Korean-to-Japanese Neural Machine Translation System using Hanja Information

Hwichan Kim **Tosho Hirasawa** **Mamoru Komachi**
Tokyo Metropolitan University
6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan
{kim-hwichan, hirasawa-tosho}@ed.tmu.ac.jp, komachi@tmu.ac.jp

Abstract

In this paper, we describe our TMU neural machine translation (NMT) system submitted for the Patent task (Korean→Japanese) of the 7th Workshop on Asian Translation (WAT 2020, Nakazawa et al., 2020). We propose a novel method to train a Korean-to-Japanese translation model. Specifically, we focus on the vocabulary overlap of Korean Hanja words and Japanese Kanji words, and propose strategies to leverage Hanja information. Our experiment shows that Hanja information is effective within a specific domain, leading to an improvement in the BLEU scores by +1.09 points compared to the baseline.

1 Introduction

The Japanese and Korean languages have a strong connection with Chinese owing to cultural and historical reasons (Lee and Ramsey, 2011). Many words in Japanese are composed of Chinese characters called Kanji. By contrast, Korean uses the Korean alphabet called Hangul to write sentences in almost all cases. However, Sino-Korean[1] (SK) words, which can be converted into Hanja words, account for 65 percent of the Korean lexicon (Sohn, 2006). Table 1 presents an example of conversions of SK words into Hanja, which are compatible with Japanese Kanji words.

In addition, several studies have suggested that overlapping tokens between the source and target languages can improve the translation accuracy (Sennrich et al., 2016; Zhang and Komachi, 2019). Park and Zhao (2019) trained a Korean-to-Chinese translation model by converting Korean SK words from Hangul into Hanja to increase the vocabulary overlap.

In other words, the meaning of a vocabulary overlap on NMT is that each corresponding word's embeddings are the same. Conneau et al. (2018) and Lample and Conneau (2019) improved translation accuracy by making embeddings between source words and their corresponding target words closer. From this fact, we hypothesize that if the embeddings of each corresponding word are closer, the translation accuracy will improve.

Korean (Hangul)	자연	언어	처리
Korean (Hanja)	自然	言語	処理
Japanese (Kanji)	自然	言語	処理
English	Natural	Language	Processing

Table 1: Example of conversion of SK words in Hangul into Hanja and Japanese translation into Kanji.

Based on this hypothesis, we propose two approaches to train a translation model. First, we follow Park and Zhao (2019)'s method to increase the vocabulary overlap to improve the Korean-to-Japanese translation accuracy. Therefore, we perform Hangul to Hanja conversion pre-processing before training the translation model. Second, we propose another approach to obtain Korean and Japanese embeddings that are closer to Korean SK words and their corresponding Japanese Kanji words. SK words written in Hangul and their counterparts in Japanese Kanji are superficially different, but we make both embeddings close by using a loss function when training the translation model.

In addition, in this study, we used the Japan Patent Office Patent Corpus 2.0, which consists of four domains, namely chemistry (Ch), electricity (El), mechanical engineering (Me), and physics (Ph), whose training, development, and test-n1[2] data have domain information. Our methods are more effective when the terms are derived from Chinese characters; therefore, we expect that the effect will be different per domain. This is because

[1] Sino-Korean (SK) refers to Korean words of Chinese origin.

[2] Here, test-n, test-n1, test-n2, and test-n3 data, and test-n consist of test-n1, test-n2, and test-n3.

Proceedings of the 7th Workshop on Asian Translation, pages 127–134

Korean (Hangul)	그래서	N의	함유량은	0.01 %이하로	한정한다.
Korean (Hanja)	그래서	N의	含有量은	0.01 %以下로	限定한다.
Japanese	そのため	Nの	含有量は	0.01 %以下に	限定する。
English	Therefore,	N	content	0.01 % or less	limit.

Table 2: Korean sentence and sentence in which SK words are converted into Hanja, together with their Japanese and English translations.

there are domains in which there are many terms derived from Chinese characters. Therefore, to examine which Hanja information is the most useful in each domain, we perform a domain adaptation by fine-tuning the model pre-trained by all training data using domain-specific data.

In this study, we examine the effect of Hanja information and domain adaptation in a Korean-to-Japanese translation. The main contributions of this study are as follows:

- We demonstrate that Hanja information is effective for Korean to Japanese translations within a specific domain.
- In addition, our experiment shows that the translation model using Hanja information tends to translate literally.

