Compact and Robust Models for Japanese-English Character-level Machine Translation

Jinan Dai
The University of Tokyo
Meguro-ku 153-8902, Tokyo, Japan
daijinan@graco.c.u-tokyo.ac.jp

Kazunori Yamaguchi
The University of Tokyo
Meguro-ku 153-8902, Tokyo, Japan
yamaguch@graco.c.u-tokyo.ac.jp

Abstract

Character-level translation has been proved to be able to achieve preferable translation quality without explicit segmentation, but training a character-level model needs a lot of hardware resources. In this paper, we introduced two character-level translation models which are mid-gated model and multi-attention model for Japanese-English translation. We showed that the mid-gated model achieved the better performance with respect to BLEU scores. We also showed that a relatively narrow beam of width 4 or 5 was sufficient for the mid-gated model. As for unknown words, we showed that the mid-gated model could somehow translate the one containing Katakana by coining out a close word. We also showed that the model managed to produce tolerable results for heavily noised sentences, even though the model was trained with the dataset without noise.

1 Introduction

In recent years, neural machine translation (NMT) has made a great progress, and its translation quality has far surpassed the conventional statistical machine translation (SMT). At first, NMT had almost relied on word-level modelling with explicit segmentation ,which brought a lot of problems such as big vocabulary (Chung et al., 2016) and frequently appeared unknown tokens. Sennrich et al. (2016) provided a subword segmentation method based on byte-pair encoding (BPE) as a solution. Character-level translation is another approach to deal with the big vocabulary and unknown words. Chung et al. (2016), Lee at al. (2017) and Cherry et al. (2018) have proved that character-level can achieve preferable translation quality without any explicit word segmentation. Though for alphabetical languages, a sentence is much longer when represented in character-level (Lee et al., 2017), Japanese can suffer less from this problem because of the existence of Kanji. However, the sequence is still relatively long, so training in character-level can still take a lot of time. The objective of this paper is to shorten the training time and reduce the storage requirement in Japanese-English translation.

In this paper, in order to increase the convergence speed, we propose two different character-level models which are a mid-gated model and a model with multi-attention, and we will examine their performances in Japanese-English translation.

Our contributions include:

- We show that mid-gated is more efficient than multi-attention in this problem.
- We show that while memory overhead is greater than subword-level translation with respect of sentence pairs used for training, the training speed can be fast in character-level Japanese-English translation.
- We show that a close transliteration can be found for unknown words in Katakana.
- We show that character-level translation can handle heavy noises with moderate performance degradation.

2 Related Work

Cherry el al. (2018) compared character-level translation methods for alphabetical languages. They studied the effect of the model capacity, the corpus size, and the compression by BPE and Multiscale architecture (Chung et al., 2017).

Following this research we tried Hierarchical Multi-scale Long Short-Term Memory (HMLSTM) (Chung et al., 2017) for character-level

Proceedings of the 6th Workshop on Asian Translation, pages 36–44
Hong Kong, China, November 4, 2019. ©2019 Association for Computational Linguistics

Japanese-English translation, but in our experiment environment[1], we did not get a preferable result. So, we omit it in our experiment for our objective is to get a compact model.

We found that HMLSTM includes $\mathrm{relu}\left(\mathbf{W}\left[\mathbf{h}_t^1; \mathbf{h}_t^2, \cdots, \mathbf{h}_t^l\right]\right)$, which is a "shortcut connection" in (He et al., 2016). Even though HMLSTM is too large for a compact model, the shortcut connection may be incorporated. So, we tested a model with the shortcut connection. The model with the shortcut connection is called a mid-gated model following the terminology of (Chung et al., 2017) in this paper.

As to BPE and its variantes, the following researches are relevent.

Chung et al. (2016) proposed a character-level decoder called Bi-Scale decoder while in their research, the encoder side uses BPE. They proved that neural machine translation can be done directly on a sequence of characters without any explicit word segmentation.

Zhang and Komachi (2018) proposed a sub-character level translation for Japanese and Chinese in which Kanji in Japanese and characters in Chinese are decomposed into ideographs or strokes. However, this approach will increase sequence length a lot and need an extra dictionary to decompose Kanji and Chinese characters into strokes or ideographs,

Costa-jussà and Fonollosa (2016) used convolution layers followed by multiple high-way layers to generate character-based word embedding. Other than embedding layer of the encoder side, both the encoder and the decoder are in the word level.

