# Enhancing Machine Translation of News:<br>Japanese to English Translation

**Victor Tan**
TDT4310 / 2024
victota@stud.ntnu.no

## Abstract

Machine Translation has made significant advances in the recent years. Yet, in some domains like news translation, we would like an even better translation, to avoid misunderstanding the news. This paper examines various approaches and strategies to improve the quality and accuracy of these translations, specifically for news articles translated from Japanese to English.

This research will use a trained model and apply it on a never-seen dataset. We will focus on applying preprocessing techniques and also postprocessing ones, to determine which can improve machine translation of news. Results show an increase of around 7.4% of BLEU Score for our model.

## 1 Introduction

Once criticized for its inability to translate every possible text perfectly, machine translation is nevertheless a tool used by many people nowadays. However, the goal of this study is not to revolutionize Machine Translation, but rather to investigate small-scale techniques that can contribute to improving the understanding of machine translated news documents, and try to understand how they do so.

We will try to follow the Shared Task provided by WMT23 and WMT24 (Workshop on Machine Translation): The WMT is an annual event that brings together researchers and developers in the field of Machine Translation. Its goal is to promote high-quality research and development in machine translation technologies by providing a platform for sharing ideas, methodologies, and results.

It also provides reference datasets and automatic evaluations (metrics) to measure the quality of translations fairly. Fine-tuning pretrained models would have been the best approach, in order to manipulate and create our own techniques on the models. However, it is heavily computationally expensive, making it not possible for us. Thus, we won't be able to follow all the limitations of the Shared Task. The only exception to the rules would be that we will use other pretrained models that aren't listed. Given that we won't train them ourselves, the accuracy and scores of the translation should be lower than the ones provided by WMT results. Again, our goal is only to enhance the translation given a model, so the choice of the model itself shouldn't really matter. We will use the trained model, Helsinki-NLP/opus-mt-ja-en, built on MarianMT architecture.

We will list all the possible preprocessing and postprocessing techniques, and add them one by one. By this incremental enhancements method, we aim to highlight specific challenges and nuances that can be seen in news translations, but also in the differences between Japanese and English, considering the linguistic variations between these two languages.

So, this research aims to tackle the key challenges in improving MT for news contents from Japanese to English for already built models.

## 2 Background

Given the nature of this project, machine translation between two languages with different writing systems, we will explain some specificities.

### 2.1 Japanese writing system

Hiragana ⟶ がくせい
Katakana ⟶ ガクセイ
Kanji ⟶ 学生
Romaji ⟶ gakusei
English ⟶ student

Figure 1: Example of Japanese writing system

There are 4 main Japanese writing systems: Hiragana, Katakana, Kanji and Romaji. The majority of the Japanese written sentences are a mix between the first three. Romanji is the transcription of Japanese into Latin characters, and isn’t commonly utilized within native Japanese texts, but is more common in learning materials.

Similarly to English, Hiragana, Katakana and Kanji can lead to homonymy, where the same Japanese character can have several meanings. For accurate translation, this can often be an issue we want to prevent.

Also, the punctuations in Japanese are not the same as the English ones. We can deduce some equivalents, but there don’t have exactly the same purpose:

Apostrophe ’ —— N/A
Quotation marks " —— 「」
Comma , —— 、
Dot . —— 。

Figure 2: Example of Punctuations differences

We will try to manipulate these punctuations and see if we can increase the quality of the translation.

### 2.2 Tokenization

Tokenization is the process of breaking down a text or a sentence into smaller units called "tokens", which are manageable units for computers. In English, we can generally consider each word as a token, so each space is a separator between the tokens.

However, in Japanese, there is no concept of space between characters. All the Japanese characters have to follow each other, so we can’t properly consider separator. Given that some entities, concepts or ideas need two or more characters together, it means that in Japanese, the process of tokenization is even more complex: we cannot simply separate each character as a token. And if the tokenization is wrong about a single character, the entire sentence can be changed. For instance, "二度と" (nidoto) means "never again" and has to be taken as a whole to be translated. But if the tokenization results in "二" and "度と", we could have the translation "two times", or "twice", which would change the meaning of the sentence.

