6**	**16.0**	**32.2**	**42.8**

Table 5: The ratio of the number of distinct n-grams to the total number of n-grams in German-English domain adaptation setting: We report averages of five domains.

#neighbors	h	$\lfloor h \times k \rfloor$	DP ↑	BLEU@1 ↑	BLEU@20 ↑
(1) DBS+kNN-MT+Randomize					
64	2	128	57.1	41.5	50.8
64	3	192	62.3	39.3	50.0
64	4	256	**65.6**	38.1	49.3
(2) DBS+kNN-MT					
128	-	-	44.2	**43.9**	51.1
192	-	-	44.1	**43.9**	**51.3**
256	-	-	44.2	**43.9**	51.1

Table 6: Effectiveness of *Randomize* on De-En IT domain based on DBS+kNN-MT: We compared (1) randomize k from $\lfloor h \times k \rfloor$ neighbors and (2) set number of neighbors per query to $\lfloor h \times k \rfloor$.

The ratio averages in the De-En domain adaptation setting are shown in Table 5. DBS+ and Nucleus+kNN-MT increased the ratio of the number of distinct n-grams more than DBS and Nucleus; the ratio increased substantially when perturbation was applied to it. The results show that our proposed methods generate translation candidates with more diverse vocabulary and phrases compared to the baselines.

5.4 Effectiveness of Randomization

We conducted an ablation study to investigate the effectiveness of *Randomize* on the Randomized-kNN. In the Randomized-kNN, the search space is stochastically expanded by uniformly and randomly sampling k from $\lfloor h \times k \rfloor$ neighbors to diversify the translations. We compared the following two methods to investigate the effectiveness of *Randomize*: (1) randomizing k from $\lfloor h \times k \rfloor$ neighbors with DBS+kNN-MT, i.e., DBS+kNN-MT+Randomize, and (2) retrieving $\lfloor h \times k \rfloor$ neighbors without randomizing on DBS+kNN-MT i.e., setting the number of neighbors per query of DBS+kNN-MT to $\lfloor h \times k \rfloor$.

A comparison for the De-En IT domain is pre-

Test Input: コロナに関しまして。 **Reference**: *I have a question about COVID.*	
DBS+kNN-MT+Randomize	**DBS+kNN-MT**
About corona.	*Regarding corona.*
With regards to corona.	*About corona.*
About COVID-19.	*It is about corona.*
Regarding corona.	*We are talking about corona.*
With regards to COVID-19.	*With regards to corona.*
⋮	⋮
Test Input: *Spring Summer*コレクションもセール対象商品*!* **Reference**: *The spring/summer collection is also included in the sale!*	
DBS+kNN-MT+Randomize	**DBS**
The Spring Summer collection is also a sale target product!	*The Spring Summer collection is also a sale target product!*
Items from the Spring Summer collection are also on sale!	*The Spring Summer collection is also a sale item!*
The Spring Summer collection is also a sale target product!	*The Spring Summer collection is also a sale eligible product!*
⋮	⋮
winter collection is also a sale target product!	*Our Spring Summer collections are on sale!*
The Spring Summer collection is also eligible for sale.	*The Spring Summer collection is also eligible for sale!*
Summer collection is also a sale target product!	*The Spring Summer Collection is also included in the sale!*

Figure 3: Example 20-best lists using DBS-based methods: In upper example, DBS+kNN-MT+Randomize successfully diversified list by adding a likely word, *COVID-19*, which did not appear in DBS+kNN-MT. In lower example, DBS+kNN-MT+Randomize introduced an unlikely word, *winter*, which did not appear in DBS.

sented in Table 6, where simply increasing the number of neighbors per query of DBS+kNN-MT did not improve diversity. *Randomize* from more neighbors is important for improving diversity.

5.5 Case Study

To better understand our proposed method through case studies, Figure 3 shows two qualitative examples in the general domain of Ja-En. We omitted some parts for brevity, and a full version is shown in Figure 5 in Appendix C.3.

In the upper example, Randomized-kNN improved the diversity of the candidates, which include the appropriate word *COVID-19*. This candidate never appeared in the 20-best list generated by DBS+kNN-MT, suggesting that considering more likely tokens by +Randomize with a broader search space improves diversity and maintains translation quality.

The example at the bottom shows increased diversity but also decreased translation quality, where translation *winter* is output for *spring/summer*, which does not appear in the DBS-generated candidates. Such antonyms as *winter*, *spring*, and *summer* tend to appear in the neighbors of word embedding space (Mrkšić et al., 2016), which is the primary cause of incorrect retrieval from the datastore in the broader kNN search space. We leave it as our future work of addressing the problem of retrieving unlikely words by a stochastically expanded kNN search.

6 Conclusion

We proposed methods to generate more diverse translation candidates by expanding the search space of kNN-MT. We experimentally showed that our proposed methods alleviated the overcorrection problem and outperformed the existing baselines in diversity, and also controlled the diversity and translation quality by changing the perturbation's magnitude.

Limitations

Our proposed method improves diversity by utilizing kNN-MT. Unfortunately, kNN-MT suffers from the drawbacks of high inference latency for kNN searches and requires much memory to load the datastore. Our proposed method is applicable not only to vanilla-kNN but also to many other variants; if a model is proposed in the future that solves these issues, we can combine our method with new kNN-MT variants to overcome these drawbacks.

Although our proposed method improves diversity, it might generate hallucinations, which are incorrect but fluent translations. This problem can be alleviated by filtering hallucinations by post-processing, an approach we leave for the future.

We also might need to consider the trade-off between diversity and quality depending on downstream applications, as in a number of experiments.

We showed the effectiveness of our proposed methods by evaluating the diversity and oracle translation quality, but the benefit in end-applications remains unclear. Li and Jurafsky (2016) implied that the higher diversity of translation candidates promotes the higher translation quality after reranking. Thus, the benefit in downstream applications can be shown by measuring the performance after using a reranking method such as quality-aware decoding (Fernandes et al., 2022).

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers JP21H05054, JP21K17801, and JP23H03458.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation*, pages 261–268, Trento, Italy. European Association for Machine Translation.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367, Baltimore, Maryland, USA. Association for Computational Linguistics.

Kyunghyun Cho. 2016. Noisy parallel approximate decoding for conditional recurrent language model. *arXiv preprint arXiv:1605.03835*.

Hiroyuki Deguchi, Taro Watanabe, Yusuke Matsui, Masao Utiyama, Hideki Tanaka, and Eiichiro Sumita. 2023. Subset retrieval nearest neighbor machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–189, Toronto, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. 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 and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60, Vancouver. Association for Computational Linguistics.

Kevin Gimpel, Dhruv Batra, Chris Dyer, and Gregory Shakhnarovich. 2013. A systematic exploration of diversity in machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1100–1111, Seattle, Washington, USA. Association for Computational Linguistics.

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

Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. 2019. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3752–3762, Florence, Italy. Association for Computational Linguistics.

Hui Jiang, Ziyao Lu, Fandong Meng, Chulun Zhou, Jie Zhou, Degen Huang, and Jinsong Su. 2022. Towards robust k-nearest-neighbor machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5468–5477, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qingnan Jiang, Mingxuan Wang, Jun Cao, Shanbo Cheng, Shujian Huang, and Lei Li. 2021. Learning kernel-smoothed machine translation with retrieved examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7280–7290, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

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, Maja Popović, and Mariya Shmatova. 2022. Findings of the 2022 conference on machine translation (wmt22). In *Proceedings of the Seventh Conference on Machine Translation*, pages 1–45, Abu Dhabi. 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.

Jicheng Li, Pengzhi Gao, Xuanfu Wu, Yang Feng, Zhongjun He, Hua Wu, and Haifeng Wang. 2021. Mixup decoding for diverse machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 312–320, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiwei Li and Dan Jurafsky. 2016. Mutual information and diverse decoding improve neural machine translation. *arXiv preprint arXiv:1601.00372*.

Yuxian Meng, Xiaoya Li, Xiayu Zheng, Fei Wu, Xiaofei Sun, Tianwei Zhang, and Jiwei Li. 2022. Fast nearest neighbor machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 555–565, Dublin, Ireland. Association for Computational Linguistics.

