Using Transformers for Japanese-English Translation

Felice Forby
CSPB 4830 NLP
4/26/2025

Abstract

For machine translation, transformer-based models have been shown to perform better than RNNs and LSTM architectures. However, AI models often struggle with translating non-European and minor languages, especially when dealing with idiomatic expressions, niche topics, or vastly different grammar systems. In this paper, I look specifically at the Japanese-English language pair. First, I evaluate the performance of current open-source transformer models, in particular smaller models, in translating from Japanese to English. Second, I experiment with how fine-tuning a transformer model using a specialized, custom dataset could improve its performance for a specific topic. I also look into how the number of training samples provided during fine-tuning affect the fine-tuning results. The models were scored using BLEU and chrF metrics and training through the Hugging Face `transformers` library. The results suggest that the use of a curated dataset for fine-tuning could improve machine translation quality for a specific topic and that model performance generally increases with the volume of training data.

Introduction

Machine translation has seen remarkable progress since the introduction of transformer-based models, which have largely replaced models using recurrent neural networks (RNNs) and long short-term memory (LSTMs) networks. Despite such advances, there are still many challenges in translating between languages with different grammar systems, especially when one or both languages are low-resource languages or when the subject matter is highly specialized. The Japanese-English language exemplifies many of these challenges with its differing writing system, grammar structures, cultural contexts, and low-context nature.

This project investigates the performance of open-source transformer models in Japanese-English translation and how targeted fine-tuning may improve translations on a particular topic. I focus on smaller, more accessible models that do not use a prohibitive amount of computing resources, while, although LLMs such as ChatGPT can outperform smaller models, their vast size often makes them impractical for use in environments with limited resources and difficult to fine-tune for specialized domains. Thus smaller models are still relevant in many situations, especially when further training is required.

Building on my professional experience as a Japanese-English translator, I aim to evaluate multiple open-source models using three different datasets and two established quantitative metrics—BLEU and chrF—as well as qualitative human assessment. Through human

assessment, I intend to identify common machine translation error patterns related to how the models handle structural differences, idiomatic expressions, and specialized vocabulary.

Furthermore, this project examines whether targeted fine-tuning using domain-specific training data can meaningfully improve the translation quality within that domain. In particular, I use a custom built dataset based on the Ruby programming language documentation. I also explore how the relationship between the volume of training data and the fine-tuning performance improvements. The findings have practical implications for developing domain-specific models where the base transformer models fall short.

By evaluating both current capabilities and limitations of transformer models for Japanese-English translation, this project hopes to increase our understanding of currently available models and insights into effective strategies for improving translation quality through fine-tuning, especially for structurally divergent languages.

Related Work

Transformer Models for Machine Translation

Introduced by Vaswani et al. (2017), the transformer architecture largely replaced LSTMs and RNNs with a design based solely multi-headed self-attention, which eliminated the need for sequential processing and allowed for parallelization during training. The transformer's encoder-decoder structure was able to outperform previous machine translation models on English-to-German and English-to-French translation tasks, achieving both superior BLEU scores and reduced training costs (Vaswani et al., 2017). This architecture has since become the standard for choice for building large language models for a wide variety of NLP tasks (Jurafsky and Martin, 2025).

Transformer Models for Asian Languages

Ngo et al. (2019) investigated the transformer architecture's effectiveness for character-based tokenization in Japanese-Vietnamese translation. Their research highlighted that traditional recurrent neural networks struggled with handling long sequences typical of character-based translation and lacked parallel computation ability. These limitations are particularly problematic for character-based languages like Chinese and Japanese where whitespace is not used between words (Ngo, et al., 2019). They compared recurrent neural network and transformer models using both word-based tokenization and character-based tokenization techniques and found character-based tokenization combined with transformer architecture achieved the best performance based. The findings demonstrate transformers' superior capability in handling long-term dependencies the long sequences of character-based languages (Ngo et al., 2019).

