Sakura at SemEval-2025 Task 2: Enhancing Named Entity Translation with Fine-Tuning and Preference Optimization

Alberto Poncelas and Ohnmar Htun
Rakuten Institute of Technology
Rakuten Group, Inc.
{alberto.poncelas,ohnmar.htun}@rakuten.com

Abstract

Translating name entities can be challenging, as it often requires real-world knowledge rather than just performing a literal translation. The shared task "Entity-Aware Machine Translation" in SemEval-2025 encourages participants to build machine translation models that can effectively handle the translation of complex named entities. In this paper, we propose two methods to improve the accuracy of name entity translation from English to Japanese. One approach involves fine-tuning the model on entries, or lists of entries, of the dictionary. The second technique focuses on preference optimization, guiding the model on which translation it should generate.

1 Introduction

The translation of Named Entities is a challenging aspect in machine translation (MT) field, as translating proper names, locations, etc. is not always straightforward.

For instance, "カールじいさんの空飛ぶ家" (meaning "Carl Grandpa's Flying House") is the Japanese version of the name of the film "Up" (see entry "Q174811"[1] in Wikidata). Machine Translation (MT) systems would not be able to translate the title without having explicit knowledge of such name entity.

The "SemEval-2025 Task 2: Entity-Aware Machine Translation" (Conia et al., 2024, 2025) is a task[2] that challenge the participants to develop MT models capable to translate sentences containing complex named entities.

This task becomes even more difficult when translating into Japanese due to the variations in script. Japanese writing uses three scripts, i.e. kanji, hiragana and katakana, each with its own set of rules and purposes. Some words might be written in kanji, while others may require hiragana or katakana.

For instance, in the previously-mentioned film title, カール (*Carl*) is written in katakana, which is typically used for foreign words or names; じいさん (*Grandfather*) appears in hiragana, reserved for native Japanese words and grammatical elements; and 空飛ぶ家 (*flying house*) is in kanji, employed for more complex or meaningful words.

Typically, to integrate new knowledge into an Large Language Model (LLM) techniques like Supervised Fine-Tuning (SFT) are employed. However, doing SFT only with dictionaries may lead to overfitting, as the model might become too focused on single-word translations.

We participated in the Entity-Aware Machine Translation (team *sakura*) to address these challenges. In this paper, we describe and compare different methods of integrating these dictionaries into the training process.

2 Related Work

Several efforts have been made to influence the generation process so that the models produce words that are closer to the desired ones. Many techniques involve fine-tuning the model with biased data. This can be done through data selection (Biçici and Yuret, 2011; Parcheta et al., 2018; Poncelas et al., 2019) or synthetic data generation (Hämäläinen and Alnajjar, 2019).

Dictionaries are also used during decoding by either adding lexical constraints (Hokamp and Liu, 2017; Susanto et al., 2020) or incorporating the dictionary directly into the prompt (Ghazvininejad et al., 2023).

In this paper, we explore entity translation as an LLM alignment (Wang et al., 2024; Kong et al., 2025) problem. Our goal is to promote outputs that are closer to human expectations. Specifically, we apply Preference Optimization, a machine learning

[1] https://www.wikidata.org/wiki/Q174811
[2] https://sapienzanlp.github.io/ea-mt/

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 791–796

technique designed to improve models by focusing on preferences. Rather than relying on a single ground truth target for predictions, it fine-tunes the model using preference data. The objective is to learn the relative desirability of different outcomes, rather than simply predicting a label.

A policy π represents the model's strategy for choosing between different possible outcomes. Therefore, given an input x and two outputs y_w and y_l (with the first output being more desirable than the second) the goal is to find a policy π_θ so that it favors $\pi_\theta(y_w|x)$ over $\pi_\theta(y_l|x)$

An approach to achieve this is Direct Preference Optimization (DPO)(Rafailov et al., 2023) which involves adjusting a policy π_θ compared to a reference π_{ref} in order to increase the log-ratio for the preferred outcome, i.e. $r_w = log\frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)}$, and decrease for the non-preferred, i.e. $r_l = log\frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}$, minimizing the loss

$$L = -\mathbb{E}_{(x,y_w,y_l)\sim D} = [log\sigma(\beta(r_w - r_l))]$$

where β is a scaling factor hyperparameter and σ is the sigmoid function.