### 2.3 Metrics

For the task of Machine Translation, the difficulty with the metrics lies in the fact that evaluating the translation quality isn’t a straightforward binary or classification problem. There can be multiple accurate translations that are acceptable.

We will use some metrics allowed by the Shared Task of WMT23 and WMT24. But many of them were not available for us, such as Yisi or prismRef. Instead, we decided to add one more metric, which is still well-known. The metrics are:

- BLEU (Bilingual Evaluation Understudy): it measures the precision of the n-grams (sequences of n words) in the machine translation compared to the reference translations.
- COMET (Conversational Metric for Evaluation of Translation, wmt22-comet-da to be precise):it utilizes a pre-trained deep bidirectional language model and is designed specifically for evaluating translations in conversational contexts. It considers factors such as coherence, and relevance in machine translations compared to human references.
- chrF (character F-score): it measures the similarity between machine translations and reference translations based on character-level precision and recall, unlike BLEU, which operates at the word level.
- ROUGE (Recall-Oriented Understudy for Gisting Evaluation): it evaluates the quality of machine-generated summaries or translations by measuring the recall of n-grams compared to reference summaries. ROUGE is particularly useful for tasks like summarization.

The reason for adding ROUGE is that is our case, we want to translate news articles. One of the possible method is to summarize the document and then translate it. For such task, summarization score is indeed important.

## 3 Related Work

As mentioned, our results will be compared to the findings of WMT23, as WMT24 results are not still available. But we will use the limitations provided by both WMT23 and WMT24, and their allowed models. With the assumption that WMT23 and WMT24 results should be similar, we will have a framework to adhere to and have similar comparisons with the findings papers Kocmi et al. (2023), using the metrics given by Freitag et al. (2023) paper.

First, we wanted to get a general insight of the expected results. Originally, we wanted to trained and fine-tune the models allowed by WMT, which were Mistral 7B, Llama2 7B, mBART and BERT. However, Mistral and Llama were allowed only since 2024, so we don't have insight for these models. Yet, we thought we could use Hendy et al. (2023) paper about GPT models to have some metrics idea: According to the paper, GPT models, with or without zero-shot translation, should perform relatively well but still not as good as the best WMT models. The paper also provides Prompt Selection Strategies that could have been applied to Mistral.

But we were not able to train any of the mentioned models properly, probably because of computational power. So we decided to use a trained model instead.

We also used preprocessing and postprocessing techniques for Japanese. All of them were presented during my teacher's lectures, Björn Gambäck, and in Ekaterina (2022)'s book. We also tried to referring to the insights from Hlaing et al. (2022) paper for Part-Of-Speech Tagging for Machine Translation, but POS tagging also requires fine-tuning, so we couldn't use POS tagging.

## 4 Architecture

Because we couldn't train nor fine-tuned our models, we had to use a trained model and apply it on a never-seen dataset. But first, we should introduce the dataset used:

### 4.1 Dataset

We used one of the allowed data of the WMT Shared Task: JParacrawl V3.0. According to their paper Morishita et al. (2020) and website, it was the largest bilingual Japanese-English corpus available, created by crawling the web and automatically aligning parallel sentences. This dataset was chosen because it covers a wide range of fields, randomly, without any specific topic. However, it means that the dataset itself is not perfect, as the reference text could contains mistakes. With 22 millions of sentences, it is initially intended to be used as both training and testing sets. But because we are not training our models, we will only use 150 000 randomly selected sentences as a test set. For reproducibility, our random seed will be 42. Each pair of sentences has a score which determine the quality of the translation. But without reference sentences, this score isn't as meaningful as a metric in section 2.3.

