Fast and Accurate Transformer-based Translation with Character-Level Encoding and Subword-Level Decoding

Anonymous ACL submission

Abstract

The Transformer translation model is fast to train and achieves state-of-the-art results for various translation tasks. However, unknown input words at test time remain a challenge for the Transformer, especially when unknown words are segmented into inappropriate subword sequences that break morpheme boundaries. This paper improves the Transformer model to learn more accurate source representations via character-level encoding. We simply adopt character sequences instead of subword sequences as input of the standard Transformer encoder and propose contextualized character embedding (CCEmb) to help character-level encoding. Our CCEmb contains information about the current character and its context by adding the embeddings of its contextual character n-grams. The CCEmb causes little extra computational cost and we show that our model with a character-level encoder and a standard subword-level Transformer decoder can outperform the original pure subword-level Transformer, especially for translating source sentences that contain unknown (or rare) words.

1 Introduction

Neural machine translation (NMT) (Bahdanau et al., 2014; Vaswani et al., 2017) is capable of open-vocabulary translation via automatic subword segmentation (Sennrich et al., 2016; Wu et al., 2016; Xu et al., 2021). However, automatic subword segmentation algorithms frequently produce inappropriate subword segmentation that breaks morpheme boundaries and affects the performance of subword-level NMT, especially for translating unknown (or rare) input words (Ataman and Federico, 2018). For example, in Table 1, assume the word “stumbled" is an unknown word (it never occurred in the training data). Even if the subword-level translation model has learned how to translate a similar word “stumble", the model still does not know how to translate “stumbled" as the automatically segmented subword sequences of these two words do not share any subword tokens as shown in Table 1. Compared to subword-level NMT, character-level NMT (Lee et al., 2017; Cherry et al., 2018; Gao et al., 2020), which trains translation models with character sequences, naturally does not suffer from inappropriate subword segmentation and has the potential to learn more accurate word representations. Cherry et al. (2018) showed that RNN-based character-level NMT models can outperform identical models that are trained with subword-level sequences.

Word	Subword	Character
stumble	stum@@ ble	s t u m b l e
stumbled	st@@ umb@@ led	s t u m b l e d

Table 1: Subword sequences and character sequences for the two words “stumble" and “stumbled". The two subword sequences do not share any subword tokens while the two character sequences share the same subsequence “s t u m b l e".

Although RNN-based character-level NMT models (Cherry et al., 2018; Chung et al., 2016; Lee et al., 2017; Ataman et al., 2019; Luong and Manning, 2016; Costa-jussà and Fonollosa, 2016) have shown promising results, the computational cost of training such a model is high as long RNNs are slow to train (Cherry et al., 2018). For efficiency, Gao et al. (2020) trained Transformer-based character-level NMT models with self-attention, but found that using a standard Transformer model to learn character-level translation achieved worse translation quality than subword-level Transformer models. Gao et al. (2020) improved the Transformer model to perform more accurate character-level translation by adding extra convolutional layers into the Transformer encoder, but their model (character-level ConvTransformer) still obtained worse translation quality compared to the original subword-level Transformer translation model.

In this paper, we improve the Transformer model to perform more effective character-level encoding with contextualized character embedding (CCEmb). Our CCEmb captures information about the current character and its context by adding the embeddings of its contextual character n-grams. We show that,

1. our CCEmb effectively improves the Transformer model for character-level encoding and requires significantly less computational cost compared to Gao et al. (2020)'s method which used extra convolutional layers.

2. while previous Transformer-based models focused on pure subword-level translation (Vaswani et al., 2017) or pure character-level translation (Gao et al., 2020), our model combining a character-level encoder with a standard subword-level Transformer decoder can achieve higher translation quality.

3. our model with a character-level encoder is particularly useful for translating infrequent words compared to the pure subword-level Transformer.

2 Our Approach

The Transformer translation model (Vaswani et al., 2017) can be directly trained with character sequences instead of subword sequences to perform character-level translation. We improve the Transformer encoder to perform more effective character-level encoding by replacing the standard character embedding with contextualized character embedding (CCEmb).

