An Empirical Study of Tokenization Strategies for Various Korean NLP Tasks

Kyubyong Park[1,*] **Joohong Lee**[2,*] **Seongbo Jang**[2,3,*] **Dawoon Jung**[2,*]
[1]Kakao Brain, [2]Pingpong AI Research, Scatter Lab,
[3]Pohang University of Science and Technology
kyubyong.park@kakaobrain.com
{joohong,dawoon}@scatterlab.co.kr
jang.sb@postech.ac.kr

Abstract

Typically, tokenization is the very first step in most text processing works. As a token serves as an atomic unit that embeds the contextual information of text, how to define a token plays a decisive role in the performance of a model.

Even though Byte Pair Encoding (BPE) has been considered the *de facto* standard tokenization method due to its simplicity and universality, it still remains unclear whether BPE works best across all languages and tasks. In this paper, we test several tokenization strategies in order to answer our primary research question, that is, "What is the best tokenization strategy for Korean NLP tasks?"

Experimental results demonstrate that a hybrid approach of morphological segmentation followed by BPE works best in Korean to/from English machine translation and natural language understanding tasks such as KorNLI, KorSTS, NSMC, and PAWS-X. As an exception, for KorQuAD, the Korean extension of SQuAD, BPE segmentation turns out to be the most effective.

Our code and pre-trained models are publicly available at https://github.com/kakaobrain/kortok.

1 Introduction

Tokenization is the very first step in most text processing works. Not surprisingly, tremendous academic efforts have been made to find the best tokenization method for various NLP tasks. For the past few years, Byte Pair Encoding (BPE) (Gage, 1994) has been considered the *de facto* standard tokenization technique since it was reintroduced by Sennrich et al. (2016a). Besides the fact that BPE turns out to be very effective in the machine translation task, another important reason BPE has gained such popularity is that BPE is a data-driven statistical algorithm so it is independent of language. However, it is still not clear whether BPE works best across all languages, irrespective of tasks.

In this paper we study various tokenization strategies for Korean, a language which is morphologically by far richer than English. Concretely, we empirically examine what is the best tokenization strategy for Korean to English / English to Korean machine translation tasks, and natural language understanding (NLU) tasks—machine reading comprehension (MRC), natural language inference (NLI), semantic textual similarity (STS), sentiment analysis, and paraphrase identification. We are particularly interested in how complementary BPE and linguistically motivated segmentation are.

2 Background

2.1 MeCab-ko: A Korean Morphological Analyzer

MeCab (Kudo, 2006) is an open-source morphological analyzer based on Conditional Random Fields (CRFs). It is originally designed for Japanese, but also serves generic purposes so it can be applied to other languages. MeCab-ko[1], a Korean extension of MeCab, started from the idea that MeCab can be easily extended to the Korean language due to the close similarity between Japanese and Korean in terms of morphology or syntax.

MeCab-ko trained its model on the Sejong Corpus (Kang and Kim, 2001), arguably the largest Korean corpus morphologically annotated by many experts, using MeCab. Ever since released in 2013, MeCab-ko has been widely used for many Korean NLP tasks due to its high accuracy and good usability. For example, the Workshop on Asian Transla-

*Equal contribution.

[1]https://bitbucket.org/eunjeon/mecab-ko

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 133–142
December 4 - 7, 2020. ©2020 Association for Computational Linguistics

Tokenization	Tokenized Sequence
Raw Text	나랑 쇼핑하자.
CV (4.1)	ㄴ/ ㅏ/ㄹ/ ㅏ/ㅇ/⋆/ㅅ/ㅛ/ㅍ/ ㅣ/ㅇ/ㅎ/ ㅏ/ㅈ/ ㅏ/.
Syllable (4.2)	나/랑/⋆/쇼/핑/하/자/.
Morpheme (4.3)	나/랑/⋆/쇼핑/하/자/.
Subword (4.4)	▁나랑/▁쇼/핑하/자/.
Morpheme-aware Subword (4.5)	▁나/▁랑/⋆/▁쇼/핑/▁하/▁자/▁.
Word (4.6)	나랑/쇼핑하자/.

Table 1: An input sentence 나랑 쇼핑하자. 'Let's go shopping with me.' is differently tokenized depending on the various tokenization strategies. Slashes (/) are token separators.

tion (WAT) has adopted it as the official segmentation tool for evaluating Korean machine translation results since 2015. (Nakazawa et al., 2015, 2016, 2017, 2018, 2019).