### 4.2 Model

The model used is Helsinki-NLP/opus-mt-ja-en, available on HuggingFace and based on MarianMT transformer model. It was trained on Opus corpus, and has never seen JParacrawl sentences, according to their website. However, given that JParacrawl was created by crawling the web, we won't know exactly if Opus corpus used or not the same sentences. This obviously is a drawback, as overfitting can occur if the sentence has already been seen during training. Fine-tuning the model from scratch would have been better. Even if it is not allowed by WMT limitations, it was the only trained model suited for us.

This model is linked to its own Tokenizer, *MarianTokenizer*, which has several purposes when paired with the model:

- It segments the sentence into token (words segmentation). In our case, it will try to correctly separate the Japanese characters in order to preserve the meaning,
- It will use its vocabulary to attribute a number to each token (Japanese to ID). It is called the encoding. For MarianMT, an End-Of-String token is written at the end of the sentence which is "</s>" with ID 0,
- The model can now use the tokenized version of the sentence and generate a translation, but it is still encoded in IDs,
- The tokenizer can decode the generated translation with another vocabulary (ID to English).

We can see that points 2, 3 and 4 have to be done together. The encoding, generating and decoding phases can only be done with the same model and Tokenizer. However, the first point, about separating the Japanese characters, can be done by another tokenizer: We will try to use several tokenizers and see if the translation would be better or not.

| Architecture | Dropout | Attention Dropout | Num beams | Hidden Layers | Activation Function | BOS Token | EOS Token | Pad Token |
|---|---|---|---|---|---|---|---|---|
| MarianMT | 0.1 | 0.0 | 4 | 6 | swish | N/A (id: 0) | </s> (id: 0) | <pad> (id: 60715) |

Table 1: Model used

## 5 Experiments and Results

We tried to follow the limitations of WMT23 and WMT24, but given our computational power, we couldn't effectively train the models provided by WMT.

### 5.1 Original experiments

We wanted to first fine-tune a Mistral and an mBart model. Our Mistral model could translate some sentences, but could randomly decide to not stop generating text even when provided with End-Of-String Tokens. Our mBART model couldn't translate anything at all and would only repeat back the input sentence in Japanese. We got a BLEU Score of 0.33 / 100 after days of training for Mistral (and obviously 0 for mBart) and decided not to fine-tune any model. The results are both due to computational power, but also due to our lacking knowledge on transformers coding practices.

### 5.2 Experimental Setup

All the experiments were made with the dataset loaded into a pandas dataframe, from which we took a sample of 150 000 entries with random state of 42. A lower number of entries would have been better. However, considering our dataset which is not made with an accurate corpus, we thought that our preprocessing steps would greatly reduce this number. The input text is the Japanese sentences, and the reference text is the English sentences. The prediction is the translation produced from the input text.

We used maximum length padding, 256 maximum output length, and number of beams of 4. Given our computational power, a greater number of beams would be inefficient, so we only tried with 3, 4 and 5, where 4 gave us the same results as 5.

The baseline method considers neither preprocessing nor postprocessing task. It could produce relatively good translations sometimes but many mistakes are due to bad preprocessing.

We also tried to see what would happen when using a model without its intended tokenizer. We used mBart50 Tokenizer, which is supposed to be similar to MarianMT Tokenizer. This will be experiment *mBart50 Tokenizer*. But they don't use the same Beginning-Of-String Token so nothing is translated.

### 5.2.1 Preprocessing

If we try to change the Japanese punctuations into English ones, experiment *Punctuations-ja-to-en*, we actually decrease the quality of the translation. The encoding of the punctuations will differ, and the model, which is not used to them, won't generate proper sentences, or even not translate part of the sentences.

Japanese sentences don't use spaces. In experiment *No spaces*, we try to delete every spaces. However, some dates needed spaces, so we still saw an slight decrease in the scores, so decided not to use this method for all future experiment.

