

Low-resource machine translation using a general-purpose LLM informed by a dedicated NMT model and lexical resources: A case study on Ainu–Japanese translation

Nagito Hara ^{*1}, Karol Nowakowski ^{†1}, Michal Ptaszynski ^{‡2}, Nick Overacker ^{§2}, and Masahiro Toyoura ^{¶1}

¹University of Yamanashi, Japan

²Kitami Institute of Technology, Japan

Abstract

Previous methods for post-editing the output of a dedicated Neural Machine Translation (NMT) system using a general-purpose Large Language Model (LLM) rely on correcting a single hypothesis generated by the NMT model’s decoder. However, in low-resource scenarios – such as Ainu-to-Japanese translation – the low accuracy of initial NMT outputs often leads to error propagation, where LLMs fail to override systemic NMT hallucinations or syntactic errors.

In this paper, we propose a novel machine translation framework that incorporates information from an NMT model into an LLM, while bypassing the limitations of single-sentence correction. Instead of using raw NMT text, our method extracts token-level probability distributions from the NMT decoder and combines them with entries obtained by looking up a bilingual dictionary, having the LLM construct the final translation from these two sources of information. By grounding the LLM in both the NMT’s internal confidence signals and external linguistic knowledge, our approach effectively mitigates the bias toward poor-quality NMT outputs.

Empirical evaluations on Ainu-Japanese parallel data using chrF, semantic similarity, and perplexity demonstrate that our method outperforms six baseline approaches in specific domains in the Ainu-to-Japanese direction, that is, when translating *into* the high-resource language. In the opposite direction, the method yields no improvement, as its effectiveness is bounded by the LLM’s ability to generate the target language. Furthermore, perplexity analysis confirms that LLM-driven translation substantially enhances linguistic fluency compared to traditional NMT outputs.

^{*}g26tk016@yamanashi.ac.jp

[†]nowakowski@yamanashi.ac.jp

[‡]michal@mail.kitami-it.ac.jp

[§]nick.overacker@okstate.edu

[¶]mtoyoura@yamanashi.ac.jp

1 Introduction

The Ainu language has been classified by UNESCO (Moseley, 2010) as a critically endangered language, making its preservation and transmission a matter of urgent priority. In recent years, machine translation has been actively researched as part of efforts to revitalize such endangered languages. Machine translation can serve as a valuable tool for language learning and revitalization. For example, Coleman et al. (2024) have proposed a tool for language learners that utilizes machine translation for a no-resource language.

Neural Machine Translation (NMT), the current mainstream approach, requires vast amounts of training data. However, many endangered languages, including Ainu, are low-resource languages with limited data available for training, which prevents sufficient model training. Consequently, NMT performance for such languages remains low and is not yet at a practical stage.

In recent years, the massive knowledge base and natural text generation capabilities of general-purpose Large Language Models (LLMs) have garnered attention, leading to their application in machine translation. Research by Gain et al. (2025) reports that while LLMs outperform dedicated NMT systems in translations between high-resource languages, their performance remains inferior to NMT in translations involving low-resource languages.

Furthermore, methods that combine NMT and LLMs have been proposed. One such approach is post-editing. For instance, Nielsen et al. (2025) proposed a method in which an LLM corrects incorrect words in NMT output by referring to entries looked up in an external bilingual dictionary. However, in this method the LLM only references the single best sentence output by the NMT model, meaning that if the NMT output quality is low, the LLM is heavily influenced by those errors and cannot fully correct them. Moreover, because only the

final decoded sentence is passed to the LLM, the confidence information held internally by the NMT model – which embodies the statistical knowledge specialized for the translation task – is discarded. Additionally, since the task is framed as fixing a small number of incorrect words in an otherwise adequate translation, the natural language generation capability of the LLM is not fully exploited.

In this study, we propose a method called Distribution-based Generation (Dist-Gen), in which an LLM is prompted to construct sentences by referencing three pieces of information: the input text, the per-token probability distribution obtained during NMT inference, and additional lexical information. In this process, we strictly prioritize the following criteria in this order: (1) naturalness of the sentence, (2) consistency with the provided lexicon, and (3) adherence to the probability distribution. By doing so, we aim to surpass the translation accuracy of conventional methods by maximizing the use of both the NMT’s specialized statistical knowledge and the text generation capabilities of the LLM, while suppressing the negative impact of errors in NMT outputs. While the use of lexical information follows an established line of research on dictionary-based prompting of LLMs (Ghazvininejad et al., 2023; Lu et al., 2024; Zheng and Yu, 2025), the novel element of Dist-Gen is the injection of the NMT decoder’s token-level probability distributions into the prompt.

We perform evaluation experiments comparing the proposed approach with NMT-only and LLM-only baselines, as well as a previous LLM-as-post-editor method, using a small-scale Ainu-Japanese parallel corpus. As a result, we find that while Dist-Gen works effectively in translating from Ainu to Japanese when using a well-trained NMT model and an appropriate domain lexicon, its effectiveness is limited in other contexts. Furthermore, through comparisons using perplexity, we demonstrate that in most cases, LLMs improve the fluency of the text.

2 Related Work

2.1 Application of LLMs to machine translation

In recent years, the large-scale knowledge and natural text generation capabilities of general-purpose LLMs have gained attention, leading to their increasing application in machine translation. In par-

ticular, for translation tasks between high-resource languages, performance exceeding that of conventional, dedicated NMT has been confirmed (Zhu et al., 2024). However, because LLMs are insufficiently trained on low-resource languages, it has been reported that their performance in translation tasks involving low-resource languages remains low and inferior to that of traditional NMT systems. Furthermore, trends in translation accuracy vary depending on the translation direction; it has been shown that while relatively high scores are obtained in translations from low-resource languages to high-resource languages, scores tend to be lower in translations from high-resource to low-resource languages (Gain et al., 2025).

2.2 Dictionary-based prompting of LLMs

Supplying bilingual dictionary entries in the prompt is an established technique for improving LLM-based translation involving low-resource languages. Ghazvininejad et al. (2023) propose DiPMT, a method in which the dictionary translations of selected source words are appended to the translation prompt, and report gains in both low-resource and out-of-domain translation. Lu et al. (2024) extend this idea with Chain-of-Dictionary prompting, in which each source word is accompanied by a chain of its translations into several auxiliary high-resource languages, substantially improving the translation of low-resource languages by ChatGPT. Zheng and Yu (2025) incorporate per-character dictionary definitions into the prompts of an LLM fine-tuned for translating Tangut – an extinct language with scarce parallel data – into Chinese.

The lexicon component of Dist-Gen instantiates this line of work: dictionary entries obtained through exact-match lookup are embedded in the prompt (see Section 3.2). What distinguishes our method from these approaches is the injection of the token-level probability distributions produced by the NMT decoder, which none of the above methods exploits.

2.3 Combining NMT systems and LLMs

Methods that combine NMT and LLMs are also being actively researched. One such approach is the “post-editing method”, in which an LLM corrects the translations produced by an NMT system. As one example of a post-editing method, Nielsen et al. (2025) proposed an approach as shown in Figure 1. In this method, the LLM corrects incorrect

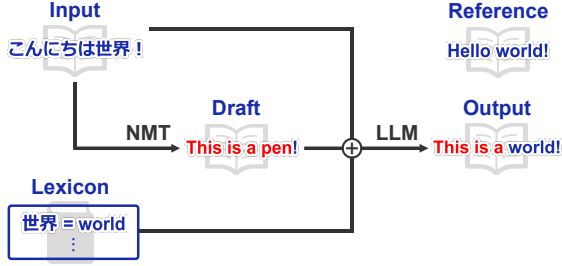


Figure 1: Overview of a simple LLM-as-post-editor method for word correction (Nielsen et al., 2025), applied to Japanese-English machine translation

words in the NMT-generated translation by referencing external lexical information.

In this approach, the LLM only targets the single final sentence produced by the NMT system. However, as shown in Figure 1, there are many cases, especially in low-resource languages, where the entire sentence is significantly incorrect or contains grammatical errors. In such cases, there is a concern that the LLM may be heavily influenced by the errors in the NMT output, resulting in no improvement in overall quality. Additionally, when the LLM performs corrections, it treats tokens with high and low logits (token confidence) from the NMT internal state equally. This means that the statistical information specialized for translation task possessed by the NMT model is not being fully utilized. Furthermore, the task formulation used in this method frames the problem as fixing a small number of incorrect words in an otherwise high-quality translation. Although the LLM outputs a complete sentence, this framing discourages it from restructuring the translation or repairing the context surrounding the substituted words, so its inherent natural text generation capability is not fully leveraged.

