Character Decomposition for Japanese-Chinese Character-Level Neural Machine Translation

Jinyi Zhang
Graduate School of Engineering
Gifu University
Gifu, Japan
Email: zhang@mat.info.gifu-u.ac.jp

Tadahiro Matsumoto
Faculty of Engineering
Gifu University
Gifu, Japan
Email: tad@gifu-u.ac.jp

***Abstract*—After years of development, Neural Machine Translation (NMT) has produced richer translation results than ever over various language pairs, becoming a new machine translation model with great potential. For the NMT model, it can only translate words/characters contained in the training data. One problem on NMT is handling of the low-frequency words/characters in the training data. In this paper, we propose a method for removing characters whose frequencies of appearance are less than a given minimum threshold by decomposing such characters into their components and/or pseudo-characters, using the Chinese character decomposition table we made. Experiments of Japanese-to-Chinese and Chinese-to-Japanese NMT with ASPEC-JC (Asian Scientific Paper Excerpt Corpus, Japanese-Chinese) corpus show that the BLEU scores, the training time and the number of parameters are varied with the number of the given minimum thresholds of decomposed characters.**

***Keywords*-character decomposition; neural machine translation; Japanese-Chinese; character-level; LSTM; encoder; decoder;**

I. Introduction

Machine translation's performance has greatly improved from Statistical Machine Translation (SMT), due to the appearance of Neural Machine Translation (NMT). For NMT, one problem is handling of the low-frequency words/characters in the vocabulary of the training data [1]. For the NMT models, as the vocabulary size increases, the computational complexity becomes enormous. Therefore, in a general word-level NMT model, the vocabulary size (the number of different characters) is usually limited to about tens of thousands of words, and the remaining low-frequency words are uniformly treated as unknown words. The increasing of unknown words leads to reduce translation performance, therefore the handling of low frequency words is a big problem in NMT.

Byte Pair Encoding (BPE) made the NMT model capable of open-vocabulary translation by encoding low-frequency and unknown words as sequences of subword units, was proposed by Sennrich et al. [2], to be used to solve the low frequency words' problem.

However, Chinese mainly uses Chinese characters (Hanzi) which are logograms. Many Chinese words are written with one or two Chinese characters, as a result, it is difficult to divide a Chinese word into high-frequency subword units. Therefore, it is considered that the character-level is suitable for NMT between Japanese and Chinese. For character-level NMT, there is also an advantage that errors and fluctuations do not occur in the process of dividing sentences into words (word segmentation).

Compared with the word-level NMT, the vocabulary size is kept small in character-level NMT, but there are still many characters of extremely low-frequency in the vocabulary. At word-level, the method of replacing a low-frequency word having low statistical reliability with another word of related high-frequency has been attempted [3], but such a substitution is difficult for characters. Therefore, we devised a method for reducing low-frequency characters for character-level NMT between Japanese and Chinese by dividing low-frequency Chinese characters into constituent elements of the character (radicals: traditionally recognized components of Chinese characters) and pseudo partial characters. We investigated the effects of the method on translation results, and the number of the parameters of the model by experiments.

We used Luong's NMT system as the base system [4], which follows an encoder-decoder architecture with global attention at the character level. In our case, we chose the character-level NMT as the baseline, because the character-level NMT between Japanese and Chinese has better translation performance than the word-level NMT.

The main contributions of this paper are the following. We created a Chinese character composition table for finding its constituent elements. We demonstrate the possibility to improve the translation performance of NMT systems by dividing the Chinese and Japanese characters into constituent elements and share them with the other characters in the vocabulary, without changing the neural network architecture. We believe this capability makes our approach applicable to different NMT architectures.

In the remainder of this paper, Section II presents the related work of this paper. Section III gives a brief explanation of the architecture of the NMT that we used as the base system and ASPEC-JC (Asian Scientific Paper Excerpt Corpus, Japanese-Chinese) corpus. Section IV describes the proposed method, how to divide the Chinese and Japanese characters into constituent elements and share them with the other characters in the vocabulary. Section V reports the experimental framework and the results obtained in the Japanese-Chinese and Chinese-Japanese translation (with ASPEC-JC [5]). Finally, Section VI concludes with the contributions of this paper and

978-1-7281-5014-7/19/$31.00 ©2019 IEEE

further work.