2.2 Byte Pair Encoding

Byte Pair Encoding (BPE) is a simple data compression technique that iteratively replaces the most frequent pair of bytes in text with a single, unused byte (Gage, 1994). Since Sennrich et al. (2016b) successfully applied it to neural machine translation models, it has been regarded as the standard tokenization method across languages.

Korean is not an exception; Park et al. (2019b) applied BPE to the Korean text in the Korean to Japanese task of WAT 2019 and ranked first. In addition, most recent Korean neural language models (e.g., KoBERT[2]) used BPE to tokenize the training text.

3 Related Work

There have been extensive studies about tokenization techniques for machine translation. Several papers claimed that a hybrid of linguistically informed segmentation and a data-driven method like BPE or unigram language modeling performs the best for non-English languages. Banerjee and Bhattacharyya (2018) combined an off-the-shelf morphological segmenter and BPE in Hindi and Bengali translations against English. Tawfik et al. (2019) used a retrained version of linguistically motivated segmentation model along with statistical segmentation methods for Arabic. Pinnis et al. (2017) adopted linguistic guidance to BPE for English-Latvian translation. Particularly (Park et al., 2019a) is close to ours, but their main focus is on preprocessing techniques for neural machine translation like parallel corpus filtering rather than on tokenization strategies per se.

Compared with the tokenization studies for machine translation, those for NLU tasks have gained less attention. Among them is Bostrom and Durrett (2020), which compared the fine-tuning task performance of BERT (Devlin et al., 2019) pre-trained with BPE and unigram language modeling. Moon and Okazaki (2020) proposed a novel encoding method for Korean and showed its efficiency in vocabulary compression with a few Korean NLU datasets.

4 Tokenization Strategies

We introduce assorted Korean tokenization strategies arranged from the smallest to the largest unit. Each of them induces different tokenization results, as illustrated in Table 1.

4.1 Consonant and Vowel (CV)

In Hangul, the standard Korean writing system, consonants and vowels, called *Jamo* in Korean, corresponding to Latin letters are assembled to form a syllable character. For example, a Hangul consonant ㅎ /h/ (U+314E) is combined with a vowel ㅏ /a/ (U+314F) to make a syllable character 하 /ha/ (U+558). Readers who are not familiar with such a mechanism can think of Jamo and syllables as atoms and molecules respectively. As a molecule $\mathtt{H_2O}$ can be decomposed into two $\mathtt{H}$ atoms and an $\mathtt{O}$ atom, a syllable 하 /ha/ can be decomposed into its constituent consonant ㅎ /h/ and vowel ㅏ /a/. The first syllable 나 /na/ of the raw text in Table 1 is tokenized into ㄴ /n/ and ㅏ /a/, and the second syllable 랑 /lang/ is tokenized into ㄹ /l/, ㅏ /a/, and ㅇ /ng/, and so on. A whitespace is replaced by a special symbol ⋆.

[2]https://github.com/SKTBrain/KoBERT

4.2 Syllable

We can tokenize a sentence at the syllable level. A whitespace is replaced by the special symbol $\star$.

4.3 Morpheme

MeCab-ko provides a convenient tokenization option in the command line interface[3]. For example, it returns `A`, `B`, and `C` given an input text `AB C`, where `A`-`C` represent morphemes. Note that the original space between `AB` and `C` is missing in the output token list. Accordingly, it is NOT possible to recover the original text from the tokenized result.

This can be problematic in some tasks that require us to restore the input text such as machine translation whose target language is Korean, or machine reading comprehension where we are expected to suggest a certain phrase in the given text as the answer.

For this reason, we insert a special token $\star$ (U+2B51) to the original whitespace position. As a result, in the above example, the tokenized sequence looks like `A`, `B`, $\star$, and `D`.

4.4 Subword

We learn and apply BPE using the SentencePiece (Kudo and Richardson, 2018) library. It prepends '▁' (U+2581) to every word to mark the original whitespace, then tokenizes text into subword pieces. As seen in Table 1, 나랑 쇼핑하자. can be split into ▁나랑, ▁쇼, 핑하, 자, and . (period).