In contrast to standard character embedding which only contains information about a single character, our CCEmb captures information about the current character and its context by adding the embeddings of its contextual character n-grams. Given a sequence[1] of L characters $x_1, ..., x_i, ..., x_L$ as the input of the Transformer encoder, we compute the CCEmb for the ith character as,

$$C_i = concat\left(\sum_{n=1}^{5} E(x_{i-n+1}^{i}), \sum_{n=1}^{5} E(x_{i}^{i+n-1})\right) \quad (1)$$

where $E(x_{i-n+1}^{i})$ is the embedding of the n-gram x_{i-n+1}^{i} which represents the left-side context of x_i; $E(x_{i}^{i+n-1})$ is the embedding of the n-gram x_{i}^{i+n-1} which represents the right-side context of x_i. We learn an embedding vector for each character n-gram in the same way as learning an embedding vector for a character or a subword token. The longest n-grams used in our experiments are 5-grams. We only use n-grams that do not cross word boundaries as shown in Figure 1. And we only use embeddings of the most frequent 32k n-grams[2] contained in the source-side of the parallel NMT training data. Because the CCEmb is obtained by concatenating left-side contextual n-gram embedding and right-side contextual n-gram embedding, the size of n-gram embedding is set to be half of the Transformer model size.

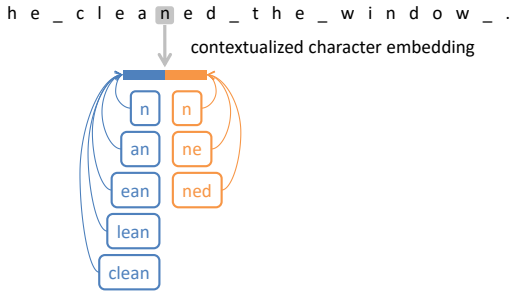

Figure 1: An example of CCEmb.

Unlike Vaswani et al. (2017)'s subword-level Transformer encoder, our character-level encoder does not suffer from inappropriate subword segmentation and can learn more accurate source representations, especially for unknown/rare words as showcased in Table 1. Compared to the subword-level Transformer encoder, the subword-level Transformer decoder is less influenced by inappropriate subword segmentation, because at test time, we only need to apply automatic subword segmentation algorithms to segment the source words, not the target words (i.e., at test time, the target-side subword tokens are generated by the translation model, not by automatic subword segmentation algorithms). In our experiments, a subword-level Transformer decoder obtained significantly higher translation quality compared to a character-level Transformer decoder and therefore we train our model to perform character-to-subword translation with our character-level encoder and a standard subword-level Transformer decoder.

[1] In character sequences, we use a special space token to indicate word boundaries.

[2] To obtain the most frequent 32k character n-grams, we apply moses scripts *tokenizer.perl* and *truecase.perl* to the source-side of the parallel training data, then compute the frequency of character n-grams that occur in the source sentences and are not longer than 5-grams, and finally use the most frequent 32k n-grams.

		DeEn		FiEn			
		BLEU	chrF	BLEU	chrF	Parameters	Speed
B2B (Vaswani et al., 2017)	Transformer	31.78	57.29	20.90	49.51	97M	1248
C2C (Gao et al., 2020)	ConvTransformer	31.01	56.85	20.63	49.42	82M	139
C2B (Ours)	Transformer	31.44	57.12	20.13	48.80	79M	312
	+CCEmb	32.20†	57.66†	20.93	50.03†	87M	292
	ConvTransformer	32.20†	57.65†	21.86†	50.57†	117M	201
	+CCEmb	32.51†	57.94†	21.92†	50.83†	125M	191

Table 2: Translation results. B2B: BPE-to-BPE; C2C: Character-to-Character; C2B: Character-to-BPE. Speed: numbers of sentence pairs being processed per second during training. † represents significantly better (Koehn, 2004) at the $p < 0.01$ level compared to the B2B Transformer.

3 Experiments

We conducted experiments on German-to-English (DeEn) and Finnish-to-English (FiEn) translation tasks. We used training data of WMT 2015 NEWS translation task for both language pairs (4.5M and 2.1M sentence pairs for DeEn and FiEn, respectively). For the DeEn task, we combined WMT NEWS test sets, newstest2010 to newstest2020, as test data (28K sentence pairs); for the FiEn task, we combined newstest2015 to newstest2019 as test data (12K sentence pairs).

