Korean Neural Machine Translation Using Hierarchical Word Structure

Jeonghyeok Park and Hai Zhao*
Department of Computer Science and Engineering, Shanghai Jiao Tong University
Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai, China
117033990011@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

***Abstract*—Korean neural machine translation may significantly suffer from low-resource issues. We thus propose an enhancement method that fully exploits the hierarchical Korean word embedding structure from source representation. To our best knowledge, this is the first attempt for such Korean NMT tasks. Every Korean word can be decomposed into character- and jamo-level (sub-character) units. Therefore, We merge the character- and jamo-level representations with word embeddings to capture important combining word meaning. And then the merged representations are fed into NMT model. Our simple and novel method achieves BLEU improvements (up to 0.6) compared to word-based NMT baselines on Korean-to-Chinese and Korean-to-English translation tasks.**

***Index Terms*—Machine Translation, Hierarchical Word Structure, Korean Language.**

I. Introduction

For different language pairs, traditional word-based neural machine translation (NMT) models suffer from the out-of-vocabulary (OOV) issue as they can only model a limited number of words. Character-based representations were proposed as a solution to overcome OOV problems, but it may too fine-grained to miss some important information. And Byte Pair Encoding (BPE) [16] demonstrated extremely competitive performance by providing effective subword segmentation for NMT systems. Though the technique has solved the OOV problem efficiently, it still misses the semantic and syntactic information of the word itself. In this paper, we introduce a simple and novel method of supplementing additional information to the encoder of the NMT model by utilizing a unique compositional structure of the Korean language.

In this work, we focus on the Korean language. Unlike other languages, the Korean language has a unique compositional structure because it has both the features of the alphabetic and syllabic writing systems. Moreover, the decomposition of Korean syllables is deterministic [19]. Korean word is constructed by a regular hierarchy, so no special markers or measures are needed for the decomposition. *Jamo*s, the Korean alphabet, are a set of the smallest unit that forms the language.

*Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), Key Projects of National Natural Science Foundation of China (U1836222 and 61733011), Huawei-SJTU long term AI project, Cutting-edge Machine reading comprehension and language model.

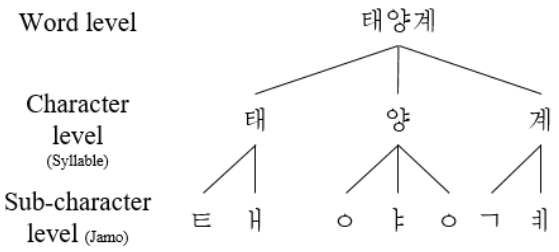

Fig. 1. The hierarchical structure of Korean language.

Every Korean word can be formed in the following hierarchy: A Korean word consists of a sequence of syllables, and a syllable (character) is arranged in a square, two-dimensional space with 2 or 3 jamo letters (sub-character). In other words, Korean words can be decomposed into two stages sequentially. As shown in Figure 1, the word 태양계 (solar system, *taeyanggye*) is a composition of three characters {태 (*tae*), 양 (*yang*) and 계 (*gye*)}, and the character 태 is a composition of {ㅌ (*t*), ㅐ (*ae*)}. The unique compositional structure of the Korean language are often ignored in most Korean NLP. Recently there have been studies that have proved that both characters and jamo letters contain important semantic and syntactic information in Korean language [19], [20]. Motivated by these works, we propose a hierarchical representation that merges word-/character-/jamo-level representation for Korean NMT. In our experiment, we demonstrate that the proposed method improves about 0.5 BLEU score of Korean NMT on average.