4.5 Morpheme-aware Subword

Motivated by the combined methods of data- and linguistically-driven approaches (Banerjee and Bhattacharyya, 2018; Park et al., 2019a; Pinnis et al., 2017; Tawfik et al., 2019), we apply MeCab-ko and BPE in sequence to make morpheme-aware subwords. According to this strategy, since BPE is applied *after* the original text is split into morphemes, tokens spanning multiple morphemes (e.g., 핑하 in the Section 4.4) are not generated. Instead, the BPE algorithm further segments morphemes into frequent pieces.

4.6 Word

We can simply split text by whitespaces. Note that punctuation marks are split into separate tokens. Check that 나랑 쇼핑하자. is tokenized into 나랑, 쇼핑하자 and . (period) in Table 1.

[3]% mecab -O wakati

Lang Pair	Vocab Size	Korean BPE Training Data	Dev	Test
Ko-En	32K	AI Hub (130MB)	35.79	36.06
		Wiki (613MB)	**39.05**	**38.69**
En-Ko	32K	AI Hub (130MB)	**37.19**	**36.98**
		Wiki (613MB)	37.11	**36.98**

Table 2: BLEU scores of Korean to English (Ko-En) and English to Korean (En-Ko) translation models with different BPE training data. Note that the English sentences are tokenized using a 32K BPE model trained on the English Wiki.

5 Experiments

5.1 Korean to/from English Machine Translation

5.1.1 Dataset

To date, there have yet been few open source benchmark datasets for Korean-English machine translation, not to mention that Korean is not in the language list of WMT[4] or IWSLT[5]. Park et al. (2019a) used OpenSubtitles (Lison and Tiedemann, 2016), a collection of crowd-sourced movie subtitles across 65 different languages, for English to Korean translation, but they are too noisy to serve as a translation benchmark dataset.[6]

Recently, a Korean-English parallel corpus was publicly released by AI Hub[7], which was gathered from various sources such as news, government web sites, legal documents, etc. We download the news data, which amount to 800K sentence pairs, and randomly split them into 784K (train), 8K (dev), and 8K (test).

5.1.2 BPE Modeling

Prior to training, we do simple preliminary experiments to decide which dataset to use for learning BPE.

There are two choices: AI Hub news training data and open source large text such as Wiki. AI Hub training data is relatively small in size (130 MB), but can be optimal as its lexical distribution will be close to that of the test data, considering both of them are from the same source. On the other hand, Wiki is larger, but it is not news per se, so can be not as appropriate as AI Hub data for

[4]`https://www.aclweb.org/anthology/venues/wmt`

[5]`http://iwslt.org/doku.php?id=start`

[6]Park et al. (2019a) reported BLEU scores of 7-12.

[7]`http://www.aihub.or.kr/aidata/87`

Tokenization	Vocab Size	Ko-En		En-Ko		OOV Rate (%)	Avg. Length
		Dev	Test	Dev	Test		
CV	166	39.11	38.56	36.52	36.45	0.02	142.75
Syllable	2K	39.30	38.75	38.64	38.45	0.06	69.20
Morpheme	8K	31.59	31.24	32.44	32.19	7.51	49.19
	16K	34.38	33.80	35.74	35.52	4.67	49.19
	32K	36.19	35.74	36.51	36.12	2.72	49.19
	64K	<u>37.88</u>	<u>37.37</u>	<u>37.51</u>	<u>37.03</u>	1.40	49.19
Subword	4K	39.18	38.75	<u>38.31</u>	<u>38.18</u>	0.07	48.02
	8K	39.16	38.75	38.09	37.94	0.08	38.44
	16K	<u>39.22</u>	<u>38.77</u>	37.64	37.34	0.10	33.69
	32K	39.05	38.69	37.11	36.98	0.11	30.21
	64K	37.02	36.46	35.77	35.64	0.12	27.50
Morpheme-aware Subword	4K	39.41	38.95	39.29	39.13	0.06	65.17
	8K	39.42	39.06	39.78	39.61	0.06	56.79
	16K	39.84	39.41	40.23	40.04	0.07	53.30
	32K	**41.00**	**40.34**	**40.43**	**40.41**	0.07	51.38
	64K	39.62	39.34	38.63	38.42	0.07	50.27
Word	64K	7.04	7.07	18.68	18.42	26.20	18.96

Table 3: BLEU scores of Korean to English (**Ko-En**) and English to Korean (**En-Ko**) translation models of various tokenization strategies. Note that we use an 32K Subword model for English for all of them. The OOV rate values in the table are obtained from the test set, but there is no meaningful difference between the test and the dev set in terms of the OOV rate. The best BLEU scores in each column (global) and group (local) are bold-faced and underlined, respectively.