2 Related Work

Several studies have been conducted on Korean–Japanese neural machine translation (NMT). Park et al. (2019) trained a Korean-to-Japanese translation model using a transformer-based NMT system with relative positioning, back-translation, and multi-source methods. There have been other attempts that combine statistical machine translation (SMT) and NMT (Ehara, 2018; Hyoung-Gyu Lee and Lee, 2015). Previous studies on Korean–Japanese NMT did not use Hanja information, whereas we train a Korean-to-Japanese translation model using data in which SK words were converted into Hanja words.

Sennrich et al. (2016) proposed byte-pair encoding (BPE), i.e., a sub-word segmentation method, and suggested that overlapping tokens by joint BPE is more effective for training the translation model between European language pairs. Zhang and Komachi (2019) increased the overlap of tokens between Japanese and Chinese by decomposing Japanese Kanji and Chinese characters into an ideograph or stroke level to improve the accuracy of Chinese–Japanese unsupervised NMT. Following previous studies, we convert Korean SK words from Hangul into Hanja to increase the vocabulary overlap.

Conneau et al. (2018) proposed a method to build a bilingual dictionary by aligning the source and target word embedding spaces using a rotation matrix. They showed that word-by-word translation using the bilingual dictionary can improve the translation accuracy in low-resource language pairs. Lample and Conneau (2019) improved the translation accuracy by fine-tuning a pre-trained cross-lingual language model (XLM). The authors observed that the bilingual word embeddings in XLM are similar. Based on these facts, we hypothesize that if the embeddings of each corresponding word become closer, the translation accuracy will improve. In this study, we use Hanja information to make the embeddings of each corresponding word closer to each other.

Some studies have focused on exploiting Hanja information. Park and Zhao (2019) focused on the linguistic connection between Korean and Chinese. They used the parallel data in which the Korean sentences of SK words were converted into Hanja to train a translation model and demonstrated that their method improves the translation quality. In addition, they showed that conversion into Hanja helped translate the homophone of SK words. Yoo et al. (2019) proposed an approach of training Korean word representations using the data in which SK words were converted into Hanja words. They demonstrated the effectiveness of the representation learning method on several downstream tasks, such as a news headline generation and sentiment analysis. To train the Korean-to-Japanese translation model, we combine these two approaches using Hanja information for training the translation model and word embeddings to improve the translation quality.

In domain adaptation approaches for NMT, Bapna and Firat (2019); Gu et al. (2019) trained an NMT model pre-trained with massive parallel data and retrained it with small parallel data within the target domain. In addition, Hu et al. (2019)

Model	Partition	Sent.	Korean		Japanese	
			Tokens	Types	Tokens	Types
Baseline	train	999,758	31,151,846	21,936	31,065,360	24,178
	dev	2,000	106,433	6,475	104,307	6,098
	test-n	5,230	272,975	9,530	269,876	9,018
	test-n1	2,000	108,327	6,425	106,947	6,137
	test-n2	3,000	153,195	7,522	151,253	6,656
	test-n3	230	11,453	1,666	11,676	1,644
Hanja-conversion	train	999,755	32,066,032	26,046	30,474,136	27,541
	dev	2,000	109,460	6,944	103,994	6,119
	test-n	5,230	298,404	9,832	269,146	9,166
	test-n1	2,000	111,543	6,941	106,653	6,162
	test-n2	3,000	175,131	6,410	150,844	6,717
	test-n3	230	11,730	1,759	11,649	1,634

Table 3: Statistics of parallel data after each pre-processing.

proposed an unsupervised adaptation method that retrains a pre-trained NMT model using pseudo-in-domain data. In this study, we perform domain adaptation by fine-tuning a pre-trained NMT model with domain-specific data to examine whether Hanja information is useful, and if so, in which domains.

3 NMT with Hanja Information

Our model is based on the transformer architecture (Vaswani et al., 2017), and we share the embedding weights between the encoder input, decoder input, and output to make better use of the vocabulary overlap between Korean and Japanese. We do not use language embedding (Lample and Conneau, 2019) to distinguish the source and target languages. We propose two models using the Hanja information described below.

3.1 NMT with Hanja Conversion

We expect that the translation accuracy will improve by converting Korean SK words into Hanja to increase the source and target vocabulary overlap. In the Hanja-conversion model, we converted SK words written in Hangul into Hanja via pre-processing. Table 2 presents an example of the Hanja conversion. This conversion can increase the number of superficially matching words with Japanese sentences. We trained the translation model after the conversion.