Fine-Tuning Machine Translation Models

Ahmed et al. (2023) explored three techniques for fine-tuning English-to-Japanese transformer models specifically for patent translation, a domain that is especially challenging due to highly technical terminology, legal language, and unique style. They also highlight the difficulties of Japanese-English translation, citing different character sets, different word orders, and the varying degrees of context used (Japanese being high-context with implied information versus English being low-context and explicit). They achieved the best results by fine-tuning a Hugging Face model with a specialized patent dataset and optimizing the model's hyper-parameters, improving the baseline BLEU scores by 41%.

For low-resource languages, Cruz and Cheng (2019) investigated transfer learning as a solution for Filipino. Their approach involved building a large, unlabeled Filipino text corpora for pre-training larger LLMs followed by fine-tuning with a smaller labeled dataset for sentiment classification. Cruz and Cheng compared a transformer model BERT and an AWD-LSTM model UMLFiT using gradually smaller fine-tuning dataset sizes to simulate data resource constraints. The results show error rates increased as training data decreased, with BERT having a lower error increase but requiring significantly more computing resources compared to the LSTM-alternative (Cruz and Cheng, 2019).

Machine Translation Evaluation

While human evaluation remains the gold standard for machine translation, automatic metrics have been develop to reduce bottlenecks and efficiency and cost constraints. One of the most widely used evaluation methods, BLEU (Papineni et al., 2002), counts the word-level n-gram matches between machine translation outputs and references, implementing modified precision and length penalties to prevent inflated scores. The metric is efficient, language-independent, word-order-independent, and correlates highly to human evaluations (Papineni et al., 2002). Though BLEU is widely used for comparison across research, BLEU does not consider recall and struggles with morphological variations (Lee et al., 2023).

chrF (Popović, et al., 2015) offers an alternative character-based approach that utilizes both precision and recall in an F-score calculation. chrF is efficient, does not require additional tooling, is both language- and tokenization-independent, and performs particularly well character-based languages like Chinese, Japanese, and Korean (Lee et al., 2023). Popović, et al. (2015) also found that it correlates well with human evaluations and demonstrated comparable or better performance than other metrics including BLEU, TER, and METEOR.

More advanced alternatives include embeddings-based metrics (BERTSCORE, MEANT) and supervised metrics (BEER, COMET), which show better correlation with human evaluations, but face practical limitations in that they require trained models, human-annotated datasets, and are computationally expensive (Lee et al., 2023).

Data

I used the following open-source datasets from the Hugging Face Hub:

Tatoeba
Tatoeba is a multilingual sentence database maintained by volunteers. Most sentences are short, simple, and cover general topics. The English and Japanese subset contains approximately 200,000 sentence pairs. I used the version on Hugging Face provided by the University of Helsinki NLP research department.

The Kyoto Free Translation Task (KFTT):
KFTT contains around 200,000 professionally translated Japanese-English sentence pairs from Wikipedia articles related to Kyoto. Originally developed by the National Institute for Information and Communication Technology (NICT) and later processed by the KFTT team for use in machine translation systems, KFTT features longer, more complex sentences with specialized historical and cultural vocabulary.

I did not preprocess or clean either of dataset, but I did inspect them and found the Tatoeba data to be high quality and the KFTT dataset has already undergone preprocessing by the creators.

In addition to the above datasets, I built a custom dataset based on the Ruby programming language documentation to introduce domain-specific content that was less likely to have been used in the base model training. This dataset includes a total of 403 Japanese-English sentence pairs, with 229 taken from bilingual documentation websites and 174 generated by Claude.ai. I decided to use synthetic data due to time-limitations to increase the amount of data available for fine-tuning. The sentences are not overly complex but contain technical terms related to Ruby and the Ruby on Rails

Methodology

The first goal of this project was to evaluate current open-sourced transformer-based models for Japanese-English translation. I searched on the Hugging Face Hub to find models built specifically for machine translation and small to run without an excessive amount of compute resources. Four models matched these criteria:

- Helsinki NLP opus-mt-ja-en
- FuguMT
- ElanMT
- Mbart-ja-en

The Helsinki NLP opus-mt-ja-en, FuguMT, and ElanMT models MarianMT-based, which is an efficient transformer similar to the encoder-decoder architecture from the “Attention is All You Need” paper by Vaswanti et al. (Junczys-Dowmunt et al. 2018). Marian MT features 8 attention heads and 6 encoder and decoder layers and is optimized for speed and efficiency (Junczys-Dowmunt et al. 2018).