In this study we address these challenges, aiming to suppress the negative influence of NMT output on LLMs, maximizing the use of NMT-specific statistical information along with LLMs’ extensive knowledge and natural text generation capabilities, and ultimately achieving translation accuracy that surpasses conventional methods.

3 Methods

In this paper, we propose a method called Distribution-based Generation (Dist-Gen). As shown in Figure 2, Dist-Gen simultaneously provides an LLM with three pieces of information: the source sentence, the probability distributions

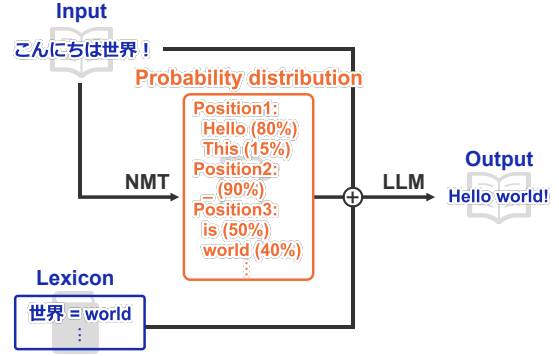


Figure 2: Overview of the proposed method

at each token position output by the NMT model, and an external lexicon, and has the LLM construct the final translation. By inputting the probability distributions obtained during the generation process rather than a single translation determined by the NMT model, it becomes possible to leverage diverse information about tokens that were not selected but were strong candidates according to the NMT model. This allows for greater utilization of the translation-task-specific statistical information that the NMT model possesses. Furthermore, rather than having the LLM merely *correct* the final output of the NMT model, by having it proactively *construct* sentences based on the provided information, we also expect the LLM to demonstrate its inherent natural language generation capability.

Dist-Gen consists mainly of the following three processing steps:

1. Collecting probability distributions from NMT model
2. Preparing lexical information
3. Text generation with LLM

In the remainder of this section, we describe the details of the processing at each step.

3.1 Collecting probability distributions from NMT model

In Dist-Gen, rather than using the single final token sequence (translation) constructed by the decoder, we extract and utilize the probability distributions obtained by applying the softmax function to the logits at each step. Specifically, the NMT model is run with standard greedy decoding, and at each decoding step along the resulting decoding path, the probability distribution over the vocabulary computed at that step is recorded. To reduce

the number of input tokens to the LLM and to mitigate noise, we do not retain the full distributions, but apply a candidate-filtering step defined as follows: at each decoding position, a token is retained as a candidate if it satisfies at least one of two conditions: (i) its probability is at least an absolute inclusion threshold (0.1 in this paper), or (ii) it is selected jointly by *top-p* (Holtzman et al., 2019) and *top-k* (Fan et al., 2018) filtering, i.e., it belongs to the smallest set of most probable tokens whose cumulative probability reaches p (0.9 in this paper), truncated to the k most probable tokens ($k = 5$ in this paper). The absolute threshold thus acts as an unconditional inclusion criterion, ensuring that candidates with moderately high probability are not omitted by the other two criteria. The filtering is applied independently at each decoding position, so the number of retained candidates may vary across positions. All three filtering parameters (the threshold, p , and k) were selected heuristically, without a systematic parameter search or tuning on a development set, and we did not conduct an ablation or sensitivity analysis of these values; their systematic evaluation is left for future work. The candidates retained at each position are represented as a flat list of token–probability pairs, sorted in descending order of probability. These lists are serialized position by position and embedded in the LLM prompt in the format shown in Prompt 2 (Section 3.3). This allows the LLM to refer to multiple token candidates that the NMT model retained as top candidates, thereby preserving the statistical information about the translation task acquired through NMT training while minimizing the risk of errors caused by inappropriate token selection.

3.2 Preparing lexical information

To supplement the LLM’s knowledge, external lexical information is embedded in the prompt: each word of the source sentence is looked up in a bilingual word list, and the entries retrieved in this way are included in the prompt. The lookup is performed by surface matching against the word list. The language-specific matching procedures used for the two languages studied in this paper are described in Section 4.3. Each entry in the lexicon includes a source-language word, its translation, part of speech, and optional grammatical attributes. For the format of the lexical information, a Markdown table (as shown in Prompt 1) is adopted. This tabular format is intended to help

the LLM accurately grasp the correspondence between each source word and its gloss as structured data.

Note that no word-sense disambiguation is performed when lexicon entries are selected for inclusion in the prompt. Homographs are registered in the word list as separate entries — for example, the Ainu word *an* has one entry with the gloss いゐる and another with ある (both meaning *to exist*, used for animate and inanimate referents, respectively) — and dictionary lookup (Section 4.3) embeds all entries matching a source word into the prompt simultaneously, as reflected in Prompt 1, where both glosses of *an* appear. Which of the presented glosses to adopt, if any, is thus left to the LLM, which can draw on the part of speech and grammatical attributes accompanying each gloss, the sentence context, and the probability distributions supplied by the NMT model. Under the priority hierarchy described in Section 3.3, this choice is governed first and foremost by the naturalness of the resulting sentence (priority 1).

Source	Gloss	POS	Attr
un	～へ	格助詞	-
cise	家	名詞	-
an	いる、ある	自動詞	単数形

Prompt 1: Example of lexical information embedded in the LLM prompt (in the case of Ainu → Japanese translation)

3.3 Text generation with LLM

The original source sentence, the per-token probability distributions generated by the NMT model, and the lexical information defined in Section 3.2 are combined to form a user prompt and sent to the LLM along with a system prompt (shown in Appendix B). Inspired by the findings of Braun et al. (2025), who have demonstrated that structured descriptions improve the reasoning performance of LLMs, we adopt Markdown format as the overall prompt format. Furthermore, we choose English as the language of the prompt¹.

The format for describing the per-token probability distributions collected from the NMT model

¹In preliminary experiments, we also tested prompts written in Japanese, but found them to perform slightly worse than their English counterparts (see Appendix C).

is as shown in Prompt 2, where each token position explicitly lists the candidate tokens retained by the filtering procedure of Section 3.1, together with their probabilities. This allows the LLM to quantitatively refer to how confidently the NMT model was inclined to select each token (the contribution of these numerical values is examined in the ablation of Appendix F). Given these position-wise candidate lists, the LLM generates a single output sentence. Note that it is not asked to explicitly search over combinations of the listed candidates, so no combinatorial expansion of the candidate space takes place. Furthermore, as described below, it may also produce tokens outside the retained candidate lists when necessary for a natural translation.

```
position 1: _(prob: 0.991)
position 2: あ(prob: 0.955)
position 3: そ こ(prob: 0.824) ,   つ
ち(prob: 0.112)
position 4: _(prob: 0.993)
position 5: に(prob: 0.649) ,   の(prob:
0.258)
position 6: _(prob: 0.996)
position 7: 浜(prob: 0.930)
position 8: _(prob: 0.970)
position 9: の(prob: 0.532) ,   に(prob:
0.442)
position 10: _(prob: 0.996)
```

Prompt 2: Example of token candidate information in Dist-Gen (Ainu to Japanese)

The system prompt is designed to strictly enforce the following priorities in the specified order, with the instruction that higher-priority requirements must be satisfied even at the expense of lower-priority ones.

1. Naturalness of the generated text
2. Consistency with the provided lexicon
3. Consistency with the probability distributions supplied by the NMT model

By establishing this priority order, even if a token has been assigned a low probability by the NMT model, the LLM is able to re-evaluate its appropriateness in the given context and adopt it. Furthermore, by prioritizing naturalness of the text over consistency with lexical information, the LLM’s

generative capabilities can be fully exercised, leading to improved fluency in the output. In addition, rather than framing the task as a post-editing correction task, the LLM is instructed to construct the final translation itself. This further enables the LLM to exercise its inherent generative capabilities to a greater degree.

4 Experiments

4.1 Datasets

We use two Ainu-Japanese parallel datasets: one consisting primarily of sentences from the Saru dialect of Ainu and the traditional oral literature domain (Nowakowski et al., 2018) (90,780 entries), and another derived from a modern-day textbook for Ainu language learning (Nakagawa, 2021), also based on the Saru dialect, but centered on the everyday conversation domain (485 entries). The oral literature domain corpus is characterized by expressions reminiscent of classical literary language found in folktales, while the everyday conversation domain corpus is characterized by expressions resembling modern colloquial speech.