If we look into the Japanese input texts, we can see that there are many words that are in English. In experiments *English deletion*, we delete all the sentences that contains Latin characters in the Japanese sentences. This is mainly due to the fact that some English words are better when not translated, such as Named Entities, websites, or domain specific terms (failback, save, ...). One way to solve this problem could be to only translate English words to Japanese then translate the sentence back into English again, but that would add another layer of errors and would be ineffective. This preprocessing step of deleting all Japanese sentences that contains Latin characters is applied for all future experiments.

If we look into the English reference texts, we can also see some errors. Many words are not in English, but in Japanese or even in Chinese. In experiment *Japanese deletion*, we delete all the sentences that contains Japanese or Chinese characters in the English sentences. If not, our model would have generated an English sentence but we wouldn't be able to compare it with the reference text that should have been in English. The metrics would have been near 0 for such sentences. This preprocessing step is also applied for all future experiments.

Considering our task, where we want to get a translation of news, we thought about translating the summary of the news instead. In experiment *Summarized*, we first summarized both English and Japanese sentences then tried to generate the translation. However, this task was not efficient in our case, because our dataset is already composed of independent sentences (not several successive sentences), and summarizing a line of the dataset would often result in the same line without any changes. Worse case, the summarizing cuts the sentences and we lose the meaning. This method could have been better if instead of single sentences, the dataset had entire news articles. This preprocessing step is not applied for all future experiments.

Then, we decided to add several other tokenizers for separating the Japanese Characters. In experiments *MeCab*, *Spacy ja_core_news_md* and *KyTea*, we use different tokenizers for word segmentation. We will still need MarianTokenizer for encoding and decoding. Considering that, in our case, the quality of the translation decreases, so this preprocessing step is not applied for all future experiments.

Finally, we wanted to have better reference English texts, by modifying the sentences. As this is probably not expected by WMT (or any translation shared task), we know it could be considered as cheating. The results are not comparable with WMT results, but could provide us the expected results if the dataset was more accurate. In experiment *Google translated + Best method*, we used Google Translation in order to change the reference English texts, and use the best method. To be precise, we first google translated the Japanese input texts into Google translated sentences and used them as reference English texts when computing metrics score with our previous best model.

### 5.2.2 Postprocessing

Postprocessing is heavily dependent on the model and the data used, but also on the sentence itself: In the same dataset, one problem could occur for some sentences but not for others, so we couldn't build a rule-base method, and try to manually select certain errors in the dataset after translation instead.

The major errors were:

- Dates: The formats of dates might differ between "MM/YYYY", "MM/YY" or "Month Year", etc... No rule-base methods could be applied.
- Special characters: Some special characters like "€", "%", "-", ... are not correctly understood by our model, no matter the method. It could be because it was not trained enough on these characters. One method we thought was to replace the characters by their full-text translation. For instance, "€" would be "ユーロ" (Romaji: Yūro). However, the translation would sometimes be worse, so we didn't applied it globally.
- Repetitive broken translation: Sometimes, the translation won't work and repeat the same sentence or word. Examples are "etc., etc., etc., etc., etc., etc., etc., etc., etc.", or "It's a lot of wind, but it's a lot of wind, and it's a lot of wind.". One rule could be to delete successive n-grams, but we find that for these two examples, the BLEU score was 0 before and after this method (BLEU score with max order > 1), so the translation itself was not accurate enough.
- Punctuation: Our metrics disregard punctuation, so did not use normalization. However, when the Japanese input sentence contains a Japanese comma "、", the model would sometimes delete (or forget) everything that came before the comma. We checked the attention mask, which is fine, but couldn't understand why the first part of the sentences is deleted. Deleting the comma would either solve the issue, not do anything or make it worse. So we couldn't provide a method.