BPE modeling.

To investigate this, first we train a 32K Korean BPE model (**A**) using SentencePiece with the Korean sentences in the AI Hub training data. Then we download the latest Wikipedia Korean[8]/English[9] dumps, and extract plain texts using WikiExtractor [10]. Next, we make 32K BPE models for Korean (**B**) and English (**C**) with them. Finally, we train Korean to English (Ko-En) and English to Korean (En-Ko) translation models on the AI Hub training data with the two different Korean BPE models (**A, B**). The training details are explained in Section 5.1.3. For comparison, we use the same English BPE model (**C**) for both.

The results are shown in Table 2. For Ko-En translation, the Wiki-based BPE model performs better in both dev and test sets by 2-3 points. For En-Ko translation, there is no practical difference in performance between the Wiki and AI Hub-based models. It is also worth considering the BPE models are used for NLU tasks as well as machine translation. All things taken together, we opt for the Wiki-based BPE model.

5.1.3 Training

We test the tokenization strategies in Section 4 with various vocabulary sizes on the AI Hub news dataset.

We use the Transformer (Vaswani et al., 2017), the state-of-the-art model for neural machine translation. We mostly follow the base model configuration: 6 blocks of 512-2048 units with 8 attention heads. We run all of our experiments using FAIRSEQ [11] (Ott et al., 2019), a PyTorch based deep learning library for sequence to sequence models.

Each model is trained using a Tesla V100 GPU with batch size 128, dropout rate 0.3, label smoothing 0.1, and the Adam (Kingma and Ba, 2015) optimizer. We set the learning rate to 5e-4 with the inverse square-root schedule. We train all models for 50 epochs and save the checkpoint files at every epoch.

5.1.4 Results

After all training stages are finished, we evaluate the saved checkpoint files of each model on

[8]https://dumps.wikimedia.org/kowiki

[9]https://dumps.wikimedia.org/enwiki

[10]https://github.com/attardi/wikiextractor

[11]https://github.com/pytorch/fairseq

Vocab Size	# Tokens	# Tokens Spanning Morpheme Boundaries
4K	387,088	25,458 (6.58%)
8K	309,360	50,029 (16.17%)
16K	271,334	62,861 (23.17%)
32K	242,736	73,609 (30.26%)
64K	221,530	82,324 (37.16%)

Table 4: Number of tokens spanning morpheme boundaries in Subword models.

the dev set to find the best one, which is subsequently used for the final test. In Table 3 we report BLEU scores on both the dev and test sets using the Moses[12] `multi-bleu.perl` script. Following WAT 2019 (Nakazawa et al., 2019), Moses tokenizer and MeCab-ko are used for tokenizing the evaluation data.

For both Ko-En and En-Ko, overall, the Subword models (35.64-39.22) and the Syllable models (38.45-39.30) are superior to the Morpheme models (31.59-37.37) or the Word models (7.04-18.42) in performance. It is highly likely to come from the lower OOV rates of the Subword models (0.07-0.12) and the Syllable models (0.06) compared to those of the Morpheme models (1.40-7.51) and the Word models (26.20). While BPE tends to split rare words into subword pieces, MeCab-ko is ignorant of statistics so it splits words into morphemes by linguistic knowledge instead. That the Morpheme and Word models generate many OOVs suggests Korean has so large types of morphemes or word forms that even 64K vocabulary is not enough to cover them all.

CV models are tiny in vocabulary size (166) so they show the lowest OOV rate (0.02). However, their performance is not as good as the Syllable or Subword models. We speculate this is because a single consonant or vowel must bear too much contextual information in the CV models.

Morpheme-aware Subword 32K models achieve the best BLEU scores. Each Subword model, as shown in Table 4, contains 6-37% of tokens spanning morpheme boundaries in the test set, which implies that subword segmentation by BPE is not optimal and morpheme boundaries are meaningful in tokenization.

To sum up, morpheme-aware subword tokenization that makes the best use of linguistic knowledge and statistical information is the best for Korean machine translation.

