Illini's Japanese ↔ English News Translation for WMT 2021

Giang Le and **Shinka Mori** and **Lane Schwartz**
Department of Linguistics, University of Illinois
gianghl2@illinos.edu, shinkam2@illinois.edu, lanes@illinois.edu

Abstract

This system paper describes an end-to-end NMT pipeline for the Japanese ↔ English news translation task as submitted to WMT 2021, where we explore the efficacy of techniques such as tokenizing with language-independent and language-dependent tokenizers, normalizing by orthographic conversion, creating a politeness-and-formality-aware model by implementing a tagger, back-translation, model ensembling, and n-best reranking. We use parallel corpora provided by WMT 2021 organizers for training, and development and test data from WMT 2020 for evaluation of different experiment models. The preprocessed corpora are trained with a Transformer encoder-decoder neural network model. We found that combining various techniques described herein, such as language-independent BPE tokenization, incorporating politeness and formality tags, model ensembling, n-best reranking, and back-translation produced the best translation models relative to other experiment systems.

1 Introduction

Despite recent advances in machine translation made possible by neural networks with attention mechanism (Bahdanau et al., 2014; Luong et al., 2015), the Japanese-English pair remains one of the toughest language pairs for machine translation systems to handle. Challenges posed by this language pair are multifaceted, starting from seemingly trivial differences in orthographic representations to deep structural divergence in syntax. This paper describes an end-to-end neural machine translation system and related experiments dedicated to the News Translation Shared Task where the target language pair is Japanese ↔ English, as part of a submission to the Sixth Conference on Machine Translation - WMT 2021. In our experiments, we explored the efficacy of techniques such as tokenizing with language-independent and language-dependent tokenizers, normalizing by orthographic conversion, creating a politeness-and-formality-aware model by implementing a tagger, back-translation, model ensembling, and n-best reranking. We found that normalizing the text by orthographic conversion did not improve over the baseline but controlling for politeness and formality levels of the text increased BLEU by 0.4 points for the en→ja direction, and other techniques such as back-translation, model ensembling, n-best reranking also produced improvements.

The paper gives a detailed review of prior work, with a particular focus on WMT 2020 submissions, and then proceeds to describe our data, model architecture, experiments, results, and discussion of their implications.

2 Prior Work

In this section, techniques and development in neural machine translation will be reviewed with a focus on the techniques and implementation most recently used for the Japanese-English language pair. General techniques deployed across papers submitted to WMT 2020 are bitext data filtering, back-translation, fine tuning with in-domain data, knowledge distillation, rule-based reranking, transfer learning, co-reference processing, hyperparameter search, segmenting by subword units, BPE dropout, model ensembling, pre-training with monolingual data, experimenting with different word segmentation methods, context word embedding, domain adaptation, using related languages in joint training, domain tagging, reranking using backward and forward scores, and dual conditional cross-entropy filtering

(Barrault et al., 2020). In subsequent subsections, representative methods and techniques will be described and the impacts of these methods presented, in so far as they are applicable to the Japanese-English pair.

2.1 Data Preprocessing

Data filtering, cleaning, and normalizing are essential steps in an NMT pipeline, due to the noisy nature of text corpora. A cursory glance at some of the given parallel corpora shows that our data could benefit from additional filtering and cleaning. For instance, the Paracrawl corpus contains a fair amount of duplicates or near duplicates and about 6 percent of the WikiMatrix corpus contains texts outside the source and target language.

Previous submissions to WMT 2020 utilized a mix of language-independent and language-dependent data preprocessing methods to prepare the corpora for training. Researchers also noted a few issues in the parallel corpora requiring special attention; for example, Kiyono et al. (2020) remarked that their translation output contains additional transliteration in brackets after names already transliterated into *katakana*, because these patterns are very common in the KFTT training corpus. They advised that this issue be handled during preprocessing, because postprocessing clean-up, while possible, tended to hurt brevity. Following this suggestion, we incorporated a preprocessing step (described in section 3) to handle these patterns.

2.2 Tokenization

Tokenization is an indispensable step in many natural language processing (NLP) applications. Byte-Pair-Encoding (BPE) by Sennrich et al. (2016c) is a popular compression algorithm that takes care of re-analyzing and splitting words into subword units based on how frequent these units are. The main idea of BPE is to recover smaller subwords that are meaningful and recurring in fuzzy 'word' boundaries in order to compress the vocabulary and decomposes rare words into known subwords. BPE is an effective solution to the issue of rare words, open vocabulary, and agglutinating or polysynthetic morphology in some languages. BPE iteratively replaces the most frequent pair of bytes in a sequence with a single, unused byte. The algorithm works by splitting all words into individual characters, adding them to a vocabulary, and then iteratively merging the most frequency subword pairs and adding them to the vocabulary.