3.2 NMT with Hanja Loss

In the Hanja-loss model, we make the embeddings of the SK word and its corresponding Japanese Kanji word closer to each other. We use a loss function to achieve this goal as follows:

$$L = L_{\mathrm{T}} + \sum_{n=1}^{N} (1 - \mathrm{Sim}(\mathrm{E}(S_n), \mathrm{E}(K_n)) \quad (1)$$

where L is the loss function per-batch and L_{T} is the loss of the transformer architecture. In addition, S and K are the lists of SK words and its corresponding Japanese Kanji words in the batch, respectively, and N is the length of the S and K lists (e.g., when the sentence is the example of Table 2 and the batch size is one, S = (함유량, 이하, 한정), K = (含有量, 以下, 限定) and N = 3.). Here, E is a function that converts words into embedding, and Sim is a cosine similarity function. Therefore, the Hanja-loss function (Equation 1) decreases when the SK word and Japanese Kanji word vectors become more similar.

We extract Kanji words in Japanese sentences to obtain K, and then normalize Kanji into traditional Chinese and convert it into Hangul using a Chinese character into Hangul conversion tool. If the conversion tool cannot convert normalized Kanji into Hangul, we remove the Kanji word from K. To obtain S, we search for the same Hangul words from the parallel Korean sentence. The reason for using the Kanji-to-Hangul conversion is the ambiguity of Hangul-to-Hanja conversion. Conversion of Kanji into Hangul is mostly unique[3]. For example, the SK word "산" can be converted into "山 (mountain)" or "酸 (acid)" in Hanja and Kanji,

[3]Kanji-to-Hangul conversion has certain ambiguity owing to the initial sound rule in the Korean language.

Model		Ch	El	Me	Ph
Baseline	Tokens / Percentage	270,406 / 3.9	86,262 / 1.1	63,837 / 0.6	93,516 / 1.1
	Types / Percentage	5,681 /31.0	5,070 / 29.3	4,202 / 22.8	5,442/ 29.6
Hanja-conversion	Tokens / Percentage	1,697,758 / 24.2	1,525,433 / 20.1	1,678,554 / 17.7	1,530,997 / 19.0
	Types / Percentage	10,803 / 49.5	9,010 / 45.0	8,449 / 39.3	10,165 / 46.8

Table 4: Statistics on vocabulary overlap between Korean and Japanese per-domain training data. The tokens and types are the numbers of overlap words and their types, and percentage is the percentage of their numbers to all data.

respectively, and the SK word into Hanja word conversion has certain ambiguity. By contrast, the Kanji word "山" can be converted uniquely into the SK word "산."

4 Domain Adaptation

We examine the effect of domain adaptation, which uses domain-specific data for retraining the pre-trained model trained by all training data. We translate the test data for each domain using a domain-specific translation model.

For training and validation, we split the training and development data into four domains: chemistry, electricity, mechanical engineering, and physics using domain information. We use these data to build domain-specific translation models.

For testing, we use the domain information annotated with the test-n1 data. However, the test-n2 and test-n3 data do not have domain information. Therefore, we train a domain prediction model by fine-tuning Korean or Japanese BERT (Devlin et al., 2019) using the labeled training data of the Japan Patent Office Patent Corpus 2.0 to predict the domain information of test-n2 and test-n3 data.

5 Experimental Settings

5.1 Implementation

We use the fairseq[4] implementation of the transformer architecture for the baseline model and the Hanja-conversion model and extend the implementation for the Hanja-loss model. Table 7 presents some specific hyperparameters that are used in all models.

To train the domain prediction model for domain adaptation (Section 4), we used the BidirectionalWordPiece tokenizer, character model of KR-BERT[5] (Lee et al., 2020) as the Korean BERT and the bert-base-japanese-whole-word-masking model[6] as the Japanese BERT.

5.2 Data

To train the Korean-to-Japanese translation model, we used the Korean↔Japanese dataset of the Japan Patent Office Patent Corpus 2.0, which consists of training, development, test-n, test-n1, test-n2, and test-n3 data. We apply the following preprocess for each model.

Baseline model We tokenize Korean sentences using MeCab[7] with the mecab-ko dictionary[8], and Japanese sentences with the IPA dictionary. After tokenization, we delete sentences with more than 200 words from the training data and apply shared byte-pair encoding (BPE, Sennrich et al., 2016) with a 30k merge operation size. Table 3 presents the statistics of the pre-processed data.

