IEICE TRANS. INF. & SYST., VOL.E107–D, NO.4 APRIL 2024

LETTER

Sense-Aware Decoder for Character Based Japanese-Chinese NMT

Zezhong LI[†a)], ***Nonmember and*** **Fuji REN**[††], ***Fellow***

SUMMARY Compared to subword based Neural Machine Translation (NMT), character based NMT eschews linguistic-motivated segmentation which performs directly on the raw character sequence, following a more absolute end-to-end manner. This property is more fascinating for machine translation (MT) between Japanese and Chinese, both of which use consecutive logographic characters without explicit word boundaries. However, there is still one disadvantage which should be addressed, that is, character is a less meaning-bearing unit than the subword, which requires the character models to be capable of sense discrimination. Specifically, there are two types of sense ambiguities existing in the source and target language, separately. With the former, it has been partially solved by the deep encoder and several existing works. But with the later, interestingly, the ambiguity in the target side is rarely discussed. To address this problem, we propose two simple yet effective methods, including a non-parametric pre-clustering for sense induction and a joint model to perform sense discrimination and NMT training simultaneously. Extensive experiments on Japanese↔Chinese MT show that our proposed methods consistently outperform the strong baselines, and verify the effectiveness of using sense-discriminated representation for character based NMT.

key words: *NMT, sense-discriminated, Japanese-Chinese*

1. Introduction

In recent years, Neural Machine Translation (NMT) has achieved enormous success on various translation tasks [1]. A typical NMT system adopts a sequence-to-sequence architecture operating on subword-level. It relies on the language-dependent segmentation algorithm, and may produce suboptimal segmentation, which can hurt the translation performance and also violate the spirit of NMT for learning everything in an end-to-end manner [2]. This deficiency becomes more serious for MT between Japanese and Chinese, both of which use consecutive logographic characters to construct words and sentences without explicit word boundaries. A promising way is to use character granularity for NMT, which could circumvent the segmentation problem completely. There has been a bundle of works comparing the pros and cons between the subword and character based models [2]–[4]. Yet most of these works focus on the alphabetic language rather than the logographic language, suggesting that the comparison results may be inconsistent. For example, a criticism for character models is that the character sequence is much longer than the subword sequence, which leads to the training inefficiency; in contrast, Japanese and Chinese suffer less from this problem due to the existence of Chinese characters. However, there is still one drawback to be addressed, that is, character is a less meaning-bearing unit than the subword, and a character tends to have more distinct senses than a subword containing it. This indicates that the character models should pay more attention to the sense discrimination.

Diving into the sense ambiguity issue in the character NMT, we can find that there are two types of sense ambiguities existing in the source and target language, separately. With the former, we can hypothesize that it has been greatly alleviated in an end-to-end fashion implicitly attributing to the powerful contextualization capability in the deep encoder like Transformer, through which an identical character with multi-sense can have different representations following the changed contexts. There are also efforts on plugging explicit disambiguation module to differentiate the multi-sense before feeding it into the encoder [5]–[7]. Although they are originally used for word sense disambiguation, they can transfer to the character models seamlessly, but with the later, i.e. the ambiguity in the target side, it is rarely addressed. The ambiguity arises when the decoder generates the next character through a softmax operation on the inner product between decoding hidden states and output embeddings. Compared to the deep multi-layers for encoding, the output embedding is a shallow single layer without the ability of sense discrimination. Apparently, the single prototype representation without considerations of multiple senses hurts the performance of character based NMT potentially.

In this paper, we focus on the multi-sense representation in the target language for character-based Japanese-Chinese NMT. Specifically, we propose two simple yet effective methods for multi-sense representation: the first method induces the multiple senses in advance via performing non-parametric clustering on the outputs of BERT [8], and then the target character sequence is transformed into a sense sequence for training and inference; in the second method, we propose a joint model to perform sense discrimination and NMT training simultaneously, i.e. using multi-sense embedding in the output embedding layer rather than single prototype character embedding.

Extensive experiments on Japanese↔Chinese MT show that our proposed methods substantially outperform the strong baselines in terms of BLEU scores, and verify the effectiveness of using sense-discriminated representation for

Manuscript received September 6, 2023.
Manuscript revised November 15, 2023.
Manuscript publicized December 11, 2023.

