데이터사이언스석사 학위논문

Analysis of Tokenizers for Various Korean NLP Tasks

(한국어 자연어 처리를 위한 토크나이저에 대한 분석)

2023년 2월

서울대학교 대학원

데이터사이언스학과

이 소 람

Analysis of Tokenizers for Various Korean NLP Tasks

(한국어 자연어 처리를 위한 토크나이저에 대한 분석)

지도교수 이 재 진

이 논문을 데이터사이언스석사 학위논문으로 제출함

2022년 12월

서울대학교 대학원

데이터사이언스학과

이 소 람

이 소 람의 데이터사이언스석사 학위논문을 인준함

2032년 1월

위 원 장	이 승 근	(인)
부 위 원 장	이 재 진	(인)
위 원	이 준 석	(인)

Abstract

Although there have been studies on Korean tokenizers intensively, there are not many studies that have reflected the grammatical characteristics of Korean, which is classified as an agglutinative language. In Korean, unlike inflectional languages such as English, where each word segment is a single word, a word segment is not a single word, but a combination of several words or a word and its corresponding grammatical elements. Therefore, it is unreasonable to use tokenizers developed based on English for Korean NLP (Natural Language Processing).

By comparing and analyzing 7 tokenizers (Mecab-ko, BPE, WordPiece, Unigram) that are currently mainly used in Korean NLP, we raise the necessity for a new Korean tokenizer that reflects Korean grammatical characteristics. And we summarize the Korean grammar characteristics that a new tokenizer should reflect.

Key words: Tokenization, Word Segmentation
Student Number: 2021-20066

Contents

Chapter 1

Introduction

Tokenization is a natural language preprocessing task that separates raw text into semantic units to make computers better understand natural language. Tokenization is being studied in various ways, but most studies focus on English. Thus, applying it directly to languages such as Korean, Japanese, and Turkish with a different morphological structure from English is difficult.

Korean	**English**
그는 매일 점심으로 샐러드를 **먹는다.**	He **eats** salad for lunch every day.
그는 어제 저녁에 피자를 **먹었다.**	He **ate** pizza last night.

Table 1.1: Inflected language v.s. agglutinative language

English is morphologically an *inflected language*, in which the form or ending of a word changes according to its grammatical characteristics in sentences. Unlike English, Korean is morphologically an *agglutinative language*, in which words are formed by combining smaller morphemes with different meanings or functions. It is possible to generate hundreds of different forms from a given root word.

For example, as shown in Table 1.1, different meanings of a word in Korean are expressed by adding the present ('-는다') and past tense endings ('-었다') after '먹' that is a root and means to eat. On the other hand, in English, the word 'eat' changes to 'eats' when it is a singular verb and to 'ate' when it is a past tense verb.

In the case of English, the unit of white space is one word, but in the

case of Korean, a tokenization technique is additionally required because a unit of white space is formed by combining suffixes or endings to stems. As the example in Table 1.1, in the case of '점심으로(for lunch)', in English, 'for lunch' can be tokenized to 'for' and 'lunch' as a white space. On the other hand, in the case of Korean, it is necessary to separate even a single word '점심으로' into '점심' and '으로'.

1.1 Purpose of Research

Recently, subword tokenizers that are easy to learn, such as BPE or Unigram, and can be used regardless of a language type, are often used when training Korean models. However, there are few studies on why such tokenizers should be used and whether better performance is guaranteed compared to existing morphological unit tokenization. So, we will analyze how each tokenizer affects the Korean language model by pretraining the GPT-2 model for each tokenizer, which tokenizer is suitable for each task (MRC, sentiment analysis, sentence similarity).

Subword tokenizer has the advantage of being able to adjust the vocab size but has a limit that the result of tokenizing depends on the training data. The morpheme tokenizer reflects the grammatical characteristics of Korean language well, but it cannot be used to encode the tokenization result as it is because the number of morphemes is too large. For example, Mecab-ko provides more than 2 million morphemes (containing 650,000 morphemes consisting only of Korean), so it is not suitable to use all of them for a language model. Finally, we will define what is needed to create a new tokenizer that compensates for the shortcomings of these two tokenizers.