We think that the methods of these researches may complicate the model and are not suitable to our objective.

In Cherry el al. (2018), the multi-headed attention was not used. But because a simple multi-headed attention may cause a mild overhead, we tried a model with multi-attention in our experiment.

3 Proposed Model

We propose two different models for the character-level translation. These model use six bidirectional LSTMs for encoder and six LSTMs for decoder. We use the multiplicative attention mechanism proposed by Luong et al. (2015) instead of additive attention proposed by Bahdanau et al. (2015) because we found out that it will greatly reduce memory consumption during training.

3.1 Basic Model

The basic model is a simple multi-layer attentional encoder-decoder (Cho et al., 2014; Bahdanau et al., 2015) model. Figure 1 shows the structure of the model. For decoder, only the first-layer LSTM takes context vectors as one of its input. The context vector and the hidden state of the last layer in the decoder are used to predict the next character.

3.2 Mid-Gated Model

We adopt a shortcut in the recurrent network by Chung et al. (2017) and Ákos Kádár (2018) which is originally for three HMLSTM layers. We call the model with the shortcut a mid-gated model.

The mid-gated model is similar to the basic model except that the input of 4th layer $\mathbf{m}_t$ of both encoder and decoder is calculated by

$$\mathbf{m}_t = \mathrm{relu}\left(\mathbf{W}_m\left[\mathbf{h}_t^1; \mathbf{h}_t^2; \mathbf{h}_t^3\right]\right) \quad (1)$$

where $\mathbf{W}_m \in \mathbb{R}^{\dim(\mathbf{m}_t)\times\sum_{l=1}^{3}\dim(\mathbf{h}_t^l)}$ is a matrix to map the concatenation of three vectors into one vector, and for encoder $\mathbf{h}_t^l$ is the concatenated output of both direction of lth layer, i.e. $\mathbf{h}_t^l = [\overleftarrow{\mathbf{h}_t^l}; \overrightarrow{\mathbf{h}_t^l}]$, and for decoder, the output of lth layer. Equation 1 can be considered as a shortcut from the first three layers to the 4th layer.

We tried changing the size and location of the shortcut, and we also tried adding another shortcut on the last layer, but we did not get further improvement in these attempts.

3.3 Multi-Attention Model

Usually, word-level and subword-level translation use only one attention layer. But for character-level translation, because of the fine temporal granularity, multi-attention may work well. Thus we tried a multi-attention model as shown in Figure 2.

The encoder side of multi-attention model is the same as the basic model. The decoder side contains six recurrent layers. We use four attention layers for the trade-off between performance and overheads. We put attention layers on the 1st and 6th recurrent layers to ensure the first recurrent

[1] 2 NVIDIA 2080s. Cherry el al. (2018) used 16 NVIDIA P100s.

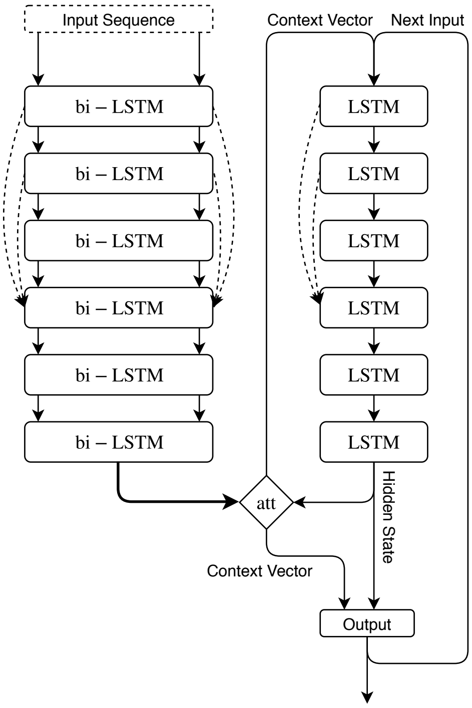


Figure 1: The basic model (without dashed connections) and mid-gated model (with the dashed connections).