Kudo and Richardson (2018) implemented BPE in SentencePiece, an unsupervised toolkit for word segmentation. A language-agnostic tokenizing and detokenizing algorithm that implements subword unit BPE (Sennrich et al., 2016c) and unigram language model (Kudo, 2018) to tokenize the data, SentencePiece also provides a convenient interface to quickly tokenize and detokenize the data, because its implementation of BPE treats the sentences as sequences of Unicode characters, does not rely on language-dependent logic, and allows training from raw texts. SentencePiece's design makes sure that detokenization is an inverse operation of tokenization and therefore achieves lossless tokenization, for all the information needed to restore the tokenized text is preserved in the encoder's or tokenizer's output. In order to achieve this, the toolkit uses a meta symbol (U+2581) to escape the whitespace character and thereby retains the whitespace in the tokenized text. The detokenizing step is then simply a replacement operation of the meta character back to the whitespace. The developers of SentencePiece experimented their toolkit with and without pre-tokenization for an English-Japanese translation task, and found that the performance of training on raw texts is comparable to training with pre-tokenization.

Previous submissions to WMT 2020 are divided when it comes to which method was preferred for tokenization. Three submissions (Kiyono et al., 2020; Oravecz et al., 2020; Marie et al., 2020) used SentencePiece and three submissions (Kim et al., 2020; Shi et al., 2020; Zhang et al., 2020) used language-specific tokenizers to preprocess Japanese (MeCab) and English (Moses) corpora. MeCab is a popular lattice-based tokenizer for Japanese. It builds a graph-like data structure to hold possible tokens in the text and then uses the Viterbi algorithm to find the best path through the graph. Moses is a well-known statistical machine translation toolkit; its perl scripts are often used to preprocess En-

glish corpora for NMT training (Koehn et al., 2007). We experimented with both SentencePiece and language-dependent tokenizers prior to submission. The details will be outlined in section 5.1 of this report.

2.3 Model Architecture and Hyperparameters Tuning

Most of the papers submitted to WMT 2020 used the Transformer Big settings described in Vaswani et al. (2017) for their NMT model architecture (Marie et al., 2020; Kiyono et al., 2020; Shi et al., 2020; Oravecz et al., 2020; Zhang et al., 2020).

Prior to the publication of *Attention is All You Need*, prominent approaches to sequence-to-sequence modeling include recurrent neural networks, long short-term memory (Hochreiter and Schmidhuber, 1997), and gated recurrent neural networks. All of these approaches suffer from computational bottleneck due to their sequential nature, which prevents parallelization within training examples. The Transformer did away with convolution and recurrence and focused on attention mechanisms, allowing for modeling of long-distance dependencies in parallel. Subsequently, it has been proven to be very successful at handling long distance dependency in natural language, as it allows the model to focus attention on particular source tokens via computation of an attention score. The attention score can be determined by way of different methods, such as a (scaled) dot product (implemented in Vaswani et al. (2017)), bilinear functions, or multi-layer perceptrons. The Transformer achieved state-of-the-art results in English $\leftrightarrow$ French and English $\leftrightarrow$ German translation tasks while cutting down on training time thanks to parallelization.

2.4 Back-Translation

Back-translation is a commonly used method in NMT to augment bitext training data, especially for low-resource languages, by creating an additional synthetic parallel corpus from monolingual corpora (Sennrich et al., 2016b). For this, a model that translates from the target to source language and a model that translates from source to target language is required. Monolingual corpus is translated from the target to source language. The monolingual corpus and the translated synthetic data is appended to the original training data to perform back-translation. It is ideal to have a lower ratio of synthetic data to parallel corpus. As the amount of bitext corpora available for the Japanese-English pair is well under 20 million sentence pairs, Japanese-English can be considered to be a medium-resource language pair and additional back-translated data could help improve translations.