Chapter 2

Background and Related Work

2.1 Tokenizer

2.1.1 BPE

Byte-Pair Encoding is a data compression algorithm, in NLP, it was first used as a tokenizer in Machine Translation[8]. The vocabulary list is created by finding a pair of consecutively most frequent characters and merging them into a single symbol. 'Since the vocabulary list is created based on frequency, it does not reflect the characteristics of the language, and it can be applied to any language.

2.1.2 Unigram

In the case of Unigram, a probability-based language model is trained for tokenizers. By calculating the likelihood loss that occurs when each subword is removed from the corpus, the subwords that have the least influence are removed.

2.1.3 WordPiece

Wordpiece model is a subword tokenizer proposed in [1]. BPE merges based on frequency, whereas wordpiece model creates vocabulary by merging pairs by maximizing the likelihood of the language model.

Sentence : "오늘은 금요일이고 날씨가 정말 좋다." **Vocabulary Size : 32000**	
Mecab	'오늘', '은', '금요일', '이', '고', '날씨', '가', '정말', '좋', '다', '.'
Unigram	'오늘', '은', '금요일', '이고', '날씨', '가', '정말', '좋다', '.'
BPE	'오늘은', '금요일', '이고', '날씨가', '정말', '좋다', '.'
WordPiece	'오늘은', '금요일', '##이고', '날씨가', '정말', '좋다', '.'
Mecab + BPE	'오늘', '은', '금요일', '이', '고', '날씨', '가', '정말', '좋', '다', '.'
Mecab + Unigram	'오늘', '', '은', '금요일', '이', '고', '날씨', '가', '정말', '좋', '다', '', '.'

Table 2.1: Tokenizer 별 문장 tokenizing 예시

2.1.4 Mecab

It is a tokenizer[2] released as an open source in Japan and provides not only tokenizing but also pos(part-of-speech) tagging. Unlike the previous subword tokenizers, it tokenizes sentences in morpheme units. The tokenizer is trained using a conditional random fields (CRFs) model. Eunjeon released mecab-ko by adding a suitable function for Korean language characteristics, and it is the most commonly used among Korean morpheme tokenizers. Unlike the previous three subword tokenizers, the vocabulary size is not limited (basically, the size of the provided vocabulary consisting only of Korean is about 650,000), and the vocabulary list can be managed by adding morphemes and corresponding part-of-speech information.

Table 2.1 is an example of tokenizing when the vocabulary size of the above tokenizers is 32000. In the case of Unigram, it can be seen that the stem and the ending are separated for all words, and in the case of BPE and WordPiece, the tokenizing results are the same.

2.2 GPT-2

GPT-2 [7] is a language model published by OpenAI and consists of only the decoder of the transformer model. It is trained by predicting the next word and is trained using a large dataset that is not oriented to a specific task. If GPT-1 [6] performed fine tuning with supervised learning for specific task data, GPT-2 has the characteristic of being able to perform downstream tasks with zero-shot.

2.3 Related Work

[4] analyzed how tokenizers of various units such as CV(consonant and vowel), syllables, morpheme, subwords, and words affect the Korean NLP task. The combination of subword and morpheme unit tokenizing showed the highest performance except for MRC. In Korean, when the morphological analysis is performed, words are sometimes transformed, such as '반가워요' → '반갑'+'어요'. In [11], the morpheme prototype is restored by learning the Seq2Seq model, and then the segmentation step is performed in units of morphemes.

In [12], to create a rule-based Korean morpheme analyzer, types of Korean word segments were defined and a morpheme analysis rule system suitable for each type was defined. A word segment in Korean is composed of several words, or words and their corresponding grammatical elements. The combination of 'noun' + 'postposition' is the most common, and 'verb' + 'ending', 'adverb' + 'postposition' combinations, etc.

Chapter 3

Experiments

3.1 Dataset

Three datasets are used for training the pretraining model. A total of about 15.7 GB of data set is used, including two dump files for Korean Wikipedia and Namu Wiki, and a crawled data set (7 GB). The sampled data above is also used to learn the subword tokenizer.

As the dataset for the downstream task, the KLUE dataset, and NSMC dataset are used. In the KLUE dataset, the NLI dataset that classifies the relationship between two sentences, the STS dataset that calculates the similarity between two sentences are used, and the Naver Sentiment Movie Corpus (NSMC) dataset is used to analyze the sentiment of movie reviews.

