The Korean Morphologically Tight-Fitting Tokenizer for Noisy User-Generated Texts

Sangah Lee and **Hyopil Shin**
Graduate School of Data Science
Seoul National University
{visualjan, hpshin}@snu.ac.kr

Abstract

User-generated texts include various types of stylistic properties, or noises. Such texts are not properly processed by existing morpheme analyzers or language models based on formal texts such as encyclopedias or news articles. In this paper, we propose a simple morphologically tight-fitting tokenizer (K-MT) that can better process proper nouns, coinages, and internet slang among other types of noise in Korean user-generated texts. We tested our tokenizer by performing classification tasks on Korean user-generated movie reviews and hate speech datasets, and the Korean Named Entity Recognition dataset. Through our tests, we found that K-MT is better fit to process internet slangs, proper nouns, and coinages, compared to a morpheme analyzer and a character-level WordPiece tokenizer.

1 Introduction

User-generated texts such as internet forums, product reviews, blog posts, tweets, and comments include various types of noise coming from their stylistic properties. Noise such as internet slang, spelling errors, and emojis may not be properly processed by traditional morpheme analyzers and language models, since such models are trained on formal texts such as books, encyclopedias, or news articles. User-generated texts that include such properties are useful to analyze public opinion on a variety of topics, products, policies, and so on.

Noise in texts may vary in their forms with different languages, even though they are common to some degree. For example, emphasized opinion can be written capitalized in English, while many exclamation marks are used in Korean as in an example: 하지마hacima!!!!!!!! “DON’T DO IT!”. Such noise has to be processed, rather than removed, depending on the purpose.

Accordingly, we analyzed various Korean user-generated texts and obtained an unexhaustive list of noise types observed in such texts: spacing errors, spelling errors, grammatical errors, special characters and emojis, Korean grapheme characters, foreign language characters, newly-listed proper nouns, coinage, and internet slang. Korean grapheme characters such as ‘ㅋㅋ,’ ‘ㅠㅠ’ are often used and function as emojis.

In this paper, we focus on proper nouns, coinages, and internet slang among other types of noise from user-generated texts. We propose a Korean Morphologically Tight-Fitting Tokenizer, K-MT, which is expected to be robust in processing such types of noises. We compare the performance of our K-MT with a Korean morpheme analyzer and a character-based WordPiece tokenizer (Wu et al., 2016) through experiments on Korean movie reviews, a hate speech dataset, and a named entity recognition dataset.

Our major contributions are:

- A noise-friendly tokenizer using Korean morphological knowledge
- A simple algorithm using a dictionary of grammatical morphemes and a rule-based method

2 Related Work

2.1 Studies on Noisy Texts

Noises can be defined differently according to tasks, types of datasets, domains, and purposes. Michel and Neubig (2018) provided types of noises observed in social media texts and focused on spelling and grammatical errors, emojis, and profanities to perform machine translation. Sjöblom et al. (2018) defined noises for subtitles including misspellings, misalignments, and sentence segmentation errors. Agarwal et al. (2020) summarized noise types in free-text answers written by Hindi children, such as punctuations, emojis, translated/transliterated text, missing space between words, and so on. Some other works focused on orthographical noises (Karpukhin et al., 2019; Kumar et al., 2020).

In the case of Korean, there has not been an explicit trend of research on Korean noisy texts. Han et al. (2019) performed a bias classification task on user-generated news comments, while they did not consider noisy linguistic features in the text. However, several user-generated corpora have recently been released, including Naver Sentiment Movie Corpus (NSMC)[1], Hate Speech Dataset (Moon et al., 2020), and various corpora released by the National Institute of Korean Language. In addition, some BERT-based models were trained on such corpora (Lee, 2020; Lee and Shin, 2021), whose vocabularies include domain-specific tokens and stylistic features such as internet slangs and emojis, different from tokens in formal texts like books or news articles.

2.2 Korean Tokenization

For Korean text processing, morpheme-level tokenization is commonly used because a morpheme is the smallest meaningful unit in Korean (Park et al., 2018, 2019). And following BERT (Devlin et al., 2019), character-based subword tokens of the WordPiece model have been popular in Korean (KoBERT[2], KorBERT[3], KR-BERT (Lee et al., 2020)). However, such subword tokens are too fine-grained and sometimes do not properly reflect meanings and properties of the Korean language (Lee et al., 2020). To alleviate this issue in noisy texts and in former works, sub-character tokens smaller than syllables are considered (Lee et al., 2018, 2020; Moon and Okazaki, 2020), since Korean morphemes' boundaries are not only between syllables but also graphemes.

Morpheme-based WordPiece tokenizers are also used and show better performance than character-based BERT models (HanBERT[4], KorBERT). Park et al. (2020) reported that morpheme-aware subword tokenization shows the best performance among various units of tokens. However, it requires the addition of an external morpheme analyzer before the application of language models or BERT tokenizers. Therefore, we attempt to imitate the morpheme-based tokenization with our morpheme-like tokenizer, which is simply designed based on grammatical morphemes and their combination rules, while considering stylistic properties and noises of user-generated texts.