2.5 Model Reranking

Zhang et al. (2020) implemented model reranking following Ng et al. (2019). N-best reranking scores and chooses a translation hypothesis from a list of n-best hypotheses from a source to target model. This method is based on a noisy channel model and Bayesian theorem of conditional probability in log scale.

$$\log P(f|e) + \lambda_1 \log P(e|f) + \lambda_2 \log P(f) \quad (1)$$

The weights λ_1 and λ_2 were learned from fine tuning a validation set. For decoding, they used beam search to generate an n-best candidate list and chose the candidate hypothesis that maximizes (1) as the best hypothesis.

Besides the noisy channel approach described above, reranking can be done using various criteria, such as distortion score, word penality, phrase penality, and so on. Shi et al. (2020) generated n-best candidates by model ensembling of forward translation models, backward translation models, and language models of the target language and then apply K-batched MIRA (Cherry and Foster, 2012) or noisy channel (Yee et al., 2019) to score them. Kiyono et al. (2020) generated n-best candidates from Source-to-Target L2R, R2L models, Target-to-Source L2R, R2L models, Unidictionary Language models, and Masked Language models to compute the scores for reranking. We reranked translation hypotheses using perplexity as a criteria.

3 Data

Our system was trained, developed, and tested fully on data provided by the WMT 2021 organizers, making it a constrained submission. Details of the raw parallel corpora prior to sub-

Corpus	Sentences (M)
JParacrawl 2.0	10.12
News Commentary v16	0.0019
Wiki Titles v3	0.757
WikiMatrix	3.6448
Subtitle Corpus	2.8013
KFTT	0.4438
Ted Talks	0.4462
Total	**18.215**

Table 1: Size of parallel corpora before filtering

Hyperparameters	T-Base	T-Big
Encoder layers	6	6
Decoder layers	6	6
Hidden layers	8	16
RRN	512	1024
d_{ff}	2048	4096
Dropout	0.1	0.3
Optimization	Adam	Adam
Decay	noam	noam
Learning rate	2	2
Warmup steps	8,000	8,000
Train steps	20,000	300,000

Table 2: Model Hyperparameters

stantial filtering[1] used in our baseline and experiment models can be viewed in Table 1.

We used the WMT 2020 development and test sets to compare various experiment models against the baseline: 1998 sentences in the development set in both directions, 1000 test sentences for the en→ja direction, and 993 sentences for the ja→en direction.

From the raw datasets, we applied data filtering to remove noisy data based on two main criteria, alignment confidence and language identification. An alignment score is available for both JParacrawl and WikiMatrix corpora; we chose 0.6 and 1.0 as the threshold for alignment confidence in JParacrawl and WikiMatrix respectively. We used fasttext (Joulin et al., 2017) and its pre-trained language identification model to identify the language of our text sentence-by-sentence, and then we filtered sentence pairs where the language identification confidence score is less than 0.8. We also applied on-the-fly filtering of sentences longer than 100 tokens during training.

According to Kiyono et al. (2020), the KFTT corpus contained instances of having Japanese names followed by its English equivalent in parentheses, which caused their model to append English names after the Japanese name in the translation output, for example キャシディ・ステイ (Cassidy Stay)). To avoid this, we filtered out English translations of names in Japanese source text, specifically WikiMatrix and KFTT, so that any English names in parentheses following its Japanese equivalent were removed. The amount of parallel training data after filtering was 12.7 M for training our submission models.

[1]The original raw WikiMatrix corpus contains 3.8M sentences. We obtained 3.6M after eliminating sentence pairs that do not have the correct language codes in the corpus. That is the only filtering applied to the bitext corpora in Table 1

4 Model Architecture

We trained the parallel corpora using the Transformer base and Transformer big settings as described in Vaswani et al. (2017), presented in Table 2. Pre-submission experiments were trained under the Transformer Base setting while all submission models were trained under the Transformer Big setting. We used the same optimization settings in the Transformer big model as in the Transformer base model. We utilized the OpenNMT toolkit (Klein et al., 2017) with a Pytorch backend to train our models. Most submission models took about 7 days to train on one single NVIDIA GeForce GTX 1080 GPU under the Transformer Big setting.

5 Experiments

5.1 SentencePiece and Language-Dependent Tokenizers

We compared two methods of tokenization for our system. The first is a tokenization method based on BPE and SentencePiece, as described in 5.1. We used SentencePiece (Kudo and Richardson, 2018) to train SentencePiece models for Japanese and English with 32,000 as the vocabulary size. SentencePiece is used to create a tokenizer that depends on subword units, similar to Byte Pair Encoding (BPE). This method of tokenization is especially effective for languages such as Japanese which does not use whitespace to separate words, has

agglutinating morphology, and contains many compound words. Using SentencePiece helps extract subwords within compound words and create a more robust tokenizer. The tokenizer model was used with OpenNMT, which performed tokenization on-the-fly. SentencePiece was used again to detokenize by removing the meta symbols from the output translation.