3.2 Tokenizer Training

A total of 7 tokenizers are compared to analyze the impact of tokenizers on each downstream task. Mecab-ko, a morpheme tokenizer, BPE, WordPiece, and Unigram corresponding to subword tokenizers, and Mecab-ko and subword tokenizers mentioned above are used in combination as a morpheme-aware tokenizer in [4].

Huggingface's Tokenizers library is used to train subword tokenizers. It is trained using a 4 GB dataset, and the vocab size is set to 32,000 and 64,000. In the case of the morpheme tokenizer, when tokenizing the training data, vocabularies with the highest frequency of appearance are selected to create and use a vocab list, and tokens that did not belong to the vocab are treated

as unknown tokens. Morpheme-aware tokenizer used subword tokenizer after pre-application of morpheme tokenizer as in [4].

3.3 GPT Pretraining

For each of the seven tokenizers, GPT-2 models are trained. The architecture of the GPT-2 small model is used. The number of layers is 12, the hidden layers of 768 dimensions, and the vocabulary size is 32,000 and 64,000. The hyperparameters of pretraining are as Table 3.1, and the training is conducted using Megatron-LM and 4 NVIDIA RTX 3090 GPUs.

Name	**Value**
Num layer	12
Hidden size	768
Sequence length	512
Learning rate	1e-5
Vocab Size	32000, 64000

Table 3.1: Hyperparameters for GPT Training

3.4 GPT Finetuning

3.4.1 KLUE

KLUE Benchmark is the Korean version of GLUE, which is widely used in English NLP. According to [5], datasets for 8 tasks are included, and among them, in this paper, we use the KLUE-NLI dataset, which classifies the relationship between two sentences into three (entailment, contradiction, neutral), and the KLUE-STS dataset that predicts semantic similarity between two sentences. The similarity is represented by a value between 1 and 5, and fine-tuning is performed to predict the similarity. For evaluation, if the similarity is greater than 3, it is labeled as 1, otherwise, it is labeled as 0.

3.4.2 NSMC

NSMC (Naver Sentiment Movie Corpus) is a dataset that classifies whether a movie review is positive (1) or negative (0). There is a total of 200K data, and there are 100K data for each of the two classes.

Finetuning is performed for each downstream task. NLI measures performance with accuracy as classification, STS measures performance based on f1-score, and NSMC uses accuracy. The hyperparameters for finetuning each task are set in Table 3.2.

	KLUE-STS	**KLUE-STS**	**NSMC**
Num Epochs	5	10	10
Dropout	0.1	0.1	0.1
Learning rate	1e-4	1e-5	1e-5
Sequence Length	512	512	512

Table 3.2: Hyperparameters for fine-tuning

3.5 Results

The performance of downstream tasks for each tokenizer is compared in Table 3.3 and Table 3.4.

In the KLUE-NLI and NSMC tasks, the highest accuracy is obtained when Mecab, a morpheme tokenizer, is used, and in the KLUE-STS task, the highest performance is obtained when Mecab-WordPiece tokenizer is applied. Unigram tokenizer shows the lowest performance in all tasks, and in the case of the KLUE-STS task, the morpheme-aware subword tokenizers show better performance than the subword tokenizers.

Table 3.5 shows the tokenization results and prediction results of the data sample of KLUE-STS, Table 3.6 of KLUE-NLI, and Table 3.7 of NSMC. In Table 3.5, the two sentences have similar meanings. For these two sentences to be predicted similarly, it is important that these tokens, which have similar meanings of '담요(blanket)' and '이불(blanket)', '전혀 (not at all)' and '아예(not at all)', and '안했어요(didn't)' and '않았어요(didn't)', are tokenized

	KLUE-STS (F1-score)	**KLUE-NLI (Accuracy)**	**NSMC (Accuracy)**
BPE	0.7093	45.5667	84.4491
WordPiece	0.6962	46.4333	84.5991
Unigram	0.6745	42.2000	83.2930
Mecab	0.7261	48.0000	86.2692
Mecab+BPE	0.7322	44.3000	84.4151
Mecab+WordPiece	0.7484	45.0333	85.6291
Mecab+Unigram	0.7000	41.9333	83.2830