[†]The author is with the Faculty of Computer Science, Zhejiang University of Water Resources and Electric Power, China.
[††]The author is with the Faculty of Computer Science, University of Electronic Science and Technology of China, China.
a) E-mail: lizezhonglaile@163.com

DOI: 10.1587/transinf.2023EDL8059

character based NMT.

2. Models

In this section, we first introduce the basic concepts, then propose two methods for multiple sense discrimination in character level NMT.

2.1 Neural Machine Translation

A typical NMT model adopts an encoder-decoder architecture. Functionally, the encoder maps the source sentence into continuous representations, based on which the decoder generates the target sentence token by token. Let x be the source sentence, $y_{<t}$ be a prefix of the target sentence y that has been generated before the step t, the probability of generating next token y_t is computed as:

$$\begin{aligned} p(y_t|x, y_{<t}) &= \text{softmax}(h_t^{\mathrm{T}} W_{y_t}) \\ &= \frac{\exp(h_t^{\mathrm{T}} W_{y_t})}{\sum_{y'_t \in V} \exp(h_t^{\mathrm{T}} W_{y'_t})} \end{aligned} \tag{1}$$

where $h_t \in R^d$ is the the decoder's output state at current step, V is the target vocabulary, and $W \in R^{d \times |V|}$ is the output embedding matrix.

During training, the loss function on individual token is the negative log of Eq. (1), that is

$$\mathcal{L} = -\log p(y_t|x, y_{<t}) \tag{2}$$

The overall loss is the sum of all the token losses in the full training data.

2.2 Sense Pre-Clustering

The most intuitive way for sense discrimination in NMT is to modify the training data with predicted senses, so the cross entropy loss for training becomes sense-aware, which is optimized by predicting the sense-specified characters instead of only characters. For example, the Chinese character "属" has distinct meanings in the Chinese word "属于 (belong)" and "金属 (metal)", which are converted into "属#1 于" and "金 属#2" respectively, where suffix "#1" and "#2" is for distinguishing multiple senses. In this way, each target sentence in the training data is re-formatted as a sequence of sense-specified characters. During inference, the output is also sense-specified characters, we remove the suffix with "#" as a post-processing step for evaluation. The critical question is how to induce the multiple senses for each character type. Therefore, we propose a simple clustering algorithm for sense discrimination, in which we combine NP-MSSG [9] with the contextualized representation from BERT. Compared with the original NP-MSSG which is built upon a shallow skip-gram model, our method can also induce an unfixed number of senses for per character type, and benefits from the representation capability of BERT.

We describe the clustering procedure for each character c as follows. First, we randomly sample ψ sentences containing c from the training data. Second, we obtain ψ contextualized representations by inputting the ψ sentences into a pretrained BERT, and use the last layer's output of BERT. Let the representation for the i^{th} occurrence be g_i. Third, we create the first cluster with its first occurrence, i.e. using the contextualized representation g_0; for the next occurrence, we merge it into the nearest cluster if the similarity is above σ; otherwise, the occurrence is allocated to a new cluster. Formally, the i^{th} occurrence is allocated to the cluster s_i by:

$$s_i = \begin{cases} k_{max}, & \text{sim}(\mu(c, k_{max}), g_i) > \sigma \\ N(c) + 1, & \text{otherwise} \end{cases} \tag{3}$$

where k_{max} denotes the index of the cluster with the nearest distance to g_i, $\mu(c, k)$ is the centroid of the k^{th} cluster $C_{c,k}$ for character c, which equals to the mean of the contextualized representations in the cluster, sim denotes cosine similarity function, and $N(c)$ is the number of clusters already allocated for character c.

Last, we remove the clusters without sufficient occurrences (less than 3). Each cluster left corresponds to a distinct sense for the character, and the centroid is used as the sense embedding. Using these sense embeddings, we can infer the specific sense for per character in the target sentence in terms of the similarity between its contextualized representation and sense embeddings.