NMT with Hanja Conversion To train the Hanja-conversion model, we convert South Korean words into Hanja using a Hanja-tagger[9] and normalize Hanja and Kanji in parallel sentences into traditional Chinese using OpenCC[10]. After conversion, we apply the same pre-processing with the baseline model. Table 3 also presents the statistics of pre-processed data[11] for the Hanja-conversion model. In addition, Table 4 presents the statistics on the overlap of tokens between Korean and Japanese per-domain training data.

NMT with Hanja Loss We use the same pre-processed data as the baseline model. To extract the set of SK words and Kanji (Section 3.2) for the Hanja-loss model, we normalize Kanji into traditional Chinese using OpenCC and convert it into Hangul using Hanja[12].

[4]https://github.com/pytorch/fairseq
[5]https://github.com/snunlp/KR-BERT
[6]https://github.com/cl-tohoku/bert-japanese
[7]http://taku910.github.io/mecab/
[8]https://bitbucket.org/eunjeon/mecab-ko-dic/src/master/
[9]https://github.com/kaniblu/hanja-tagger
[10]https://github.com/BYVoid/OpenCC
[11]The number of tokens differs from the baseline because we apply the tokenization after Hanja conversion.
[12]https://pypi.org/project/Hanja/

	Korean BERT				Japanese BERT			
Partition	Ch	El	Me	Ph	Ch	El	Me	Ph
test-n2	645	589	1,131	635	538	782	1,094	586
test-n3	17	43	64	106	19	60	49	102

Table 5: The test-n2 and test-n3 data size of each domain, which are predicted by the domain prediction models (Section 4). Korean BERT and Japanese BERT are the models used to train the domain prediction models.

Model	dev	test–n	test-n1	test-n2	test-n3
Baseline	68.41±0.11	71.84±0.18	72.46±0.09	73.42±0.25	**45.12±0.50**
Hanja-conversion	67.86±0.14	63.70±0.24	71.96±0.23	56.68±0.43	44.60±0.44
Hanja-loss	**68.47±0.07**	**71.96±0.14**	**72.60±0.07**	**73.55±0.22**	44.85±0.50

Table 6: BLEU scores of each single model. These BLEU scores are the average of the four models. The Hanja-loss model achieves the highest scores in the test-n, test-n1, and test-n2 data.

Hyperparameter	Value
Embedding dimension	512
Attention heads	8
Layers	6
Optimizer	Adam
Adam betas	0.9, 0.98
Learning rate	0.0005
Dropout	0.1
Label smoothing	0.1
Max tokens	4,098

Table 7: Hyperparameters.

Domain Adaptation We split the training, development, and test-n1 data using domain information, where the distribution of each domain is equal. We use the domain prediction model to split the test-n2 and test-n3 data. After splitting the data, we apply the same pre-processing as the baseline model. In addition, we use the same BPE model as the baseline model. Table 5 presents the test-n2 and test-3 data sizes of each domain.

5.3 Results

Table 6 presents the BLEU scores of a single model. We indicate the best scores in bold. In a single model, the Hanja-loss model achieves the highest scores for the test-n, test-n1, and test-n2 data. The test-n data reveals an improvement of +0.12 points from the baseline model.

The test-n2 data indicate that the Hanja-conversion model cannot translate well on test-n2 data. The reason is that all words in the Korean sentences of test-n2 data are written without any segmentation, which causes many errors in Hanja conversion.

Table 8 presents the BLEU and RIBES scores of the ensemble of four models and the domain adaptation ensemble models. When we use Japanese BERT to predict the domain, there is a slight improvement in the test-n and test-n2 data when compared with the baseline model[13]. In addition, Table 8 reveals no difference between the baseline and Hanja-loss models in the ensemble models.

Table 9 presents the per-domain dev and test-n1 BLEU scores of each ensemble model. The Hanja-loss model is not better than the baseline model for all data, and there is no difference between the baseline and Hanja-loss models.

6 Discussion

6.1 Hanja-conversion Model versus Hanja-loss Model

In this section, we compare the Hanja-conversion model and the Hanja-loss model. Table 6 indicates that the Hanja-loss model is better than the Hanja-conversion model in terms of the BLEU scores. The reason for this result is that Hangul into Hanja word conversion errors reduce the translation accuracy in the Hanja-conversion model.