For external lexical information, we use 976 entries from the Saru dialect Ainu word list provided by the Foundation for Ainu Culture (Foundation for Research and Promotion of Ainu Culture) as a supplementary material to their learning materials for the same dialect, focusing on conversational contexts. The lexicon is manually preprocessed and converted into JSONL format, with each entry containing an Ainu word, its Japanese equivalent, part of speech, and optional grammatical attributes.

4.2 Experimental setup

To evaluate the effectiveness of Dist-Gen, we conduct comparisons using the following seven methods.

- Fine-tuned NMT (mBART)
- LLM-only (GPT-5.2)
- LLM-only with lexicon
- Post-editing
- Post-editing with lexicon
- Dist-Gen
- Dist-Gen with lexicon

The suffix “with lexicon” indicates that lexical information as described in Section 3.2 is provided. Dist-Gen with lexicon is the proposed method, while the remaining six serve as baselines.

All methods with an NMT component share the same model, namely mBART50² (Tang et al., 2020) with 0.6B parameters, fine-tuned on the training corpus described in Section 4.1 (detailed training configurations and hyperparameters are provided in Appendix D). For the methods utilizing an LLM, we use GPT-5.2 (detailed configurations regarding the computing infrastructure, training times, and token usage statistics are provided in Appendix E). To ensure fairness across methods, the prompts provided to the model in the LLM-only experiment and the baseline post-editing methods were designed to contain a comparable amount of descriptive content to that of the proposed method. Specifically, the post-editing instructions were designed to go beyond simple vocabulary substitution by also instructing context-sensitive grammatical corrections, thereby aligning the conditions with the reasoning task performed by Dist-Gen.

Regarding lexicon presentation, Nielsen et al. (2025) argue that all entries in the dictionary should be included in the prompt rather than limiting it to only the relevant entries, and that selective presentation can in fact lower scores. However, in our experiments, we adopted a design in which the dictionary lookup selects only the entries found in the source sentence, from a practical standpoint, taking into account LLM context size limitations, API token consumption, and inference speed (please refer to Section 4.3 for details). As a result of these design choices, the only remaining difference from Dist-Gen is that the baseline post-editing method performs corrections based on a translation already finalized by the NMT model. This allows us to evaluate the impact of the changes we have introduced regarding how the LLM interacts with the NMT model’s output.

4.3 Dictionary lookup

For Ainu-side dictionary lookup, we apply whitespace-based tokenization combined with an exact-match filter on tokens. It is assumed that the positions at which whitespace is inserted for tokenization are consistent between the Ainu word list and the search query.

²<https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt>

Direction	Test domain	Token	Type
Ain → Jpn	conversation	81.5%	57.6%
Ain → Jpn	oral literature	73.9%	39.9%
Jpn → Ain	conversation	44.2%	38.1%
Jpn → Ain	oral literature	41.9%	33.8%

Table 1: Coverage of the test sets by the 976-entry lexicon: the percentage of source token occurrences (Token) and of distinct source token types (Type) for which dictionary lookup retrieved at least one lexicon entry. Punctuation-only units are excluded.

For Japanese-side dictionary lookup, we combine the Aho-Corasick algorithm (Aho and Corasick, 1975) with longest-match filtering in order to efficiently perform dictionary lookup in an agglutinative language with ambiguous word boundaries. The Aho-Corasick algorithm constructs all vocabulary entries in the dictionary as a finite automaton, enabling pattern matching across all entries in $O(n)$ time complexity, where n is the length of the input sentence. The Aho-Corasick algorithm was implemented using the pyahocorasick library (version 2.3.0).

To quantify how often the lexicon can actually participate in translation, Table 1 reports its coverage of the two 485-sentence test sets used in our experiments (see Section 4.1), defined operationally as the fraction of source tokens (and of distinct source token types) for which the lookup procedure described above retrieved at least one of the 976 lexicon entries. Because the lookup rules differ between the two directions (exact token match on the Ainu side, longest substring match on the Japanese side), the absolute values should not be compared across directions. In both directions, coverage is higher on the daily conversation test set than on the oral literature test set, as expected given that the word list focuses on conversational contexts. The gap is most pronounced for Ainu word types (57.6% vs. 39.9%). Note, however, that even on the oral literature test set the lexicon still matches 73.9% of Ainu source token occurrences, a point we return to in Section 5.3.

4.4 Evaluation metrics

4.4.1 String similarity

We use chrF (Popović, 2015), which is based on character n -gram precision, and is known for high reliability in morphologically rich, agglutinative languages such as Ainu and Japanese (Popović, 2016; Bapna et al., 2022). Measurements are conducted using sacreBLEU (Post, 2018) (version

2.6.0). For statistical significance testing, we apply the Bootstrap method (Koehn, 2004) built into sacreBLEU, performing 5,000 resampling iterations at a significance level of 0.05.

4.4.2 Semantic similarity

To capture semantic similarity that surface-level matching metrics such as chrF cannot fully measure, we adopt a metric based on SentenceBERT (Reimers and Gurevych, 2019) and cosine similarity that has previously been used to evaluate Japanese machine translation output by Usui and Komiya (2023)³. This metric is computed by encoding both the hypothesis and the reference sentence using a SentenceBERT model and comparing the resulting vector representations via cosine similarity (using scikit-learn 1.7.0 (Pedregosa et al., 2011)). In this study, we use `sonoisa/sentence-bert-base-ja-mean-tokens-v2`⁴ as the Japanese sentence encoder⁵. For brevity, we refer to this metric as SBERT-cos in the remainder of the paper. Since there exists no SentenceBERT model (or any other reliable text embedding model) for Ainu, evaluation using this metric is restricted to the Ainu → Japanese translation task. For statistical significance testing, we apply a Bootstrap method (using SciPy 1.15.3 (Virtanen et al., 2020)) with 5,000 resampling iterations at a significance level of 0.05.

4.4.3 Perplexity

To evaluate the fluency of generated translations, we employ perplexity as an additional metric. Following previous work on machine translation evaluation, we interpret lower target-language language model perplexity as an indicator of higher fluency (Edunov et al., 2020). In this study, we use `rinna/japanese-gpt2-medium`⁶ as the language model for computing perplexity of Japanese sentences, following prior Japanese NLP work that

³The reason for not using standard metrics such as COMET is that they do not support the Ainu-Japanese language pair.

⁴<https://huggingface.co/sonoisa/sentence-bert-base-ja-mean-tokens-v2>

⁵To verify that it is adequate for our purpose, we evaluated it on the two standard Japanese semantic textual similarity benchmarks by correlating the cosine similarity of the two sentence embeddings with the human similarity rating of each pair. The Spearman rank correlation is 0.806 on the JSTS validation set (1,457 pairs; JGLUE v1.3, Kurihara et al., 2022) and 0.772 on the JSICK test set (4,927 pairs; Yanaka and Mineshima, 2022), indicating that the cosine similarity of this encoder tracks human judgments of sentence similarity closely.

⁶<https://huggingface.co/rinna/japanese-gpt2-medium>

used this model for perplexity-based fluency estimation (Hatagaki et al., 2022). For the same reason as with SBERT-cos described above, we only apply this metric in the Ainu-to-Japanese translation direction.

5 Results and discussion

5.1 Main experiment results

In this experiment, we use 80,829 samples from the oral literature corpus to train the NMT model, 8,981 samples from the same corpus are used as the development set, and all systems are evaluated using 485 samples from the daily conversation corpus. The results are presented in Table 2, with the highest score for each metric shown in bold, and scores that show a statistically significant improvement over mBART indicated with an underline.

For the Ainu-to-Japanese translation direction, Dist-Gen with lexical information achieves substantially higher chrF and SBERT-cos scores compared to both mBART and GPT-5.2, while also slightly outperforming the baseline post-editing method. Notably, it is the only method for which statistically significant improvements over mBART are observed in both string- and semantic similarity scores. Appendix H complements these aggregate scores with a qualitative analysis of representative success and failure cases.