Table 3.3: Results of Experiments in case of 32000 Vocabulary Size

	KLUE-STS (F1-score)	**KLUE-NLI (Accuracy)**	**NSMC (Accuracy)**
BPE	0.7059	44.8333	85.3165
WordPiece	0.7536	47.7333	86.9451
Unigram	0.7417	45.0333	85.2925
Mecab	0.7484	46.4333	86.8550
Mecab+BPE	0.7111	42.5000	84.8978
Mecab+WordPiece	0.7541	48.7333	87.3137
Mecab+Unigram	0.6532	41.6000	83.9984

Table 3.4: Results of Experiments in case of 64000 Vocabulary Size

well. WordPiece and Mecab, and the combination of Mecab and WordPiece, well tokenized '전혀 (not at all)' and '아예(not at all)', 안('not') and 않('not'), which contain negative meanings.

In Table 3.6, the two sentences have opposite meanings. For these two sentences to be classified as the opposite, 기적('miracle') and 평범('ordinary') are key tokens, so it is important to tokenize them well. In the combination of WordPiece and Mecab and Mecab and WordPiece that correctly predicted, it can be seen that 기적 ('miracle') and 평범 ('ordinary') are included in the token.

The previous two tasks compare two sentences, but in the case of sentiment analysis, the relationship between tokens within a sentence or the meaning of the token itself is important. In Table 3.7, it can be seen that whether the word 후회('regret') with a negative meaning is well tokenized affected the prediction results.

"Label" : 3.4 (1) **"Premise": "진짜 그래서 이불은 사용 아예 안했어요."** **"Hypothesis": "그래서 저는 담요를 전혀 사용하지 않았어요."**		
Tokenizer	**Prediction**	**Tokenized Results**
Vocab Size : 32000		
BPE	**1**	진/짜/그래/서/이/불은/사용/아/예/안/했어요. 그래/서/저는/담/요를/전/혀/사용/하지/않/았/어요.
WordPiece	**1**	진짜/그래서/이/##불은/사용/아예/안/##했어요/. 그래서/저는/담/##요를/전혀/사용하지/않았/##어요/.
Unigram	**0**	진/짜/ /그/래/서/ /이/불/은/ /사/용/ /아/예/ /안/했/어요. 그/래/서/ /저/는/ /담/요/를/ /전/혀/ /사/용/하지/ /않/았/어요.
Mecab	**1**	진짜/그래서/이불/은/사용/아예/안/했/어요/. 그래서/저/는/담요/를/전혀/사용/하/지/않/았/어요/.
Mecab + BPE	**1**	진/짜/그래/서/이/불/은/사용/아/예/안/했/어요/. 그래/서/저/는/담/요/를/전/혀/사용/하/지/않/았/어요/.
Mecab + WordPiece	**1**	진짜/그래서/이/##불/은/사용/아예/안/했/어/##요/. 그래서/저/는/담/##요/를/전혀/사용/하/지/않/았/어/##요/.
Mecab + Unigram	**0**	진/짜/그/래/서/이/불/은/사/용/아/예/안/했/어/요/. 그/래/서/저/는/담/요/를/전/혀/사/용/하/지/않/았/어/요/.
Vocab Size : 64000		
BPE	**1**	진/짜/그래/서/이/불은/사용/아/예/안/했어요. 그래/서/저는/담/요를/전/혀/사용/하지/않/았/어요.
WordPiece	**1**	진짜/그래서/이불/##은/사용/아예/안/##했어요/. 그래서/저는/담/##요를/전혀/사용하지/않았어/##요/.
Unigram	**1**	진/짜/ /그래/서/ /이/불/은/ /사용/ /아/예/ /안/했어요./ 그래/서/ /저/는/ /담/요/를/ /전/혀/ /사용/하지/ /않/았/어요.
Mecab	**1**	진짜/그래서/이불/은/사용/아예/안/했/어요/. 그래서/저/는/담요/를/전혀/사용/하/지/않/았/어요/.
Mecab + BPE	**0**	진/짜/그래/서/이/불/은/사/용/아/예/안/했/어/요/. 그래/서/저/는/담/요/를/전/혀/사/용/하/지/않/았/어/요/.
Mecab + WordPiece	**1**	진짜/그래서/이불/은/사용/아예/안/했/어요/. 그래서/저/는/담요/를/전혀/사용/하/지/않/았/어요/.
Mecab + Unigram	**1**	진짜/그래서/이/불/은/사/용/아/예/안/했/어요/. 그래서/저/는/담/요/를/전/혀/사/용/하/지/않/았/어요/.