II. Related Work

The characters used in a language are usually much fewer than the words of the language. Character-level neural language models [6] and MT are explored and achieved respective results. Previous works, such as POS tagging [7], name entity recognition [8], parsing [9], learning word representations [10], and character embeddings [11], shown different advantages of using character-level information in Natural Language Processing (NLP).

Besides, subword-based representations (the middle of word-based and character-based representations) have been explored in NMT [2], and are applied to English and other western languages, where most of the words consist of several or a dozen characters. Contrastingly, Chinese characters, which are used in Chinese, Japanese and some other Asian languages, are typical logograms. A logogram is a character that represents a concept or thing, namely a word; and thus, it is difficult to split those words into subwords. Recently, Meng et al. [12] found that character-based models consistently outperform subword-based and word-based models for deep learning-based Chinese NLP works.

For Chinese-Japanese NMT, the sub-character level information improved the translation performance [13], by using sub-character sequences on either the source or target side. However, about their character decomposition, it still needs to be explored. Du and Way [14] trained factored NMT models using "Pinyin" sequences on the source side. Pinyin, is the official romanization system for Chinese. This work only applied to Chinese source-side NMT. Zhang and Matsumoto [15] also attempted to use a factored encoder for Japanese-Chinese NMT system using radical information. They did not achieve good results in Chinese-to-Japanese NMT. Wang et al. [16] directly applied a BPE algorithm to sequences before building NMT models. This method has only been tested in the Chinese-English direction and is not comprehensive enough.

III. Neural Machine Translation and ASPEC-JC Corpus

A. Neural Machine Translation

NMT completely adopts the neural network approach to compute the conditional probability $p(y|x)$ of the target sentence y for the given source sentence x. We follow the NMT architecture by Luong et al. [4], which we will briefly describe here. This NMT system is implemented as a global attentional encoder-decoder neural network with Long Short-Term Memory (LSTM), and we simply use it at the character level.

The encoder is a bi-directional neural network with LSTM units that reads an input sequence $x = (x_1, \ldots, x_m)$ and calculates a forward sequence of hidden states $(\overrightarrow{h}_1, \ldots, \overrightarrow{h}_m)$ and a backward sequence $(\overleftarrow{h}_1, \ldots, \overleftarrow{h}_m)$. The hidden states $\overrightarrow{h_j}$ and $\overleftarrow{h}_j$ are concatenated to obtain the annotation vector h_j.

The decoder is a recurrent neural network with LSTM units that predicts a target sequence $y = (y_1, \ldots, y_n)$. Every word (or character in case of character-level NMT) y_i is predicted based on a recurrent hidden state s_i, the previously predicted word (or character) y_{i-1}, and a context vector c_i. c_i is computed as the weighted sum of the annotations h_j. Finally, the weight of each annotation h_j is computed through an alignment (or attention) model α_{ij}, which models the probability that y_i is aligned to x_j. The forward states of the encoder is expressed as below:

$$\overrightarrow{h}_j = \tanh(\overrightarrow{W} E x_j + \overrightarrow{U} \overrightarrow{h}_{j-1}) \tag{1}$$

where $E \in \mathbb{R}^{m \times V_x}$ is a word embedding matrix, $\overrightarrow{W} \in \mathbb{R}^{n \times m}$ and $\overrightarrow{U} \in \mathbb{R}^{n \times n}$ are weight matrices; m, n and V_x are the word embedding size, the number of hidden units, and the vocabulary size of the source language, respectively.

B. ASPEC-JC Corpus

We implement our system with the ASPEC-JC corpus, which was constructed by manually translating Japanese scientific papers into Chinese [5]. The Japanese scientific papers are either the property of the Japan Science and Technology Agency (JST) or stored in Japan's Largest Electronic Journal Platform for Academic Societies (J-STAGE).