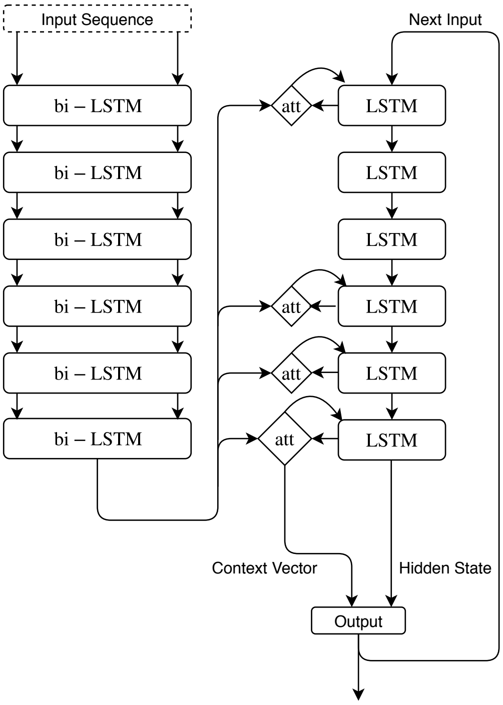


Figure 2: The multi-attention model

layer taking context as input and the sixth recurrent layer outputting context, and we found out that it is optimal to put other two attention layers on the 4th and 5th recurrent layers in our preliminary experiments. We tried the combination of the multi-attention model and mid-gated model, but we did not find any improvement in the combination.

	ASPEC-JE	NTCIR-JE
Pairs (train)	1,000,000	1,387,713
Pairs (dev)	1790	2000
Pairs (devtest)	1784	-
Pairs (test)	1812	2300
Vocab (ja)	3084	2966
Vocab (en)	291	98

Table 1: Numbers of sentence pairs and vocabulary of ASPEC-JE and NTCIR-JE.

4 Experiments Design

4.1 Datasets and Preprocessing

We used ASPEC (Nakazawa et al., 2016) and NTCIR (Goto et al., 2013) as out datasets. The ASPEC dataset contains three training sets `train-1.txt`, `train-2.txt` and `train-3.txt`. We only used the first training set because of our limited hardware resources.

Table 1 shows the sizes of the training set of both datasets. Note that the vocabulary in this paper refers to the number of different characters in the training sets.

For ASPEC dataset, we appended a space at both the beginning and end of each sentence of both languages. Note that this will not influence the final result. We did not perform any other preprocessing. We did not eliminate long sentences. We kept all numbers, characters, punctuations in Japanese side of the datasets as is. We used OpenNMT-tf's built-in character tokenizer for tokenization.

4.2 Training

The model were trained using sentence-level cross entropy loss. Batch sizes were capped at 12,800 tokens, and each batch was divided between two GPUs running synchronously. The dimension of character embedding of Japanese was 512 and for English, 128. All other vector dimensions were 512. The basic and mid-gated models were trained using two NVIDIA's GeForce 1080Ti's, while the multi-attention model was trained using two NVIDIA's RTX 2080Ti's.

We initialized parameters randomly with a uniform (-0.1,0.1) distribution. We used Adam's Optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon =$

Models	ASPEC-JE	NTCIR-JE
Basic	26.89 (22.47)	40.82 (36.76)
Mid-Gated	**27.63 (22.98)**	**41.32 (37.34)**
Multi-Attention	27.06 (22.35)	41.11 (37.08)
Yamagishi et al. (2017)	(18.78)	(29.80)
Morishita et al. (2017)	27.62	-
NMT with Attention (Cho et al., 2014; Bahdanau et al., 2015)	26.91	-
Transformer (Vaswani et al., 2017)	28.06	-

Table 2: BLEU scores for various models. Scores calculated by Travatar are shown in parentheses. Scores with references are from the literatures.

10^{-8} (Kingma and Ba, 2015). Gradient norm was clipped to 5.0 (Pascanu et al., 2012). The dropout rate was set to 0.2 for all models. Dropouts were taken place in all bidirectional LSTMs and LSTMs. The initial learning rate was 0.0002, and it decayed with rate 0.9 for every 10k batches after 20k batches. Training stopped when dev set perplexity had not decreased for 6k batches. We implemented the mid-gated and multi-attention models on OpenNMT-tf (1.20.0) for training. The inference was done on version 1.24.0.

Except where mentioned below, the inference used beam search with 4 hypotheses, and the strictness of length normalization was set to 0.2 (Wu et al., 2016).