3 Proposal

Our goal is to identify an effective method for integrating a dictionary of named entity translations into the knowledge of an MT model. Generally, to adapt a model for specific translation tasks, it is fine-tuned using in-domain data. However, fine-tuning with dictionaries could lead to the model producing shorter sentences, which might hurt the overall translation performance.

3.1 Fine-Tune Lists of Words

We first explore how the performance of the model changes when, instead of providing training instances as pairs of individual named entities, we present them as a list of entities. For example, instead of (source,target) pairs such as (*Kazinform*,カズインフォルム) which correspond to an individual entry in the dictionary. As an alternative we may have "*Kazinform* | *Yasuo Kamon* | *Hinda District* | *Yū Aosawa*" in the English side and "カズインフォルム | 嘉門安雄 | ヒンダ郡 | 蒼澤悠" in the Japanese side. Both sides contain words that are mapped one-to-one with each other, but these name entities are provided as a list. By doing this, we expect the model to not be biased towards translating individual words or concept, but longer sequence.

3.2 Align the Model to a Dictionary

As mentioned, fine-tuning a model with such data may not be a good idea. Therefore, as an alternative, we also explore the behavior of the model when instead of fine-tuning, we use preference-based feedback. Instead of teaching the model new knowledge, our proposal is to alter the translation probabilities of the name entity so the model generates those indicated by the dictionary. We expect to rerank the possible translation candidates so those in the dictionary are promoted.

An example is presented in the diagram of Figure 1. The *base-sft* model translates the term "Akegawa" as 阿部川. However, according to the "Q11515045" entry in Wikidata, it should be translated as 曙川. During Preference Optimization, we train the model to promote the translation of the dictionary over the current output. Consequently, the model can generate the desired output.

In order to do Preference Optimization, the training set consist of triplets of (source,chosen,rejected) as shown in Table 1. We provide more details on how this dataset has been built in Section 4.3.

4 Experimental Settings

4.1 Evaluation

The performance of the models can be evaluated in different aspects:

- Entity Translation Accuracy: The models should translate the name entities in the source sentence accurately, according to the entries in Wikidata. The metric used for this is Manual Entity Translation Accuracy (M-ETA) (Conia et al., 2024), which computes the proportion of entities that are correctly (exact match) translated and is computed as M-ETA $= \frac{\text{\# correctly translated entities}}{\text{\# entities in the reference translations}}$
- Overall Translation Quality: Models should generate accurate translations of the provided English sentence. In order to measure this, we use both Character n-gram F-score (CHRF) (Popović, 2015), which is based on character overlap, and Cross-lingual Optimized Metric for Evaluation of Translation (COMET)[3] (Rei et al., 2022) metrics.