[12] http://www.statmt.org/moses

Hyper-param	KorQuAD	KorNLI	KorSTS	NSMC	PAWS
Epoch	5	3	5	3	5
Batch	16	64	64	64	64
η	5e-5	1e-4	5e-5	5e-5	1e-4
Dropout	0.1	0.1	0.1	0.1	0.1
Warm-up	0.1	0.1	0.1	0.1	0.1
Max Seq.†	128	128	128	128	128

Table 5: Fine-tuning hyper-parameters for NLU tasks. η: learning rate. †: Max sequence length is 256 for CV models in all tasks.

5.2 Korean Natural Language Understanding Tasks

Large pre-trained language models have proven their effectiveness in many downstream tasks (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019). We pre-train BERT (Devlin et al., 2019) models with various tokenization strategies, and fine-tune them on five different Korean NLU tasks.

5.2.1 Machine Reading Comprehension: KorQuAD 1.0 Dataset

The KorQuAD 1.0 dataset (Lim et al., 2019) is a Korean adaptation of SQuAD 1.0 (Rajpurkar et al., 2016), a popular reading comprehension dataset. KorQuAD 1.0 consists of 10,645 passages and their paired 66,181 questions (60,407 for training + 5,774 for development[13]). Like SQuAD 1.0, KorQuAD 1.0 involves answering a question given a passage. The answer must be a phrase within the passage.

5.2.2 Natural Language Inference: KorNLI Dataset

The KorNLI Dataset (Ham et al., 2020) is a Korean NLI dataset sourced from three different NLI datasets: SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), and XNLI (Conneau et al., 2018).

It is composed of 950,354 sentence pairs: 942,854 for training, 2,490 for development, and 5,010 for test. A model receives a pair of sentences—a premise and a hypothesis—and classifies their relationship into one out of three categories: *entailment*, *contradiction*, and *neutral*.

5.2.3 Semantic Textual Similarity: KorSTS Dataset

The KorSTS Dataset (Ham et al., 2020) is a Korean STS dataset translated from the STS-B dataset (Cer et al., 2017). It comprises 8,628 sentence

[13] The test dataset is not included.

Tokenization	Vocab Size	KorQuAD Dev (EM/F1)	KorNLI Dev	KorNLI Test	KorSTS Dev	KorSTS Test	NSMC Dev	NSMC Test	PAWS-X Dev	PAWS-X Test
CV	166	59.66 / 73.91	70.60	71.20	77.22	71.47	87.97	87.89	58.00	55.20
Syllable	2K	69.10 / 83.29	73.98	73.47	82.70	75.86	88.94	89.07	68.65	67.20
Morpheme	32K	68.05 / 83.82	74.86	74.37	82.37	76.83	87.87	88.04	69.30	67.20
	64K	70.68 / 85.25	75.06	75.69	83.21	77.38	88.72	88.88	73.40	68.65
Subword	4K	71.48 / 83.11	74.38	74.03	83.37	76.80	89.08	89.30	72.00	69.60
	8K	72.91 / 85.11	74.18	74.65	83.23	76.42	89.08	89.19	73.45	69.00
	16K	73.42 / 85.75	74.46	75.15	83.30	76.41	88.89	88.88	73.40	70.70
	32K	**74.04** / 86.30	74.74	74.29	83.02	77.01	89.39	89.38	74.05	70.95
	64K	**74.04** / **86.66**	73.73	74.55	83.52	77.47	88.80	89.19	75.85	72.10
Morpheme-aware Subword	4K	67.53 / 81.93	73.53	73.45	83.34	76.03	88.93	89.32	69.75	67.45
	8K	70.90 / 84.57	74.14	73.95	83.71	76.07	89.37	89.29	73.40	71.30
	16K	69.47 / 83.36	75.02	74.99	83.22	76.59	89.33	89.41	75.05	71.70
	32K	72.65 / 86.35	74.10	75.13	83.65	**78.11**	89.53	89.65	74.60	71.60
	64K	69.48 / 83.73	**76.39**	**76.61**	**84.29**	76.78	**89.82**	**89.66**	**76.15**	**74.00**
Word	64K	1.54 / 8.86	64.06	65.83	69.00	60.41	70.10	70.58	58.25	55.30

Table 6: Performance of various models on several Korean natural language understanding tasks. The evaluation metrics are as follows: KorQuAD: Exact Match/F1, KorNLI: accuracy (%), KorSTS: 100 × Spearman correlation, NSMC: accuracy (%), PAWS-X: accuracy (%). The best scores in each column (global) and group (local) are bold-faced and underlined, respectively.

pairs—5,749 for training, 1,500 for development, and 1,379 for test. The task assesses the gradations of semantic similarity between two sentences with a scale from 0 to 5.