5 Results

5.1 BLEU Scores

We report our BLEU scores for the three models in Table 2. For ASPEC, we preprocessed the inference result by removing spaces at the beginning and end of translated sentences. For NTCIR, we kept the inference result as is. We used Moses-tokenized case-sensitive BLEU[2] score as our evaluation metric. We report the test-set scores on the checkpoints having lowest perplexity on the dev set. As we can be seen in the table, the mid-gated model produces the best result among the three models. The parenthesized scores are calculated by bootstrap resampling implemented in Travatar[3].

	Time	GPU1	GPU2
Basic	43h	4GB	4GB
Mid-Gated	40h	8GB	4GB
Multi-Attention	37h	8GB	4GB

Table 3: The actual training time and GPU overhead of each model. Note that Tensorflow tend to occupy more memory than needed.

The organizer's results of WAT 2018 [4] in vanilla encoder-decoder with attention model (Cho et al., 2014; Bahdanau et al., 2015) and Transformer (Vaswani et al., 2017) are also shown.

We also include the best scores in a single model reported by Yamagishi el al. (2017) and Morishita et al. (2017).

The BLEU scores of our models are similar to the subword-level model of Morishita et al. (2017). However our training is much simpler. It takes 10.1 million sentence pairs to train the basic model, and 8.0 million pairs for mid-gated model, and 6.8 million pairs for multi attention model, while Morishita et al. (2017) used 60 million pairs for training in the experiment with the best result in a single model. The actual memory overheads and training time are shown in Table 3. Morishita et al. (2017) used batch of 128 sentence pairs. But in our experiments, setting batch size of each GPU to more than 40 sentence pairs without limiting the sentence length during training caused out-of-memory error. Thus we consider character-level translations uses more memory than subword-level translation while the training speed can be fast with respect to sentence pairs. The BLEU scores of our model are slightly inferior to that of Transformer, but our model has less parameters and is trained easily.

5.2 Translation Examples

We choose two examples from the test set to show the difference of the three models in translation. As shown in Table 4, the translation is the same

[2] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation2/automatic_evaluation_systems/automaticEvaluationEN.html

[3] http://www.phontron.com/travatar/evaluation.html

[4] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/list.php?t=2&o=4

for the simple first sentence, but in the second example, the mid-gated model is superior on fluency and accuracy. As for the word "演えきシステム", which means "deduction system", none of the models translates exactly the same as the reference, while the results by the basic and mid-gated model are only different in articles and suffices.

We also want to check how multi-attention works. As shown in Figure 3, the first two attention layers barely catch the right alignment. The third attention layer got some alignments in the middle of the sentence. In the forth attention layer, when the length of English word is longer than the corresponding Japanese word, the model tend to align the first N characters to the corresponding Japanese characters, where N is the length of the Japanese word, and the remaining characters to the beginning of the sentence.

5.3 Noise

Algorithm 1 AddNoise$_{dropRate,insertRate}$

```
for $sentence$ in $testset$ do
    for $char$ in $sentence$ do
        $drop \leftarrow$ true for probability of $dropRate$
        $insert \leftarrow$ true for probability of $insertRate$
        if $drop =$ true and $insert =$ true then
            Replace $char$ with a random character
        else if $drop =$ true and $insert =$ false then
            Drop $char$ in $sentence$
        else if $drop =$ false and $insert =$ true then
            Add a randomly chosen character before $char$
        end if
    end for
end for
```

We tested whether the models can handle noise. We added noise to the ASPEC's test set by randomly dropping and inserting characters to the Japanese side. The inserted characters are chosen randomly from the vocabulary. The insert and drop rate ranges over $5\%, 4\%, 3\%, 2\%, 1\%, 0.1\%, 0.01\%, 0.001\%,$ and 0%. Algorithm 1 shows this noising procedure. For each insert and drop rate pair, we built three test set for each drop-insert rate pair and averaged the BLEU scores.

The result is shown in Figure 4. We notice that even with a heavy noise with drop rate of 5% and insert rate of 5%, the three models still managed to yield a tolerable result. Also, we can conclude that dropping characters can cause more decrease in BLEU scores compared to inserting. We speculate that although both inserting and dropping will interfere the inference, the information loss caused by dropping has more impact. Table 5 shows a noise-added example and its translations.

5.4 Beam Width and Length Normalization

As suggested by Morishita et al. (2017) and Wu et al. (2016), we use length normalization with strictness of 0.2. Figure 5 shows how BLEU score changes when increasing beam width.