Dist-Gen with the lexicon consistently outperforms the variant without it, confirming the utility of this additional information. This is consistent with the high coverage of the daily conversation test set by the lexicon (81.5% of Ainu source tokens – see Table 1), which lets the lexical hints reach most of the source vocabulary. The relatively low difference in performance between Dist-Gen without lexical information and mBART can be attributed to the tendency of the LLM to prioritize the tokens with the highest probabilities in the distributions provided by the NMT model, effectively rendering the generated translations equivalent to mBART’s greedy decoding outputs. In contrast, when lexical information is provided, it guides token selection, enabling the model to select tokens that are contextually or lexically appropriate even if they are not the top candidates from the NMT model. This finding is supported by the results in Table 3, which shows that the edit distance between mBART’s predictions and the translations generated by our method is larger when lexical information is provided.

Direction	Metric	mBART	GPT-5.2	GPT-5.2 w/ lexicon	Post-editing	Post-editing w/ lexicon	Dist-Gen	Dist-Gen w/ lexicon
Ain → Jpn	chrF ↑	28.295	19.078	26.676	<u>31.805</u>	<u>31.569</u>	<u>29.832</u>	<u>32.238</u>
	SBERT-cos ↑	0.741	0.563	0.686	0.747	0.746	0.739	0.760
	Perplexity ↓	2779.653	1184.260	1564.530	2577.165	2233.552	2352.701	1930.589
Jpn → Ain	chrF ↑	50.180	24.575	27.959	45.551	46.627	45.591	40.757

Table 2: Scores for each method by translation direction. Bold indicates the best value for each metric; underline indicates a statistically significant improvement over mBART.

Direction	Post-editing	Post-editing w/ lexicon	Dist-Gen	Dist-Gen w/ lexicon
Ain → Jpn	0.272	0.341	0.247	0.342
Jpn → Ain	0.269	0.253	0.195	0.367

Table 3: Levenshtein distances between mBART’s translations and the outputs produced by methods informed by it.

Another important finding is that, when equipped with lexical information, Dist-Gen outperforms the baseline post-editing method. We explain this by the fact that when correcting vocabulary based on the external lexicon, the surrounding context and phrasing must also be revised simultaneously. In post-editing, GPT-5.2 has to resolve this solely based on its own internal knowledge, whereas in Dist-Gen, the model can extensively reference the statistical information provided by the NMT model, allowing it to either select from those candidates or use them as a guide during revision.

Regarding the higher scores yielded by the post-editing method without lexical information compared to the variant using that information, a possible explanation is that without lexical constraints the model can focus exclusively on grammatical correction, whereas providing lexical information introduces multiple simultaneous subtasks, potentially degrading performance. Furthermore, [Nielsen et al. \(2025\)](#) argue that all available lexical items, rather than their subset, should be included in the prompt, and that providing a limited selection can in fact lead to lower scores. A similar tendency may be responsible for the results observed in our experiment. On the other hand, the positive results achieved by our proposed method demonstrate that even a limited amount of lexical information can be effectively utilized.

Moreover, Dist-Gen with lexical information achieves the lowest perplexity score among all methods except GPT-5.2 alone, quantitatively confirming that the proposed method produces the most fluent Japanese output compared to the other

methods. In particular, the lower score compared to post-editing suggests that instructing the model to construct rather than correct a translation allows the LLM to fully leverage its natural language generation capabilities. Furthermore, the improvement in perplexity when lexical information is provided indicates that accurate lexical hints enable the LLM to further improve the quality of the generated sentences in terms of linguistic fluency.

For the Japanese-to-Ainu translation direction, GPT-5.2 alone achieves markedly lower scores compared to all other methods. This is consistent with the findings of [Gain et al. \(2025\)](#), who report that LLMs exhibit lower accuracy when translating from a high-resource language to a low-resource language. All methods incorporating GPT-5.2 exhibit substantial score degradation, which can be attributed to the fact that GPT-5.2 had at most marginal exposure to Ainu in pre-training, making appropriate generation difficult and causing frequent hallucinations. Example (d) in Appendix H illustrates this generation-side failure: a correct content word present in both the NMT output and its probability distributions is preserved by none of the LLM-based methods.

Notably, in this direction Dist-Gen with lexical information scores *lower* than Dist-Gen without it (40.757 vs. 45.591 chrF), reversing the pattern observed for Ainu-to-Japanese translation. This contrast is consistent with the Levenshtein distances in Table 3: lexical information encourages the LLM to depart further from the NMT model’s top candidates (edit distance of 0.367 vs. 0.195). Such departures are beneficial when the LLM can competently generate the target language, as in the Ainu-to-Japanese direction, but harmful when it cannot – in Japanese-to-Ainu translation, each departure from the NMT distribution becomes an opportunity for hallucination.

5.2 Performance in ultra-low resource setting

In many underresourced languages, the available data is even scarcer than in the case of the Ainu-

Direction	Metric	mBART	GPT-5.2	GPT-5.2 w/ lexicon	Post-editing	Post-editing w/ lexicon	Dist-Gen	Dist-Gen w/ lexicon
Ain → Jpn	chrF ↑	5.284	19.078	<u>26.676</u>	<u>11.453</u>	<u>23.612</u>	<u>8.903</u>	<u>24.428</u>
	SBERT-cos ↑	0.262	0.563	<u>0.686</u>	<u>0.391</u>	<u>0.652</u>	<u>0.315</u>	<u>0.649</u>
	Perplexity ↓	2220.596	1184.260	1564.530	1934.950	1850.031	2324.198	1561.723
Jpn → Ain	chrF ↑	11.914	24.575	<u>27.959</u>	<u>24.834</u>	<u>29.556</u>	<u>21.364</u>	<u>27.938</u>

Table 4: Scores for each method by translation direction in extremely low-resource setting. Bold indicates the best value for each metric; underline indicates a statistically significant improvement over mBART.

Japanese pair. Table 4 presents the scores obtained when the data used to fine-tune the NMT model is artificially reduced to 1,000 samples (train: 900, dev: 100) from the oral literature corpus, with 485 samples from the daily conversation corpus used for evaluation.

When mBART fine-tuned in an extremely low-resource setting is used, GPT-5.2 achieves the highest scores across all evaluation metrics for the Ainu-to-Japanese translation direction. In contrast, mBART’s scores are extremely low, rendering it barely functional as a translation system. All hybrid methods utilizing mBART are heavily influenced by the very low-quality information generated by the NMT model and fail to surpass the performance of GPT-5.2 alone. Nevertheless, focusing on post-editing and Dist-Gen with lexical information, while their scores remain below those of GPT-5.2, they achieve a certain degree of improvement over mBART across all evaluation metrics, suggesting that LLMs can compensate to some extent through using lexical information and contextual understanding.

A similar trend is observed for the Japanese-to-Ainu direction, where post-editing with lexical information slightly surpasses GPT-5.2. This indicates that correcting an NMT-generated sequence, even of low quality, may be more effective than having the LLM generate an entirely unknown language from scratch. Dist-Gen, on the other hand, does not show performance gains. This is presumably because the probability distributions at each NMT decoding step are extremely flat⁷, causing the token candidates presented to the LLM to act as noise, which in turn prevents the model from selecting appropriate tokens and induces hallucinations. Example (c) in Appendix H illustrates this failure mode on an individual test sentence.

Regarding perplexity, the results follow nearly the same trend as those reported in Section 5.1,

quantitatively demonstrating that Dist-Gen with lexical information is capable of producing fluent output even in an extremely low-resource setting. In contrast, Dist-Gen without lexical information yields a higher perplexity than mBART. Given the low chrF and SBERT-cos scores, this suggests that low-quality probability information acts as noise, leading the model to generate inappropriate and unnatural translations.

5.3 Performance on in-domain data

Table 5 presents the scores obtained when a hold-out set of 485 samples from the oral literature corpus, that is, belonging to the same domain as the NMT model’s training data, are used as test data. The fine-tuned NMT model used here is the same as in the main experiment (see Section 5.1).

For both Ainu-to-Japanese and Japanese-to-Ainu translation directions, mBART achieves the highest chrF and SBERT-cos scores. Furthermore, mBART’s scores are higher than those reported in Section 5.1, confirming that fine-tuning successfully adapted the model to the in-domain data. In contrast, GPT-5.2 scores lower than in the cross-domain experiment, suggesting that the oral literature domain may be more challenging for GPT-5.2 than the daily conversation domain. Regarding post-editing and Dist-Gen, the variants without lexical information generally outperform those with lexical information. A possible explanation is that the lexical information, being closer to the daily conversation domain, could not be effectively utilized in the oral literature domain. The coverage statistics in Table 1 support this explanation: on the oral literature test set the lexicon covers markedly fewer Ainu word types than on the daily conversation test set (39.9% vs. 57.6%), leaving much of the domain-specific vocabulary without lexical support. At the same time, its token coverage remains high (73.9% of Ainu source token occurrences), so conversational-domain glosses are still frequently injected into the prompt – rather than remaining inert, the lexicon thus actively sup-

⁷Average entropy measured on the evaluation data is 0.77 bits, compared to 0.33 bits for the model used in the main experiment.