The Mbart-ja-en model is based on Facebook’s mbart-large-cc25, which has 16 attention heads and 12 encoder and decoder layers, and was trained using a multilingual de-noising technique (Liu et al. 2020). Although I would have liked to use a larger Mbart model, it was too resource intensive and thus I decided to exclude it from the project.

Each model was evaluated with 100 random sentences sampled using the same random seed from the above three datasets. Outputs were scored using BLEU and chrF metrics per dataset to observe how sentence characteristics (simple, complex, or domain-specific) affected the performance. Additionally, I manually analyzed the outputs to identify common errors.

BLEU and chrF were chosen because they are both most widely used and have complementary strengths. BLEU allows for comparison to other studies and chrF better handles character-based languages like Japanese.

After evaluation, I fine-tuned the Helsinki NLP opus-mt-ja-en model using the Ruby dataset. I chose this model because it performed relatively well with simple sentences, but had lower scores for Ruby-specific data than some of the other models. To gain insight into how volume of data can impact fine-tuning results, I fine-tuned four different versions of the model using 100, 200, 300, and 353 training samples, setting aside 50 samples for evaluation.

I used the Hugging Face platform and API in all stages of the project including model selection, dataset download and splitting, evaluation, and fine-tuning.

Results

Base Model Evaluation

The figure below shows model performance across the three datasets. The Marian-based models (Helsinki Opus MT, ElanMT, and FuguMT) performed well on Tatoeba, scoring above 0.40 for BLEU and ~60 for chrF, while the Mbart model did noticeably worse. All models struggled with the more complex KFTT dataset. On the Ruby dataset, Helsinki Opus and Mbart scored between their Tatoeba and KFTT results. ElanMT and FuguMT were able to achieve higher chrF scores than on Tatoeba sentences, though the BLEU scores were slightly lower.

Overall, Mbart performed the worst, while ElanMT and FuguMT had the best results. ElanMT also had the fastest translation times, indicating better optimization for speed.

JA-EN Machine Translation Transformer Model Evaluation

Model	Dataset	BLEU	chrF	Translation time (sec)
Helsinki-NLP/opus-mt-ja-en	Tatoeba	0.429	**60.66**	18.8
	KFTT	0.054	26.64	88.6
	Ruby	0.227	55.79	49.5
Mitsua/elan-mt-bt-ja-en	Tatoeba	**0.438**	60.49	14.3
	KFTT	0.191	**47.67**	52.1
	Ruby	0.345	64.58	26.66
staka/fugumt-ja-en	Tatoeba	0.427	59.34	19.4
	KFTT	**0.208**	47.37	94.8
	Ruby	**0.391**	**67.64**	44.5
ken11/mbart-ja-en	Tatoeba	0.261	48.41	24.7
	KFTT	0.033	23.00	62.5
	Ruby	0.128	44.24	30.9
	* Evaluated with the same 100 random samples from each dataset			

Common Errors

After automatic evaluation, I read through the translation outputs of each models and identified the following common mistakes:

1. Failing to understand the implicit context within the Japanese sentence
2. Awkward word-for-word literal translations
3. Inability to handle long, complex sentences
4. Using incorrect words or misspelling when dealing with specialized topics (in this case, history and programming languages)

Below are some samples of the translation outputs from each dataset:

A) 太陽電池で動く自動車を望んでいる。

Reference: I want a car that runs on solar power.
Model: I want a car that works on solar cells.
Interpretation: Literal translation of 太陽電池, “solar cells” or “solar batteries,” when solar power would have been more appropriate.

B) 私の時計は１０時です。

Reference: It is ten o'clock by my watch.

Model: My watch is ten o'clock.

Interpretation: Unable to understand the implicit context. In Japanese, the subject is "my watch," but it is implied that actual subject is the *time* on the watch.