Table 3.5: Tokenized Examples of KLUE-STS Task

"Label" : Contradiction (0) **"Premise": "내가 사랑하는 사람이 날 사랑하는 건 기적이다."** **"Hypothesis": "내가 사랑하는 사람이 날 사랑하는 건 평범한 사실이다."**		
Tokenizer	**Prediction**	**Tokenized Results**
Vocab Size : 32000		
BPE	**1**	내가/사랑/하는/사람이/날/사랑/하는/건/기/적이다. 내가/사랑/하는/사람이/날/사랑/하는/건/평/범한/사실/이다.
WordPiece	**0**	내가/사랑하는/사람이/날/사랑하는/건/기적/##이다/. 내가/사랑하는/사람이/날/사랑하는/건/평범한/사실이다/.
Unigram	**2**	내/가/ /사/랑/하는/ /사/람/이/ /날/ /사/랑/하는/ /건/ /기/적/이다./ 내/가/ /사/랑/하는/ /사/람/이/ /날/ /사/랑/하는/ /건/ /평/범/한/ /사/실/이다.
Mecab	**0**	내/가/사랑/하/는/사람/이/날/사랑/하/는/건/기적/이/다/. 내/가/사랑/하/는/사람/이/날/사랑/하/는/건/평범/한/사실/이/다/.
Mecab + BPE	**2**	내/가/사랑/하/는/사람/이/날/사랑/하/는/건/기/적/이/다/. 내/가/사랑/하/는/사람/이/날/사랑/하/는/건/평/범/한/사실/이/다/.
Mecab + WordPiece	**0**	내/가/사랑/하/는/사람/이/날/사랑/하/는/건/기적/이/다/. 내/가/사랑/하/는/사람/이/날/사랑/하/는/건/평범/한/사실/이/다/.
Mecab + Unigram	**2**	내/가/사/랑/하/는/사/람/이/날/사/랑/하/는/건/기/적/이/다/. 내/가/사/랑/하/는/사/람/이/날/사/랑/하/는/건/평/범/한/사/실/이/다/.
Vocab Size : 64000		
BPE	**2**	내가/사랑/하는/사람이/날/사랑/하는/건/기적/이다. 내가/사랑/하는/사람이/날/사랑/하는/건/평/범한/사실/이다.
WordPiece	**0**	내가/사랑하는/사람이/날/사랑하는/건/기적/##이다/. 내가/사랑하는/사람이/날/사랑하는/건/평범한/사실이다/.
Unigram	**2**	내가/ /사랑/하는/ /사람/이/ /날/ /사랑/하는/ /건/ /기/적이다./ 내가/ /사랑/하는/ /사람/이/ /날/ /사랑/하는/ /건/ /평/범/한/ /사실/이다.
Mecab	**0**	내/가/사랑/하/는/사람/이/날/사랑/하/는/건/기적/이/다/. 내/가/사랑/하/는/사람/이/날/사랑/하/는/건/평범/한/사실/이/다/.
Mecab + BPE	**2**	내/가/사랑/하/는/사람/이/날/사랑/하/는/건/기/적/이/다/. 내/가/사랑/하/는/사람/이/날/사랑/하/는/건/평/범/한/사/실/이/다/.
Mecab + WordPiece	**0**	내/가/사랑/하/는/사람/이/날/사랑/하/는/건/기적/이/다/. 내/가/사랑/하/는/사람/이/날/사랑/하/는/건/평범/한/사실/이/다/.
Mecab + Unigram	**0**	내/가/사랑/하/는/사람/이/날/사랑/하/는/건/기/적/이/다/. 내/가/사랑/하/는/사람/이/날/사랑/하/는/건/평/범/한/사/실/이/다/.