Direction	Metric	mBART	GPT-5.2	GPT-5.2 w/ lexicon	Post-editing	Post-editing w/ lexicon	Dist-Gen	Dist-Gen w/ lexicon
Ain → Jpn	chrF ↑	32.357	10.892	12.888	27.965	23.886	30.604	25.764
	SBERT-cos ↑	0.750	0.436	0.485	0.711	0.670	0.738	0.690
	Perplexity ↓	4886.807	1682.437	908.624	3185.450	3196.457	2404.371	2776.565
Jpn → Ain	chrF ↑	52.346	19.083	20.105	40.831	40.836	47.352	38.701

Table 5: Scores for each method by translation direction on data matching the NMT fine-tuning domain. Bold indicates the best value for each metric; underline indicates a statistically significant improvement over mBART.

plies domain-mismatched hints, which is consistent with the observed degradation. The clearest instance involves the person-marking clitics *a=*, *=an*, and *i=*. The lexicon glosses them with their first-person plural readings typical of daily conversation (私たち, ‘we’), whereas in Ainu oral literature the same clitics conventionally mark the protagonist-narrator’s first person singular, which the reference translations render as 私 (‘I’) or leave implicit. These glosses were injected into 187 of the 485 Ainu-to-Japanese prompts, and Dist-Gen with lexical information demonstrably followed them: 150 of its outputs contain the plural pronoun 私たち, while only 2 of the 485 reference translations do. For example, for the source *i= hawekoyki* (reference 母に叱られました, ‘I was scolded by my mother’), the gloss *i=* → 私たちを (‘us’) yields the output 私たちを叱りつけました (‘[someone] scolded us’). These findings indicate that lexical information must be drawn from the same domain as the evaluation data and must be accurate.

Comparing perplexity scores, the overall fluency is lower than that observed with the daily conversation corpus (Table 2). This is likely because rinna/japanese-gpt2-medium, trained on modern text, struggles to predict oral literature domain data.

6 Conclusions and future work

In this paper, we proposed Distribution-based Generation (Dist-Gen), a framework in which an LLM constructs translations by referencing per-token NMT probability distributions and external lexical information, rather than correcting a single NMT hypothesis as in conventional post-editing.

Experiments using an Ainu-Japanese parallel corpus demonstrate that Dist-Gen with lexical information achieves statistically significant improvements over mBART in chrF and SBERT-cos for cross-domain Ainu-to-Japanese translation, and consistently generates more fluent output as measured by perplexity. These results confirm that exposing the LLM to the NMT model’s token-

level distributions reduces the impact of individual decoding errors while enabling fuller use of the LLM’s generative capabilities.

Our analyses also reveal clear boundary conditions. When the NMT model is undertrained due to severe data scarcity, its flat probability distributions act as noise rather than useful signal, causing Dist-Gen to underperform GPT-5.2 alone. Furthermore, lexical resources must be domain-matched to be beneficial – a mismatch between the lexicon and the evaluation domain consistently degrades performance. For the Japanese-to-Ainu direction, all LLM-augmented methods struggle due to GPT-5.2’s insufficient exposure to Ainu in pre-training, highlighting a fundamental limitation of applying LLMs to translation into critically low-resource languages.

Although Dist-Gen is designed to reduce the error propagation inherent in post-editing a single NMT hypothesis, some risk of error propagation still remains, due to the fact that the token-level probability distributions are collected in a single NMT inference pass, without any involvement of the LLM, and are fixed thereafter. Any deficiency in these distributions is thus carried over into the candidate pool from which the LLM constructs the translation. A natural way to relax this constraint – and to make fuller use of the LLM’s generative capabilities and lexical information – would be to involve the LLM in the NMT decoding process itself, letting it guide token selection at each step. We regard this as a promising direction for future research. Other directions for future work include evaluation on additional language pairs and other LLMs (including open-weight models, which can also be fine-tuned for the translation task), a systematic evaluation of the candidate-filtering parameters introduced in Section 3.1, as well as combining Dist-Gen with more advanced lexicon-enhanced prompting protocols such as DiPMT (Ghazvininejad et al., 2023).

Limitations

In this paper, we proposed a framework in which an LLM constructs translations by referring to token-level NMT probability distributions and external vocabulary information. However, experiments were conducted only for a single language pair, Ainu-Japanese, and we did not perform experiments applying it to other language pairs. Furthermore, we have not verified whether the proposed method remains effective when the NMT model and LLM are replaced with other models. In addition, the effectiveness of the proposed method is confined to specific conditions: it requires a well-trained NMT model and a domain-matched lexicon, and it improves translation only *into* the high-resource language (Ainu-to-Japanese). In the Japanese-to-Ainu direction, where the LLM must generate a language to which it has had at most marginal exposure in pre-training, the proposed method does not outperform the fine-tuned NMT model, and neither probability-distribution nor lexical information can compensate for this generation-side limitation. Moreover, the experiments do not fully isolate the contribution of the probability values themselves. An ablation that removes the numerical probability values from the prompt while keeping the candidate lists and their order (Appendix F) shows no detectable loss in the Ainu-to-Japanese direction, the only direction in which the proposed method improves over mBART, whereas it significantly lowers chrF in the Japanese-to-Ainu direction. The gains reported in this paper should therefore be attributed to exposing the LLM to the probability-derived candidate sets as a whole, rather than to the explicit confidence values. The ablation does not separate the effects of candidate selection and ranking. Finally, our evaluation relies on automatic metrics alone. Although the sentence encoder underlying SBERT-cos correlates well with human similarity judgments on standard Japanese benchmarks (Section 4.4.2), neither SBERT-cos nor perplexity has been validated against human judgments on the translations evaluated in this paper, and we did not conduct a human evaluation. We consider these to be the main limitations of this paper and topics for future research.