We can find out that the BLEU scores decrease drastically as beam width increases after 4 or 5 if length normalization is not adopted. While with length normalization, the BLEU scores only decrease by less than 0.7, this is different from BPE translation shown by Morishita et al. (2017) where the scores stay increasing even after beam width of 25.

In character-level translation, we observed that all three models tended to produce a few empty sentences, but with layer normalization with strictness of 0.2, this tendency is suppressed.

Note that the largest beam width is 221 because we employ the character-level translation. We do not try stricter length normalization since in our preliminary experiments, more strictness would decrease the performance with a large beam width.

5.5 Unknown Words

Like BPE, character-level translation is also hoped for predicting candidates for unknown words. In this paper, we define unknown words as follow:

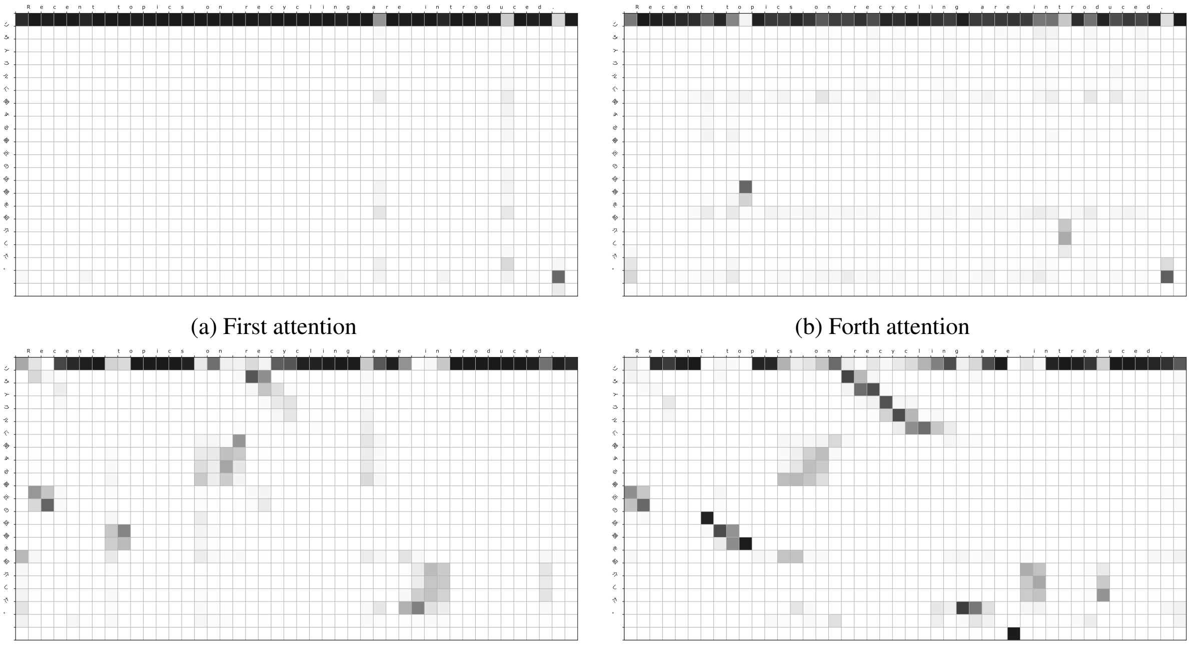
Definition A string is an unknown word if and only if

1. it is a token of a tokenized sentence outside the training dataset, and,
2. it is not substring of any sentence in the training dataset.

For example, the Japanese word "データベース" which means "database" in English, is a token of the second sentence in Table 4 after tokenization. In the training set, the sentence is not included, but there exists some other sentence, one of whose substring is the token, thus this is not unknown

Src	リサイクルに関する話題を紹介した
Ref	Recent topics on recycling are introduced.
Basic	Recent topics on recycling are introduced.
Mid-Gated	Recent topics on recycling are introduced.
Multi-Attention	Recent topics on recycling are introduced.
Src	超伝導材料開発のためのデータベースを構築し,材料設計用演えきシステムの開発を行った。
Ref	A database for development of superconducting material was constructed, and deduction system for material design was developed.
Basic	The database for the development of superconducting material was constructed, and deductive system for material design was developed.
Mid-Gated	A database for the development of superconducting materials was constructed and deduced system for material design was developed.
Multi-Attention	The database for the superconducting material development was constructed, and the development system for material design was developed.

Table 4: Translation example by three models.