II. Related works

Since the dawn of NLP, there have been a variety of studies that encode more information that extracts from subword-level units into the representation of neural models. Many morpheme-based and character-based models have been proposed [5]–[10]. Among them, [9] proposed a neural language model that uses a convolutional neural network (CNN) to produce high-quality character-level representation. Partially motivated by this work, we further extend the range from character level to the sub-character level. For Korean, some previous studies exploit information that extracts from jamo

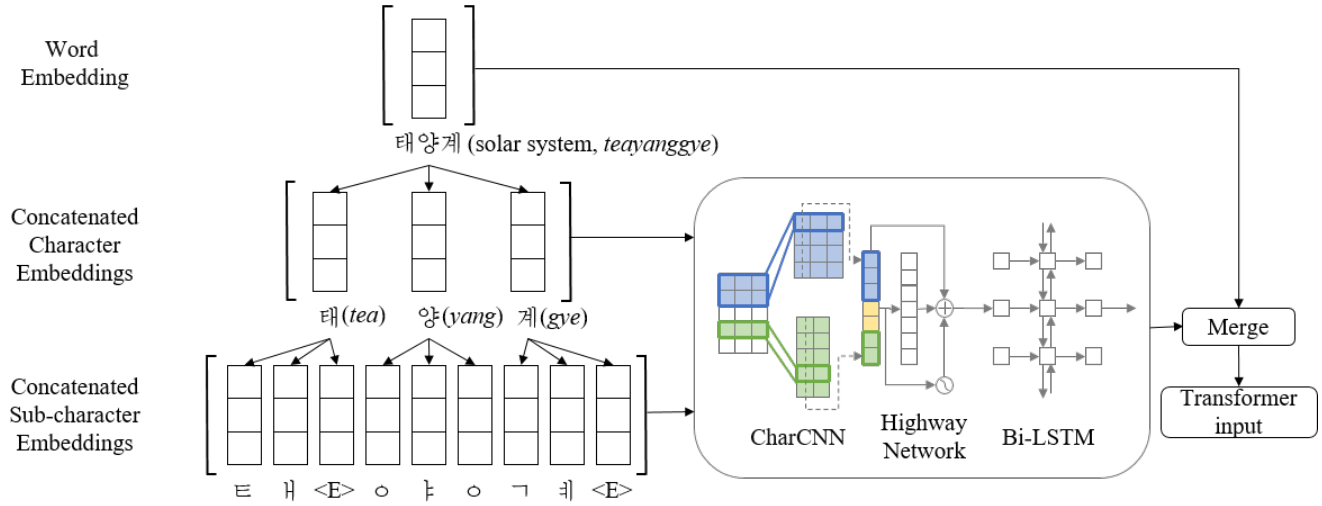

Fig. 2. Hierarchical structure of Korean language and overall architecture of our model. $\langle E \rangle$ denote a empty symbol of the final consonant.

letters to enhance the accuracy and richness of word-level representations [19], [20]. Reference [17] showed that successfully produces syllable-based representation for Korean language using a CNN model.

For machine translation, various studies have demonstrated that sub-character-level information could further enhance performance [23]–[25]. Reference [25] integrates the Chinese radicals into the NMT model to address this OOV issue on Chinese-to-English translation task. Reference [23], [24] successfully improved the performance of NMT models and UNMT models by leveraging decomposed sub-character level information for logographic languages (e.g., Chinese, Japanese). On the other hand, there are several approaches that introduce character embedding or linguistic feature information for western languages such as English and German that cannot be decomposed into sub-character units to solve OOV issue and improve translation performance [12]–[15]. Reference [12], [13] proposed the character-level NMT model without explicit segmentation. Reference [15] proved the usefulness of linguistic features by fusing various linguistic features into the embedding layer of the encoder in the NMT model.

III. Benefits of Character-/Jamo-level Representations

The low-level unit representation can not only alleviate the problem of data sparsity but also effectively address OOV problem caused by misspelling. There are also some additional benefits: Korean syllables have semantic meanings. About 60 percent of Korean words belong to the Sino-Korean vocabulary that is of Chinese origin and can be converted into corresponding Chinese characters. Since Chinese characters are primarily morphosyllabic, a single Chinese character conveys certain meaning (or perform a certain grammatical function). Hence, Korean syllables, especially those that can be converted into Chinese, have specific meanings in themselves. For example, the Korean word 태양계 (solar system) in Figure 1 can be written as 太陽系, which also means the solar system, in Chinese characters and is a composition of 태 (great, 太), 양 (light, 陽) and 계 (system, or series, 系). Reference [27] demonstrated that the performance of the MT models can be improved by utilizing the Sino-Korean word as a translation pivot in Korean-to-Chinese translation. Moreover, The Korean language, which is an official agglutinative language, has over 600 affixes (prefixed and suffixes). The verb endings attached to any verb stem serve various functions such as tense, questions, and honorific. Lastly, jamo letters are often used as suffix for grammatical purposes. For example, as in 갈 (will go), 찾을 (will find) and 날 (will fly), ㄹ is attached to a verb to indicate a future tense. Therefore, we utilize character-/jamo-level representation as well as word representation to capture the above-mentioned additional semantic and syntactic information.