ASPEC-JC is composed of three parts: training data (672,315 sentence pairs), development data (2,090 sentence pairs), development-test data (2,148 sentence pairs) and test data (2,107 sentence pairs) on the assumption that it would be used for machine translation research.

ASPEC-JC contains both abstracts and some parts of the body texts. ASPEC-JC only includes "Medicine", "Information", "Biology", "Environmentology", "Chemistry", "Materials", "Agriculture" and "Energy" 8 fields because it was difficult to include all the scientific fields. These fields were selected by investigating the important scientific fields in China and the use tendency of literature databases by researchers and engineers in Japan. In these fields, sentences belonging to the same article are not included.

Compared with other language pairs such as English-French, which usually comprises millions of parallel sentences. ASPEC-JC corpus only has about 672k sentences. Moreover, LSTMs+attention model is usually more robust than the transformer model [17] on smaller datasets, due to the smaller number of parameters [12].

IV. Reduction of low-frequency characters by character decomposition

During the training and translation process, the training data contains many low-frequency characters that the NMT model cannot translate. The low-frequency characters affect translation performance.

In this research, we decomposed low-frequency characters (mainly Chinese characters) by using high-frequency characters and pseudo-characters, and sharing pseudo-characters among multiple low-frequency characters. We devised a method to remove characters below a certain frequency and checked the effect on translation performance in the experiment.

The method of decomposing low-frequency characters will be described below.

A. Character Decomposition

Chinese characters are logograms, but some different types could be identified, based on the manner in which they are formed.

They include:

- pictographs: 日 (sun), 月 (moon), 人 (person), 木 (tree),
- simple ideograms: 一 (one), 二 (two), 上 (up), 下 (down),
- compound ideographs: 林 (woods ← tree+tree), 休 (rest ← person+tree), and
- phono-semantic compounds: 銅 (copper ← semantic 金 (metal) + phonetic 同), 河 (river ← semantic 水 (water) + phonetic 可).

Phono-semantic compounds, together with compound ideographs, form over 90% of Chinese characters; accordingly, most Chinese characters consist of two or more (sub-)characters.

Even if a character is rare, its component may be a high-frequency character. For example, 楡 (elm) appears only 16 times in the Japanese sentences of the ASPEC-JC training data, whereas its component 木 (tree) appears 7780 times. If there are other low-frequency characters that have 木 as their components, the frequency of 木 increases more by decomposing the low-frequency characters. In most cases, the higher frequency component (such as 木) of a compound character is a radical, which is related to the meaning of the character.

Our method decomposes low-frequency characters into two partial characters by using the Chinese character decomposition table (Section IV-B). If a character has three or more parts, the method decomposes it into the first part and the rest. Comparing the two components of a character, the component appearing less frequently in the training data is replaced with a pseudo-character, such as s_1, s_2, ..., s_n.

The appearance frequency of the pseudo-partial characters are increased by sharing them among low-frequency characters as follows:

楡 (elm tree) → [木 (wood), s_1]
枡 (a square wooden box used to measure rice)
→ [木 (wood), s_2]
炒 (fry) → [火 (fire), s_1]
焔 (flame) → [火 (fire), s_2]

To balance the frequency of the pseudo-characters, we set an upper limit of the number of pseudo-partial characters that are paired with each genuine character component.

If the number exceeds the limit, the method decomposes the character into two pseudo-characters as follow:

榊 (sakaki tree) → [s_{13}, s_{16}]
枷 (cangue) → [s_{19}, s_{22}]

If a low-frequency character cannot be decomposed, it is replaced with a pair of a pseudo-character and 漢 (han) (for Chinese characters), 仮 (assumed) (for Japanese Kana) or 符 (symbol) (for symbols and other characters).

In this way, the method replaces every low-frequency ($\leq k$) character with a pair of a high-frequency character and a pseudo-character or pairs of two pseudo-characters, in order to eliminate such low-frequency characters in the training data.

The mappings from the low-frequency characters to the character pairs are separately created for Japanese and Chinese training data. The training is conducted with the decomposed data. In testing time, low-frequency characters in the source sentences are decomposed with the mapping for the source language before translation. The translated sentences are reconstructed (decoded) with the mapping for the target language. If the reconstruction of a character is failed, the character pair is replaced with the space character.