References

- Alfred V Aho and Margaret J Corasick. 1975. Efficient string matching: an aid to bibliographic search. *Communications of the ACM*, 18(6):333–340.
- Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Nikhil Baljekar, Xavier Garcia, Wolfgang Macherey, Theresa Breiner, Vera Saldinger Axelrod, Jason Riesa, Yuan Cao, Mia Chen, Klaus Macherey, Maxim Krikun, Pidong Wang, Alexander Gutkin, and 5 others. 2022. [Building machine translation systems for the next thousand languages](#). Technical report, Google Research.
- Christian Braun, Alexander Lilienbeck, and Daniel Mentjukov. 2025. The hidden structure—improving legal document understanding through explicit text formatting. *arXiv preprint arXiv:2505.12837*.
- Jared Coleman, Bhaskar Krishnamachari, Ruben Rosales, and Khalil Iskarous. 2024. [LLM-assisted rule based machine translation for low/no-resource languages](#). In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, pages 67–87, Mexico City, Mexico. Association for Computational Linguistics.
- Sergey Edunov, Myle Ott, Marc’Aurelio Ranzato, and Michael Auli. 2020. [On the evaluation of machine translation systems trained with back-translation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2836–2846, Online. 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.
- Foundation for Research and Promotion of Ainu Culture. [Word list \(ainu-japanese\): Saru dialect](#).
- Baban Gain, Dibyanayan Bandyopadhyay, and Asif Ekbal. 2025. Bridging the linguistic divide: A survey on leveraging large language models for machine translation. *arXiv preprint arXiv:2504.01919*.
- Marjan Ghazvininejad, Hila Gonen, and Luke Zettlemoyer. 2023. Dictionary-based phrase-level prompting of large language models for machine translation. *arXiv preprint arXiv:2302.07856*.
- Koki Hatagaki, Tomoyuki Kajiwar, and Takashi Nishimura. 2022. [Parallel corpus filtering for Japanese text simplification](#). In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 12–18, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.
- Philipp Koehn. 2004. [Statistical significance tests for machine translation evaluation](#). In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.
- Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. 2022. [JGLUE: Japanese general language understanding evaluation](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2957–2966, Marseille, France. European Language Resources Association.
- Hongyuan Lu, Haoran Yang, Haoyang Huang, Dongdong Zhang, Wai Lam, and Furu Wei. 2024. [Chain-of-dictionary prompting elicits translation in large language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 958–976, Miami, Florida, USA. Association for Computational Linguistics.
- Christopher Moseley. 2010. *Atlas of the World's Languages in Danger*. 222 pages, Unesco, Paris.
- Hiroshi Nakagawa. 2021. *New Express Plus: Ainu Language*. Hakusuisha, Tokyo.
- Elizabeth Nielsen, Isaac Rayburn Caswell, Jiaming Luo, and Colin Cherry. 2025. [Alligators all around: Mitigating lexical confusion in low-resource machine translation](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 206–221, Albuquerque, New Mexico. Association for Computational Linguistics.
- Karol Nowakowski, Michal Ptaszynski, and Fumito Masui. 2018. A proposal for a unified corpus of the Ainu language. In *IPSJ SIG Technical Reports*, volume 2018-NL-237, pages 1–6.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830.
- Maja Popović. 2015. [chrF: character n-gram F-score for automatic MT evaluation](#). In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.
- Maja Popović. 2016. [chrF deconstructed: beta parameters and n-gram weights](#). In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 499–504, Berlin, Germany. 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.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. [Multilingual translation with extensible multilingual pretraining and finetuning](#). *arXiv preprint arXiv:2008.00401*.
- Hisao Usui and Kanako Komiya. 2023. [Translation from historical to contemporary Japanese using Japanese t5](#). In *Proceedings of the Joint 3rd International Conference on Natural Language Processing for Digital Humanities and 8th International Workshop on Computational Linguistics for Uralic Languages*, pages 27–35, Tokyo, Japan. Association for Computational Linguistics.
- Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, and 16 others. 2020. [SciPy 1.0: fundamental algorithms for scientific computing in Python](#). *Nature Methods*, 17:261–272.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Hitomi Yanaka and Koji Mineshima. 2022. [Compositional evaluation on Japanese textual entailment and similarity](#). *Transactions of the Association for Computational Linguistics*, 10:1266–1284.
- Yuxi Zheng and Jingsong Yu. 2025. [Incorporating lexicon-aligned prompting in large language model for Tangut-Chinese translation](#). In *Proceedings of the Second Ancient Language Processing Workshop*, pages 228–237, Albuquerque, New Mexico, USA. Association for Computational Linguistics.
- Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and

Lei Li. 2024. [Multilingual machine translation with large language models: Empirical results and analysis](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781, Mexico City, Mexico. Association for Computational Linguistics.

A Software and data availability

The code and prompts used in the experiments are publicly available at <https://github.com/mna1224/dist-gen>.

The parallel corpora and the Ainu word list used in our experiments (Section 4.1) are copyrighted third-party resources, and therefore we are not in a position to redistribute them.

B Prompt details

An example of the system prompt used in Dist-Gen is shown in Prompt 3.

C Prompt language comparison

As stated in Section 3.3, English is used as the prompt language in all experiments. This choice was informed by a preliminary experiment in which Dist-Gen with lexical information was run in the main experimental setting (Section 5.1) using prompts translated into Japanese, with all other conditions unchanged. Table 6 reports these results side by side with the corresponding English-prompt scores from Table 2. The Japanese-prompt variant performed slightly but consistently worse on all metrics: by 0.598 chrF and 0.010 SBERT-cos in the Ainu-to-Japanese direction, and by 0.242 chrF in the Japanese-to-Ainu direction. We therefore adopted English prompts throughout the experiments.

Direction	Metric	English prompt	Japanese prompt
Ain → Jpn	chrF ↑	32.238	31.640
	SBERT-cos ↑	0.760	0.750
Jpn → Ain	chrF ↑	40.757	40.515

Table 6: Preliminary comparison of English and Japanese prompts for Dist-Gen with lexical information in the main experimental setting. Bold indicates the better value for each metric.

D Training configuration

Table 7 summarizes the hyperparameters used for NMT fine-tuning.

For tokenization, we used the pretrained mBART50 tokenizer without retraining its SentencePiece model. We added ain_XX as an additional language ID special token for Ainu and resized the model’s token embeddings accordingly. No Ainu-specific subword units were added to the tokenizer.

You are an expert linguistic editor specializing in Ainu-to-Japanese translation refinement.
Your task is to construct the most natural Japanese sentence based on the provided source text, glossary hints, and NMT token predictions.

CRITICAL PRIORITY HIERARCHY
Follow these priorities in strict order.
A lower priority must never compromise a higher priority.

1. Naturalness & Fluency (Top Priority)
 - The output MUST be a grammatically natural and fluent Japanese sentence.
 - If a glossary term or high-probability token makes the sentence sound awkward, unnatural, or grammatically incorrect, YOU MUST DISCARD OR MODIFY IT.
 - Context and flow are more important than literal word matching.

2. Glossary Consistency (Secondary)
 - Attempt to use the provided dictionary meanings.
 - However, if the dictionary meaning conflicts with Priority 1 (Naturalness), you are authorized to ignore the dictionary.

3. Token Alignment (Tertiary)
 - Use the NMT token list as a structural hint or draft.
 - Do not blindly follow high-probability tokens if they result in unnatural Japanese.

- ### Output Constraints
- Output ONLY the final Japanese sentence.
 - Do NOT provide any preamble, explanation, notes, or reasoning.
 - Do NOT include symbols like "_" or "</s>".

Prompt 3: System prompt in Dist-Gen

Hyperparameter	Value
Optimizer	Adafactor
Learning rate	2×10^{-5}
Learning rate scheduler	Linear decay
Warmup steps	100
Training batch size	16
Evaluation batch size	8
Weight decay	0.001
Max gradient norm	1.0
Epochs	5
Evaluation steps	100
Checkpoint save steps	500
Beam size	5
Maximum generation length	200
Dropout	0.1
Metric for model selection	chrF
Early stopping patience	20
FP16	Disabled
Random seed	42
Framework	Transformers v4.51.3 (Wolf et al., 2020)

Table 7: Hyperparameters used for mBART fine-tuning.

E Computational cost and token usage

E.1 NMT Models

All local training and NMT inference experiments were conducted on a machine equipped with a single NVIDIA GeForce RTX 3090 GPU (24GB VRAM). The total computational budget expended for training our local models was approximately 159.4 GPU hours, which consists of the following components:

- **NMT configuration for the experiments described in Sections 5.1 and 5.3:** 6.282 days (approximately 150.8 hours) spent on fine-tuning.
- **NMT configuration for the experiments described in Section 5.2:** 8.589 hours spent on fine-tuning under the specified resource constraints.

In contrast to the training phase, the computational overhead required for local NMT inference during the evaluation phase was negligible, consuming only a few minutes to process the entire test dataset.

E.2 Large Language Model (GPT-5.2)

All LLM calls used GPT-5.2 (gpt-5.2-2025-12-11) via the official API. Temperature was set to 0.0 to ensure reproducibility. Max output tokens was set to 100 to suppress wasted token consumption caused by repetition loops. All other parameters were left at their defaults.

Table 8 shows the average total number of tokens per sentence and the total number of tokens for each method and translation direction across all three experiments ($n = 485$ sentences per experiment).

F Ablation: removing the numerical probability values

In Dist-Gen, the candidate list at each decoding position conveys information from the NMT distribution in two ways: (i) through which tokens are retained and in which order they are listed (Section 3.1), and (ii) through the numerical probability attached to each candidate (see Prompt 2). To isolate the contribution of the latter, we conducted an ablation in the main experimental setting (Section 5.1) in which the numerical values are removed from the prompt while everything else is held fixed: each candidate is presented as token instead of `token(prob: 0.xxx)`, and the source sentences, the candidate sets, their order, the system prompt, and the lexicon condition are identical to those of the corresponding Dist-Gen run. Note that this ablation does not remove all information derived from the NMT distributions: the selection and the ranking of the candidates are still determined by their probabilities. What is removed is thus the explicit magnitude of the probabilities and of the differences between them.

Table 9 reports chrF for both translation directions, and Table 10 reports SBERT-cos for the Ainu-to-Japanese direction. In both tables, differences are computed from the unrounded scores as the ablated condition minus the original Dist-Gen condition. Statistical significance is assessed with the same procedures as in Section 4.4.1 and Section 4.4.2: for chrF, the paired bootstrap resampling test of sacreBLEU (with 5,000 resampling iterations), reported as p -values; for SBERT-cos, a paired bootstrap of the mean difference (with 5,000 resamples), reported as 95% confidence intervals.