2.3 Joint Model

Instead of conducting pre-clustering in the first method, inspired by [10], we propose a joint model which can perform sense discrimination in real time during NMT training. Specifically, we convert the sense-agnostic loss in Eq. (2) into a sense-aware loss by predicting the next character and its sense jointly:

$$\mathcal{L} = -\log p(y_t, s_t|x, y_{<t}) \tag{4}$$

$$\begin{aligned} p(y_t, s_t|x, y_{<t}) &= \text{softmax}(h_t^{\mathrm{T}} S_{y_t, s_t}) \\ &= \frac{\exp(h_t^{\mathrm{T}} S_{y_t, s_t})}{\sum_{y'_t \in V} \sum_{s'_t=1}^{K} \exp(h_t^{\mathrm{T}} S_{y'_t, s'_t})} \end{aligned} \tag{5}$$

where s_t it the selected sense for y_t. It's worth noting that the output embedding layer is modified by replacing W with multiple sense embeddings $S \in R^{d \times |V| \times K}$, in which each character c has K separate sense embeddings, i.e. $(S_{c,1}, S_{c,2}, \ldots, S_{c,K})$. To optimize Eq. (4), we need a sense prediction module to obtain the most likely sense s_t, which is unobserved in the training data. The most intuitive solution is to set s_t as $\text{argmax}_k h_t^{\mathrm{T}} S_{y_t,k}$. However, this argmax operation would cause the differential problem in training, so we turn to a weighted loss using a soft sense distribution:

$$\mathcal{L} = -\log \sum_{k=1}^{K} \alpha(s_t = k)\, p(y_t, s_t = k|x, y_{<t}) \tag{6}$$

where $\alpha(s_t = k)$ is the probability of selecting the k^{th} sense, which is obtained from a sense prediction network. Formally, it is computed by

$$\alpha(s_t = k) = \frac{\exp\left(f(h_t, S_{y_t,k})\right)}{\sum_{k'=1}^{K} \exp\left(h_t^{\mathrm{T}} S_{y'_t,k'}\right)} \tag{7}$$

$$f(h_t, S_{y_t,k}) = v^{\mathrm{T}} \tanh(U_1 h_t + U_2 S_{y_t,k}) \tag{8}$$

where U_1, U_2, and v are the parameters in the sense prediction network f.

During inference, the next character and its sense are predicted jointly, that is $y_t^*, s_t^* = \mathrm{argmax}_{y_t,s_t}\ p(y_t, s_t | x, y_{<t})$. Note here, our focus is to strengthen the sense discrimination ability in the thin output embedding layer of the decoder, so we only use senses as an auxiliary prediction task, and not use them as the input condition, which is also consistent with the training procedure.

3. Experiments

3.1 Dateset

We conduct Japanese-to-Chinese (J→C) and Chinese-to-Japanese (C→J) translation experiments on JPO Patent Corpus, which consists of approximately 1 million parallel Japanese-Chinese sentences in the patent domain. The size of training set is 1 million, and validation set is 2,000. There are four test sets including 2K, 3K, 204 and 5K sentences respectively (noted as test-n1, test-n2, test-n3, and test-n4). To fasten training, we filter out too long examples in the training data, resulting in 970K examples left.

3.2 Setup

For pre-clustering, we set $\psi = 100$, $\sigma = 0.75$; for the sake of clustering accuracy and efficiency, we only cluster the Chinese characters with a frequency greater than 200 in the training data both for Chinese and Japanese (we exclude Kana characters in Japanese). For the joint model, we set $K = 3$.

For translation task, we implement all the methods on the top of Transformer in the toolkit of fairseq [11]. We utilize a model architecture consisting of 6 encoder and decoder layers, with each layer incorporating a multi-head attention comprising 8 heads. The word embedding and high-level representation dimensions are set at 512, while the FFN layer is set at 2048. We employ Adam optimizer, with a warm-up step of 8000 and a dropout probability of 0.1, and incorporated uniform label smoothing with 0.1 uncertainty. During inference, we select the averaged model of the last 5 epoch checkpoints to perform beam search decoding with a beam width of 5.