[3]Unbabel/wmt22-comet-da

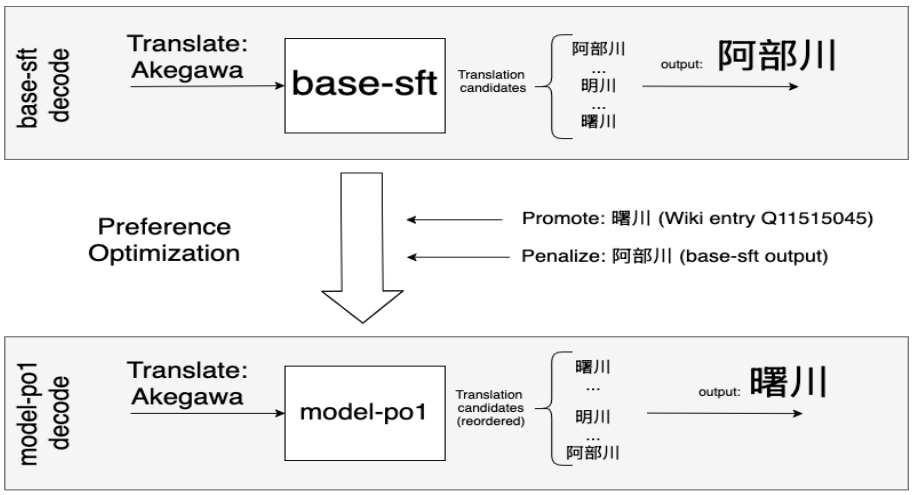

Figure 1: Overview of Preference Optimization. The model *base-sft* produces a translation that does not match the entry in the Wikidata dictionary (above). During Preference Optimization, we specify which terms should be promoted (i.e. the correct translation from the dictionary), and which should be penalized (i.e. the incorrect output from the *base-sft* model). The resulting model, *model-po1*, is able to reorder the translation candidates and produce the correct translation (bottom).

Wiki id	Source	Chosen	Rejected
Q5760427	Hikaru Museum	光ミュージアム	光美術館
Q407486	Air Madrid	エア・マドリード	エアマドリッド
Q11515045	Akegawa	曙川	阿部川

Table 1: Example of preference data.

We report the performance scores on two sets provided by the organizers[4], i.e. *valid* (723 lines) and *test* (5108 lines).

4.2 Baseline Model

In the first stage, we build a strong model in the English to Japanese direction. We decided to use the *RakutenAI-7B-chat*[5] model (Rakuten Group, Inc. et al., 2024) as it has been specially tailored for various Natural Language Processing (NLP) tasks in both English and Japanese languages. In addition to that, it has demonstrated strong performance on translation (Htun and Poncelas, 2024).

We fine-tune this model on English-Japanese parallel sentences in order to build a model specialized in the translation task. For this, we use the *mintaka*[6] dataset (Sen et al., 2022) provided by the organizers of the shared task which contains 7K parallel sentences. By doing this, we increase the performance of the model on the translation task (for example, the performance of in the test set increased from 31.4 CHRF points to 45.1). We will use this fine-tuned model, i.e. *base-sft*, for our experiments.

The model has been fine-tuned on this dataset for one epoch. This approach was applied to all the models presented in this paper, and each was tuned for one epoch.

4.3 Experiments

To explore how to incorporate dictionary knowledge into a model, we follow the approaches described in Section 3 to build new models.

We use *Paranames* (Sälevä and Lignos, 2022) dataset[7] which contains a list of terms and their translations in Japanese according to Wikidata. In total it contains 1.1M terms in Japanese. We translate these terms using *base-sft*, and remove those entries that our model is already capable of translating accurately. We keep the entries where the target Japanese and our translation is not an exact match. After this process, the size of the filter dictionary is

[4] https://huggingface.co/datasets/sapienzanlp/ea-mt-benchmark

[5] https://huggingface.co/Rakuten/RakutenAI-7B-chat

[6] https://github.com/amazon-science/mintaka

[7] https://huggingface.co/datasets/bltlab/ParaNames

850K.

We use this filtered dictionary to create sets of parallel data as described in Section 3.1. In particular, we build three datasets: (i) individual dictionary entries; (ii) lists of five entries, and (iii) lists of ten entries. We fine-tune *base-sft* model with these datasets to build three models: *model-sft1*, *model-sft5* and *model-sft10*.

Additionally, we build preference data as triplets of (source, chosen, rejected) as described in Section 3.2. The source consists of the English-side of the dictionary. We use the Japanese-side of *Paranames* as "chosen" and the translation provided by *base-sft* as "rejected". This is because we want to promote the original entry in the dictionary and downgrade those generated by our model.

We build three versions of the preference data: by grouping the terms in lists of sizes 1, 5 and 10. We use each dataset to build another three models: *model-po1*, *model-po5* and *model-po10* following Preference Optimization technique.