In the Ainu-to-Japanese direction, removing the numerical values has no detectable effect: chrF changes by -0.009 without and by -0.367 with lexical information, neither of which is significant, and both SBERT-cos confidence intervals include zero. The absence of a measurable difference is not due to the LLM ignoring the change in the prompt: only 34.6% (without lexicon) and 38.1% (with lexicon) of the ablated outputs are string-identical to the corresponding Dist-Gen outputs. The outputs

Direction	Method	Main experiment (Section 5.1)	Ultra-low resource (Section 5.2)	In-domain (Section 5.3)	Overall
Ain → Jpn	Dist-Gen w/ lexicon	820.36 (397873)	1058.43 (513340)	718.04 (348247)	865.61 (1259460)
	Dist-Gen	667.04 (323513)	905.45 (439143)	583.68 (283085)	718.72 (1045741)
	GPT-5.2 w/ lexicon	386.63 (187514)	386.63 (187514)	365.91 (177465)	379.72 (552493)
	GPT-5.2	231.88 (112460)	231.88 (112460)	229.49 (111304)	231.08 (336224)
	Post-editing w/ lexicon	503.17 (244037)	503.80 (244344)	477.51 (231591)	494.83 (719972)
	Post-editing	402.00 (194972)	402.93 (195420)	394.85 (191501)	399.93 (581893)
Jpn → Ain	Dist-Gen w/ lexicon	798.58 (387310)	1132.09 (549063)	713.86 (346224)	881.51 (1282597)
	Dist-Gen	654.82 (317586)	989.50 (479907)	583.18 (282844)	742.50 (1080337)
	GPT-5.2 w/ lexicon	402.90 (195406)	402.90 (195406)	385.64 (187035)	397.15 (577847)
	GPT-5.2	263.59 (127841)	263.59 (127841)	260.98 (126577)	262.72 (382259)
	Post-editing w/ lexicon	557.22 (270250)	557.33 (270304)	534.29 (259133)	549.61 (799687)
	Post-editing	479.99 (232796)	481.55 (233553)	471.86 (228853)	477.80 (695202)

Table 8: Average number of input tokens per sentence, with total token counts in parentheses, for each method and translation direction across all three experiments ($n = 485$ sentences per experiment).

Direction	Lexicon	Dist-Gen	w/o prob. values	Diff.	p
Ain → Jpn	w/o	29.832	29.824	-0.009	0.421
	w/	32.238	31.871	-0.367	0.151
Jpn → Ain	w/o	45.591	40.933	-4.659	<0.001
	w/	40.757	35.934	-4.824	<0.001

Table 9: chrF of Dist-Gen with and without the numerical probability values in the prompt (main experimental setting). $p < 0.001$ corresponds to the smallest value attainable with 5,000 resamples.

Lexicon	Dist-Gen	w/o prob. values	Diff.	95% CI
w/o	0.739	0.738	-0.002	[-0.010, 0.006]
w/	0.760	0.756	-0.004	[-0.012, 0.004]

Table 10: SBERT-cos of Dist-Gen with and without the numerical probability values in the prompt (Ainu → Japanese, main experimental setting).

do change, but to translations of statistically indistinguishable quality under both metrics.

In the Japanese-to-Ainu direction, in contrast, removing the numerical values lowers chrF by about 4.7 points in both lexicon conditions, and only 9.9% (without lexicon) and 12.2% (with lexicon) of the outputs remain identical. The numerical values thus appear to keep the LLM’s output closer to the NMT model’s preferred tokens, which is beneficial in this direction, where every departure of the LLM-based systems from mBART’s output tends to lower chrF (Table 2) because of the LLM’s limited command of Ainu. Even with the numerical values, however, Dist-Gen (45.591) remains below mBART (50.180; Table 2), so the values mitigate a degradation rather than produce an improvement.

The ablation indicates that in the direction where Dist-Gen improves over mBART, the improvement cannot be attributed to the explicit confidence val-

Direction	Post-editing	Post-editing w/ lexicon	Dist-Gen	Dist-Gen w/ lexicon
Ain → Jpn	0.473	0.763	0.675	0.826
Jpn → Ain	0.673	0.660	0.608	0.716

Table 11: Levenshtein distances between mBART’s translations and the outputs produced by methods informed by it, in ultra-low-resource setting.

Direction	Post-editing	Post-editing w/ lexicon	Dist-Gen	Dist-Gen w/ lexicon
Ain → Jpn	0.318	0.433	0.290	0.418
Jpn → Ain	0.335	0.298	0.181	0.367

Table 12: Levenshtein distances between mBART’s translations and the outputs produced by methods informed by it, on data matching the NMT fine-tuning domain.

ues. It is consistent with being carried by the probability-derived candidate sets and their ranking, by the lexical information, and by the generation (rather than post-editing) formulation of the task – the present ablation does not separate these factors, and an ablation that additionally removes the ranking (e.g., by shuffling the candidate order) is left for future work.

G Levenshtein distance

We report Levenshtein distances between mBART translations and the outputs of post-editing and Dist-Gen for the ultra-low resource and in-domain settings.

Table 11 shows the results for the ultra-low resource setting, and Table 12 shows the results for the in-domain setting.

H Qualitative analysis

This appendix complements the aggregate scores with four representative examples: two success

cases and two failure cases. The examples were identified by inspecting the outputs of all compared systems in all experimental settings and both translation directions.

For each example we report the source, the reference, and the outputs of all compared systems together with sentence-level chrF (computed with sacreBLEU 2.6.0, as in Section 4.4.1), the lexicon entries retrieved for the source sentence where relevant, and the part of the NMT token-level probability distributions most relevant to the phenomenon in question (probabilities rounded to three decimal places; “_” marks the SentencePiece word-boundary marker, as in Prompt 2). Two caveats apply throughout. First, chrF is a surface string-similarity metric and does not by itself establish semantic or grammatical correctness, so we discuss the correctness of content words separately from the scores. Second, each case is a single example: we restrict ourselves to differences that are directly observable in the outputs and refrain from causal claims about how the models internally used particular signals.

H.1 Example (a): avoiding a mistranslation fixed in the 1-best hypothesis

Table 13 shows an Ainu-to-Japanese example from the main experiment (Section 5.1). The source contains the personal name *Cikiriaskay*. mBART renders it by its literal meaning as 「足が達者だ」 (roughly “being good on one’s feet”), and both post-editing variants retain this reading, which is fixed in the single NMT hypothesis they receive. Dist-Gen with lexical information instead produces the transliteration チキリアスカイ (one syllable away from the reference チキリアシカイ) and obtains the highest sentence-level chrF among all systems. The initial character of the transliteration, チ, occurs in the NMT distribution only as the third-ranked candidate at position 2 (probability 0.027), far below the literal reading 足 (0.493). The candidate that initiates the transliteration is thus visible to Dist-Gen through the retained NMT candidate list, but not to post-editing, which receives only the 1-best hypothesis.

Two observations bound what this example can show. First, both GPT-5.2 variants, which see the source (where the name appears in Latin script) but no NMT information, produce the same transliteration; the example therefore demonstrates an output-level difference between constructing a translation from the distribution and correcting the 1-best hy-

pothesis, but cannot by itself establish that Dist-Gen causally exploited the lower-ranked candidate. Second, the retrieved lexicon entries contain no translation of the name itself, yet Dist-Gen *without* lexical information retains the literal reading; in this instance, the departure from the 1-best hypothesis thus occurred only under the lexicon-informed prompt.

H.2 Example (b): content-word correction supported by lexical information

Table 14 shows an Ainu-to-Japanese example from the main experiment. mBART renders the bird name *eyami* (カケス, ‘Eurasian jay’) as 刺繍 (‘embroidery’), and the NMT distribution supports this error with high confidence (刺 0.472 at position 7, 繍 0.942 at position 8), while the first character of the correct word, カ, is only the second-ranked candidate (0.181). The retrieved lexicon, however, contains the entry *eyami* → カケス. All three lexicon-informed systems produce the identical, correct output この鳥はカケスだ。 , whereas none of the corresponding no-lexicon variants recovers the bird name: GPT-5.2 transliterates the source word (エヤミ), and both post-editing and Dist-Gen follow the NMT reading 刺繍.