B. Creation of Chinese Character Composition Table

We created a table for decomposing Chinese characters, based on the Chinese character decomposition table of cjklib[1], the Kanji structure information table of the CHISE project[2], Jigen[3] and the distribution data of the Kanji database project[4]. Our table was created manually.

If there are multiple Chinese characters having the same constituent elements as shown in Table I and Table II, they are distinguished by numbering as follows:

Examples: 暈 (dizzy) → 日軍1
暉 (sunshine) → 日軍2
柰 (crab-apple) → 木示1
标 (label) → 木示2

As the Table III and Table IV show, if the components are decomposed in a simple form, the meaning becomes weak, they are excluded from the table so that decomposition is not performed. In the experiments with the ASPEC-JC corpus, we excluded 102 Kanji out of 3,802 Kanji contained in Japanese sentences, and 204 Hanzi out of 5,576 Hanzi contained in Chinese sentences.

We have manually confirmed the Chinese character composition table, and also uploaded this table to github, hoping that interested people can come up with suggestions for improvement [5].

V. Evaluation and Translation Results

A. Experiment Settings

We implemented our system using the OpenNMT toolkit [18] with the ASPEC-JC corpus which had already introduced in Section III-B.

[1] http://cjklib.org

[2] http://www.chise.org

[3] http://jigen.net

[4] http://kanji-database.sourceforge.net

[5] https://github.com/zhang-jinyi/Chinese-Character-Composition-Table

Table I
EXAMPLES OF JAPANESE KANJI THAT HAVING THE SAME CONSTITUENT ELEMENTS.

Constituent elements	Japanese Kanji
弓丨	引 (draw), 弔 (hang)
朿朿	棘 (sour jujube), 棗 (jujube)
日軍	暈 (dizzy), 暉 (sunshine)
木口	束 (bind), 杏 (apricot)
土襄	壌 (soil), 壤 (soil)
口貝	唄 (song), 員 (member)
山夆	峰 (peak), 峯 (peak)

Table II
EXAMPLES OF CHINESE HANZI THAT HAVING THE SAME CONSTITUENT ELEMENTS.

Constituent elements	Chinese Hanzi
冂土	由 (by), 田 (field), 冉 (tender)
木示	标 (label), 柰 (crab-apple)
冂人	贝 (shellfish), 内 (inside)
口八	叭 (horn), 只 (only)
亻直	值 (value), 値 (value)
口贝	呗 (to chant), 员 (member)
日军	晖 (sunshine), 晕 (dizzy)

Our models have one LSTM layer, with 512 cells, and embedding size is 512. The parameters are uniformly initialized in $(-0.1, 0.1)$, using plain SGD, starting with a learning rate of 1 until epoch 6, and after that, 0.5 times for each epoch. The max-batch size is 100. The normalized gradient is rescaled whenever its norm exceeds 1. The dropout probability is set to 0.5 to avoid overfitting. Decoding is performed by beam search with a beam size of five. The maximum length of a sentence is 250 by default, but it is set to 500 because it becomes much longer at the character level.

We segment the Chinese and Japanese sentences into words by Jieba[6] and Mecab[7], respectively.

Table III
EXAMPLES OF JAPANESE KANJI THAT EXCLUDED FROM THE TABLE.

Constituent elements	Japanese Kanji
乚一	七 (seven)
一乂	丈 (measure)
一卜	下 (below)
丿乚	儿 (son)
月一	且 (even)
丨丶	卜 (divination)
十一	土 (soil)

Table IV
EXAMPLES OF CHINESE HANZI THAT EXCLUDED FROM THE TABLE.

Constituent elements	Chinese Hanzi
丿厶	么 (for interrogatives and adverbs)
一夕	歹 (bad)
厶月匕匕	能 (ability)
七十	车 (vehicle)

BiLingual Evaluation Understudy (BLEU) is an algorithm for evaluating the quality of text that has been machine-translated from one natural language to another [19]. BLEU score is calculated with multi-bleu.perl attached to OpenNMT after the word segmentation. In other words, we took the word-level evaluation.