Makoto Morishita, Katsuki Chousa, Jun Suzuki, and Masaaki Nagata. 2022. JParaCrawl v3.0: A large-scale English-Japanese parallel corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6704–6710, Marseille, France. European Language Resources Association.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California. Association for Computational Linguistics.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASPEC: Asian scientific paper excerpt corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2204–2208, Portorož, Slovenia. European Language Resources Association (ELRA).

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

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

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

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.

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.

Matīss Rikters, Ryokan Ri, Tong Li, and Toshiaki Nakazawa. 2019. Designing the business conversation corpus. In *Proceedings of the 6th Workshop on Asian Translation*, pages 54–61, Hong Kong, China. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Raphael Shu, Hideki Nakayama, and Kyunghyun Cho. 2019. Generating diverse translations with sentence codes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1827, Florence, Italy. Association for Computational Linguistics.

Zewei Sun, Shujian Huang, Hao-Ran Wei, Xin-yu Dai, and Jiajun Chen. 2020. Generating diverse translation by manipulating multi-head attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8976–8983.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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).

Dexin Wang, Kai Fan, Boxing Chen, and Deyi Xiong. 2022. Efficient cluster-based k-nearest-neighbor machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2175–2187, Dublin, Ireland. Association for Computational Linguistics.

Xuanfu Wu, Yang Feng, and Chenze Shao. 2020. Generating diverse translation from model distribution with dropout. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1088–1097, Online. Association for Computational Linguistics.

Zhixian Yang, Renliang Sun, and Xiaojun Wan. 2022. Nearest neighbor knowledge distillation 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 5546–5556, Seattle, United States. Association for Computational Linguistics.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343, Florence, Italy. Association for Computational Linguistics.

Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. 2021. Adaptive nearest neighbor 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 2: Short Papers)*, pages 368–374, Online. Association for Computational Linguistics.

Corpus	Src-Tgt	#train	$\|\mathcal{D}\|$	#test
		Domain Adaptation		
Koran	De-En	14,979	450K	2,000
IT		177,795	3.10M	2,000
Medical		206,804	5.70M	2,000
Law		447,701	18.4M	2,000
Subtitles		12,409,630	154M	2,000
ASPEC	Ja-En	2,000,000	68.3M	1,812
KFTT		440,288	15.2M	1,160
TED talk		223,108	5.24M	1,285
BSD		20,000	256K	2,120
		General Domain		
WMT'19	De-En	32,278,623	916M	2,000
WMT'22	Ja-En	32,104,268	874M	2,008
WMT'22	Uk-Cs	12,621,881	192M	2,812

Table 7: Statistics of dataset

A Detailed Experimental Settings

A.1 Statistics of Dataset

Table 7 shows the dataset's statistics. $|\mathcal{D}|$ is the size of the datastore (identical to the number of target-side tokens of the training data). #train and #test are the number of sentences in the training and the test data.

A.2 Model Settings

Table 11 shows the hyperparameters we used in the experiments.

Nucleus sampling We tuned hyperparameter p from $p \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$ based on the validation data.

kNN-MT We used squared-L2 distance as a distance function. For efficiency, we quantized the datastore with IVFPQ and we set the code size to 64. We used the 1024-dimensional representation input to the final layer feedforward network as the key. For the domain adaptation settings, we used 1M keys with 4096 clusters. For the general-domain settings, we used 5M keys with 65536 clusters. For inference, neighbors were searched from the nearest 32 clusters in the datastore. For the De-En domain adaptation setting, we used the same settings as Khandelwal et al. (2021) for k, λ, and τ. For the Ja-En domain adaptation setting, we used the same k as Khandelwal et al. (2021) and tuned λ and τ from $\lambda \in \{0.1, 0.2, ..., 0.9\}$, $\tau \in \{10, 100, 1000\}$ with validation data. For the general-domain settings, we tuned hyperparameters k, λ, and τ from $k \in \{16, 32, 64, 128\}$, $\lambda \in \{0.1, 0.2, ..., 0.9\}$, $\tau \in \{10, 100, 1000\}$ with validation data.

Proposed method For the DBS+* settings, we used the same parameters as the baseline. We tuned p from $p \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$ for Nucleus+* settings without +Perturbation and from $p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ for setting with +Perturbation. For the +Static settings, we computed the mean d_m and standard deviation d_s of the distance to the nearest neighbors on the validation data in advance, and set $h_m = h'_m \times d_m$ and $h_s = h'_s \times d_s$, where h'_m and h'_s are tuned parameters from $h'_m \in \{0.025, 0.05, 0.1, 0.2, 0.4, 0.8\}$ and $h'_s \in \{0.025, 0.05, 0.1, 0.2, 0.4, 0.8\}$ on the validation data. For the +Adaptive settings, we tuned the hyperparameter from $h'_m \in \{0.025, 0.05, 0.1, 0.2, 0.4, 0.8\}$, $h'_s \in \{0.025, 0.05, 0.1, 0.2, 0.4, 0.8\}$. For the +Randomize settings, we tuned the hyperparameters from $h \in \{1.1, 1.2, ..., 4.0\}$. Note that the hyperparameters for +kNN-MT, such as k, λ, and τ, on the proposed methods are identical to the standard kNN-MT.

A.3 Metric Settings

The detailed metric settings are as follows:

BLEU is calculated with `sacrebleu` (Post, 2018). The signature for the corpus-wise BLEU is `nrefs:1|case:mixed|eff:no|tok:13a|smooth:none|version:2.2.1`, and for the sentence-level BLEU is `nrefs:1|case:mixed|eff:yes|tok:13a|smooth:add-k[1.00]|version:2.2.1`.

MedBLEU is the corpus-wise BLEU score computed by the median sentence-level BLEU score for each N-best candidates. When N is even, we selected the sentence with the highest sentence-level BLEU between the two sentences in the middle.

PLL is a metric of the fluency and is computed for sentence $\boldsymbol{y} = (w_1, \ldots, w_{|\boldsymbol{y}|})$:

$$\mathrm{PLL}(\boldsymbol{y}) = \sum_{t=1}^{|\boldsymbol{y}|} \log P_{\mathrm{MLM}}(w_t | \boldsymbol{y}_{\setminus t}), \quad (7)$$

where $\boldsymbol{y}_{\setminus t}$ is a sentence with masked token w_t at time step t and $P_{\mathrm{MLM}}(w_t|\boldsymbol{y}_{\setminus t})$ is the probability that the MLM model predicts original token w_t from masked sentence $\boldsymbol{y}_{\setminus t}$.

DP is formally defined for N-best candidates for source sentence set $\mathcal{X}$ as $\mathbb{H} = \{\mathbf{H}_1, \ldots, \mathbf{H}_N\}$, where $\mathbf{H}_n = \{\hat{\boldsymbol{y}}_1^n, \ldots, \hat{\boldsymbol{y}}_M^n\}$, is calculated as follows:

$$\mathrm{DP}(\mathbb{H}) = \frac{1}{N(N-1)} \times \sum_{\mathbf{H}\in\mathbb{H}} \sum_{\mathbf{H}'\in\mathbb{H}, \mathbf{H}'\neq\mathbf{H}} 1 - \mathrm{BLEU}(\mathbf{H}, \mathbf{H}'). \quad (8)$$

Note that $\mathrm{BLEU}(\mathbf{H}, \mathbf{H}')$ is the corpus-wise BLEU of hypothesis $\mathbf{H}$ for reference $\mathbf{H}'$.

We also used the following metrics to further evaluate the proposed methods in detail. The results are in Appendix B.

MeanLen is the mean sentence length ratio of the candidates to the reference translations. The closer this metric is to 1, indicating that the model outputs sentences of more appropriate length.

COMET@N is the system-level COMET (Rei et al., 2020) score computed by the largest sentence-level COMET score for each N-best candidates. We use `wmt22-comet-da` model[12] for evaluation, and report COMET@1 and COMET@20 in our experiment.

BERTScore@N is the system-level BERTScore (Zhang et al., 2020) computed by the largest sentence-level BERTScore for each N-best candidates. We report BERTScore@1 and BERTScore@20 in our experiment. The hashcode for BERTScore is `roberta-large_L17_idf_version=0.3.12 (hug_trans=4.22.2)-rescaled`.

Speed is the inference speed (tokens/s) logged by `fairseq` when using a single GPU (GeForce RTX 3090).

B Detailed Results

The results for each domain of the De-En domain adaptation setting are shown in Table 12 to Table 16. The results for each domain of the Ja-En domain adaptation setting are shown in Table 17 to Table 20. The results for each language pair of the general-domain setting are shown in Table 21 to Table 23.