This illustrates the value of accurate, domain-matched lexical information for content words. Note that the effect is shared by all three lexicon-informed methods, so it is not a Dist-Gen-specific advantage.

H.3 Example (c): failure under highly uncertain NMT distributions

Table 15 shows a Japanese-to-Ainu example from the ultra-low-resource setting (Section 5.2). Here the NMT distributions are extremely flat: the top-1 probability is below 0.5 at 11 of the 15 decoding positions, with a mean of 0.384 across positions. Moreover, the reference content words *yupo* (‘elder brother’) and *hemanta* (‘what’) never appear among the retained candidates, and *kusu* surfaces only through the lower-ranked subword candidate *_kus* (positions 8, 9, and 11). Dist-Gen with lexical information assembles its output from the lexicon hints (*te*, *ari*) and from candidates recurring across positions (*a=*, *kor*, *an*, *pe*, *ne*, *na*); the reference words *yupo* and *hemanta* and the sequence *e= a wa e= an* are all absent from its output, whose sentence-level chrF (10.99) falls below both GPT-5.2 variants and both post-editing variants.

For the same source sentence in the main setting,

	Output	chrF
Source	Cikiriaskay sekor ku= rehe an .	—
Reference	チキリアスカイ と言います。	—
mBART	足が達者だと私の名前です。	5.24
GPT-5.2	「チキリアスカイ」と私は名乗っている。	19.84
GPT-5.2 w/ lexicon	「チキリアスカイ」と私の名前がある。	18.95
Post-editing	足が達者だというのが私の名前です。	6.13
Post-editing w/ lexicon	「足が達者だ」と私の名前がある。	2.45
Dist-Gen	「足が達者だ」と私の名前です。	5.08
Dist-Gen w/ lexicon	「チキリアスカイ」と言うのが私の名前です。	22.86
Lexicon	sekor → へと; ku= → 私; rehe → への名前; an → いる、ある	—

position 2: 足 0.493, 千 0.259, チ 0.027, ま 0.026, 道 0.024
position 6: 達 0.487, 利 0.050, 悪い 0.036, よ 0.036, き 0.034

Table 13: Example (a): a proper name in the source (main setting, Ainu-to-Japanese; abbreviations as in Table 2). chrF is sentence-level chrF against the reference. The box shows the retained NMT candidates at the two positions corresponding to the literal reading 足...達者; the transliteration-initial チ is the third-ranked candidate at position 2.

	Output	chrF
Source	tan cikap eyami ne .	—
Reference	この鳥はカケスである。	—
mBART	この鳥は刺繍してある。	26.58
GPT-5.2	この鳥はエヤミというものだ。	17.31
GPT-5.2 w/ lexicon	この鳥はカケスだ。	55.02
Post-editing	この鳥は刺繍してある。	26.58
Post-editing w/ lexicon	この鳥はカケスだ。	55.02
Dist-Gen	この鳥は刺繍したものだ。	18.00
Dist-Gen w/ lexicon	この鳥はカケスだ。	55.02
Lexicon	tan → この; cikap → 鳥; eyami → カケス; ne → へだ	—

position 7: 刺 0.472, カ 0.181, 玉 0.043, 山 0.020, 杭 0.017
position 8: 繡 0.942

Table 14: Example (b): the domain content word *eyami* (main setting, Ainu-to-Japanese; abbreviations as in Table 2). chrF is sentence-level chrF against the reference. The box shows the retained NMT candidates at the positions where mBART generates 刺繡 ('embroidery').

where the mean top-1 probability is 0.737, Dist-Gen with lexical information produces *Ku=yupo! Makanak wa te ta arok wa an?* (chrF 28.95). This contrast is consistent with the entropy-based explanation in Section 5.2: flat, low-quality distributions act as noise rather than signal.

H.4 Example (d): loss of a correct content word in generation into Ainu

Table 16 shows a Japanese-to-Ainu example from the main experiment. The reference contains the content word *wentarap* ('to dream'), mBART's output preserves it, and the NMT distributions contain the corresponding subword sequence with high confidence (*_went* ranked first at position 2 with probability 0.285, followed by the near-deterministic continuations *a* 0.973 and *rap* 1.000). The correct content word was therefore available to the LLM-based systems both in the 1-

best hypothesis (for post-editing) and in the probability distributions (for Dist-Gen). Nevertheless, none of the six LLM-based outputs preserves the contiguous string *wentarap*: five render the source word 夢 as (a fragment of) the romanized Japanese *yume*, and Dist-Gen without lexical information deforms the sequence into *Wen a=rap*. Accordingly, mBART obtains the highest sentence-level chrF (62.47).

This example shows that supplying correct NMT evidence does not guarantee its preservation when the LLM must generate into the low-resource language (this generation-side limitation was also discussed in Section 5.1 and in the Limitations section).

	Output	chrF
Source	兄さん！どうしてここで座っているの？	—
Reference	ku= yupo ! Hemanta kusu teta e= wa e= an ?	—
mBART	a= kor ne kusu ?	10.06
GPT-5.2	Nispa! hemanta an pe ne wa ekotta ta e=rok an?	26.48
GPT-5.2 w/ lexicon	Nispa! Hemanta te ari e=rok wa e=an?	39.10
Post-editing	Nispa! hemanta an roka ta a=rok wa an?	21.66
Post-editing w/ lexicon	Nispa! hemanta kusu te ta apkas oka?	42.38
Dist-Gen	A=kor an pe ne na?	6.87
Dist-Gen w/ lexicon	A=kor te ari a=an pe ne na?	10.99
Lexicon	～して → wa; ここ → te; いる → okay; ～で → ari	—

```

position 1: _a 0.291, _ne 0.063, _se 0.061
position 2: = 0.970
position 3: _kor 0.275, _on 0.059, _e 0.052
position 4: _a 0.098, _an 0.078, _ne 0.057
position 5: = 0.603, sin 0.115, hun 0.113
position 6: _kor 0.262, _e 0.072, _ko 0.034
position 7: _a 0.086, _an 0.056, _ni 0.052
position 8: = 0.458, _kus 0.105, hun 0.084
position 9: _kor 0.207, _e 0.074, _kus 0.056
position 10: _an 0.073, _a 0.059, _ni 0.055
position 11: _pe 0.196, _siri 0.104, _kus 0.058
position 12: _ne 0.628, _? 0.146, _an 0.055
position 13: _na 0.188, _? 0.164, _a 0.116
position 14: _? 0.432, _ 0.216, _e 0.180
position 15: </s> 0.995

```

Table 15: Example (c): failure under highly uncertain NMT distributions (ultra-low-resource setting, Japanese-to-Ainu; abbreviations as in Table 2). chrF is sentence-level chrF against the reference. The box lists, for every decoding position, at most the three highest-probability retained candidates; the top-1 probability is below 0.5 at 11 of the 15 positions (mean 0.384).

	Output	chrF
Source	「そのような夢を見た」と、父が言った。	—
Reference	" neno wentarap =an " sekor onaha hawean .	—
mBART	" ene wentarap =an . " sekor a= ona hawean .	62.47
GPT-5.2	"Sonno yō na yum o ku=ramu" sekor a=ona hawean.	33.92
GPT-5.2 w/ lexicon	"Nerok kor yume ku=nu" newa mici hawean.	23.45
Post-editing	"ene ne yume ku=ramu an" sekor a=ona hawean.	45.20
Post-editing w/ lexicon	"nea okere yume ku=nu=an." sekor a=mici hawean.	32.91
Dist-Gen	"Wen a=rap=an ruwe ne." se kor a=ona hawean.	45.52
Dist-Gen w/ lexicon	"Nerok na yume a-rap=an" newa mici se kor a=ona hawean.	42.19
Lexicon	その → nerok; ～よ → na; ～と → newa; 父 → mici	—

```

position 2: _went 0.285, _nen 0.203, _en 0.202, _tap
0.153, _ne 0.023
position 3: a 0.973
position 4: rap 1.000
position 5: _ = 0.612, _pa 0.132, ' 0.109, _ya 0.035
position 6: an 1.000

```

Table 16: Example (d): the content word *wentarap* (‘to dream’) is present in the NMT 1-best output and its distributions, but preserved by none of the LLM-based systems (main setting, Japanese-to-Ainu; abbreviations as in Table 2). chrF is sentence-level chrF against the reference. The box shows the retained NMT candidates at positions 2–6, covering the subword sequence *_went+a+rap*.