IV. Model

We produce character-level and jamo-level representation through the character-level neural language model (CharCNN) [9], highway network [11], and LSTM [1]. Since CharCNN effectively aggregates character-level information about one word regardless of the length of the word, we adopted CharCNN as a core model for generating low-level unit representation. The overall architecture is shown in Figure 2.

Composition Model Let unit u be the component of a word $k \in \mathcal{V}$ where $\mathcal{V}$ is the vocabulary of words, d^u be the dimensionality of the unit embedding, and $e^u \in \mathbb{R}^{d^u}$ be embedding for each unit. Suppose that word k is made up of l units, $[u_1, u_2, ..., u_l]$. Then word k is represented by concatenating unit vectors as a column vector: $e^{unit}_{concat} = [e^{u_1}, e^{u_2}, ...e^{u_l}] \in \mathbb{R}^{d^u \times l}$. We first apply padding to the beginning and the end of e^{unit}_{concat}.Then we apply a narrow convolution between e^{unit}_{concat} and a convolution filter $H \in \mathbb{R}^{d_u \times w}$ of width w to obtain a

TABLE I
THE CORPUS STATISTICS FOR THE DIFFERENT PARALLEL CORPORA.

Corpus	Train	Vali	Test	# of word/char/jamo (KO)	# of word (EN&ZH)
Dong-A	247755	5000	5000	32567/1858/168	33854
AIHub	379897	5000	5000	32909/1986/210	32314

feature map $f^k \in \mathbb{R}^{l-w+1}$. We use multiple filters $H_1, ..., H_h$ of varying widths to aggregate various unit-level information. Given a word k, the i-th element of feature map f^k is given by:

$$f_i^k = \tanh(\langle e_{concat}^{unit}[*, i : i + w - 1], H\rangle + b)$$

where $\langle A, B\rangle = Tr(AB^T)$ is Frobenius inner product. We then apply a max pooling to capture the most important feature for the filter H and feed the output into a highway network and a bidirectional LSTM. And we get the unit-level representation $y^k = Comp(e^{u_1}, e^{u_2}, ...e^{u_l}; \omega)$ for word k, where ω is learned jointly with the overall objective.

Given a word 태양계, we first decompose into character-level units {태, 양, 계} and jamos-level units {ㅌ, ㅐ, $\langle E\rangle$, ㅇ, ㅑ, ㅇ, ㄱ, ㅖ, $\langle E\rangle$}. If some characters such as 태 and 계 lack final consonant, we add an empty symbol $\langle E\rangle$ such that the model can learn about the absence of the final consonant. We do not apply jamo-decomposition to characers such as English word, numeral. And BPE symbol is handled as a padding.

As shown in Figure 2, we produce a pair of the concatenated character- and jamo-level embeddings:

- Concatenated character-level embeddings: $e_{concat}^{char} = [e^{c_1}, ..., e^{c_n}] \in \mathbb{R}^{d^c \times n}$
- Concatenated jamo-level embeddings: $e_{concat}^{jamo} = [e^{j_1}, ..., e^{j_m}] \in \mathbb{R}^{d^j \times m}$

Then the concatenated embeddings are fed to the proposed model to generate the character- and jamo-level representation:

- character-level representation: $C^{word} = Comp(e_{concat}^{char}; \omega_{char})$
- Jamo-level embeddings: $J^{word} = Comp(e_{concat}^{jamo}; \omega_{jamo})$