(a) First attention

(b) Forth attention

(c) Fifth attention

(d) Sixth attention

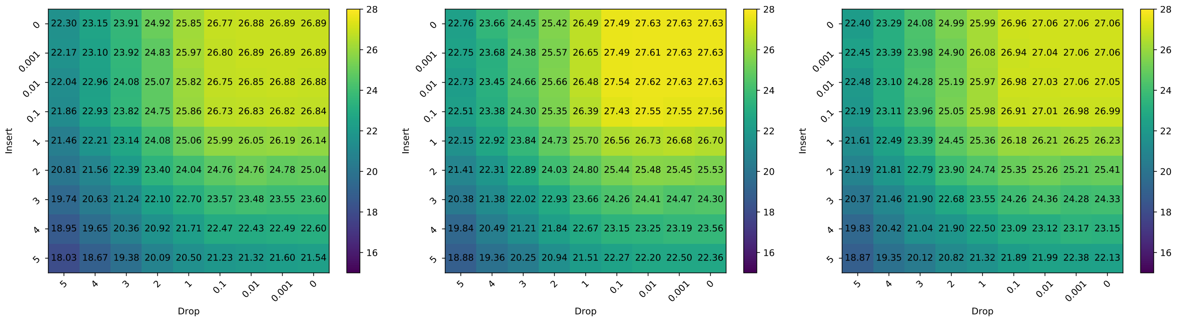
Figure 3: Attentions of the multi-attention model for the first sample in Table 4.

word. Further, the string "超電導材料開発のためのデータベースを構築し" is not a substring of any sentence in the training set, but it is not a token of the tokenized sentence, so this is not unknown word either.

Further, we categorize unknown words into three types:

1. Words only containing Katakana, which is usually transliteration of other language.
2. Words only containing Hiragana and Kanji that is in the character vocabulary.
3. Words containing unseen Kanji.

In order to find sentences with unknown words, we first tokenized the Japanese source sentences in dev, devtest, an test sets using MeCab and constructed vocabulary. For each word in vocabulary, we check if it is a substring of any sentence in the train set. Finally, we eliminated all words with



(a) Basic Model

(b) Mid-Gated Model

(c) Multi-Attention Model

Figure 4: The translation scores for different drop-insert probability pairs.

Src	材料製造プロセスでは,物質の融解・凝固・急冷などの熱的現象を,精密に制御することが必要である。
Ref	For material production, it is necessary to precisely control thermal phenomena such as fusion, solidification, and rapid cooling of a substance.
Noised	材料製造プロセスで談は,物質の融解・凝べ固・急ρ冷卒な坂どの熱的現象を,精密に制御すること必要である。
Basic	In the material manufacturing process, it is necessary to precisely control the thermal phenomenon of melting, compatible solution, and proper ρ compound sake of materials.
Mid-Gated	It is necessary to precisely control the thermal phenomenon of melting, solidification, and rapid ρ collapse of materials in the material manufacturing process.
Multi-Attention	In the material manufacturing process, it is necessary to precisely control the thermal phenomenon of melting and flocculation of the material, and thermal phenomenon of rapid ρ cooling sake.

Table 5: A translation result of the noised sentence. The boxed characters in the Src sentence is the one dropped out, and the boxed characters in the Modified sentence is the one inserted. The words translated wrongly are underlined.

Type	**Src**	**Ref**	**Basic**	**Mid-Gated**	**Multi-Attention**
1	アンタゴニスティック	antagonistic	antagonistic	antagonistic	antagonistic
	エコマティAX	Ecomatie AX	ecomater AX	Ecomatey AX	ecomate AX
2	福島大学	Fukushima University	Fukushima University	Fukushima University	Fukushima University
	長谷山俊郎	Toshio Hasegawa	Nagayama Yamato	**IGNORED**	Nakayama
	友ケ島	Tomoga Island	Fiken Island	**IGNORED**	**IGNORED**
3	嘔気・嘔吐	nausea and vomiting	air and vomiting	air and vomiting	air and vomiting
	捏造	fabrication	structure	**IGNORED**	construction

Table 6: Examples of translations of unknown words. The cases of each words are kept as is. Unknown Kanji, i.e., Kanji that do not exist in the vocabulary, are underlined. The translations of words "嘔気" and "嘔吐" were not contained in the reference, so we show the dictionary meaning of these words in Ref.

only alphabets and numbers since it is trivial to translate these "words".