6.2 Baseline Model versus Hanja-loss Model

In this section, we compare the baseline model and the Hanja-loss model. Tables 6, 8 and 9 indicate no difference between the baseline and Hanja-loss models in terms of BLEU and RIBES scores.

Table 10 presents the results of human evaluation. These figures are adequacy scores evaluated

[13]However, in Korean-to-Japanese translation, we cannot use Japanese BERT to predict the domain.

	Model	dev	test–n	test-n1	test-n2	test-n3
	Baseline	69.36 / -	73.40 / 0.9504	73.76 / **0.9495**	74.70 / 0.9543	53.70 / **0.9066**
	Hanja-loss	**69.40** / -	73.40 / 0.9504	73.81 / **0.9495**	74.67 / 0.9544	**53.73** / 0.9056
Korean BERT	Baseline	69.20 / -	73.39 / 0.9503	73.85 / 0.9494	74.66 / 0.9540	53.24 / 0.9063
	Hanja-loss	69.26 / -	73.38 / **0.9505**	**73.90** / **0.9495**	74.61 / 0.9545	53.24 / 0.9060
Japanese BERT	Baseline	- /-	**73.45** / 0.9502	- /-	**74.77** / 0.9542	53.23 / 0.9049
	Hanja -loss	- /-	73.41 / **0.9505**	- /-	74.66 / **0.9546**	53.34 / 0.9052

Table 8: BLEU/RIBES scores of each ensemble of four models. The bottom two rows are the scores of the ensemble models retrained by each domain-specific data. Korean BERT and Japanese BERT represent the experimental results using the domain prediction models.

		Ch		El		Me		Ph	
	Model	dev	test-n1	dev	test-n1	dev	test-n1	dev	test-n1
Domain adaptation	Baseline	69.57	**73.92**	**68.45**	75.83	**69.16**	70.17	**71.53**	76.34
	Hanja-loss	**70.08**	72.67	68.26	**76.15**	69.10	**70.25**	71.41	**76.45**
	Baseline	**63.29**	66.67	**62.68**	69.14	65.23	65.83	64.53	68.93
	Hanja-loss	63.23	**66.78**	62.20	**69.15**	**65.29**	**66.06**	**65.27**	**70.12**

Table 9: The per-domain dev and test-n1 BLEU scores of each domain adaptation model and each single model trained by per-domain training data.

Model	Adequacy
Baseline	**4.71**
Hanja-loss	4.70

Table 10: Human evaluation of each domain adaptation ensemble models in test-n data. We use Japanese BERT for domain prediction model.

by WAT 2020 organizers. In human evaluation, the baseline model is better than the Hanja-loss model. However, the improvement in scores is less than +0.01 points. Our experiment using all the training data reveals that there is little difference between the baseline model and the Hanja-loss model.

6.3 Effect of Domain in the Hanja-loss Model

In this section, we examine the effect of the Hanja-loss model in the domain-specific data.

BLEU Scores Table 9 presents the BLEU scores of each model trained by domain-specific data. The Hanja-loss model achieves the highest scores in the Me and Ph domains. Specifically, the test data in the field of physics reveals an improvement of +1.09 points for the baseline model.

By contrast, in the Ch domain, there are no improvements in either the domain adaptation model or the model trained by per-domain training data. In the domain adaptation model of the Hanja-loss model, the test data of Ch indicates a deterioration of -1.25 points for the baseline model. As the reason for this result, we consider that Hanja information is not necessary for the Ch domain because there is more vocabulary overlap than the other domains even without Hanja conversion (Table 4).

Outputs Table 11 presents a successful output example of the Hanja-loss model. The baseline model cannot translate the word "단사 섬도," which means "single yarn fiber," in the source sentence well, but the Hanja-loss model can translate it correctly. We also found that the Hanja-loss model tends to translate literally. In the output of Table 11, the baseline model translates the word "사용, "which means "use," into "用い," whereas the Hanja-loss model translates it into "使用." The word "사용" can translate into both "用い" and "使用," but "使用" is the Hanja form of the SK word "사용."

In Table 12's output, the Hanja-loss model translates the word "개," which means "piece" into "個," which is the Hanja form of the SK word of "개," but the translated word in the reference is "つ." Therefore, in the Hanja-loss model, if the reference sentence is not a literal translation, the BLEU scores are low.