Finally, the character- and jamo-level representation are merged with the word-level lookup embedding e^{word} for a word in three ways: Concatenating, Averaging, and merging with Gate network. Concatenation is to concatenate several representations at an equal rate:

$$R_{Concat}^{word} = [e^{word}; C^{word}; J^{word}]$$

where R_{Concat}^{word} is the final representation that is fed into the NMT model and the operator $[;]$ denotes concatenation. And Averaging denotes a simple average of all embedding [22]:

$$R_{Aver}^{word} = (e^{word} + C^{word} + J^{word})/n$$

where n is number of representations to merge. Reference [22] demonstrated that averging word embeddings can provide an approximation of the performance of concatenation without increasing the dimension of the embeddings. Lastly, merging with Gate network is leveraging gating vectors to induce a merged representation:

$$R_{gate}^{word} = \alpha \odot e^{word} + \beta \odot C^{word} + \gamma \odot J^{word}$$

where α, β, γ are gating vectors and their sum is 1 and $\odot$ denotes element-wise multiplication. Then the merged representations that integrate word-/character-/jamo-level representation with the three methods are fed into a NMT model.

V. EXPERIMENT

To verify that the proposed method is effective, we perform experiments on both Korean-to-English and Korean-to-Chinese translation. For all translation tasks, we evaluate the BLEU [2] score with SacreBLEU[1] [26] which aims to standardize BLEU evaluation. All the model trainings are on one NVIDIA Geforce GTX 1080 Ti GPU.

A. Experimental Settings

Datasets For Korean-to-English translation, we carry out experiments on a corpus of casual conversation that provided by AIHub[2]. For Korean-to-Chinese translation, we use news corpus, which is collected from the online Dong-A newspaper[3] by us. Table I shows the statistics of datasets used in our experiments. Given the full parallel corpus, we first tokenize Korean, Chinese, and English sentences by using KoNLPy[4], jieba[5], and Moses [3], respectively. And then, we employ BPE with 32K merge operations for all datasets before training NMT model. For jamo-level decomposition, we use the open toolkit to decompose a character into jamo letters[6].

Model Settings After analyzing the two Korean corpora, we found that most Korean words consist of 2.5 syllables or 5 jamos, so we set the filters of the CharCNN as follows: filters of width $[1, 2, 3, 4, 5, 6]$ of size $[25, 50, 75, 100, 125, 150]$ for a total of 525 filters. And the size of the BiLSTM's hidden state is set to 256. The character- and jamo-level embedding dimensions are set to 50 and 300, respectively.

[1]Our SacreBLEU signatures are $BLEU+case.mixed+lang.ko-zh+numrefs.1+smooth.exp+tok.zh+version.1.4.3$ (KO→ZH) and $BLEU+case.mixed+lang.ko-en+numrefs.1+smooth.exp+tok.13a+version.1.4.3$ (KO→EN)

[2]http://www.aihub.or.kr/

[3]http://www.donga.com/

[4]https://konlpy.org/en/latest/

[5]https://github.com/fxsjy/jieba

[6]https://github.com/bluedisk/hangul-toolkit

TABLE II
EXPERIMENTAL RESULTS ON KOREAN-TO-CHINESE TRANSLATION AND KOREAN-TO-ENGLISH TRANSLATION.

Features	SacreBLEU Score	
	Dong-A (KO→ZH)	AIHub (KO→EN)
jamo	40.2	28.6
char	39.7	29.3
word	41.8	30.6
Merge methods	Concat/Aver/Gate	Concat/Aver/Gate
jamo, char	40.9/41.8/41.6	29.6/30.0/29.8
jamo, word	**42.3**/42.0/41.9	**31.2**/30.7/30.6
char, word	**42.3**/42.2/**42.3**	31.0/31.0/30.9
jamo, char, word	42.2/**42.3**/41.8	**31.2**/31.0/30.9

We use the 6-layer base Transformer architecture as a baseline model. For all experiments, the size of the embedding vector of source-/target-side is set to 512, and the initial learning rate is set to 0.2. We optimize the model parameters using Adam optimizer [4] with $\beta_1 = 0.9$ and $\beta_1 = 0.998$, and Noam learning rate decay [21] with 8000 warm-up steps. All our experiments were performed using the open source Torch-based toolkit OpenNMT [18].