The second tokenization method that we experimented with is language-dependent. We tokenized English using Moses, following the steps described in Hieber et al. (2018), namely normalizing punctuation in the raw data with normalize-punctuation.perl, removing non-printing characters with remove-non-printing-char.perl, and tokenizing by tokenizer.perl.

For Japanese, we tokenized the data with *fugashi* (McCann, 2020), a Python wrapper of the MeCab morphological analyzer described in 2.2. After tokenization, we applied BPE (Sennrich et al., 2016c) on both Japanese and English with 25,000 merge operations to constrain the vocabulary size.

For this comparison, we used a mid-sized corpus to save time and resources instead of the full 18M corpus. The number of sentences after filtering and preprocessing is 6.4M sentences. We trained the models using the Transformer Base settings, as described in Table 1.

5.2 Politeness and Formality Tagger

Previous work showed that controlling politeness levels has a positive impact on machine translation systems. Feely et al. (2019) implemented a formality-aware tagging method for en→ja NMT. The authors classified formality levels into three categories (informal, polite, and formal) and found that using a heuristics-based tagger improved the system's performance. Similar to Feely et al. (2019), Sennrich et al. (2016a) and Yamagishi et al. (2016) improved on the stylistics of the output (politeness and honorific forms, respectively), by applying a side-constraint approach where target and source suffixes were added during training to add more meta-textual information to the corpora. We tested the effectiveness of this technique on an en→ja translation system.

The news genre is frequently written in fairly formal Japanese. Makino (2008) described politeness and formality in Japanese as orthogonal concepts. It's possible to use polite but informal language in daily polite conversations as well as formal language devoid of polite conjugations such as in news articles, academic papers, and so on. While the given parallel corpora are generally of the latter type, the subtitles corpus contains mostly colloquial language and the Ted talks corpus contains polite endings not intended to be used in news articles.

Due to the presence of mixed writing styles in the training data, we developed a politeness and formality tagger that works in conjunction with the Kytea tokenizer (Neubig, 2011) to address this issue, because we observed that our initial translation outputs often contained polite forms not commonly used in the news genre. Our tagger appends a <polite> or <formal> tag to the beginning of the source (English) side based on the schema in A. Applying this tagger on a 6.4M training corpus results in 33.34% tagged as polite and 5.45% tagged a formal. We tokenized the data using SentencePiece transforms, implemented in the OpenNMT toolkit. We also filtered out sentence pairs longer than 100 tokens. We trained the models using the Transformer Base settings, as described in Table 1.

5.3 Normalizing by Orthographic Conversion

The Japanese writing system uses a combination of three distinctive orthographic scripts: *kanji*, *hiragana*, and *katakana*. *kanji* are Chinese characters, used to write content words such as nouns, verb stems, adjectives, and so on. *hiragana* was derived from *kanji*. It is a phonetic syllabary, typically used to write conjugational endings, particles, and grammatical words. *katakana*, also a phonetic syllabary much like *hiragana*, is typically reserved to write foreign words, loan words, or strengthen the emotive content of the texts. In modern times, the Latin alphabet also has increased visibility due to the popularity of English, and the Japanese language can be transliterated using this alphabet as well. This way of writing Japanese is called *romaji*.

We were interested in examining if converting the raw training texts to other orthographic scripts such as *hiragana* and *romaji*

affects the translation quality of the output. Because *hiragana* and *katakana* have a one-to-one correspondence, it sufficed to experiment with either one of them. Converting the raw text to *hiragana* has a normalizing effect as what it does is reducing the logographic/ideographic *kanji* characters to their pronunciation, the moraic units written in the *hiragana* syllabraries. In that sense, it helps reduce variability in the data and perhaps is beneficial. However, normalizing also strips the text off many contextual cues that would be helpful in translation. The dispersion of *hiragana* in between the content words written in *kanji* is arguably systematic enough for our model to learn that one is used to represent grammatical particles and the other is used to represent objects, names, actions, and so on. Similarly, converting the raw text to *romaji* has a normalizing effect at the quasi-phonemic level. In a related manner, Du and Way (2017) looked at how a model trained on *pinyin* performed on a Chinese → English translation task. They found that using *pinyin* can help alleviate the problem of rare words, although it can introduce ambiguities.