Table 3.6: Tokenized Examples of KLUE-NLI Task

"Label" : 0 (Negative) **"Text": "오랜만에 영화본걸 후회하게만든 영화**		
Tokenizer	**Prediction**	**Tokenized Results**
Vocab Size : 32000		
BPE	**1**	오/랜/만에/영화/본/걸/후/회/하게/만/든/영화
WordPiece	**0**	오랜만에/영화/##본/##걸/후회/##하게/##만/##든/영화
Unigram	**1**	오/랜/만에/ /영화/본/걸/ /후/회/하게/만/든/ /영화
Mecab	**0**	오랜만/에/영화/본/걸/후회/하/게/만든/영화
Mecab + BPE	**1**	오/랜/만/에/영화/본/걸/후/회/하/게/만/든/영화
Mecab + WordPiece	**1**	오랜/##만/에/영화/본/걸/후회/하/게/만든/영화
Mecab + Unigram	**1**	오/랜/만/에/영화/본/걸/후/회/하/게/만/든/영화
Vocab Size : 64000		
BPE	**1**	오/랜/만에/영화/본/걸/후/회/하게/만/든/영화
WordPiece	**0**	오랜만에/영화/##본/##걸/후회/##하게/##만/##든/영화
Unigram	**1**	오랜만에/ /영화/본/걸/ /후/회/하게/만/든/ /영화
Mecab	**0**	오랜만/에/영화/본/걸/후회/하/게/만든/영화
Mecab + BPE	**1**	오/랜/만/에/영화/본/걸/후/회/하/게/만/든/영화
Mecab + WordPiece	**1**	오랜만/에/영화/본/걸/후회/하/게/만든/영화
Mecab + Unigram	**1**	오/랜/만/에/영화/본/걸/후/회/하/게/만/든/영화

Table 3.7: Tokenized Examples of NSMC Task

Tokenizer	**Text : 동대구 · 김천구미 · 신경주역에서**	**Text : 동대구역, 김천구미역, 신주역을**
BPE	동/대구/ · 김/천/구미/ · /신경/주/역에서	동/대/구역/,/김/천/구미/역,/신/주/역을
WordPiece	동/##대구/ · /김천/##구미/ · /신경/##주/##역에서	동/##대구/##역/,/김천/##구미/##역/,/신주/##역을
Unigram	동/대/구/ · /김/천/구/미/ · /신경/주/역/에서	동/대/구역/,/ /김/천/구/미/역/,/ /신/주/역을
Mecab	동대구/ · /김천구/미/ · /신경주역/에서	동대구역/,/김천구/미역/,/신/주역
Mecab + BPE	동/대구/ · /김/천/구/미/ · /신경/주/역/에서	동/대/구역/,/김/천/구/미/역/,/신/주/역/을
Mecab + WordPiece	동/##대구/ · /김천/##구/미/ · /신경/##주/##역/에서	동/##대구/##역/,/김천/##구/미역/,/신/주역/을
Mecab + Unigram	동/대/구/ · /김/천/구/미/ · /신경/주/역/에서	동/대/구역/,/김/천/구/미/역/,/신/주/역/을

Table 3.8: Tokenized Examples of Sentences Containing Proper Nouns

Chapter 4

Conclusion

As a result of comparing experiments of various types of tokenizers, we observe that Mecab, a morpheme tokenizer that reflects the grammatical characteristics of Korean, shows generally high performance for all tasks. In the case of subword tokenizers, they can not separate postpositions or endings in Korean well. However, even in the case of Mecab, as shown in Table 3.8, tokenizing does not work well when proper nouns or new words are included in the sentence.

It seems natural to expect that there would be a suitable tokenizer for each downstream task. However, we can check that tokenizers show similar performance with respect to types of tasks. A new Korean tokenizer has to separate things like postpositions and endings well and create a vocabulary list by adjusting the vocabulary size well so that one word is not tokenized into syllables.

4.1 Analysis

We analyzed existing tokenizers from two perspectives and derived limitations.