For the first type of unknown words, all models can easily predict the translation and the mid-gated

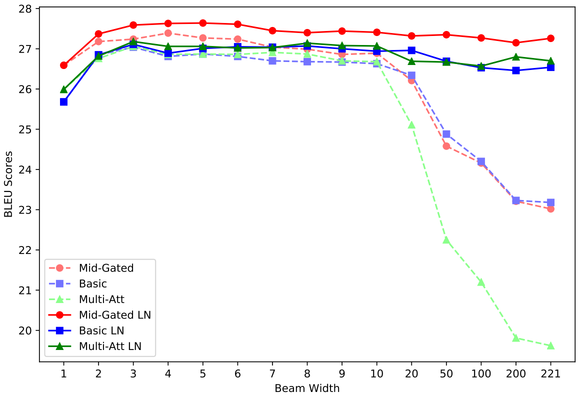


Figure 5: BLEU scores for different beam widths. Here "LN" stands for length normalization.

model can predict translation better for it can also identify proper nouns such that the first character is in the upper case.

For the second type, all models can somehow predict the translation, while as for people's name and hard-to-read name of places, the mid-gated model tends to ignore them while the other two models are trying to translate in their own way.

For the third type of unknown words, the models tend to predict the translation using only known characters. Table 6 gives some examples. Due to limited space, we only give the unknown words and their translations in the reference set and translation results.

The fact that the mid-gated model tends to ignore the second and third types of unknown words does not contradict to the result in Table 2, since even though other models translate the second and third type in their own way, the result is not exactly the correct answer and it is ignored in the BLEU scores. The number of the first type of unknown words in dev, devtest and test sets are twice as many as the sum of numbers of other two types of unknown words, and for the first type of unknown words, mid-gated model tends to predict them better, as shown in Table 6.

6 Conclusion

The objective of this paper is to get a computationally and spatially cheaper character-level translation model while keeping performance in BLEU scores. We proposed three models and showed that one of the models, the mid-gated model, was much better in speed and space consumption than the previous models with similar BLEU scores. We also showed that a relatively narrow beam of width 4 or 5 was sufficient for the mid-gated model.

In character-level translation, no word is made unknown because the vocabulary, which is a set of characters in character-level translation, is small and there is no need to limit vocabulary. Still occurring unknown word in character-level translation is unseen transliteration, an unseen word containing Hiragana and Kanji, or a word with unseen Kanji. Such an unknown word is difficult to translate, but we showed that, as to an unseen transliteration, the mid-gated model could somehow translate it by coining out a close word.

We also showed that the model managed to produce tolerable results for heavily noised sentences. Remarkable here is that the model was trained with the dataset without noise.

For future work, we want to explore a way to correctly translate an unknown word containing Hiragana and Kanji and a word with unseen Kanji. We want to handle typos including conversion error and swapping as well as comparing their performance against word-level and subword-level translations. We also want to investigate the mid-gated model's ability in translating alphabetical languages.

7 Acknowledgment

We also appreciate the reviewers for their insightful comments. This work was supported by JSPS KAKENHI Grant Number 17H01837.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In (Bengio and LeCun, 2015).

Yoshua Bengio and Yann LeCun, editors. 2015. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4295–4305, Brussels, Belgium. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. 2017. Hierarchical multiscale recurrent neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany. Association for Computational Linguistics.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany. Association for Computational Linguistics.

Isao Goto, Ka Po Chow, Bin Lu, Eiichiro Sumita, and Benjamin Tsou. 2013. Overview of the patent machine translation task at the ntcir-10 workshop.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Ákos Kádár, Marc-Alexandre Côté, Grzegorz Chrupala, and Afra Alishahi. 2018. Revisiting the hierarchical multiscale LSTM. *CoRR*, abs/1807.03595.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In (Bengio and LeCun, 2015).

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

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.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2017. NTT neural machine translation systems at WAT 2017. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 89–94, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. Aspec: Asian scientific paper excerpt corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2204–2208, Portorož, Slovenia. European Language Resources Association (ELRA).

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. On the difficulty of training recurrent neural networks. In *ICML*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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.

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

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2017. Improving Japanese-to-English neural machine translation by voice prediction. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 277–282, Taipei, Taiwan. Asian Federation of Natural Language Processing.

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, Belgium, Brussels. Association for Computational Linguistics.