To investigate the question of what impact normalizing the Japanese source text in *hiragana* and *romaji* does, we experimented training three ja→en models where the source text is written in three orthographic scripts, the regular mixed style (baseline), the normalized moraic level *hiragana*, and the normalized quasi-phonemic level *romaji*. Each training corpus contained 4M sentence pairs, after being filtered by setting the language identification score threshold at 0.85 and sampled. The data were preprocessed with SentencePiece and trained under the Transformer Base setting, as described in Table 1.

5.4 Back-Translation

For back-translation, we preprocessed a subset of 4M sentences from the monolingual Newscrawl corpus in the same manner described in 3. The filtered corpus was 3,344,628 lines each. We then used the previously trained ja→en and en→ja model to translate the monolingual data to create synthetic data, setting a beam size of 1 during decoding. We obtained 2.4M and 2.6M sentences of Japanese and English synthetic data from back-translation. This was combined with the existing parallel data to create a corpus of approximately 15M sentences.

Tokenizers	BLEU
SentencePiece ja→en	14.0
Moses and fugashi ja→en	9.9
SentencePiece en→ja	16.0
Moses and fugashi en→ja	9.9

Table 3: Tokenizer Comparison

5.5 Model Ensembling and N-Best Reranking

For n-best reranking, we used a script by Xu Song, `bert-as-a-language-model`[2], which calculates the probability of tokens and perplexity of sentences given a corpus. Using OpenNMT's option to produce n-best translations from an emsemble of several high-performing checkpoints, we created 10 best translations, and used `bert-as-a-language-model` to pick the hypothesis with the highest perplexity score. This method ensures the selected hypothesis has maximized fluency compared to other candidates.

6 Results and Discussion

6.1 SentencePiece and Language-Dependent Tokenizers

We obtained the BLEU scores in Table 3 for our models. The comparison is not entirely fair because the amount of data trained for the moses and fugashi tokenizer to translate in the ja→en direction is 7.3M instead of 6.4M like other models. Additionally, the number of BPE merge operations learned for the language-dependent tokenizer case should have been set to the same as that of SentencePiece for a more equitable comparison.

Using SentencePiece appears to yield better BLEU result in this experiment; however, we also did not keep the other factors constant across the different models under comparison. Nonetheless, this experiment's result led us to adopt SentencePiece as our preferred method for segmentation in other experiments.

[2]https://github.com/xu-song/bert-as-language-model

Models	BLEU
Baseline en→ja	16.0
With tagger en→ja	16.4

Table 4: Tagger Comparison

Orthographic scripts	BLEU
Mixed scripts (baseline) ja→en	14.2
hiragana ja→en	12.6
romaji ja→en	12.8

Table 5: Orthographic Scripts Comparison

6.2 Politeness and Formality Tagger

BLEU scores from a baseline model and a tagger model as seen in Table 4 shows that using a formality and politeness aware model improves the model's performance.

The result of this experiment is very encouraging to us, although the margin of improvement is quite small. The subset of the training data used for this experiment contains 6.4M sentence pairs.

6.3 Normalizing by Orthographic Conversion

We obtained the BLEU scores in Table 5 for our models. It can be seen from the results that training with normalized data by orthographic conversion does not improve the models over the baseline. The models trained on normalized data also have similar performances.

Result of this experiment suggests that normalizing by orthographic conversion might have removed too many contextual cues for the model to perform well. Possible work for future experiments include investigating whether normalizing *katakana* into *hiragana* could have a positive impact, because doing so would remove variability but would not introduce ambiguity to the extent it might have done when the content words in *kanji* were also normalized. Another direction for future research involves looking at training NMT models using sub-character units such as radicals or strokes, as was done in Zhang and Komachi (2018).

Models	BLEU
Baseline en→ja	18.3
With BT data en→ja	18.6
Baseline ja→en	18.8
With BT data ja→en	19.1

Table 6: Back-Translation vs. Baseline

Models	BLEU
Baseline en→ja	32.9
n-best reranking en→ja	34.0
Baseline ja→en	17.6
n-best reranking ja→en	18.6

Table 7: N-best reranking vs. Baseline

6.4 Back-Translation

Using back-translated data significantly improved the results, as shown in table 6. The results reinforce previous findings that back-translation generally improves translation quality, and for languages with low resources it can be especially useful. For the Ja-En pair, the parallel data for news-specific corpus was very scarce, so using the newscrawl and newscorpus was beneficial to the model learning.