5 Results and Analysis

The results of the models are shown in Table 2 for the valid set, and Table 3 for the test set. After the incorporation of the dictionary we see improvements in terms of M-ETA. However this also impact the translation quality.

5.1 Fine-Tuned Models

The most notable improvement in M-ETA scores occurs when the model is fine-tuned with a dictionary containing single -name entities, i.e. *model-sft1*. This model achieves the highest M-ETA scores, with 25.7 points for the valid set and 29.6 for the test set. However, this comes at the cost of the biggest decline in translation quality, making it the only model that is more than 1 COMET points behind.

Although no models fine-tuned with dictionaries show an improvement in translation quality, we find that fine-tuning with larger entry lists, such as *model-sft5* and *model-sft10*, leads to a smaller reduction in quality. As the list length grows, the decrease in translation quality becomes less pronounced. However, this results in smaller gains in M-ETA scores, and in some cases, such *model-sft10* in the *valid* set, Table 2, it shows lower M-ETA than the baseline. The optimal list length remains unclear, as using 10 entries results in a decrease in M-ETA for the valid set, but the test set shows a score comparable to that achieved when trained with 5 entries.

5.2 Preference Optimization Models

Regarding the models where Preference Optimization technique was used, we observe that the translation quality is similar to the baseline. There is a discrepancy between COMET and CHRF metrics, while CHRF indicates a slight decline, COMET shows either same or improved quality. In any case, the differences compared to the baseline are minimal (less than 1 point difference for both metrics).

In terms of entity translation accuracy, models with Preference Optimization generally show increased the M-ETA scores over *base-sft*. However, these scores are still lower compared to those of fine-tuned models.

These models seem to be unaffected by the number of name entities in the dictionary. Increasing the number of entries does not have an impact on the performance.

6 Conclusion

Our system demonstrated competitive results in both M-ETA and COMET metrics. The leaderboard[8] shows that its performance can be comparable to some of the larger models.

In this paper, it has been shown that fine-tuning the model on the dictionary can improve translation accuracy. However, this comes at the cost of reduced quality. Therefore, we proposed two alternatives that achieve a balanced trade-off between translation quality and entity translation accuracy. The first approach involves fine-tuning with lists of named entity pairs, which helps mitigate the quality decline while improving M-ETA scores. The second alternative utilizes preference optimization, which also results in improved M-ETA scores, while maintaining a similar level of translation quality.

In this study, we utilized the *RakutenAI-7B-chat* model, originally developed for Japanese and English. Consequently, our experiments focused on these languages only. Nonetheless, we believe the proposed approach can be generalized to other language pairs. Furthermore, we want to investigate whether this is applicable to bigger models, like those presented in the leaderboard of the workshop.

[8] https://huggingface.co/spaces/sapienzanlp/ea-mt-leaderboard

Model	M-ETA	Δ	COMET	Δ	CHRF	Δ
base-sft	24.1	-	91.2	-	42.5	-
model-sft1	25.7	1.6	88.0	-3.1	36.1	-6.4
model-sft5	24.2	0.1	90.5	-0.7	39.4	-3.1
model-sft10	22.7	-1.4	90.9	-0.3	41.7	-0.8
model-po1	25.3	1.2	91.3	0.1	42.2	-0.3
model-po5	23.7	-0.4	91.2	0.0	41.8	-0.7
model-po10	25.0	0.9	91.2	0.0	41.7	-0.8

Table 2: Entity translation accuracy and translation quality evaluated in the *valid* set. The column Δ indicates the score difference between the model and the baseline *base-sft*.