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

Method	DP	BLEU @20	BLEU @40	BLEU Mrg@40
Baseline	31.4	42.6	44.4	-
DBS	35.9	40.0	41.6	43.8
Nucleus	48.0	42.1	43.7	44.6
kNN-MT	32.3	**51.8**	**53.6**	**53.5**
DBS+kNN-MT	42.0	48.6	-	51.8
+Static	55.2	49.0	-	52.0
+Adaptive	53.7	49.0	-	52.1
+Randomize	54.4	48.4	-	51.5
Nucleus+kNN-MT	51.6	50.4	-	52.8
+Static	55.0	49.9	-	52.5
+Adaptive	55.6	49.8	-	52.4
+Randomize	**59.4**	49.2	-	52.0

Table 8: Ablation study for MergedBLEU@40 in the De-En domain adaptation setting: DP, BLEU@20 and MergedBLEU@40 are the scores when beam size is set to 20, and BLEU@40 is the score when beam size is set to 40. We report averages of five domains.

C Further Analysis

C.1 Ablation Study for MergedBLEU@N

In §4, we evaluated MergedBLEU@40, the oracle translation quality when merged with Baseline, and showed that the proposed methods' MergedBLEU@40 are comparable to baselines (Baseline, DBS, Nucleus and kNN-MT) in the Ja-En domain adaptation and general-domain settings. However, it is not obvious whether the proposed methods' MergedBLEU@40 is also comparable to the oracle quality of baselines with a larger beam size. Thus, we conducted an ablation study.

Tables 8, 9, and 10 show the oracle BLEU results for the 40-best (BLEU@40) when the baselines' beam size are set to 40.[13] We found that the MergedBLEU@40 of our proposed methods even shows comparable performance to BLEU@40 of baselines in the Ja-En domain adaptation (Table 9) and general-domain (Table 10) settings. These results support our hypothesis that our diversified methods generate high-quality candidates.

C.2 Tuning kNN Diversified Decoding

Figure 4 shows the relationship of the magnitude of the perturbation against DP and BLEU of DBS+kNN-MT+Perturbation in the De-En IT domain. This result shows the trade-off between the DP and the BLEUs for all the perturbation types, indicating that the proposed methods adjust the diversity and the translation quality by varying the

[13] For evaluating BLEU@40, we used the same hyperparameters as in §4 and Appendix A.2 except for beam size.

Method	DP	BLEU @20	BLEU @40	BLEU Mrg@40
Baseline	38.0	26.0	28.2	-
DBS	54.9	24.8	26.9	28.2
Nucleus	63.9	26.5	28.4	28.9
kNN-MT	37.4	**29.7**	**31.8**	31.5
DBS+kNN-MT	60.7	27.9	-	30.9
+Static	66.5	28.3	-	31.3
+Adaptive	66.8	28.4	-	31.3
+Randomize	65.9	27.8	-	30.9
Nucleus+kNN-MT	66.6	29.0	-	**31.7**
+Static	64.0	28.9	-	31.5
+Adaptive	64.0	28.8	-	31.5
+Randomize	**74.8**	28.7	-	31.4

Table 9: Ablation study for MergedBLEU@40 in the Ja-En domain adaptation setting: DP, BLEU@20 and MergedBLEU@40 are the scores when beam size is set to 20, and BLEU@40 is the score when beam size is set to 40. We report averages of four domains.

Method	DP	BLEU @20	BLEU @40	BLEU Mrg@40
Baseline	37.5	41.2	43.9	-
DBS	51.3	37.9	40.2	43.1
Nucleus	62.8	40.4	43.1	44.2
kNN-MT	37.3	**41.4**	**44.0**	42.6
DBS+kNN-MT	52.6	38.2	-	43.4
+Static	54.8	38.3	-	43.4
+Adaptive	54.3	38.3	-	43.4
+Randomize	53.9	38.2	-	43.4
+Uniquify	54.9	37.8	-	43.2
+Static	55.9	37.7	-	43.1
+Adaptive	55.7	37.8	-	43.2
+Randomize	55.8	37.7	-	43.2
Nucleus+kNN-MT	64.5	40.5	-	**44.3**
+Static	52.4	38.9	-	43.4
+Adaptive	52.8	38.9	-	43.4
+Randomize	62.3	38.9	-	43.5
+Uniquify	**70.8**	39.3	-	44.0
+Static	55.4	38.7	-	43.3
+Adaptive	55.7	38.7	-	43.4
+Randomize	67.7	38.4	-	43.3

Table 10: Ablation study for MergedBLEU@40 in the general-domain setting: DP, BLEU@20 and Merged-BLEU@40 are the scores when beam size is set to 20, and BLEU@40 is the score when beam size is set to 40. We report averages of three language pairs.

perturbation's magnitude. The effect of temperature τ on kNN probability of DBS+kNN-MT is also shown in Figure 4 (d). Both DP and BLEU peak around τ from 1 to 10. Unlike the perturbation's magnitude, we found no trade-off between DP and BLEU for the temperature adjustment.

C.3 Detailed Quantitative Analysis

Figure 5 shows a detailed quantitative analysis.

D Used Data, Model, and Software

D.1 Data

De-En domain adaptation parallel corpora created by Koehn and Knowles (2017) based on OPUS (Tiedemann, 2012). License: allowed for research purpose use.

The Asian Scientific Paper Excerpt Corpus created by Nakazawa et al. (2016). License: `https://jipsti.jst.go.jp/aspec/`.

The Kyoto Free Translation Task created by Neubig (2011). Download: `http://www.phontron.com/kftt/index.html`, License: CC BY-SA 3.0.

Ted talks created by Cettolo et al. (2012). Download: `https://wit3.fbk.eu/`, License: CC BY-NC-ND.

The Business Scene Dialogue corpus created by Rikters et al. (2019), Download: `https://github.com/tsuruoka-lab/BSD`, License: CC BY-NC-SA.

WMT'19 news translation task created by Barrault et al. (2019), Download: `https://www.statmt.org/wmt19/translation-task.html`, License: allowed for research purpose use.

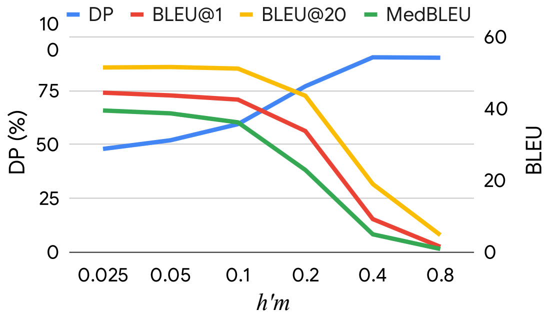
WMT'21 news translation task created by Akhbardeh et al. (2021), Download: `https://www.statmt.org/wmt21/translation-task.html`, License: allowed for research purpose use.

WMT'22 general translation task created by Kocmi et al. (2022), Download: `https://www.statmt.org/wmt22/translation-task.html`, License: allowed for research purpose use.

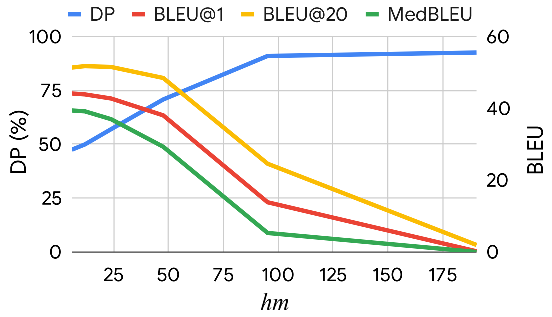
JParaCrawl v3.0 created by Morishita et al. (2022). Download: `http://www.kecl.ntt.co.jp/icl/lirg/jparacrawl/`, License: allowed for research purpose use.

D.2 Model

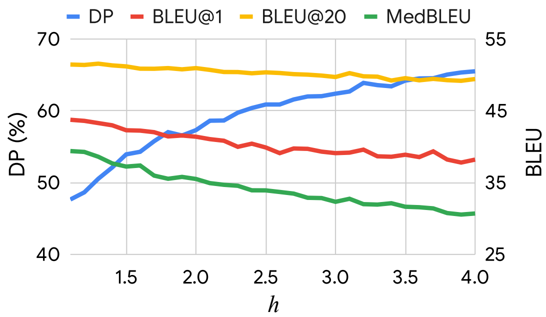
WMT'19 De-En pre-trained model trained by Ng et al. (2019). Download: `https://github.com/facebookresearch/fairseq/tree/main/examples/wmt19`, License: MIT.