We train Transformer models with CCEmb to perform character-to-subword translation for each language pair. We compare our model with the original Transformer model (Vaswani et al., 2017) and the ConvTransformer model (Gao et al., 2020) which employs extra convolutional layers in the Transformer encoder for character-to-character translation. As the Transformer model (Vaswani et al., 2017) was proposed to learn subword-to-subword translation and the ConvTransformer model (Gao et al., 2020) was proposed to learn character-to-character translation, we also train Transformer and ConvTransformer models to learn character-to-subword translation and investigate the effectiveness of Transformer and ConvTransformer for character-to-subword translation.

We applied moses scripts *tokenizer.perl* and *truecase.perl* as preprocessing for training all models. For subword segmentation, we used *byte pair encoding* (BPE) (Sennrich et al., 2016) to learn a joint source and target vocabulary of 32k for each language pair. We used the base model setting of Vaswani et al. (2017)'s work for all models in our experiments. During training, we set the max length of character sequences to be 500 and the max length of subword sequences to be 100.

Translation results, BLEU[3] and chrF (Popović, 2015), are given in Table 2. Table 2 shows that (i) the original C2C ConvTransformer model obtained worse translation quality compared to the original B2B Transformer model (ii) our C2B Transformer model with CCEmb can achieve higher translation quality compared to the B2B Transformer (iii) combining the ConvTransformer character-level encoder and a subword-level decoder can outperform the original C2C ConvTransformer, and our CCEmb can further improve the ConvTransformer character-level encoder. Table 2 also shows that our CCEmb caused little increase in computational cost while ConvTransformer added extra convolutional layers into the Transformer encoder and led to significantly more computational cost and parameters.

Character-level Encoding for Infrequent words Compared to subword-level encoding, character-level encoding can obtain better translation for unknown/rare input words that are inappropriately segmented. Table 3 gives an example: the input German word "Baufehler" is segmented into "B@@ auf@@ eh@@ ler" by BPE which clearly broke the morpheme boundaries as a semantically meaningful subword segmentation should be "Bau@@ (construction) fehler (defect)". If the word "Baufehler" occurred frequently in the parallel training data, the BPE2BPE Transformer would have learned how to translate this word even though it is segmented into semantically meaningless subword tokens. However, "Baufehler" is a rare word (only occurred twice in the training data) and therefore the BPE2BPE Transformer failed to translate it correctly. To quantify the advantage of our character-level encoding over subword-level encoding for translating infrequent words, we use the frequency of a source word occurring in the source-side of the parallel training data and then divide the DeEn test data into two parts $T_{frequent}$ and $T_{infrequent}$ by ranking all source test sentences

[3]BLEU scores are case-sensitive and computed by moses script *multi-bleu-detok.perl*.

SRC	Ermittler entdecken gefährlichen **Baufehler** in A380-Triebwerken
REF	Investigators uncover dangerous **defect** in A380 engines
BPE	Ermitt@@ ler entdecken gefährlichen **B@@ auf@@ eh@@ ler** in A3@@ 8@@ 0-@@ Trieb@@ werken
B2B	Investigators discover dangerous A380 engines
C2B	Detectors Discover Dangerous **Failures** in A380 Engines

Table 3: Translation examples. SRC: source; REF: reference; BPE: subword-level input sequence segmented by BPE; B2B: translation produced by the BPE2BPE Transformer; C2B: translation produced by the character2BPE Transformer with CCEmb.