3.3 Main Results

Table 1 reports the Japanese-Chinese translation results measured by BLEU scores on the test sets (test-n3 is not used individually due to its small size). The baseline is a character based NMT using vanilla Transformer. BPE denotes a subword based NMT with a joint vocabulary of 32K BPE tokens[††]. Different to NMT for alphabetic languages, we can observe that the BPE approach in Japanese-Chinese NMT is inferior to the baseline using characters, which is consistent with the conclusions in [12], [13]. One reason is that the words in logographic languages are far shorter than words in alphabetic languages, which causes BPE failing to solve the low frequency words' problem. Another reason is that the Chinese character is a much more meaning-bearing unit than the alphabet, which makes it suitable as the encoding and decoding unit. In contrast, our pre-clustering based method brings consistent improvements over the baseline on all the test sets, and obtains a large improvement of 0.74 BLEU points (62.06 vs. 62.80) on the overall, which is the best result among all the tested systems. The joint model also achieves preferable performance, with an improvement of 0.63 BLEU points (62.06 vs. 62.69).

Table 1 BLEU scores evaluated on test set (J→C)[†]

	test-n1	test-n2	test-n4	test-all
Baseline	56.05	57.76	72.79	62.06
BPE	54.75	56.87	71.89	61.17
Pre-cluster	56.40	58.39	**73.60**	**62.80**
Joint model	56.11	**58.50**	**73.39**	**62.69**

Table 2 BLEU scores evaluated on test set (C→J)

	test-n1	test-n2	test-n4	test-all
Baseline	59.08	58.99	74.07	63.19
BPE	59.29	58.83	72.91	62.83
Pre-cluster	**60.54**	**59.50**	74.02	**63.79**
Joint model	**60.42**	**59.73**	73.87	**63.77**

Similar trends can be found for the Chinese-to-Japanese direction, as shown in Table 2. Our pre-clustering based method obtains an improvement of 0.6 BLEU points (63.19 vs. 63.79), and the joint model achieves an improvement of 0.58 BLEU points (63.19 vs. 63.77).

3.4 Cluster Example

Figure 1 shows a pre-clustering example for the Chinese character "属". Its contextualized embeddings are grouped into 3 clusters, which roughly correspond to three distinct senses, i.e. "belong" (occurs in the context of "所属" and "属于"), "metal" (occurs in the context of "金属") and "property" (occurs in the context of "属性"), respectively.

3.5 Quantitative Analysis

To examine whether our methods can help to generate characters with multiple senses in the decoder, besides the

[†]Note: the figures in bold denote statistically that the corresponding result is better than that of the baseline ($p < 0.05$).

[††]Before conducting BPE tokenization, the Chinese and Japanese sentences are segmented with toolkits of Jieba and Mecab respectively

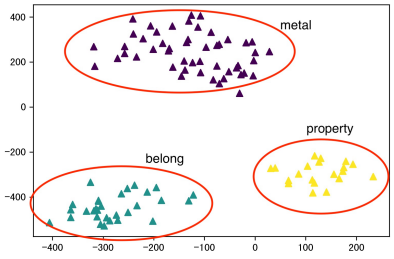

Fig. 1 2D t-SNE projection of contextualized embeddings.

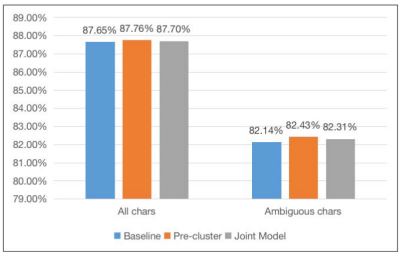

Fig. 2 Comparison of target character recall on J-C task.