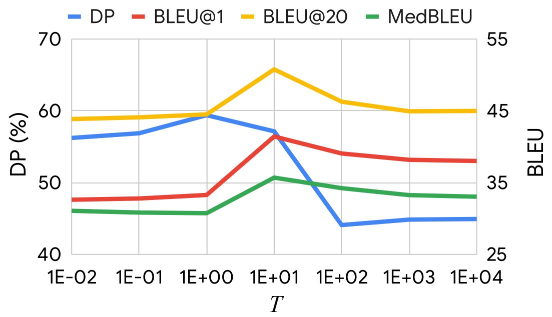
(a) Adaptive noise



(b) Static noise



(c) Randomized-kNN



(d) Temperature

Figure 4: Relationship among perturbation's magnitudes or temperature and DP/BLEU on the De-En IT domain

D.3 Software

`fairseq` created by Ott et al. (2019). Download: https://github.com/facebookresearch/fairseq, License: MIT.

`FAISS` created by Johnson et al. (2019). Download: https://github.com/facebookresearch/faiss, License: MIT.

`sacreBLEU` created by Post (2018). Download: https://github.com/mjpost/sacrebleu, License: Apache License 2.0.

`COMET` created by Rei et al. (2020). Download: https://github.com/Unbabel/COMET, License: Apache License 2.0.

`BERTScore` created by Zhang et al. (2020). Download: https://github.com/Tiiiger/bert_score, License: MIT.

Hyperparameters	De-En Domain Adaptation					Ja-En Domain Adaptation				General Domain		
	Koran	IT	Medical	Law	Subtitles	ASPEC	KFTT	BSD	TED	De-En	Ja-En	Uk-Cs
Nucleus												
p	0.6	0.8	0.7	0.7	0.8	0.6	0.6	0.6	0.5	0.7	0.6	0.6
kNN-MT												
k	64	64	64	64	64	64	64	64	64	32	16	16
τ	100	10	10	10	10	100	100	10	100	100	100	100
λ	0.8	0.7	0.8	0.8	0.7	0.7	0.6	0.3	0.6	0.2	0.2	0.2
DBS+kNN-MT+Static												
h_m	49.4	23.8	38.2	16.9	36	25.5	29.5	29	32	18.9	20.5	39.6
h_s	1.15	3.2	0.35	2	0.2	1	11.2	2.1	1.5	1.4	0.3	0.6
DBS+kNN-MT+Adaptive												
h'_m	0.2	0.1	0.1	0.05	0.1	0.1	0.1	0.1	0.05	0.025	0.1	0.1
h'_s	0.1	0.05	0.1	0.05	0.05	0.025	0.8	0.05	0.8	0.8	0.1	0.8
DBS+kNN-MT+Randomize												
h	2.9	2	2.7	3.2	3.1	3.2	3.7	3.8	1.4	3.7	3.9	3.4
Nucleus+kNN-MT												
p	0.6	0.7	0.8	0.7	0.8	0.5	0.4	0.5	0.6	0.7	0.6	0.6
Nucleus+kNN-MT+Static												
p	0.5	0.9	0.9	0.9	0.7	0.3	0.5	0.5	0.5	0.5	0.5	0.5
h_m	24.7	11.9	9.55	8.45	18	25.5	29.5	29	16	18.9	41	39.6
h_s	0.575	3.2	0.7	0.25	6.4	4	11.2	2.1	12	0.35	0.3	2.4
Nucleus+kNN-MT+Adaptive												
p	0.5	0.9	0.9	0.9	0.7	0.3	0.5	0.5	0.5	0.5	0.5	0.5
h'_m	0.025	0.05	0.05	0.05	0.1	0.1	0.05	0.1	0.025	0.2	0.05	0.2
h'_s	0.4	0.05	0.2	0.025	0.8	0.8	0.025	0.05	0.025	0.05	0.05	0.2
Nucleus+kNN-MT+Randomize												
p	0.5	0.9	0.9	0.9	0.9	0.5	0.7	0.7	0.5	0.7	0.7	0.5
h	1.3	1.3	1.5	1.1	1.1	1.8	1.1	1.5	1.6	3.4	1.4	4
DBS+kNN-MT+Uniquify+Static												
h_m	-	-	-	-	-	-	-	-	-	9.45	5.125	19.8
h_s	-	-	-	-	-	-	-	-	-	0.175	0.6	2.4
DBS+kNN-MT+Uniquify+Adaptive												
h'_m	-	-	-	-	-	-	-	-	-	0.05	0.025	0.05
h'_s	-	-	-	-	-	-	-	-	-	0.025	0.05	0.025
DBS+kNN-MT+Uniquify+Randomize												
h	-	-	-	-	-	-	-	-	-	3.4	2.4	2.8
Nucleus+Uniquify+kNN-MT												
p	-	-	-	-	-	-	-	-	-	0.7	0.5	0.7
Nucleus+Uniquify+kNN-MT+Static												
p	-	-	-	-	-	-	-	-	-	0.5	0.5	0.5
h_m	-	-	-	-	-	-	-	-	-	37.8	20.5	39.6
h_s	-	-	-	-	-	-	-	-	-	1.4	0.15	2.4
Nucleus+Uniquify+kNN-MT+Adaptive												
p	-	-	-	-	-	-	-	-	-	0.5	0.5	0.5
h'_m	-	-	-	-	-	-	-	-	-	0.4	0.1	0.1
h'_s	-	-	-	-	-	-	-	-	-	0.8	0.4	0.4
Nucleus+Uniquify+kNN-MT+Randomize												
p	-	-	-	-	-	-	-	-	-	0.7	0.7	0.5
h	-	-	-	-	-	-	-	-	-	3.8	2.8	2.3

Table 11: Hyperparameters

Method	DP	BLEU				DEQ	MLen	COMET		BERTScore		PLL			Speed
		@1	@20	Mrg	Ref			@1	@20	@1	@20	Max	Min	Mean	
Reference	-	-	-	-	-	-	-	-	-	-	-	-	-	-2.95	-
Baseline	27.4	16.9	22.5	-	16.3	0.33	0.950	0.723	0.763	0.389	0.464	-2.08	-3.48	-2.73	897.2
DBS	39.4	17.0	22.4	24.3	15.5	2.94	0.937	0.720	0.763	0.392	0.465	-1.97	-3.74	-2.77	565.2
Nucleus	52.4	16.7	23.8	25.1	15.5	5.80	0.955	0.722	0.766	0.387	0.466	-2.03	-3.57	-2.76	519.4
kNN-MT	26.2	21.0	27.4	29.0	20.0	-	0.946	0.728	0.775	0.423	0.514	-2.01	-3.48	-2.69	86.8
DBS+kNN-MT	47.2	20.5	27.0	28.8	17.9	10.22	0.945	0.723	0.775	0.424	0.511	-1.72	-3.84	-2.73	75.8
+Static	60.1	19.3	27.1	28.9	16.5	9.75	0.941	0.711	0.772	0.405	0.507	-1.54	-3.92	-2.69	55.6
+Adaptive	63.6	18.6	26.6	28.5	15.8	8.92	0.948	0.697	0.767	0.387	0.497	-1.55	-4.14	-2.76	37.6
+Randomize	55.9	19.5	26.7	28.4	16.7	8.92	0.953	0.718	0.773	0.411	0.505	-1.64	-3.86	-2.69	65.2
Nucleus+kNN-MT	74.5	18.7	27.7	29.2	15.3	10.31	0.990	0.712	0.767	0.404	0.499	-1.86	-4.17	-2.91	50.8
+Static	65.9	20.6	27.7	29.3	16.7	11.95	0.987	0.725	0.772	0.425	0.507	-1.86	-4.15	-2.85	50.3
+Adaptive	63.3	20.5	27.1	28.8	16.9	12.18	0.986	0.725	0.770	0.424	0.503	-1.90	-4.09	-2.84	33.3
+Randomize	66.2	20.7	27.2	28.9	16.4	11.20	0.989	0.726	0.770	0.425	0.504	-1.87	-4.15	-2.86	49.5