B. Experimental Results

Table II shows the sacreBLEU evaluation of our systems. For both KO→ZH and KO→EN, the results show similar patterns: (1) The performance of transformer models trained with only the jamo- or character-level representation is lower than with only the word-level representation. (2) Merging word-level representation and other representations led to the transformer model's performance improvement. (3) Among the merging methods, the method of concatenating the representations generally shows the highest improvement in the result. The results show that the merged representation for Korean achieves small but consistent sacreBLEU improvements over the baseline (only use word-level representation) on all parallel corpora.

The proposed method requires up to 10% more parameters and longer training time than the base Transformer model to generate low-level representations, but it can encode richer information than the word-based NMT model.

VI. ANALYSIS AND DISCUSSIONS

A. Ablation Study

Our composition model consists of three different networks: CNN, Highway Network, and LSTM. We perform an ablation study on Korean-to-Chiense translation task (Dong-A dataset) to understand the importance of these networks in producing low-level representations. The baseline model uses word-/character-/jamo-level representations and merge them through the Averaging method. As Table III shows, if we remove Highway Network, the translation performance decreases by 0.4 BLEU point. When LSTM is excluded from the composition model, it results in a significant drop of 0.7 BLEU point. When using only CNN models, translation performance dropped by 0.8 BLEU point. Similar to our model, [17] proposed syllable-based word embedding for Korean using CharCNN, and demonstrated good performance on the word similarity and relatedness task. However, in our preliminary experiment, we found that low-level representations produced using CharCNN only dropped performance on translation tasks. Therefore, we mitigated this issue by adding LSTM to the model.

TABLE III
ABLATION STUDY OF COMPOSITION MODEL ON KOREAN-TO-CHIENSE TRANSLATION TASK (DONG-A DATASET).

Model	SacreBLEU Score
Baseline	42.3
-LSTM	41.6
-Highway Network	41.9
-LSTM&Highway Network	41.5

B. Why use the Composition Model?

We use the composition model to integrate low-level unit representation into the NMT model. It requires the character-/jamo-level representation for each word and each word consists of units of different lengths. Hence, we have adopted a model that can extract information effectively without being constrained by the length of the word. Similar to our model, [28] (in Chinese) proposed a new approach that integrates both word embeddings for Chinese words and Chinese character stroke sequence information into NMT system. However, They average a sequence of Chinese character stroke vectors which are induced from Chinese word without using any specific model and merge with the word embedding. In the preliminary experiment, we found that the method did not work well for the Korean translation task.

C. Is jamo presentation useful in translation task?

Another observation is that jamo-level representation has a relatively lower contribution to performance either individually or in a merged manner than character-level representation. There may be several causes: (1) Jamo information is useful in speech recognition/synthesis task but may not in translation task. (2) All the datasets used in the experiment consist of formal sentences. Jamo representation may show better performance in the Ill-formed sentences commonly seen on the internet (i.e., ㅎ ㅇ is shorthand for 하ㅇ|(hi)) in formal sentences. As a future work, we have a plan to expand our model so that it can cover real-word data that are actually used on the Internet.

VII. CONCLUSION

In this paper, we have presented the merged representations that exploit the hierarchical structure of the Korean language for Korean NMT model. In our experiment, we have demonstrated that the proposed model requires additional parameters, but is competitive and efficient in capturing additional semantic and syntactic information. Furthermore, the merged representations can identify unseen verbs more effectively than

simple word-level representations. Although this proposed method seems only suitable for the Korean language, it could be easily applied to similar scenarios.

REFERENCES

[1] Sepp Hochreiter, Jürgen Schmidhuber.: Long short-term memory. In: Neural Computation, pp. 1735–1780.

[2] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu.: BLEU: a method for automaticevaluation of machine translation.In: Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics (2002), pp. 311–318.

[3] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions, pp. 177–180

[4] Diederik P. Kingma, Jimmy Ba.: Adam: A method for stochastic optimization. In: CoRR, abs/1412.6980.

[5] Thang Luong, Richard Socher, Christopher D Manning.: Better word representations with recursive neural networks for morphology. In: Proceedings of CoNLL, pp. 104–113.