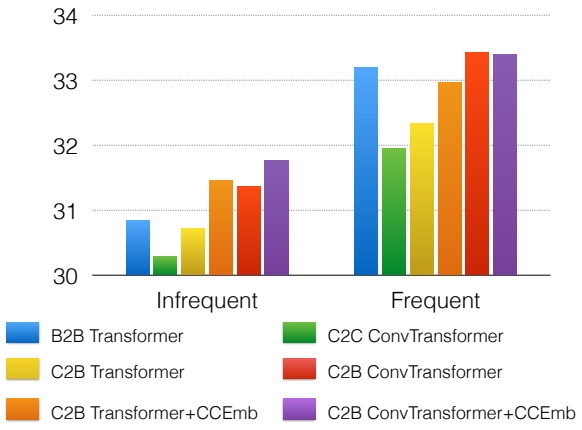

Figure 2: Translation results (BLEU) for $T_{infrequent}$ and $T_{frequent}$.

according to the frequency of the least frequent word contained in the sentence. As shown in Figure 2, compared to subword-level encoding, character-level encoding is generally more beneficial for translating $T_{infrequent}$ than for translating $T_{frequent}$. Figure 2 also shows that our CCEmb effectively improved both Transformer and ConvTransformer for translating $T_{infrequent}$.

Character-level vs. Subword-level Decoding Table 2 shows that using a subword-level Transformer decoder obtained higher translation quality than a character-level Transformer decoder in our experiments. There are two main reasons: 1. character sequences are much longer than subword sequences and a character token contains significantly less information than a subword token, which increase the difficulty of character-level decoding compared to subword-level decoding; 2. the decoding process is less influenced by inappropriate automatic subword segmentation compared to the encoding process, because at test time, only source words need to be segmented by BPE and target-side subword tokens are generated by the translation model (not segmented by BPE). Although subword-level decoding achieved higher translation quality than character-level decoding in our experiments, for future research, character-level decoding has the potential to outperform subword-level decoding as target words in the parallel NMT training data can still be inappropriately segmented by automatic subword segmentation algorithms and affect the training process of subword-level decoders.

4 Related Work

For improving character-level NMT, Libovický and Fraser (2020) showed that, initially training a subword-level translation model and then finetuning it on characters can achieve higher translation quality compared to training character-level translation models from random initialization, but their method still obtained worse overall translation quality compared to subword-level NMT models.

Other than character-level NMT, there are a number of methods (Kudo, 2018; Xiao et al., 2019; Provilkov et al., 2020) that were proposed to address the inappropriate subword segmentation problem of subword-level NMT by exploiting multiple possible subword segmentation candidates in subword-level NMT systems. However, a source/target word can have a large number of possible subword segmentation candidates, which leads to high computational cost for their methods to make use of all possible subword segmentation. Therefore, for efficiency, Kudo (2018); Xiao et al. (2019)'s methods can only use n-best subword segmentation candidates at NMT training/inference time; Provilkov et al. (2020) only used multiple subword segmentation at training time, not inference time.

5 Conclusion

This paper improves Transformer translation models to perform more effective character-level encoding with CCEmb. Our CCEmb captures not only information about the current character but also its context information by adding embeddings of its contextual character n-grams. The CCEmb causes little increase in computational cost and we show that our approach with a character-level encoder and a standard subword-level Transformer decoder can outperform previous pure subword-level (and pure character-level) Transformer-based models.

References

Duygu Ataman and Marcello Federico. 2018. An evaluation of two vocabulary reduction methods for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 97–110, Boston, MA. Association for Machine Translation in the Americas.

Duygu Ataman, Orhan Firat, Mattia A. Di Gangi, Marcello Federico, and Alexandra Birch. 2019. On the importance of word boundaries in character-level neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 187–193, Hong Kong. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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*, pages 4295–4305, Brussels, Belgium. Association for Computational Linguistics.

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.

Yingqiang Gao, Nikola I. Nikolov, Yuhuang Hu, and Richard H.R. Hahnloser. 2020. Character-level translation with self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1591–1604, Online. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. 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.

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.

Jindřich Libovický and Alexander Fraser. 2020. Towards reasonably-sized character-level transformer NMT by finetuning subword systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2572–2579, Online. Association for Computational Linguistics.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063, Berlin, Germany. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1706.03762*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Fengshun Xiao, Jiangtong Li, Hai Zhao, Rui Wang, and Kehai Chen. 2019. Lattice-based transformer encoder for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3090–3097, Florence, Italy. Association for Computational Linguistics.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary learning via optimal transport for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7361–7373, Online. Association for Computational Linguistics.