Table 12: Koran domain in German-English

Method	DP	BLEU				DEQ	MLen	COMET		BERTScore		PLL			Speed
		@1	@20	Mrg	Ref			@1	@20	@1	@20	Max	Min	Mean	
Reference	-	-	-	-	-	-	-	-	-	-	-	-	-	-4.93	-
Baseline	31.5	37.7	47.3	-	34.0	-0.21	1.004	0.821	0.873	0.606	0.729	-2.96	-7.21	-4.87	874.8
DBS	35.2	37.1	44.2	48.5	33.9	0.41	0.998	0.821	0.861	0.606	0.696	-2.97	-7.05	-4.82	597.7
Nucleus	51.8	36.4	45.5	48.8	31.9	2.42	1.011	0.807	0.856	0.574	0.684	-3.09	-6.97	-4.90	487.1
kNN-MT	32.7	45.9	55.0	57.0	39.8	-	0.974	0.829	0.891	0.645	0.782	-2.94	-7.69	-5.04	59.5
DBS+kNN-MT	44.6	43.9	50.9	55.4	37.4	4.94	0.975	0.815	0.874	0.617	0.735	-3.09	-7.60	-5.05	57.3
+Static	57.0	42.8	51.6	55.7	34.5	4.58	0.973	0.812	0.874	0.614	0.739	-2.92	-7.97	-5.07	48.9
+Adaptive	59.5	42.5	51.2	55.5	33.6	4.31	0.975	0.809	0.873	0.607	0.735	-2.89	-8.08	-5.10	32.2
+Randomize	57.1	41.5	50.8	55.4	33.5	3.85	0.972	0.808	0.871	0.606	0.730	-2.97	-7.92	-5.10	51.0
Nucleus+kNN-MT	47.6	44.7	52.3	55.9	39.3	29.80	1.002	0.824	0.873	0.637	0.733	-3.55	-6.56	-4.95	42.5
+Static	61.3	45.2	52.1	55.8	33.9	4.81	1.025	0.822	0.875	0.634	0.740	-3.17	-7.54	-5.04	31.1
+Adaptive	61.7	45.0	52.0	55.7	33.9	4.85	1.027	0.822	0.875	0.632	0.741	-3.12	-7.57	-5.03	19.0
+Randomize	63.3	44.1	52.0	55.5	32.8	4.33	1.022	0.822	0.874	0.632	0.736	-3.12	-7.59	-5.07	34.3

Table 13: IT domain in German-English

Method	DP	BLEU				DEQ	MLen	COMET		BERTScore		PLL			Speed
		@1	@20	Mrg	Ref			@1	@20	@1	@20	Max	Min	Mean	
Reference	-	-	-	-	-	-	-	-	-	-	-	-	-	-3.24	-
Baseline	27.7	40.4	49.1	-	37.3	-0.15	0.946	0.832	0.858	0.643	0.711	-2.33	-4.23	-3.17	706.0
DBS	31.4	39.9	46.0	50.2	36.3	0.16	0.938	0.831	0.854	0.641	0.697	-2.26	-4.29	-3.15	401.2
Nucleus	39.2	40.0	48.6	50.9	37.1	0.82	0.950	0.825	0.853	0.630	0.694	-2.42	-4.10	-3.20	479.4
kNN-MT	29.5	55.4	63.0	64.4	48.8	0.00	0.928	0.847	0.875	0.707	0.776	-2.31	-4.59	-3.28	17.9
DBS+kNN-MT	36.7	54.0	59.6	62.3	47.1	4.22	0.937	0.836	0.868	0.684	0.753	-2.31	-4.77	-3.31	15.7
+Static	55.8	50.6	59.3	61.9	40.6	3.20	0.934	0.830	0.867	0.669	0.752	-2.11	-5.50	-3.41	14.0
+Adaptive	49.2	52.9	60.3	62.9	43.6	3.78	0.934	0.833	0.868	0.677	0.755	-2.17	-5.20	-3.38	8.0
+Randomize	52.1	50.0	59.0	61.9	40.8	2.83	0.931	0.829	0.866	0.665	0.750	-2.21	-5.26	-3.40	14.6
Nucleus+kNN-MT	41.4	55.1	61.9	63.8	48.4	30.56	0.965	0.844	0.872	0.703	0.763	-2.47	-4.27	-3.25	32.8
+Static	48.2	54.7	61.1	63.3	45.3	5.31	0.984	0.844	0.871	0.700	0.760	-2.40	-4.74	-3.31	26.5
+Adaptive	49.5	54.6	61.2	63.2	45.0	5.32	0.984	0.844	0.871	0.699	0.762	-2.36	-4.80	-3.32	15.1
+Randomize	53.9	54.0	60.3	62.5	42.5	3.89	0.983	0.843	0.870	0.696	0.757	-2.36	-4.89	-3.34	27.7

Table 14: Medical domain in German-English

Method	DP	BLEU				DEQ	MLen	COMET		BERTScore		PLL			Speed
		@1	@20	Mrg	Ref			@1	@20	@1	@20	Max	Min	Mean	
Reference	-	-	-	-	-	-	-	-	-	-	-	-	-	−2.57	-
Baseline	19.5	46.1	52.4	-	44.1	−0.01	0.963	0.854	0.873	0.675	0.731	−1.97	−3.17	−2.50	822.4
DBS	27.4	45.0	50.2	53.6	41.7	0.46	0.936	0.851	0.869	0.670	0.717	−2.00	−3.35	−2.58	428.1
Nucleus	43.2	44.7	52.9	54.8	41.2	1.35	0.957	0.850	0.873	0.665	0.730	−1.94	−3.22	−2.54	458.4
kNN-MT	19.6	61.9	68.8	69.9	58.6	-	0.977	0.871	0.891	0.757	0.818	−2.02	−3.27	−2.57	27.2
DBS+kNN-MT	26.9	60.8	65.8	67.5	55.9	2.69	0.965	0.863	0.885	0.743	0.798	−2.06	−3.44	−2.64	26.1
+Static	33.8	60.0	66.6	68.1	54.6	3.56	0.964	0.862	0.887	0.739	0.802	−2.01	−3.55	−2.65	25.8
+Adaptive	31.7	60.6	66.5	68.0	55.2	3.55	0.964	0.864	0.887	0.743	0.802	−2.02	−3.53	−2.64	13.7
+Randomize	42.2	56.2	64.8	66.5	49.5	2.49	0.963	0.855	0.884	0.713	0.790	−1.99	−3.71	−2.69	25.5
Nucleus+kNN-MT	31.3	61.3	68.2	69.3	57.8	14.43	0.992	0.867	0.887	0.751	0.807	−2.13	−3.08	−2.56	20.5
+Static	44.6	61.7	67.4	68.7	52.3	3.95	1.002	0.868	0.888	0.755	0.806	−2.03	−3.70	−2.65	17.9
+Adaptive	45.7	61.6	67.5	68.7	52.0	3.93	1.004	0.868	0.888	0.754	0.807	−2.03	−3.75	−2.65	9.4
+Randomize	45.6	61.4	66.9	68.2	51.4	3.62	1.001	0.867	0.886	0.750	0.801	−2.05	−3.73	−2.66	18.5

Table 15: Law domain in German-English

Method	DP	BLEU				DEQ	MLen	COMET		BERTScore		PLL			Speed
		@1	@20	Mrg	Ref			@1	@20	@1	@20	Max	Min	Mean	
Reference	-	-	-	-	-	-	-	-	-	-	-	-	-	−3.07	-
Baseline	51.0	29.3	41.7	-	23.0	−1.66	0.917	0.799	0.843	0.595	0.685	−1.98	−4.65	−3.12	1095.9
DBS	45.9	29.0	36.9	42.6	24.1	−14.20	0.923	0.797	0.833	0.593	0.660	−1.96	−4.74	−3.10	857.7
Nucleus	53.2	29.0	39.8	43.3	24.6	−14.33	0.935	0.798	0.837	0.593	0.668	−2.07	−4.25	−3.04	700.4
kNN-MT	53.6	31.7	45.1	47.1	24.6	-	0.911	0.801	0.851	0.606	0.709	−1.87	−4.65	−3.04	4.5
DBS+kNN-MT	54.6	30.6	39.5	45.0	24.4	3.45	0.930	0.793	0.839	0.597	0.677	−1.70	−4.86	−3.02	4.4
+Static	69.5	29.5	40.4	45.4	21.1	4.53	0.933	0.787	0.840	0.586	0.680	−1.51	−5.19	−3.03	4.4
+Adaptive	64.4	30.4	40.5	45.6	22.7	5.63	0.932	0.790	0.841	0.594	0.683	−1.58	−5.09	−3.03	2.2
+Randomize	64.7	30.2	40.6	45.5	22.4	5.00	0.931	0.789	0.841	0.587	0.681	−1.56	−5.03	−3.02	4.4
Nucleus+kNN-MT	63.1	30.6	42.2	45.9	24.4	36.54	0.941	0.797	0.843	0.601	0.688	−1.85	−4.39	−2.98	3.5
+Static	55.0	31.6	41.3	45.6	26.2	−0.87	0.935	0.799	0.840	0.606	0.683	−1.97	−4.23	−2.96	3.6
+Adaptive	57.9	31.2	41.2	45.5	25.6	−4.36	0.939	0.799	0.841	0.604	0.684	−1.93	−4.37	−2.98	1.8
+Randomize	68.0	31.3	39.8	45.0	22.1	5.76	0.967	0.798	0.837	0.603	0.676	−1.83	−5.12	−3.10	3.2