In many cases, validation perplexity (perplexity with dev data) stopped declining in epoch 10 or 11. The average of BLEU scores from that point to epoch 16 was taken as the evaluation BLEU value. The baseline is the character-level translation with the raw training data that does not process anything.

B. Experiment Results and Discussion

Variation of BLEU scores: The low-frequency characters are deleted from the training data by the character decomposition method described in Section IV-A. Figure 1 shows the variation in BLEU scores per epoch.

The least frequency of occurrence of the baseline is 1. The upper limit of the subscript of pseudo-characters was basically set to 55 in both languages, but it was set to 60 when setting the least frequency of occurrence as 7000 to more in Chinese language data, because of the lack of pseudo-characters .

In the Japanese-to-Chinese translation, when the least frequency of occurrence was between 10 and 120, the translation results often exceeded the baseline. Improved about 0.5% when setting the least frequency of occurrence to 20. On the other hand, in the case of Chinese-to-Japanese translation, the translation result was less likely to exceed the baseline, but it improved by about 0.3% when setting the least frequency of occurrence to 150. The results above are not as good as we expected.

The type of decomposed characters are 78% of Chinese Hanzi in Chinese sentences, 47% of Japanese Kana in Japanese sentences, 36% of Japanese Kanji in Japanese sentences in the training data, respectively. It is conceivable that this difference affects the translation results. Unlike Chinese, Japanese Kanji only account for 36% in Japanese, and our method only decomposes Japanese Kanji in Japanese, which caused Japanese to be not fully decomposed. This will result in a certain decline in Japanese translation results in the direction of Chinese-to-Japanese. The decomposed sentences we used as training data became longer than before. This may be a factor that affects the translation results. There is also a possibility that the number of characters contained in the training data (the vocabulary size of the NMT system) also has a huge difference from the 6,088 characters of Chinese to the 4,249 characters of Japanese, on the ASPEC-JC corpus.

Variation of training time: Reducing low-frequency characters decreases the vocabulary size, so the number of parameters to be trained also decreases. As a result, it is expected that the amount of memory used during