## 5.3 Experimental Examples

We can consider one example and the results for single sentence translation.

| | |
|---|---|
| English | Lesson : Conversational English - 5 lesson pack This is a 5-lesson pack. |

| | |
|---|---|
| Japanese | レッスン：Conversational English - 5 lesson pack これは合計5回のパックレッスンです。 |
| Translation | Lesson: International English - 5 without pack this is a total of five pack lessons. |
| Scores | Bleu: 38.34 \| chrf: 50.96 |

If we translate the English words first and delete the hyphen, which is not commonly used in Japanese:

| | |
|---|---|
| Japanese | レッスン：コンバセーショナルイングリッシュ5レッスンパックこれは合計5回のパックレッスンです。 |
| Translation | Lesson: Conversative English 5 lesson pack. This is a total of five pack lessons. |
| Scores | Bleu: 49.25 \| chrf: 65.23 |

Table 2: Example of Translation without English words in Japanese text

So we have an increase of 28.00% to 49.20% in the score depending on the metric. However, this was only possible because we could manually and perfectly translate the English words. As explained, this is not possible for all sentences, so we prefer to delete them.

## 5.4 Experimental Results

Here is a summary of the results:

| Method | BLEU | ROUGE | chrf | COMET |
|---|---|---|---|---|
| Baseline | 13.57 | 46.69 | 39.17 | 73.93 |
| mBart50 Tokenizer | 00.01 | 00.75 | 01.04 | 00.28 |
| Punctuation-ja-to-en | 12.98 | 45.22 | 37.97 | 72.75 |
| No spaces | 13.57 | 46.63 | 39.14 | 73.89 |
| English deletion | 14.25 | 47.34 | 40.20 | 75.20 |
| Japanese deletion | **14.58** | **47.34** | **41.21** | **75.23** |
| Summarized | 14.14 | 44.39 | 39.76 | 72.91 |
| MeCab | 14.20 | 46.79 | 40.29 | 74.60 |
| KyTea | 04.01 | 24.61 | 22.67 | 51.91 |
| SpaCy ja_core_news_md | 14.21 | 46.79 | 40.31 | 74.59 |
| Google translated + Best method | 23.15 | 57.70 | 49.45 | 79.10 |

Table 3: Results of Preprocessing Methods applied to Machine Translation from Japanese to English (Green Methods are also applied for all methods below the current one)

Some hypothesis of explanations were provided in Section 5.2.1. We can see that the tokenizer used by MarianMT already provides the best scores, and that utilizing another word segmentation decreases them. We didn't know the reason why, but our hypothesis is that MarianMT encoder prefers to only use MarianMT word segmentation. With current HuggingFace documentation, we don't really know how exactly it uses a word segmentation first, which might result in invalid encoding.

Furthermore, we can see that in our task, the main challenge is to actually get a proper dataset, with reference text matching the input ones, and that using Google Translation to get better reference texts is drastically increasing the score. Yet again, it is definitely not a technique that should be used in a Shared Task, and is only here to provide insight.

For comparison, the best overall model in WMT23 was the one submitted by SKIM team in Kudo et al. (2023) paper, according to Kocmi et al. (2023).

| | | | | |
|---|---|---|---|---|
| SKIM | 24.8 | N/A | 51.1 | 84.0 |

## 6 Discussion

We also thought about other preprocessing techniques that could have been applied. One commonly used in Machine Translation metrics tasks is to control the length of the sentences. If the input text is too short or too long, the translation is expected to be bad, so we can delete those sentences. However, our task is not to get the best score overall, but to try to enhance the translations by choosing the right techniques. As short or long sentences are very common in documents, we did not use this technique of deletion, but prefer finding suitable techniques, resulting in a lower score than WMT results.