Table 16: Subtitles domain in German-English

Method	DP	BLEU				DEQ	MLen	COMET		BERTScore		PLL			Speed
		@1	@20	Mrg	Ref			@1	@20	@1	@20	Max	Min	Mean	
Reference	-	-	-	-	-	-	-	-	-	-	-	-	-	−2.90	-
Baseline	31.2	21.0	28.2	-	19.8	0.07	0.946	0.829	0.849	0.572	0.629	−1.86	−2.83	−2.31	914.3
DBS	50.1	19.5	27.4	30.6	17.0	3.12	0.904	0.826	0.850	0.563	0.629	−1.76	−3.24	−2.43	581.3
Nucleus	64.9	20.4	29.7	31.7	16.9	5.40	0.934	0.827	0.853	0.566	0.637	−1.66	−3.27	−2.40	591.7
kNN-MT	31.0	24.7	32.8	34.6	23.1	-	0.955	0.831	0.853	0.594	0.658	−1.86	−2.91	−2.33	8.2
DBS+kNN-MT	58.6	22.9	31.3	34.0	18.3	5.67	0.895	0.827	0.854	0.583	0.656	−1.56	−3.44	−2.43	7.6
+Static	63.5	22.9	32.1	34.4	17.7	6.02	0.892	0.827	0.855	0.583	0.659	−1.51	−3.46	−2.42	7.8
+Adaptive	64.6	22.6	31.9	34.3	17.6	6.06	0.893	0.827	0.854	0.582	0.657	−1.51	−3.49	−2.42	4.1
+Randomize	62.2	22.4	31.3	33.9	17.6	5.65	0.897	0.826	0.853	0.579	0.652	−1.51	−3.44	−2.41	7.7
Nucleus+kNN-MT	66.5	24.0	33.9	35.6	19.1	8.82	0.950	0.831	0.857	0.590	0.666	−1.63	−3.40	−2.43	6.8
+Static	49.9	24.0	32.2	34.6	21.0	8.89	0.958	0.830	0.853	0.592	0.654	−1.70	−3.08	−2.35	7.1
+Adaptive	53.4	24.0	32.1	34.6	20.4	8.23	0.960	0.830	0.853	0.589	0.653	−1.66	−3.18	−2.35	3.5
+Randomize	67.8	23.9	32.8	34.9	18.2	7.43	0.953	0.831	0.854	0.590	0.656	−1.64	−3.58	−2.47	6.4

Table 17: ASPEC domain in Japanese-English

Method	DP	BLEU				DEQ	MLen	COMET		BERTScore		PLL			Speed
		@1	@20	Mrg	Ref			@1	@20	@1	@20	Max	Min	Mean	
Reference	-	-	-	-	-	-	-	-	-	-	-	-	-	−2.75	-
Baseline	31.5	18.8	25.1	-	17.5	0.70	0.872	0.762	0.808	0.490	0.581	−1.88	−3.72	−2.70	758.0
DBS	52.9	18.3	24.4	27.3	15.4	4.39	0.826	0.761	0.810	0.489	0.577	−1.81	−4.08	−2.83	474.2
Nucleus	68.2	18.6	26.3	28.2	15.3	7.07	0.873	0.764	0.816	0.493	0.584	−1.85	−3.91	−2.82	444.8
kNN-MT	29.2	22.4	29.4	31.2	20.8	-	0.913	0.778	0.822	0.539	0.628	−1.79	−3.75	−2.59	20.5
DBS+kNN-MT	55.3	21.9	29.0	30.9	18.1	9.81	0.870	0.778	0.827	0.537	0.627	−1.65	−4.28	−2.75	19.4
+Static	61.4	21.3	29.2	31.1	17.3	9.15	0.862	0.777	0.826	0.534	0.624	−1.58	−4.35	−2.74	20.8
+Adaptive	62.5	21.4	29.1	31.1	17.1	8.87	0.861	0.776	0.827	0.530	0.624	−1.55	−4.40	−2.74	11.5
+Randomize	59.8	21.1	28.5	30.5	17.0	8.01	0.866	0.773	0.825	0.524	0.619	−1.55	−4.21	−2.71	20.3
Nucleus+kNN-MT	55.1	22.9	29.5	31.6	20.5	99.42	0.942	0.781	0.825	0.545	0.621	−1.93	−3.54	−2.69	18.4
+Static	68.3	22.6	30.3	32.0	18.4	16.28	0.950	0.778	0.828	0.539	0.627	−1.77	−4.02	−2.75	15.2
+Adaptive	65.1	22.6	30.2	31.9	19.1	21.20	0.947	0.780	0.828	0.542	0.628	−1.83	−3.94	−2.74	9.0
+Randomize	78.5	22.0	29.1	31.0	15.6	9.41	0.977	0.777	0.822	0.536	0.616	−1.73	−4.85	−2.97	13.3

Table 18: KFTT domain in Japanese-English

Method	DP	BLEU				DEQ	MLen	COMET		BERTScore		PLL			Speed
		@1	@20	Mrg	Ref			@1	@20	@1	@20	Max	Min	Mean	
Reference	-	-	-	-	-	-	-	-	-	-	-	-	-	−2.94	-
Baseline	37.5	12.8	18.5	-	11.8	0.37	0.888	0.772	0.814	0.462	0.544	−1.74	−3.75	−2.58	957.7
DBS	55.6	12.0	18.0	20.3	9.9	4.87	0.839	0.769	0.815	0.459	0.541	−1.56	−4.01	−2.62	592.9
Nucleus	57.4	12.4	19.2	20.9	11.1	7.64	0.898	0.772	0.815	0.459	0.538	−1.70	−3.66	−2.57	601.5
kNN-MT	36.8	15.0	21.5	22.7	13.8	-	0.913	0.775	0.818	0.485	0.564	−1.45	−3.17	−2.17	40.8
DBS+kNN-MT	62.0	14.3	20.7	22.6	11.2	9.78	0.884	0.771	0.818	0.478	0.558	−1.06	−3.47	−2.15	35.7
+Static	67.0	14.1	21.0	23.0	10.6	9.49	0.873	0.767	0.817	0.474	0.558	−.98	−3.45	−2.11	33.0
+Adaptive	64.7	14.1	21.2	23.1	10.9	9.78	0.879	0.769	0.817	0.474	0.558	−1.01	−3.43	−2.13	21.4
+Randomize	63.7	13.9	20.9	22.8	11.0	9.51	0.882	0.767	0.817	0.473	0.558	−.99	−3.43	−2.12	34.6
Nucleus+kNN-MT	84.1	13.0	21.8	23.3	9.5	11.10	0.982	0.765	0.813	0.463	0.543	−1.22	−3.76	−2.31	22.7
+Static	74.3	14.5	21.6	23.2	10.9	13.07	0.979	0.771	0.815	0.475	0.550	−1.21	−3.58	−2.22	24.0
+Adaptive	72.6	14.9	21.6	23.2	11.2	14.00	0.976	0.772	0.815	0.477	0.548	−1.26	−3.56	−2.23	15.5
+Randomize	75.3	14.9	21.6	23.3	10.9	13.07	0.982	0.771	0.816	0.476	0.552	−1.21	−3.61	−2.23	23.4