4.1.1 Grammatical

As Table 3.3, Table 3.4, the morpheme tokenizer generally shows good performance compared to other tokenizers. Especially when the vocabulary size is 32,000, better performance is shown when the subword tokenizer is not

Morpheme	**욕실/도/무척/청결/합니다/.**
Vocab Size : 32000	
WordPiece	욕/##실/##도/무척/청/##결합/##니다/.
Mecab + WordPiece	욕/##실/도/무척/청/##결/합니다/.
Vocab Size : 64000	
WordPiece	욕/##실도/무척/청/##결합/##니다/.
Mecab + WordPiece	욕실/도/무척/청결/합니다/.

Table 4.1: Examples of Tokenized Sentence

used alone. This supports the need for tokenizers to reflect grammatical features. When comparing the result of the morpheme tokenizer with subword tokenizers Table 4.1, grammatical elements such as postpositions and endings could not be properly separated. The words that should not be separated are tokenized, and tokens that are not related to the meaning of the sentence are tokenized. To reduce the division of words into syllable units (characters in the case of English), we experimented by increasing the vocabulary size to 64,000. When only the subword tokenizer is used, '##실' and '도' are combined instead of '욕' and '##실'. On the other hand, when used with Mecab, tokenization is correctly performed in units of morpheme.

4.1.2 Systemmatical

In the case of the Mecab tokenizer, it has the advantage that it can be used without training. However, since the provided vocabulary size is too large, it is necessary to select the vocabulary list to use. In addition, words such as neologisms and proper nouns require additional training.

In order to overcome the limitations of the two points of view, the new Korean tokenizer needs to consider the following things:

- Separate grammatical elements using the morpheme tokenizer
- Define the finite list of grammatical elements to manage the vocabulary size
  - Replace elements with similar meanings to one representative element
- Train a new tokenizer with the sentences where grammatical elements are removed

Bibliography

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, June 2019. Association for Computational Linguistics.

[2] T. Kudo. Mecab : Yet another part-of-speech and morphological analyzer. *http://mecab.sourceforge.jp*, 2006.

[3] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[4] Kyubyong Park, Joohong Lee, Seongbo Jang, and Dawoon Jung. An empirical study of tokenization strategies for various Korean NLP tasks. In *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, Suzhou, China, December 2020. Association for Computational Linguistics.

[5] Sungjoon Park, Jihyung Moon, Sungdong Kim, Won-Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Tae Hwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong, Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Eun-

jeong Lucy Park, Alice Oh, Jung-Woo Ha, and Kyunghyun Cho. KLUE: korean language understanding evaluation. *CoRR*, abs/2105.09680, 2021.

[6] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[7] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[8] Rico Sennrich, Barry Haddow, and Alexandra Birch. 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, August 2016. Association for Computational Linguistics.

[9] Jooyoul Lee Seungyoung Lim, Myungji Kim. Korquad: Korean qa dataset for machine comprehension. *The Korean Institute of Information Scientists and Engineers*, pages 539–541, 2018.

[10] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

[11] Jun Young Youn and Jae Sung Lee. A deep learning-based two-steps pipeline model for korean morphological analysis and part-of-speech tagging. *Journal of KIISE*, 48(4):444–452, 2021.

[12] Lee Youngmyn. Grammatical strategy to develop a korean morpheme analyser. *Journal of Korean Language Education*, 2017.

국문초록

한국어 토크나이저에 관한 연구는 계속되어 왔지만, 한국어의 문법적 특성을 반영한 토크나이저에 대한 연구는 많지 않다. 한국어는 하나의 어절이 여러 개의 단어 혹은 단어와 그에 대응되는 문법적 요소들로 구성된 교착어의 특징을 가지고 있다. 이는 하나의 어절이 하나의 단어로 구성되어 있고, 굴절어의 특징을 갖고 있는 영어와는 다르기 때문에 영어를 기반으로 개발된 토크나이저들을 한국어 자연어 처리에 사용하는 것은 적합하지 않다. 본 논문에서는 한국어 자연어 처리에서 주로 사용되는 7개의 토크나이저 (Mecab-ko, BPE, WordPiece, Unigram) 들을 비교하고 분석한다. 분석 결과를 바탕으로 한국어의 문법적 특성을 반영한 새로운 토크나이저의 필요성을 제안하고, 해당 토크나이저가 반영해야 할 요소들에 대하여 정리하였다.

주요어휘: 자연어처리, 토크나이저, 형태소분석기
학번: 2021-20066