Another preprocessing technique was to use the score provided by the JParacrawl dataset, and use a threshold to delete every pair of sentence which has a score below it. However, we didn't know how this score was computed: in the first 2 lines of the data set, we have the following.

| | |
|---|---|
| Line of the dataset | 1 |
| Japanese | 鼻・口のところはあらかじめ少し切っておくといいですね。 |
| English | And everyone will not care that it is not you. |
| Scores | 0.535 |
| Google Translation | You can cut a little off the nose/mouth beforehand. |

| | |
|---|---|
| Line of the dataset | 2 |
| Japanese | アドレス置いとくので、消されないうちにメールくれたら嬉しいです |
| English | And everyone will not care that it is not you. |
| Scores | 0.557 |
| Google Translation | I'll leave you my address and I'd be happy if you email me before it's deleted. |

Table 4: Two first lines of the dataset with the same English reference text for two Japanese texts

Both lines have the same English texts while having different Japanese texts. Furthermore, their respective translation is not close to the English one, while having an average and close score. So we decided to not consider the score provided by JParacrawl in order to avoid modifying the dataset further: Applying *English deletion* and *Japanese deletion* already deleted 51 431 sentences. We here made the assumption that the scores computed for each methods are comparable. We could also use other datasets for more coverage.

A final discussion could be focused on personal pronouns. In Japanese, some sentences omit pronouns because the context or the previous sentence doesn't necessitate their use. However, in our case, it's a problem because we only have isolated sentences without any context, and we can't do any technique to prevent it.

## 7 Conclusion and Future Work

During the research, we highlighted the main challenges that could arise in Machine Translation between two languages with different writing systems. The main factor in order to have a good Machine Translation model is to have a good dataset, with accurate bilingual corpus. Moreover, we discussed about the preprocessing and postprocessing steps that could be used in Japanese to English translations tasks.

Our techniques provided an increase of 7.44% in BLEU score, 1.39% in ROUGE score, 5.21% in chrf score and 1.75% in COMET score. The results are not totally comparable to WMT results, because we did not use the same dataset. Yet, we still have scores similar to the least competitive models in WMT23. We do not have the results for WMT24 yet (conference in November 2024), so our model would probably rank even lower.

If we could fine-tune the model, we could use POS tagging and parsing so the model could have additional context. According to Hlaing et al. (2022) paper, it is possible to fine-tune MarianMT model for POS tagging. Our hypothesis is that providing POS tags would enable the tokenizer to perform better segmentations. In some cases, it might also help for homonyms. We could also try to fine-tune Mistral again.

Yet, our primary goal is to enhance the translation of news, so document-based translation instead of sentence-level translation might yield better results for our specific use case, and we should use another dataset.

## References

Kochmar Ekaterina. *Getting Started with Natural Language Processing*. 2022. ISBN 9781617296765.

Markus Freitag, Nitika Mathur, Chi-kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Tom Kocmi, Frederic Blain, Daniel Deutsch, Craig Stewart, Chrysoula Zerva, Sheila Castilho, Alon Lavie, and George Foster. Results of wmt23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the Eighth Conference on Machine Translation*, pages 578–628, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.wmt-1.51.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. How good are gpt models at machine translation? a comprehensive evaluation, 2023.

Zar Hlaing, Ye Kyaw Thu, Thepchai Supnithi, and Ponrudee Netisopakul. Improving neural machine translation with pos-tag features for low-resource language pairs. *Heliyon*, 8:e10375, 08 2022. doi:10.1016/j.heliyon.2022.e10375.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, OndÅ™ej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja PopoviÄ‡, and Mariya Shmatova. Findings of the 2023 conference on machine translation (wmt23): Llms are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.wmt-1.1.

Keito Kudo, Takumi Ito, Makoto Morishita, and Jun Suzuki. SKIM at WMT 2023 general translation task. In Philipp Koehn, Barry Haddow, Tom Kocmi, and Christof Monz, editors, *Proceedings of the Eighth Conference on Machine Translation*, pages 128–136, Singapore, December 2023. Association for Computational Linguistics. doi:10.18653/v1/2023.wmt-1.9. URL https://aclanthology.org/2023.wmt-1.9.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3603–3609, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://www.aclweb.org/anthology/2020.lrec-1.443.