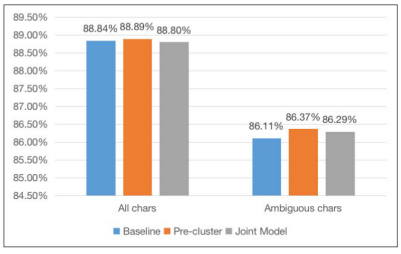

Fig. 3 Comparison of target character recall on C-J task.

sentence-level BLEU, we conduct a finer-grained analysis using target character recall (TCR). First, we quantitatively define the degree of sense ambiguity for the characters, which equals to the entropy of sense distribution derived in pre-clustering, i.e. $-\sum_k p(C_{c,k}) \log p(C_{c,k})$, where $p(C_{c,k}) = |C_{c,k}|/\sum_k |C_{c,k}|$. Then, we create a set Λ, which is comprised of 200 most sense ambiguous characters in the target language with the highest entropy values. Last, we run the word aligner AwesomeAlign [14] on the test set (i.e. between source and reference) and system output (i.e. between source and prediction) respectively, and output two character-level alignments $\mathcal{A}$ and $\mathcal{B}$. The TCR is computed as $|\mathcal{A} \cap \mathcal{B}|/|\mathcal{A}|$. As shown in Figs. 2 and 3, we compare the TCR on the overall characters and ambiguous characters (i.e. within Λ) respectively. It can be observed that in both directions, our methods have higher TCR compared to the baseline, and the improvements are more obvious for the ambiguous characters, which verifies the effectiveness of our proposed methods.

4. Conclusions

The present paper presents two simple yet effective methods for modeling sense-discriminated representation for characters based NMT, i.e., a pre-clustering for sense induction and a joint model to perform sense discrimination and NMT training simultaneously. To the best of our knowledge, this is the first work to explicitly model the sense discrimination in NMT decoding. Improvements on Japanese↔Chinese translation tasks verify the effectiveness. As our current study is centered on Japanese-Chinese NMT, we plan to validate these methods on NMT for other languages in the future.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," Proc. NIPS, Long Beach, CA, USA, pp.5998–6008, Dec. 2017.

[2] J. Libovický, H. Schmid, and A. Fraser, "Why don't people use character-level machine translation?," Proc. ACL, Dublin, Ireland, pp.2470–2485, 2022.

[3] J. Li, Y. Shen, S. Huang, X. Dai, and J. Chen, "When is char better than subword: A systematic study of segmentation algorithms for neural machine translation," Proc. ACL/IJCNLP, Virtual Event, pp.543–549, 2021.

[4] C. Cherry, G. Foster, A. Bapna, O. Firat, and W. Macherey, "Revisiting character-based neural machine translation with capacity and compression," Proc. EMNLP, Brussels, Belgium, pp.4295–4305, 2018.

[5] A.R. Gonzales, L. Mascarell, and R. Sennrich, "Improving word sense disambiguation in neural machine translation with sense embeddings," Proc. WMT 2017, Copenhagen, Denmark, pp.11–19, 2017.

[6] F. Liu, H. Lu, and G. Neubig, "Handling homographs in neural machine translation," Proc. NAACL-HLT, New Orleans, Louisiana, USA, pp.1336–1345, 2018.

[7] Z. Yang, W. Chen, F. Wang, and B. Xu, "Multi-sense based neural machine translation," Proc. IJCNN, Anchorage, AK, USA, pp.3491–3497, 2017.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," Proc. NAACL-HLT, Minneapolis, Minnesota, pp.4171–4186, 2019.

[9] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum, "Efficient non-parametric estimation of multiple embeddings per word in vector space," Proc. EMNLP, Doha, Qatar, pp.1059–1069, 2014.

[10] L. Liu, T.H. Nguyen, S.R. Joty, L. Bing, and L. Si, "Towards multi-sense cross-lingual alignment of contextual embeddings," Proc. COLING, Gyeongju, Korea, pp.4381–4396, 2022.

[11] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," Proc. NAACL-HLT, Minneapolis, MN, USA, pp.48–53, 2019.

[12] L. Zhang and M. Komachi, "Using sub-character level information for neural machine translation of logographic languages," ACM Trans. Asian Low Resour. Lang. Inf. Process., vol.20, no.2, pp.1–15, 2021.

[13] J. Zhang and T. Matsumoto, "Character decomposition for japanese-chinese character-level neural machine translation," Proc. IALP, Shanghai, China, pp.35–40, November 2019.

[14] Z.-Y. Dou and G. Neubig, "Word alignment by fine-tuning embeddings on parallel corpora," Proc. EACL, Online, pp.2112–2128, 2021.