Table 19: TED talks domain in Japanese-English

Method	DP	BLEU				DEQ	MLen	COMET		BERTScore		PLL			Speed
		@1	@20	Mrg	Ref			@1	@20	@1	@20	Max	Min	Mean	
Reference	-	-	-	-	-	-	-	-	-	-	-	-	-	**−2.42**	-
Baseline	51.8	19.9	32.4	-	16.9	−0.98	0.985	0.812	0.868	0.551	0.677	−1.54	−4.38	−2.62	1020.1
DBS	60.8	19.0	29.6	34.7	15.2	3.12	0.970	0.808	0.862	0.544	0.656	−1.51	−4.34	−2.63	690.8
Nucleus	65.1	18.9	30.8	34.9	15.8	6.28	0.992	0.811	0.860	0.546	0.651	−1.56	−4.04	−2.59	657.1
kNN-MT	52.8	21.4	35.1	37.4	17.8	-	0.962	0.817	0.879	0.561	0.711	−1.49	−4.46	−2.62	65.6
DBS+kNN-MT	67.0	19.9	30.7	36.2	14.7	4.55	0.941	0.807	0.869	0.548	0.678	−1.47	−4.88	−2.73	57.1
+Static	74.2	19.8	31.1	36.6	13.8	5.37	0.938	0.807	0.869	0.547	0.679	−1.39	−5.22	−2.76	56.0
+Adaptive	75.5	19.7	31.1	36.5	13.6	5.39	0.940	0.807	0.868	0.546	0.675	−1.38	−5.14	−2.75	43.1
+Randomize	77.8	19.2	30.6	36.3	12.8	4.97	0.933	0.803	0.868	0.541	0.672	−1.36	−5.43	−2.81	50.1
Nucleus+kNN-MT	60.9	21.3	30.9	36.3	17.3	15.75	0.979	0.815	0.863	0.559	0.658	−1.60	−3.74	−2.52	63.8
+Static	63.5	20.8	31.4	36.3	17.0	13.30	0.984	0.814	0.863	0.556	0.661	−1.54	−3.90	−2.52	52.4
+Adaptive	65.0	21.0	31.3	36.4	16.7	11.11	0.987	0.815	0.862	0.557	0.659	−1.55	−3.92	−2.53	35.3
+Randomize	77.7	20.6	31.3	36.2	13.8	6.16	1.009	0.812	0.867	0.554	0.671	−1.43	−4.61	−2.69	43.8

Table 20: BSD domain in Japanese-English

Method	DP	BLEU				DEQ	MLen	COMET		BERTScore		PLL			Speed
		@1	@20	Mrg	Ref			@1	@20	@1	@20	Max	Min	Mean	
Reference	-	-	-	-	-	-	-	-	-	-	-	-	-	-2.93	-
Baseline	32.1	39.6	51.4	-	37.0	0.59	0.981	0.846	0.876	0.714	0.787	-2.18	-3.53	-2.80	1045.7
DBS	41.3	38.9	47.6	53.0	33.9	2.60	0.969	0.844	0.873	0.710	0.772	-2.09	-3.77	-2.84	720.7
Nucleus	55.1	38.1	50.6	54.2	32.5	4.60	0.986	0.847	0.874	0.712	0.782	-2.04	-3.67	-2.80	620.3
kNN-MT	31.7	40.6	52.2	52.9	37.6	-	0.986	0.847	0.877	0.719	0.788	-2.17	-3.51	-2.79	8.9
DBS+kNN-MT	41.7	39.7	48.7	53.5	34.3	3.10	0.976	0.846	0.873	0.713	0.775	-2.03	-3.77	-2.82	9.0
+Static	42.7	39.7	48.6	53.4	34.3	3.30	0.979	0.844	0.873	0.713	0.774	-2.02	-3.78	-2.82	9.1
+Adaptive	41.9	39.6	48.7	53.5	34.3	3.07	0.977	0.845	0.873	0.713	0.774	-2.02	-3.78	-2.82	4.6
+Randomize	42.2	39.6	48.6	53.5	34.2	3.10	0.981	0.845	0.872	0.714	0.772	-1.99	-3.77	-2.81	8.9
+Uniquify	44.1	39.1	47.5	53.1	33.1	2.77	0.979	0.846	0.877	0.710	0.769	-1.99	-3.83	-2.83	8.9
+Static	44.5	39.1	47.7	53.1	33.1	2.85	0.982	0.847	0.872	0.710	0.768	-1.95	-3.86	-2.83	8.9
+Adaptive	44.3	39.1	47.7	53.1	33.1	2.83	0.981	0.845	0.877	0.711	0.768	-1.96	-3.87	-2.83	4.5
+Randomize	44.2	39.0	47.4	53.0	33.0	2.75	0.983	0.848	0.872	0.710	0.766	-1.95	-3.88	-2.83	8.9
Nucleus+kNN-MT	57.1	38.8	51.1	54.5	32.5	4.97	0.992	0.845	0.871	0.711	0.783	-2.01	-3.68	-2.79	8.1
+Static	36.7	40.6	48.2	53.1	36.9	6.74	0.988	0.844	0.871	0.719	0.770	-2.22	-3.36	-2.77	8.5
+Adaptive	39.0	40.3	48.6	53.3	36.4	6.07	0.989	0.844	0.871	0.717	0.772	-2.19	-3.41	-2.77	4.2
+Randomize	55.0	40.3	49.1	53.4	32.4	4.50	0.997	0.845	0.870	0.717	0.775	-2.05	-3.79	-2.81	7.9
+Uniquify	66.1	37.0	49.1	53.8	28.8	3.90	0.996	0.843	0.874	0.704	0.775	-1.95	-3.87	-2.83	7.7
+Static	41.9	39.6	47.9	53.0	35.2	4.29	0.987	0.846	0.872	0.715	0.768	-2.19	-3.43	-2.79	8.6
+Adaptive	43.5	39.8	48.2	53.0	34.9	4.41	0.988	0.846	0.871	0.717	0.768	-2.18	-3.54	-2.80	4.2
+Randomize	62.7	39.6	48.1	53.1	29.2	3.67	1.010	0.845	0.871	0.712	0.769	-2.02	-4.22	-2.88	7.3

Table 21: General domain in German-English

Method	DP	BLEU				DEQ	MLen	COMET		BERTScore		PLL			Speed
		@1	@20	Mrg	Ref			@1	@20	@1	@20	Max	Min	Mean	
Reference	-	-	-	-	-	-	-	-	-	-	-	-	-	-3.17	-
Baseline	40.7	20.9	30.6	-	19.0	-5.75	0.895	0.792	0.846	0.534	0.646	-2.01	-4.59	-3.12	984.3
DBS	59.8	19.6	28.3	32.8	15.6	5.85	0.854	0.786	0.840	0.526	0.630	-1.76	-4.67	-3.07	638.8
Nucleus	71.1	20.4	30.6	33.9	15.5	8.85	0.906	0.785	0.839	0.527	0.634	-1.78	-4.67	-3.07	562.0
kNN-MT	40.5	20.8	30.5	31.9	18.9	-	0.898	0.789	0.844	0.529	0.643	-1.97	-4.57	-3.09	8.9
DBS+kNN-MT	62.0	19.1	28.2	32.9	15.1	5.57	0.848	0.784	0.840	0.520	0.624	-1.62	-4.72	-3.03	9.2
+Static	64.0	19.3	28.2	32.9	14.8	5.70	0.845	0.783	0.840	0.519	0.625	-1.58	-4.73	-3.00	9.2
+Adaptive	64.4	19.2	28.3	32.9	14.7	5.73	0.846	0.782	0.840	0.519	0.626	-1.57	-4.75	-3.00	4.6
+Randomize	64.0	19.3	28.2	32.9	14.8	5.66	0.845	0.782	0.839	0.520	0.623	-1.59	-4.75	-3.01	9.1
+Uniquify	64.2	19.2	28.1	32.9	14.9	5.85	0.849	0.787	0.845	0.522	0.624	-1.63	-4.83	-3.05	9.1
+Static	65.1	19.3	28.1	32.8	14.7	5.76	0.847	0.787	0.840	0.523	0.623	-1.57	-4.84	-3.03	9.1
+Adaptive	65.1	19.1	28.1	32.9	14.6	5.73	0.845	0.784	0.843	0.521	0.622	-1.57	-4.83	-3.03	4.6
+Randomize	65.6	19.3	28.1	32.9	14.6	5.84	0.845	0.787	0.841	0.520	0.622	-1.56	-4.89	-3.03	9.1
Nucleus+kNN-MT	73.1	19.8	30.4	33.8	15.1	8.42	0.907	0.785	0.840	0.518	0.630	-1.71	-4.71	-3.04	7.7
+Static	64.5	20.7	29.7	33.2	16.2	8.73	0.906	0.784	0.840	0.527	0.627	-1.78	-4.52	-3.02	8.0
+Adaptive	62.1	20.2	29.4	33.2	16.4	8.64	0.905	0.784	0.839	0.525	0.626	-1.85	-4.47	-3.03	4.0
+Randomize	77.7	19.9	28.9	33.2	13.3	6.65	0.924	0.782	0.839	0.524	0.623	-1.68	-5.11	-3.12	7.1
+Uniquify	69.0	20.2	30.7	34.0	15.8	9.11	0.907	0.787	0.844	0.524	0.634	-1.76	-4.56	-3.03	7.9
+Static	66.0	20.6	29.5	33.2	15.8	8.28	0.910	0.787	0.841	0.526	0.627	-1.81	-4.59	-3.06	7.8
+Adaptive	66.1	20.8	29.4	33.2	15.8	8.19	0.910	0.789	0.841	0.530	0.628	-1.81	-4.60	-3.06	3.9
+Randomize	82.4	19.8	28.4	32.9	12.0	6.08	0.946	0.784	0.837	0.518	0.616	-1.68	-5.50	-3.23	6.5