C) しかし近世に至って表現豊かな楽器である三味線が渡来、完成され、更に箏、胡弓が加わり、これらによる新たな音楽が生まれた。

Reference: However, the shamisen, an instrument full of expression, introduced from abroad and reached its full potential by the early Edo period, and moreover, the koto and kokyu were added and consequently a new kind of music developed.

Model: However, in the next generation, new music has been born with a colorful musical instrument that has been completed and the bow added to it.

Interpretation: Translation does mention specific musical instruments and uses incorrect historic terminology. For example, the instrument "kokyu" (胡弓) is translated as "bow" because the second character 弓 means "bow." Overall, the translation cannot capture the complexity of the sentence and does not make sense.

D) Action Cableのチャネルを通じて、異なるタイプのリアルタイム機能向けの特定のストリームを作成できます。

Reference: Through channels in Action Cable, specific streams can be created for different types of real-time functionality.

Model: You can create a specific stream for different types of real-time functions through the ActionCile channel.

Interpretation: Using the second person "you" is somewhat off-style for technical documentation, which is often passive. Incorrect use of "functions" instead of "functionality," and misspelled the name of a particular library only used with Ruby on Rails, "Action Cable."

Model Fine-tuning

The results of fine-tuning the Helsinki Opus MT model demonstrate that additional training improves performance on domain-specific content. As shown in the figure below, increasing the training samples by 100-sentence increments consistently increased performance metrics. This was surprising, given that fine-tuning often uses thousands of training examples. The model's responsiveness may be a consequence of its small size, as larger language models would likely require much more data to show similar gains.

Helsinki Opus-mt-ja-en fine-tuned with varying number of Ruby-specific training samples

Number of samples	BLEU	chrF
Before fine-tuning	0.250	59.38
100	0.275	60.47
200	0.317	62.37
300	0.326	63.31
353 (All available)	**0.337**	**63.32**
	Evaluated with test set from Ruby-specific dataset	

The example below shows how the output changed with each training iteration. Initially, the model uses “combines” instead of the more appropriate “integrates” until the last iteration, though it still uses the incorrect noun form, “integrations.” The incorrectly pluralized “Action Cable” was fixed after the first iteration. The translation of “WebSockets” changed from “webSocks” to “webSockets” (with incorrect capitalization), then in the finally to the correct capitalization, but reverted to the incorrect term “WebSocks.”

Example output after each iteration

Action CableはWebSocketsをRailsアプリケーションの他の部分と統合し、リアルタイム機能をRubyで記述できるようにします。

EN Reference: **Action Cable integrates WebSockets** with the rest of your Rails application, allowing real-time features to be written in Ruby.

- **Original Model Prediction:**
  Action Cables combine webSocks with other parts of the Rails application and allow you to describe real-time features in Ruby.
- **After 100 samples:**
  Action Cable combines webSocks with other parts of Rails applications and allows you to describe real-time features in Ruby.
- **After 200 samples:**
  Action Cable combines webSocks with the rest of Rails applications so that they can describe real-time features in Ruby.
- **After 300 samples:**
  Action Cable combines webSockets with the rest of Rails applications so that they can describe real-time features in Ruby.
- **After all samples:**
  Action Cable integrations WebSocks with the rest of Rails applications so that they can describe real-time features in Ruby.

Discussion

Based on the fine-tuning outcomes, it is promising to see that small transformer models can be responsive to additional topic-specific training even with a small number of examples. Although smaller models are not as powerful LLMs, they may be more easily trainable for a highly specific use-case and more performant in situations where resources are limited.

One limitation of this project was the small size of the dataset used for fine-tuning. Though the results suggest that more examples will improve the performance of the model, there were not enough samples to test if there is a limit to how much improvement can be realized or if too much data would result in overfitting.

Additionally, I realized early on that it is hard to find good evaluation and training datasets that the models have not already been trained on. As I searched for models to use for this project, many of them used the same datasets for initial training, which has the potential to skew the metrics. This is one reason I decided to create my own custom dataset.