6.5 Model Ensembling and N-Best Reranking

During the decoding phase, we ensembled the highest performing checkpoints and obtained 10 best translations from those checkpoints. The best hypothesis was selected by the highest perplexity score of a language model. We found that for both directions, this method resulted in improved translations, as demonstrated in table 7.

7 Conclusion

We produced several models to tackle the task of translating Japanese to English and English to Japanese. Namely, we've used BPE, adjusting for politeness and formality, and during decoding, model ensembling and n-best reranking. Normalizing by orthographic conversion did not produce improvement compared to the baseline, but the other techniques have all proven to be effective and thus have been employed in our final submissions. We also found that for both en→ja and ja→en, adding back-

translated data improved the results. This may be explained by the fact that there is very little parallel data in the news domain, and adding synthetic data from alternative in-domain sources helped tune the model. While improvement in the BLEU score is modest, we expect the results to improve further if we increase the amount of back-translated data.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. Cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada. Association for Computational Linguistics.

Jinhua Du and A. Way. 2017. Pinyin as subword unit for chinese-sourced neural machine translation. In *AICS*.

Weston Feely, Eva Hasler, and Adrià de Gispert. 2019. Controlling Japanese honorifics in English-to-Japanese neural machine translation. In *Proceedings of the 6th Workshop on Asian Translation*, pages 45–53, Hong Kong, China. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2018. Sockeye: A toolkit for neural machine translation.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Jiwan Kim, Soyoon Park, Sangha Kim, and Yoonjung Choi. 2020. An iterative knowledge transfer NMT system for WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 139–144, Online. Association for Computational Linguistics.

Shun Kiyono, Takumi Ito, Ryuto Konno, Makoto Morishita, and Jun Suzuki. 2020. Tohoku-AIP-NTT at WMT 2020 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 145–155, Online. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Seiichi Makino. 2008. *A dictionary of advanced Japanese grammar = Nihongo bunpo jiten. Jokyu hen*, first edition. edition. The Japan Times, Tokyo.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. 2020. Combination of neural machine translation systems at WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 230–238, Online. Association for Computational Linguistics.

Paul McCann. 2020. fugashi, a tool for tokenizing japanese in python.

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

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

Csaba Oravecz, Katina Bontcheva, László Tihanyi, David Kolovratnik, Bhavani Bhaskar, Adrien Lardilleux, Szymon Klocek, and Andreas Eisele. 2020. eTranslation's submissions to the WMT 2020 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 254–261, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. pages 35–40.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Tingxun Shi, Shiyu Zhao, Xiaopu Li, Xiaoxue Wang, Qian Zhang, Di Ai, Dawei Dang, Xue Zhengshan, and Jie Hao. 2020. OPPO's machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 282–292, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2016. Controlling the voice of a sentence in Japanese-to-English neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 203–210, Osaka, Japan. The COLING 2016 Organizing Committee.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. 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 5696–5701, Hong Kong, China. Association for Computational Linguistics.

Longtu Zhang and Mamoru Komachi. 2018. Neural machine translation of logographic language using sub-character level information. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 17–25, Brussels, Belgium. Association for Computational Linguistics.

Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, Yongyu Mu, Jingnan Zhang, Xiaoqian Liu, Xuanjun Zhou, Yinqiao Li, Bei Li, Tong Xiao, and Jingbo Zhu. 2020. The NiuTrans machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 338–345, Online. Association for Computational Linguistics.

A Appendix

Tags	Predicate endings
<polite>	‘です’,
	‘ます’,
	‘でした’,
	‘ました’,
	‘まして’,
	‘ません’,
	‘ましょう’,
	‘なさい’,
	‘ください’,
	‘くださいませ’
<formal>	‘である’,
	‘であろう’,
	‘であるだろう’,
	‘であった’,
	‘であったろう’,
	‘であっただろう’,
	‘であっている’,
	‘であっていた’,
	‘であれる’,
	‘であらせる’,
	‘であられる’,
	‘であらない’,
	‘であらないだろう’,
	‘であらなかった’,
	‘であらなかっただろう’,
	‘であれない’,
	‘であらせない’,
	‘であられない’,

Table 8: Tagging Rules