5.2.4 Sentiment Analysis: NSMC Dataset

NSMC[14] is a movie review dataset scraped from Naver Movies™. It consists of 200K samples of which 150K are the training set and the rest 50K are the test set. Each sample is labeled with 0 (negative) or 1 (positive). We hold out 10 percent of the training data for development.

5.2.5 Paraphrase Identification: PAWS-X Dataset

The PAWS-X dataset (Yang et al., 2019) is a challenging paraphrase identification dataset in six languages including Korean. The Korean portion amounts to 53,338 sentence pairs (49,410 for training, 1,965 for development, and 1,972 for test). Like the NSMC dataset, each sentence pair is annotated with either 0 (negative) or 1 (positive).

For each tokenization strategy, we pre-train a BERT-Base model on a large corpus and fine-tune it on the training sets of the five NLU tasks independently.

Pre-training. Because the Korean Wiki corpus is not enough in volume, 640 MB, for the pre-training purpose, we additionally download the recent dump of Namuwiki[15], a Korean Wiki, and extract plain texts using Namu Wiki Extractor[16]. On the resulting Namuwiki corpus (5.5 GB) along with the Wiki corpus (640 MB), pre-training is performed with a Cloud TPU v3-8 for 1M steps using the official BERT training code[17], which is based on TensorFlow. We set the training hyper-parameters of all models as follows: batch size = 1024, max sequence length = 128, optimizer = AdamW (Loshchilov and Hutter, 2019), learning rate = 5e-5, warm-up steps = 10K.

Fine-tuning. After converting each of the pre-trained models in TensorFlow into PyTorch, we fine-tune it using HuggingFace Transformers[18] (Wolf et al., 2019). The hyper-parameters for each task are shown in Table 5.

5.2.6 Results

In Table 6 we report the evaluation results of the various models on the dev and test sets. Since KorQuAD lacks the test set, we report the results on the dev set only.