[6] http://github.com/fxsjy/jieba

[7] http://taku910.github.io/mecab

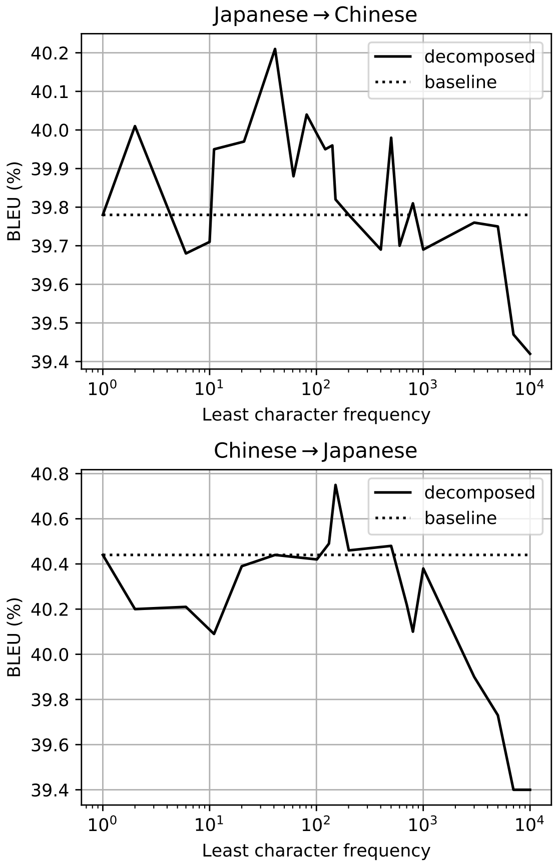

Figure 1. Variation of the BLEU scores

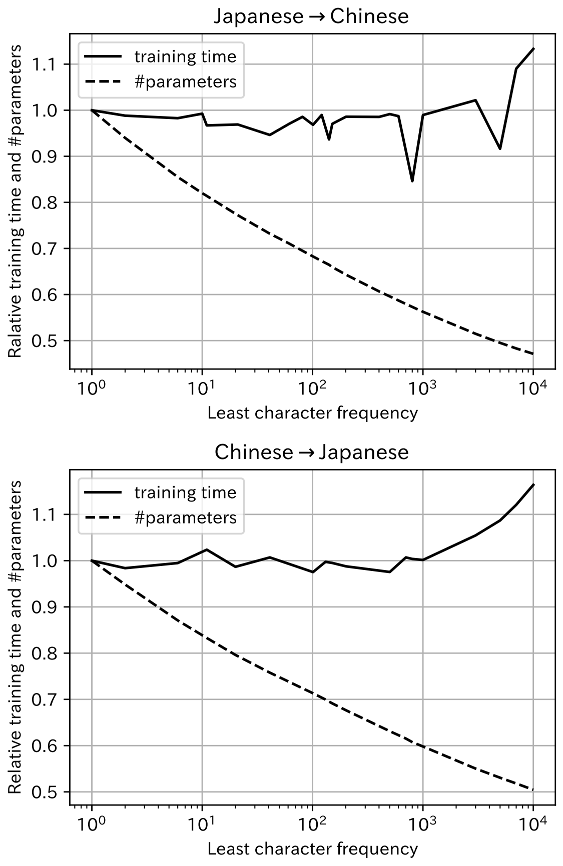

Figure 2. Variation of the training time and the number of parameters

training and the training time will be reduced. Figure 2 shows the number of parameters obtained from the log data in training and the variation in the average training time per epoch. Because experiments were conducted on multiple systems with different configurations of CPU and GPU, the values are relative to the results of the baseline training on each system.

The character decomposition reduces the number of parameters of NMT models, but increases the number of characters. In the Japanese-to-Chinese translation, the training time was always shorter than the baseline until the least frequency of 1000, and the average was 3.56% shorter in the range of the least frequency of 10 to 1000. On the other hand, the effect of shortening the training time was not seen much in Chinese-to-Japanese translation. The decomposed sentences become longer than before. This makes such a result that taking more time to calculate the models.

The results obtained above are dependent on the ASPEC-JC corpus. For different corpora, there should be different threshold (least frequency of occurrence) choices and different translation results.

VI. Conclusion

In this research, we created a Chinese character composition table and proposed a method to reduce low-frequency characters by decomposing low-frequency characters into Chinese characters' constituent elements and pseudo-characters for NMT between Japanese and Chinese.

Experiments of Japanese-to-Chinese and Chinese-to-Japanese NMT systems showed that the BLEU scores and the training time varied with the number of least frequency of decomposed characters. As a result, compared to the baseline, the BLEU value was about 0.5% higher in Japanese-to-Chinese and 0.3% higher in Chinese-to-Japanese. However, especially in the Chinese-to-Japanese, in most cases, the BLEU scores were lower than the baseline. The translation results are not very well overall. The training time was generally shorter than the baseline when the least frequency of occurrence was less than 1000 in the Japanese-to-Chinese translation experiment.

Because the decomposition of the Chinese characters causes the sentence to grow longer, we should increase the NMT model's support for long sentences, such as using the long sentence segmentation method for NMT [20].

Further, we should use the popular models to train, such as the transformer model [17].

In the future, we should improve or find a better character decomposition method to choose the appropriate least frequency of occurrence for different corpus, even at the character level translation from Chinese to other languages, or from Japanese to other languages. The use of Chinese characters' constituent elements may lead to an improvement for translation performance with fewer parameters and shorter training time.

Acknowledgment

J. Zhang was supported by China Scholarship Council (CSC) under the CSC Grant (No.201708050078). All authors wish to thank the anonymous reviewers and editors for their insightful comments and assistance.

References

[1] P. Koehn and R. Knowles, "Six Challenges for Neural Machine Translation," in *Proc. of the First Workshop on Neural Machine Translation*, ACL, Vancouver, Canada, 2017, pp 28–39.