Model	M-ETA	Δ	COMET	Δ	CHRF	Δ
base-sft	27.5	-	92.5	-	45.1	-
model-sft1	29.6	2.1	91.4	-1.1	41.2	-3.9
model-sft5	29.8	2.3	92.4	-0.1	44.0	-1.1
model-sft10	29.7	2.2	92.8	0.3	45.8	0.7
model-po1	28.3	0.8	92.6	0.1	44.4	-0.7
model-po5	29.4	1.9	92.5	0.0	44.6	-0.5
model-po10	29.5	2.0	92.7	0.2	44.3	-0.8

Table 3: Entity translation accuracy and translation quality evaluated in the *test* set. The column Δ indicates the score difference between the model and the baseline *base-sft*.

One limitation of this work is that we added the dictionary knowledge as lists of one, five, and ten words. In the future, we would like to explore what sizes are optimal to achieve the best performance. In addition, we want to explore whether adding a combination of these would lead to better results. Another way to further explore this work is to generate synthetic sentences from the dictionary instead of sticking to the list of words.

References

Ergun Biçici and Deniz Yuret. 2011. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283, Edinburgh, UK.

Simone Conia, Daniel Lee, Min Li, Umar Farooq Minhas, Saloni Potdar, and Yunyao Li. 2024. Towards cross-cultural machine translation with retrieval-augmented generation from multilingual knowledge graphs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16343–16360, Miami, Florida, USA.

Simone Conia, Min Li, Roberto Navigli, and Saloni Potdar. 2025. SemEval-2025 task 2: Entity-aware machine translation. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*. Association for Computational Linguistics.

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*.

Mika Hämäläinen and Khalid Alnajjar. 2019. A template based approach for training NMT for low-resource uralic languages-a pilot with Finnish. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, pages 520–525, Sanya, China.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada.

Ohnmar Htun and Alberto Poncelas. 2024. Rakuten's participation in WMT 2024 patent translation task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 643–646, Miami, Florida, USA.

Lingkai Kong, Haorui Wang, Wenhao Mu, Yuanqi Du, Yuchen Zhuang, Yifei Zhou, Yue Song, Rongzhi Zhang, Kai Wang, and Chao Zhang. 2025. Aligning large language models with representation editing: A control perspective. *Advances in Neural Information Processing Systems*, 37:37356–37384.

Zuzanna Parcheta, Germán Sanchis-Trilles, and Francisco Casacuberta. 2018. Data selection for nmt using infrequent n-gram recovery. In *Proceedings of*

the 21st Annual Conference of the European Association for Machine Translation, pages 239–248, Alicante, Spain.

Alberto Poncelas, Gideon Maillette de Buy Wenniger, and Andy Way. 2019. Transductive data-selection algorithms for fine-tuning neural machine translation. In *Proceedings of the 8th Workshop on Patent and Scientific Literature Translation*, pages 13–23, Dublin, Ireland.

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.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. volume 36, pages 53728–53741, New Orleans, USA.

Rakuten Group, Inc., Aaron Levine, Connie Huang, Chenguang Wang, Eduardo Batista, Ewa Szymanska, Hongyi Ding, Hou Wei Chou, Jean-François Pessiot, Johanes Effendi, Justin Chiu, Kai Torben Ohlhus, Karan Chopra, Keiji Shinzato, Koji Murakami, Lee Xiong, Lei Chen, Maki Kubota, Maksim Tkachenko, Miroku Lee, Naoki Takahashi, Prathyusha Jwalapuram, Ryutaro Tatsushima, Saurabh Jain, Sunil Kumar Yadav, Ting Cai, Wei-Te Chen, Yandi Xia, Yuki Nakayama, and Yutaka Higashiyama. 2024. RakutenAI-7B: Extending Large Language Models for Japanese. *Preprint*, arXiv:2403.15484.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid).

Jonne Sälevä and Constantine Lignos. 2022. ParaNames: A massively multilingual entity name corpus. In *Proceedings of the 4th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 103–105, Seattle, Washington.

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619, Gyeongju, Republic of Korea.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with levenshtein transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, et al. 2024. A comprehensive survey of LLM alignment techniques: RLHF, RLAIF, PPO, DPO and more. *arXiv preprint arXiv:2407.16216*.