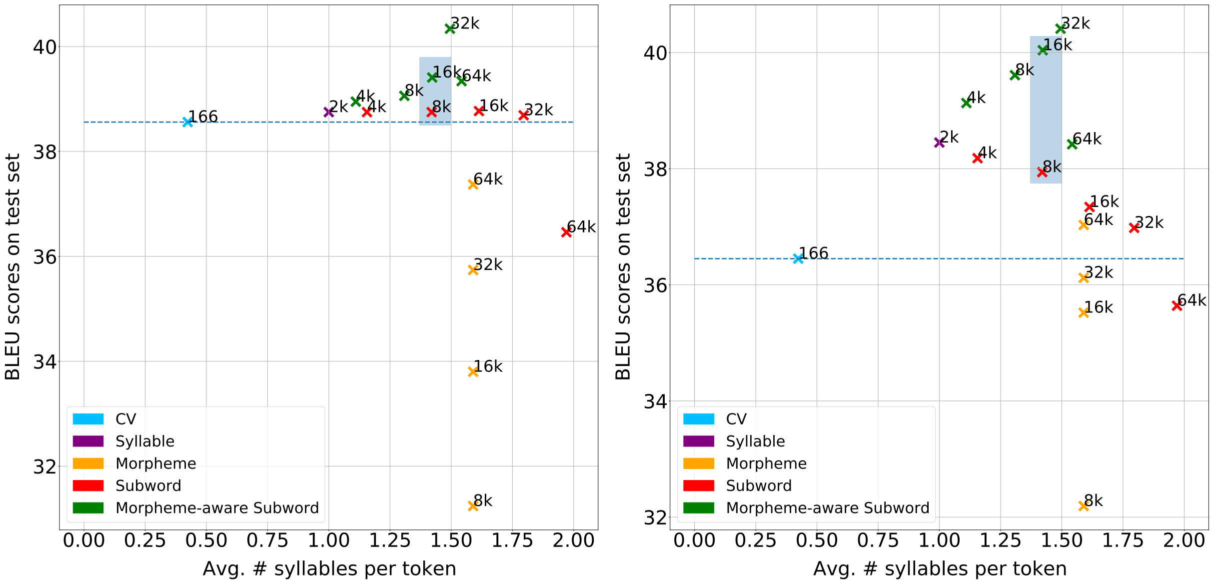
[14] https://github.com/e9t/nsmc

[15] http://dump.thewiki.kr

[16] https://github.com/jonghwanhyeon/namu-wiki-extractor

[17] https://github.com/google-research/bert

[18] https://github.com/huggingface/transformers



(a) Ko-En

(b) En-Ko

Figure 1: Translation performance over the average number of syllables per token

As for KorQuAD, Subword 64K models achieve the highest Exact Match (EM) and F1 scores. The scores in the Subword and Morpheme models increase monotonically as the vocabulary size grows. On the other hand, the 32K models outperform the others in the Morpheme-aware Subword models; no clear correlation is found between performance and vocabulary sizes in them.

For all the other four tasks, Morpheme-aware Subword 64K models show the best scores. One noteworthy phenomenon is that the scores tend to increase as the vocabulary size grows across the tokenization groups. This is discordant with the machine translation results in Section 5.1.4, where a larger vocabulary size does not guarantee better performance for the Subword and Morpheme-aware Subword models.

6 Discussion

We further examine which factors with respect to tokenization affect the Ko-En and En-Ko translation performance.

6.1 Token Length

Because tokenization involves splitting a text into shorter segments, we find it important to figure out how much information each segment bears. To this end, based on the assumption that the longer a text is, the more information it is likely to have, we plot the BLEU scores by the average number of syllables per Korean token in the translation test sets in Figure 1.

The BLEU scores of the subword models—Syllable, Morpheme, Subword, and Morpheme-aware Subword—are mostly higher than those of the CV models, which are plotted as dotted lines. In particular, the Syllable, Subword, and Morpheme-aware Subword models between 1.00 and 1.50 show the best scores both in Ko-En and in En-Ko. When a token has more than 1.5 syllables on average, the scores begin to decrease, and the Word models which has more than 2.5 syllables in a token performs the worst (7.07 for Ko-En and 18.42 for En-Ko). Note that they are not in the figures due to space constraints.

6.2 Linguistic Awareness

Obviously token length is not the only key factor in tokenization strategies. Let us compare the Morpheme-aware Subword 16K models (green markers) and Subword 8K models (red markers) in the shaded regions in Figure 1. Although they have the same average token length around 1.4, the Morpheme-aware Subword models outperform the Subword models. We believe this is evidence to support that linguistic awareness is another important factor in Korean tokenization strategies for machine translation.

6.3 Under-trained Tokens

In section 5.1.4, we pointed out high OOV rates are highly likely to degrade the performance of

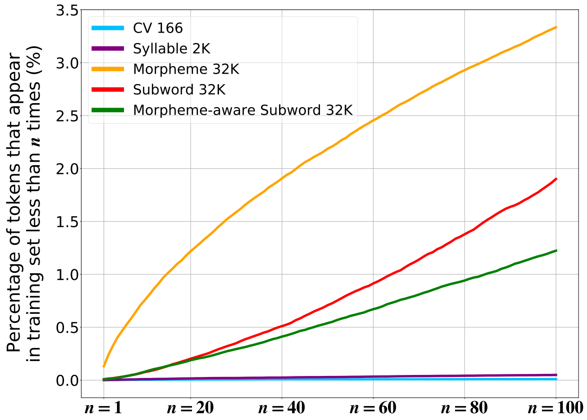


Figure 2: Percentage of under-trained tokens in various tokenization strategies

Morpheme models. It is also worth noting that in Figure 1 as most of the orange markers denoting Morpheme models are below the dotted lines.

OOVs are the tokens that appear only in the test set. They are an extreme case of under-trained tokens—test set's tokens that appear in the training set for the limited number of times. Figure 2 shows how much under-trained tokens account for in each model, ranging from $n = 1$ to $n = 100$, where n is the frequency of the under-trained tokens in the training set. Clearly, the curve of the Morpheme 32K model is far above that of the others, indicating that it suffers from the problem of under-trained tokens the most.

7 Conclusion

We explored various Korean tokenization strategies on machine translation and five NLU tasks. In machine translation Morpheme-aware Subword models with a vocabulary size worked best for both Korean to English and English to Korean settings. By contrast, there was no single best tokenization strategy for the NLU tasks. Instead, Subword 64K models showed the best performance on KorQuAD, whereas Morpheme-aware Subword 64K models turned out to be optimal for the other KorNLI, KorSTS, NSMC, and PAWS-X tasks.