[2] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annual Meeting of the Assoc. for Computational Linguistics*, Berlin, Germany, 2016, pp 1715-1725.

[3] R. Chitnis, and J. DeNero, "Variable-Length Word Encodings for Neural Translation Models," in *Proc. 2015 Conf. on Empirical Methods in Natural Language Processing*, ACL, Lisbon, Portugal, 2015, pp. 2088–2093.

[4] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. 2015 Conf. on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015, pp 1412-1421.

[5] T. Nakazawa, M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara, "ASPEC: Asian scientific paper excerpt corpus," in *Proc. 9th Int. Conf. on Language Resources and Evaluation (LREC 2016)*, European Language Resources Association (ELRA), Portorož, Slovenia, 2016, pp 2204-2208.

[6] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character aware neural language models," in *Proc. 30th AAAI Conf. on Artificial Intelligence*, Phoenix, Arizona, USA, 2016. pp. 2741-2749.

[7] C. D. Santos and B. Zadrozny, "Learning character-level representations for part-of-speech tagging," in *The 31st International Conference on Machine Learning*, Beijing, China, 2014, pp. 1818-1826.

[8] C. D. Santos and V. Guimaraes, "Boosting named entity recognition with neural character embeddings," in *Proc. Fifth Named Entity Workshop*, ACL, Beijing, China, 2015, pp. 25-33.

[9] M. Ballesteros, C. Dyer, and N. A. Smith, "Improved transition-based parsing by modeling characters instead of words with lstms," in *Proc. 2015 Conf. on Empirical Methods in Natural Language Processing*, ACL, Lisbon, Portugal, 2015, pp. 349-359.

[10] X. Chen, L. Xu, Z. Liu, M. Sun, and H. B. Luan, "Joint learning of character and word embeddings," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, AAAI Press, Buenos Aires, Argentina, 2015, pp. 1236-1242.

[11] Y. Li, W. Li, F. Sun, and S. Li, "Component-enhanced Chinese character Embeddings," in *Proc. 2015 Conf. on Empirical Methods in Natural Language Processing*, ACL, Lisbon, Portugal, 2015, pp. 829-834.

[12] Y. Meng, X. Li, X. Sun, Q. Han, A. Yuan and J.Li, "Is Word Segmentation Necessary for Deep Learning of Chinese Representations? " in *Proc. 57th Annual Meeting of the Assoc. for Computational Linguistics*, Florence, Italy, 2019, [Online]. arXiv:1905.05526.

[13] L. Zhang and M. Komachi, "Neural Machine Translation of Logographic Languages Using Sub-character Level Information," in *Proceedings of the 3rd Conference on Machine Translation (WMT)*, Brussels, Belgium, 2018, pp. 17–25.

[14] J. Du and A. Way, "Pinyin as Subword Unit for Chinese-Sourced Neural Machine Translation," in *Irish Conference on Artificial Intelligence and Cognitive Science*, Dublin, Ireland, 2017.

[15] J. Zhang and T. Matsumoto, "Improving Character-level Japanese-Chinese Neural Machine Translation with Radicals as an Additional Input Feature," in *the 21st International Conference on Asian Language Processing (IALP)*, Singapore, 2017, pp. 172-175.

[16] Y. Wang, L. Zhou, J. Zhang, and C. Zong, "Word, subword or character? an empirical study of granularity in Chinese-English NMT," in *China Workshop on Machine Translation*, Springer, Dalian, China, 2017, pp. 30–42.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *the 31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 5998-6008.

[18] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "OpenNMT: Open-source toolkit for neural machine translation," in *Proc. 55th Annual Meeting of the Assoc. for Computational Linguistics*, Vancouver, Canada, 2017, pp. 67–72.

[19] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," *in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, PA, USA, 2002, pp. 311-318.

[20] J. Pouget-Abadie, D. Bahdanau, B. V. Merrienboer, K. Cho, and Y. Bengio, "Overcoming the Curse of Sentence Length for Neural Machine Translation using Automatic Segmentation," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, Qatar, 2014, pp. 78–85.