Table 22: General domain in Japanese-English

Method	DP	BLEU				DEQ	MLen	COMET		BERTScore		PLL			Speed
		@1	@20	Mrg	Ref			@1	@20	@1	@20	Max	Min	Mean	
Reference	-	-	-	-	-	-	-	-	-	-	-	-	-	-2.40	-
Baseline	39.8	30.0	41.7	-	27.0	5.67	0.933	0.865	0.914	0.634	0.739	-1.62	-3.42	-2.40	974.0
DBS	52.8	28.2	37.8	43.6	23.4	3.64	0.916	0.853	0.911	0.624	0.713	-1.49	-3.68	-2.43	755.9
Nucleus	62.3	29.0	40.1	44.4	23.1	5.81	0.941	0.859	0.909	0.629	0.721	-1.50	-3.64	-2.42	669.7
kNN-MT	39.7	30.0	41.6	43.0	27.0	-	0.928	0.863	0.914	0.633	0.738	-1.61	-3.47	-2.41	21.0
DBS+kNN-MT	54.0	28.5	37.9	43.8	23.2	3.79	0.912	0.853	0.910	0.625	0.713	-1.45	-3.78	-2.44	21.3
+Static	57.8	28.0	38.0	43.8	22.4	3.96	0.909	0.852	0.910	0.622	0.712	-1.39	-3.96	-2.44	21.8
+Adaptive	56.5	28.1	38.0	43.8	22.7	3.96	0.911	0.852	0.910	0.622	0.711	-1.42	-3.91	-2.44	11.0
+Randomize	55.7	28.1	37.8	43.7	22.9	3.89	0.911	0.852	0.910	0.623	0.711	-1.41	-3.89	-2.44	21.1
+Uniquify	56.3	28.3	37.7	43.6	22.6	3.78	0.915	0.858	0.913	0.621	0.709	-1.45	-3.89	-2.46	21.2
+Static	58.1	28.1	37.4	43.5	22.1	3.78	0.913	0.858	0.908	0.621	0.708	-1.41	-4.02	-2.47	21.2
+Adaptive	57.8	28.2	37.5	43.6	22.3	3.84	0.913	0.856	0.911	0.623	0.707	-1.41	-3.97	-2.46	10.9
+Randomize	57.8	28.2	37.5	43.6	22.2	3.79	0.912	0.859	0.910	0.621	0.707	-1.41	-3.98	-2.46	21.1
Nucleus+kNN-MT	63.4	28.8	40.0	44.5	22.8	5.70	0.940	0.851	0.909	0.625	0.721	-1.47	-3.68	-2.41	18.1
+Static	55.9	29.4	39.0	43.9	24.2	5.91	0.942	0.852	0.909	0.632	0.715	-1.54	-3.53	-2.39	18.4
+Adaptive	57.4	29.5	38.8	43.8	23.8	5.53	0.943	0.852	0.910	0.630	0.713	-1.54	-3.62	-2.40	9.1
+Randomize	54.4	29.3	38.5	43.8	24.4	5.71	0.940	0.851	0.909	0.630	0.713	-1.56	-3.50	-2.39	18.0
+Uniquify	77.4	26.8	38.0	44.1	18.3	4.35	0.950	0.845	0.905	0.609	0.705	-1.38	-4.05	-2.47	17.2
+Static	58.2	29.1	38.6	43.8	23.5	5.36	0.944	0.857	0.909	0.628	0.713	-1.54	-3.62	-2.41	18.0
+Adaptive	57.6	29.1	38.6	43.9	23.7	5.37	0.944	0.855	0.909	0.626	0.712	-1.55	-3.63	-2.41	9.0
+Randomize	58.0	29.2	38.7	43.9	23.5	5.32	0.943	0.856	0.909	0.627	0.713	-1.54	-3.59	-2.40	17.9

Table 23: General Domain in Ukrainian-Czech

Test Input: コロナに関しまして。
Reference: *I have a question about COVID.*

DBS+kNN-MT+Randomize	**DBS**
About corona.	*Regarding corona.*
With regards to corona.	*About corona.*
About COVID-19.	*It is about corona.*
Regarding corona.	*We are talking about corona.*
With regards to COVID-19.	*With regards to corona.*
We are talking about corona.	*Regarding corona.*
Now, about corona.	*We are talking about corona.*
It is about corona.	*It is about corona.*
Regarding corona.	*Regarding corona..*
About COVID-19...	*Related to corona.*
Regarding COVID-19.	*About coronavirus.*
With regards to corona..	*It is about corona...*
During the coronavirus pandemic.	*Regarding corona.*
Concerning corona.	*About the coronavirus.*
About corona..	*Regarding corona..*
Now, regarding the coronavirus...	*About coronavirus.*
Was it a virus?	*Regarding corona.*
Regarding corona..	*About coronavirus..*
Concerning corona.	*Regarding coronavirus.*
Regarding COVID-19..	*Regarding corona..*

Test Input: *Spring Summer*コレクションもセール対象商品*!*
Reference: *The spring/summer collection is also included in the sale!*

DBS+kNN-MT+Randomize	**DBS**
The Spring Summer collection is also a sale target product!	*The Spring Summer collection is also a sale target product!*
Items from the Spring Summer collection are also on sale!	*The Spring Summer collection is also a sale item!*
The Spring Summer collection is also a sale target product!	*The Spring Summer collection is also a sale eligible product!*
Summer collection is also on sale!	*Also included in the sale is the Spring Summer collection!*
The Spring Summer collection is also included in the sale!	*The Spring Summer collection is also a sale target item!*
The Spring Summer collection is also a sale target item!	*Also on sale products for the Spring Summer collection!*
Available on sale for the Spring Summer Collection!	*The Spring Summer collection is also a sale target product!*
We also have the Spring Summer collection on sale!	*The Spring Summer collection is also on sale!*
Spring Summer collection is also a sale target product!	*Summer collection is also included in the sale!*
Items from the Spring Summer collection are also on sale!	*The Spring Summer collection is also a saleable item!*
The Spring Summer collection is also on sale!	*See you at the Spring Summer Collection!*
It is also a product subject to sale!	*The Spring Summer collection is also a sale target!*
The Spring Summer collection is also a sales target product!	*Summer collection is also a sale target product!*
The Spring Summer collection is also eligible for sale!	*Spring Summer collection is also a sale eligible product!*
winter collection is also on sale!	*The Spring Summer collection is also eligible for sale!*
The Spring Summer collection is also a sale target item!	*The Spring Summer collection is also a sale eligible product!*
We also have the Spring Summer collection on sale!	*The Spring Summer collection is also part of the sale!*
winter collection is also a sale target product!	*Our Spring Summer collections are on sale!*
The Spring Summer collection is also eligible for sale.	*The Spring Summer collection is also eligible for sale!*
Summer collection is also a sale target product!	*The Spring Summer Collection is also included in the sale!*

Figure 5: Full example 20-best lists using DBS-based methods