Acknowledgments

We are grateful to the anonymous reviewers for their valuable comments. For pre-training models, we used Cloud TPUs provided by TensorFlow Research Cloud program.

References

Tamali Banerjee and Pushpak Bhattacharyya. 2018. Meaningless yet meaningful: Morphology grounded subword-level NMT. In *Proceedings of the Second Workshop on Subword/Character LEvel Models*, pages 55–60, New Orleans. Association for Computational Linguistics.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. *arXiv preprint arXiv:2004.03720*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Jiyeon Ham, Yo Joong Choe, Kyubyong Park, Ilji Choi, and Hyungjoon Soh. 2020. KorNLI and KorSTS: New benchmark datasets for korean natural language understanding. *arXiv preprint arXiv:2004.03289*.

Beom-mo Kang and Hung-gyu Kim. 2001. 21st century sejong project-compiling korean corpora. In *Proceedings of the 19th International Conference on Computer Processing of Oriental Languages (ICCPOL 2001)*, pages 180–183.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015*.

Taku Kudo. 2006. Mecab: Yet another part-of-speech and morphological analyzer. `https://sourceforge.net/projects/mecab/`.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. KorQuAD 1.0: Korean qa dataset for machine reading comprehension. *arXiv preprint arXiv:1909.07005*.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *7th International Conference on Learning Representations, ICLR 2019*.

Sangwhan Moon and Naoaki Okazaki. 2020. Jamo pair encoding: Subcharacter representation-based extreme Korean vocabulary compression for efficient subword tokenization. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3490–3497, Marseille, France. European Language Resources Association.

Toshiaki Nakazawa, Chenchen Ding, Hideya Mino, Isao Goto, Graham Neubig, and Sadao Kurohashi. 2016. Overview of the 3rd workshop on Asian translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 1–46, Osaka, Japan. The COLING 2016 Organizing Committee.

Toshiaki Nakazawa, Nobushige Doi, Shohei Higashiyama, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. 2019. Overview of the 6th workshop on Asian translation. In *Proceedings of the 6th Workshop on Asian Translation*, pages 1–35, Hong Kong, China. Association for Computational Linguistics.

Toshiaki Nakazawa, Shohei Higashiyama, Chenchen Ding, Hideya Mino, Isao Goto, Hideto Kazawa, Yusuke Oda, Graham Neubig, and Sadao Kurohashi. 2017. Overview of the 4th workshop on Asian translation. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 1–54, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Toshiaki Nakazawa, Hideya Mino, Isao Goto, Graham Neubig, Sadao Kurohashi, and Eiichiro Sumita. 2015. Overview of the 2nd workshop on Asian translation. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 1–28, Kyoto, Japan. Workshop on Asian Translation.

Toshiaki Nakazawa, Katsuhito Sudoh, Shohei Higashiyama, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, and Sadao Kurohashi. 2018. Overview of the 5th workshop on Asian translation. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation: 5th Workshop on Asian Translation: 5th Workshop on Asian Translation*, Hong Kong. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Chanjun Park, Gyeongmin kim, and Heuiseok Lim. 2019a. Parallel corpus filtering and korean-optimized subword tokenization for machine translation. In *Proceedings of the 31st Annual Conference on Human & Cognitive Language Technology*.

Cheoneum Park, Young-Jun Jung, Kihoon Kim, Geonyeong Kim, Jae-Won Jeon, Seongmin Lee, Junseok Kim, and Changki Lee. 2019b. KNU-HYUNDAI's NMT system for scientific paper and patent tasks on WAT 2019. In *Proceedings of the 6th Workshop on Asian Translation*, pages 81–89, Hong Kong, China. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Mārcis Pinnis, Rihards Krišlauks, Daiga Deksne, and Toms Miks. 2017. Neural machine translation for morphologically rich languages with improved subword units and synthetic data. In *International Conference on Text, Speech, and Dialogue*, pages 237–245. Springer.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ahmed Tawfik, Mahitab Emam, Khaled Essam, Robert Nabil, and Hany Hassan. 2019. Morphology-aware word-segmentation in dialectal Arabic adaptation of neural machine translation. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 11–17, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-x: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.