[6] Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, Tie-Yan Liu.: Co-learning of word representations and morpheme representations. In: Proceedings of CoNLL.

[7] Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, Huanbo Luan.: Joint learning of character and word embeddings. In: Proceedings of IJCAI. (2015)

[8] Rupesh Kumar Srivastava, Klaus Greff, and Jurgen Schmidhuber.: Highway networks. arXiv preprint arXiv:1505.00387 (2015).

[9] Yoon Kim, Yacine Jernite, David Sontag, Alexander M. Rush. Character-aware neural language models. In: Proceedings of AAAI, pp. 2741–2749.

[10] Zhen Yang, Wei Chen, Feng Wang, Bo Xu.: A character-aware encoder for neural machine translation. In: Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016), pp. 3063–3070.

[11] Rupesh Kumar Srivastava, Klaus Greff, and Jurgen Schmidhuber.: Highway networks. arXiv preprint arXiv:1505.00387 (2015).

[12] Junyoung Chung, Kyunghyun Cho, Yoshua Bengio.: A character-level decoder without explicit segmentation for neural machine translation. In: Proceedings of Association for Computational Linguistics (2016)

[13] Jason Lee, Kyunghyun Cho, Thomas Hofmann. :Fully character-level neural machine translation without explicit segmentation. arXiv preprint arXiv:1610.03017 (2016)

[14] Austin Matthews, Eva Schlinger, Alon Lavie, Chris Dyer.: Synthesizing compound words for machine translation. In: Proceedings of Association for Computational Linguistics (2016)

[15] Rico Sennrich, Barry Haddow.: Linguistic input features improve neural machine translation. In: Proceedings of WMT (2016)

[16] Rico Sennrich, Barry Haddow, Alexandra Birch.: Neural machine translation of rare words with subword units. In: Proceedings of ACL.

[17] Sanghyuk Choi, Taeuk Kim, Jinseok Seol, Sanggoo Lee. A syllable-based technique for word embeddings of korean words. In: Proceedings of the First Workshop on Subword and Character Level Models in NLP, pp. 36–40.

[18] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, Alexander M. Rush.: OpenNMT: Open-Source Toolkit for Neural Machine Translation. In: Proceedings of Association for Computational Linguistics (2017)

[19] Karl Stratos.: A sub-character architecture for Korean language processing In: the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 721–726. Copenhagen, Denmark.(2017)

[20] Sungjoon Park, Jeongmin Byun, Sion Baek, Yongseok Cho, Alice Oh.: Subword-level Word Vector Representations for Korean. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2429–2438

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, Illia Polosukhin.: Attention is all you need. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), pp. 5998–6008.

[22] Joshua Coates, Danushka Bollegala.: Frustratingly easy meta-embedding—computing meta embeddings by averaging source word embeddings. In: Proceedings of NAACL-HLT. (2018)

[23] Longtu Zhang, Mamoru Komachi.: Neural machine translation of logographic language using subcharacter level information. In: Proceedings of the 3th Conference on Machine Translation: Research Papers, WMT 2018., pp. 17–25.

[24] Longtu Zhang, Mamoru Komachi.: Chinese–Japanese Unsupervised Neural Machine Translation Using Sub-character Level Information. In: Proceedings of the 33rd Pacific Asia Conference on Language, Information and Computation, pp. 309–315.

[25] Shaohui Kuang, Lifeng Han.: Apply Chinese radicals into neural machine translation: Deeper than character level. In: 30Th European Summer School In Logic, Language And Information. (2018)

[26] Matt Post.: A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation: Research Papers, pp. 186–191.

[27] Jeonghyeok Park and Hai Zhao.: Korean-to-Chinese Machine Translation using Chinese Character as Pivot Clue. In: Proceedings of 33rd Pacific Asia Conference on Language, Information and Computation (PACLIC 33), pp. 558–566. Hakodate, Japan.

[28] TAN Xin and KUANG Shaohui and ZHANG Longyin and XIONG Deyi.:Integration of Chinese character stroke sequence into neural machine translation. In: Journal of Xiamen University(Natural Science) (2019), pp. 164–169.