7 Conclusions and Future Work

In this paper, we described our NMT system submitted to the Patent task (Korean→Japanese) of the

Reference	PGA 繊維 として 、例えば 、単 糸 繊度 .... 使用する こと が できる 。
English	As PGA fiber, we can use the single yarn fiber ...
Source	PGA 섬유 로서 , 예 를 들 면 단사 섬도 ... 사용 할 수 있 다 .
Source (Hanja)	PGA 繊維 로서 , 例 를 들 면 単糸 繊度 ... 使用 할 수 있 다 .
Baseline	PGA 繊維 として は 、例えば 、単 粒 度 ... 用い て も よい 。
Hanja-loss	PGA 繊維 として 、例えば 、単 糸 繊度 ... 使用する こと が できる 。

Table 11: A successful output example of the Hanja-loss model.

Reference	一つ の 態様 で は 、R 1 及び R 2 は 1 つ 以上 の 重水素 原子 を 含む 。
English	In one form, R1 and R2 include one or more deuterium atoms.
Source	하나 의 실시 양태 에서 , R 1 및 R 2 는 1 개 이상 의 중수소 원자 를 포함 한다 .
Source (Hanja)	하나 의 実施 様態에서 , R 1 및 R 2 는 1 個 以上 의 重水素 原子 를 包含 한다 .
Baseline	一 実施 態様 で は 、R 1 および R 2 は 1 つ 以上 の 重水素 原子 を 含む 。
Hanja-loss	一 実施 態様 で は 、R 1 および R 2 は 、1 個 以上 の 重水素 原子 を 含む 。

Table 12: An unsuccessful output example of the Hanja-loss model.

7th Workshop on Asian Translation. We proposed novel methods for training the Korean-to-Japanese translation model, which uses Hanja information. We also demonstrated that the effect of our proposed method is different for all domain data.

However, some SK words are polysemous. Our proposed method treats embeddings of such SK words the same and cannot address this problem. Therefore, the problem of polysemous words is a major challenge for our proposed method.

In this study, we focused on vocabulary overlap between Korean Hanja words and Japanese Kanji words. In addition, many Hanja and Kanji words are of Chinese origin. Therefore, in the future, we will attempt to develop a translation method that takes advantage of the vocabulary overlap among Korean, Japanese, and Chinese.

Acknowledgement

This work has been partly supported by the programs of the Grant-in-Aid for Scientific Research from the Japan Society for the Promotion of Science (JSPS KAKENHI) Grant Number 19K12099.

References

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Terumasa Ehara. 2018. SMT reranked NMT(2). In *Proceedings of the 5th Workshop on Asian Translation*.

Shuhao Gu, Yang Feng, and Qun Liu. 2019. Improving domain adaptation translation with domain invariant and specific information. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. 2019. Domain adaptation of neural machine translation by lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Jun-Seok Kim Hyoung-Gyu Lee, Jaesong Lee and Chang-Ki Lee. 2015. NAVER machine translation system for WAT 2015. In *Proceedings of the 2nd Workshop on Asian Translation*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*.

Ki-Moon Lee and S Robert Ramsey. 2011. *A history of the Korean language*. Cambridge University Press.

Sangah Lee, Hansol Jang, Yunmee Baik, Suzi Park, and Hyopil Shin. 2020. KR-BERT: A small-scale Korean-specific language model. *ArXiv*, abs/2008.03979.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. 2020. Overview of the 7th workshop on Asian translation. In *Proceedings of the 7th Workshop on Asian Translation*.

Cheoneum Park, Young-Jun Jung, Kihoon Kim, Geonyeong Kim, Jae-Won Jeon, Seongmin Lee, Junseok Kim, and Changki Lee. 2019. KNU-HYUNDAI's NMT system for scientific paper and patent tasks on WAT 2019. In *Proceedings of the 6th Workshop on Asian Translation*.

Jeonghyeok Park and Hai Zhao. 2019. Korean-to-Chinese machine translation using Chinese character as pivot clue. In *33rd Pacific Asia Conference on Language, Information and Computation*.

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*.

Ho-Min Sohn. 2006. *Korean language in culture and society*. University of Hawaii Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Kang Min Yoo, Taeuk Kim, and Sang-goo Lee. 2019. Don't just scratch the surface: Enhancing word representations for Korean with Hanja. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Longtu Zhang and Mamoru Komachi. 2019. Chinese-Japanese unsupervised neural machine translation using sub-character level information. In *33rd Pacific Asia Conference on Language, Information and Computation*.