In building up my custom dataset, however, I quickly realized that it is very tedious and time-consuming to do, especially if the language pairs must be copy-and-pasted by hand. There are only a few one-to-one Japanese-to-English resources available on the Internet that I could easily find, which lead me to supplement the data with synthetically generated data.

Originally I did want to experiment with a larger model, like Facebook's newer many-to-many Mbart model, but it prohibitively resource intensive to use. I was not able to get it to translate more than a handful of examples during evaluation, let alone try to fine-tune it, so I decided to drop it from the project.

Conclusion & Future Work

One important insight that I gained from this project is that even small transformer models do a quite well with Japanese-English translation. This was a pleasant surprise for me, as I assumed only large language models could do well with this language pair. Therefore, for specific use-cases, I believe it would worth investing in fine-tuning a small model. Also, due to the lack of good JA-EN datasets, it would be a worthwhile pursuit and valuable contribution to build a high-quality custom dataset on any specialized topic, especially if it were open-sourced.

That being said, I think continuing to build up my custom dataset as future work would make it a valuable asset to the NLP and Ruby programming communities. As Ruby was originally created in Japan, the data could be used to develop MT models for documentation, conference talks, and other materials. A larger dataset would also allow me to continue fine-tuning one of the models I worked with in this project.

From my fine-tuning experiments, I concluded that the volume of data does indeed affect the training results, with even a small increase in the number of examples improving metric scores. However, the small size of my dataset prohibited me from experimenting past 400 training samples. As future work in this area, I would like to rerun the experiments with more training data, perhaps even from a different dataset, to gain further insights. One remaining question I would like to explore is whether using a large amount of fine-tuning data could negatively affect results.

Bibliography

Ahmed, M., Ouda, A., Abusharkh, M., Kohli, S., & Rai, K. (2023). An optimized approach to translate technical patents from english to japanese using machine translation models. Applied Sciences, 13(12), 7126. https://www.mdpi.com/2076-3417/13/12/7126

Cruz, J. C. B., & Cheng, C. (2019). Evaluating language model finetuning techniques for low-resource languages. arXiv preprint arXiv:1907.00409. https://arxiv.org/pdf/1907.00409

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., ... & Birch, A. (2018). Marian: Fast neural machine translation in C++. arXiv preprint arXiv:1804.00344. https://arxiv.org/pdf/1804.00344

Jurafsky, D., & Martin, J. (2025). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models (Third Edition draft). https://web.stanford.edu/~jurafsky/slp3/ed3book_Jan25.pdf

Lee, S., Lee, J., Moon, H., Park, C., Seo, J., Eo, S., ... & Lim, H. (2023). A survey on evaluation metrics for machine translation. Mathematics, 11(4), 1006. https://www.mdpi.com/2227-7390/11/4/1006

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., ... & Zettlemoyer, L. (2020). Multilingual denoising pre-training for neural machine translation. Transactions of the Association for Computational Linguistics, 8, 726-742. https://arxiv.org/pdf/2001.08210

Ngo, T. V., Ha, T. L., Nguyen, P. T., & Nguyen, L. M. (2019). How transformer revitalizes character-based neural machine translation: An investigation on Japanese-Vietnamese translation systems. arXiv preprint arXiv:1910.02238. https://arxiv.org/pdf/1910.02238

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics (pp. 311-318). https://aclanthology.org/P02-1040.Pdf

Popović, M. (2015, September). chrF: character n-gram F-score for automatic MT evaluation. In Proceedings of the tenth workshop on statistical machine translation (pp. 392-395). https://aclanthology.org/W15-3049.pdf

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Datasets

Neubig, Graham (2011). The Kyoto Free Translation Task. http://www.phontron.com/kftt

The Kyoto Free Translation Task on Hugging Face. https://huggingface.co/datasets/Hoshikuzu/KFTT

Tatoeba Dataset on Hugging Face. https://huggingface.co/datasets/Helsinki-NLP/tatoeba

Tatoeba Website. https://tatoeba.org/en/

Ruby and Ruby on Rails Dataset on Hugging Face. https://huggingface.co/datasets/morinoko-inari/ruby-rails-ja-en