3 Korean Morphologically Tight-fitting Tokenizer (K-MT)

As an agglutinative language, Korean words consist of combinations of root and affix to represent each word's grammatical function in a sentence. For example, a root word 소녀 sonye "girl" can be combined with various suffixes: 소녀 sonye 가 ka "girl (subj)" and 소녀 sonye 를 lul "girl (obj)." Each of the root and affix is a morpheme, an important basic unit representing meanings in Korean, making Korean a morphologically rich language. Therefore, morpheme-based tokens better reflect meanings and obtain better performances than other tokenization units in general. However, as morpheme analysis of texts requires extra preprocessing using external morpheme taggers, we propose our Korean Morphologically Tight-Fitting Tokenizer (K-MT) with a much simpler rule-based algorithm and coarse-grained token boundaries.

For each word segment split by whitespace, we separate suffixes by matching with a pre-defined list of grammatical morphemes in a backward direction. After all possible suffixes are separated, we obtain a root and use it as an individual token. The pre-defined suffix list consists of frequent connectives and sentence-final endings used more than 100 times extracted from Kkma morpheme analyzer[5] and postpositional particles and pre-final endings used in Mecab-ko morpheme analyzer[6]. We additionally include ending forms used in online, user-generated texts such as -여 ye and -용 yong which are originally -요 yo. The list of grammatical morphemes used in our tokenizer can be found in Appendix A.

# of Suffixes	Suffix Combination
2	었ess/았ass + *E*
3	(으u)시si + 었ess + *E*
	었ess/았ass + 겠keyss + E
4	(으u)시si + 었ess + 겠keyss + *E*

Table 1: Suffix Combination Rules (Choi, 2012)

We separate such suffixes not only by simple string matching but also using the suffix combination rules constructed referring to Choi (2012),

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

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

[3] https://aiopen.etri.re.kr/

[4] https://github.com/tbai2019/HanBert-54k-N

[5] http://kkma.snu.ac.kr/

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

as in Table 1. Choi (2012) statistically patternized grammatical morphemes that are frequently combined. In Table 1, *E* represents all the connectives and final endings in the pre-defined list and the other morphemes like 었ess are pre-final endings. While we preferentially match longer morphemes by default, we prioritize the morpheme sequences matched with the suffix combination patterns in Table 1.

According to the algorithm, our K-MT tokenizes an example sentence 나는 학교에 간다. “I go to school.” as below. This tokenized result is the same as morpheme-based tokenization.

나 는 학교 에 간다 .
na nun hakkyo ey kanta .
I (subj) school to go .

Because we regard a word segment’s root token as the remaining part of the word other than the suffix, any rare, new-listed words such as domain-specific words and coinages are not incorrectly divided into subwords. Moreover, while WordPiece tokens can include a whole word segment like 소녀가 sonyeka “girl (subj)” depending on frequency, our tokenizer always separates and preserves the form of root 소녀 sonye “girl” based on morphological knowledge. Compared to including all related forms derived from one root token, this method can make the composition of the vocabulary more diverse while maintaining the vocabulary size.

4 Experiments and Analysis

To evaluate our K-MT, we compare the tokenized results with those of Mecab-ko, a Korean morpheme analyzer, and a character-based subword tokenizer, WordPiece. Mecab-ko is an upper bound model in this paper, and our goal is to construct a simpler tokenizer imitating the elaborated morpheme analyzer as possible.

First, we selected two popular noisy, user-generated datasets: Naver Sentiment Movie Corpus (NSMC) and Hate Speech Dataset (HSD). Both datasets include most of the noise types we briefly mentioned in section 1. NSMC includes binary sentiment polarity labels, positive and negative, on each film review and consists of 13K, 2K, and 5K reviews for training, validation, and test set. HSD consists of 7,896, 471, and 974 news comments for training, validation, and test set, respectively. We performed two different tasks on HSD: bias classification (whether a comment includes ‘gender’ bias, ‘other’ types of bias, or ‘none’) and hate speech detection (whether a comment includes ‘hate’ speech, less hateful but ‘offensive’ speech, or ‘none’).

For the three tasks, we applied a simple 3-layer CNN model. We limited the size of vocabulary for all the tokenizers to 25,000 and used a batch size of 64, an embedding dimension of 128, 128 convolutional filters, and a dropout of 0.2. We used an Adam optimizer and Cross-Entropy loss and trained the model for 10 epochs. Table 2 shows the classification accuracies of the three tokenizers on the tasks. Since the labels for the HSD test set are not publicly released, so we report the performance on the validation set only.

Tokenizer	**NSMC valid**	**NSMC test**	**HSD hate**	**HSD bias**
Mecab-ko **(upper bound)**	**84.83**	**84.74**	**56.69**	**76.86**
WordPiece	83.62	83.42	44.37	74.10
K-MT (ours)	82.41	81.89	51.38	68.58

Table 2: Performances on User-Generated Text Datasets

The morpheme-level tokens obtained from Mecab-ko would contribute to the highest accuracies, since morphemes reflect the properties and meanings of the Korean language the best. WordPiece tokenizer obtained slightly lower accuracies than Mecab-ko, and our K-MT closely follows WordPiece, reporting accuracies of 82.41 and 81.89 for the validation set and the test set, respectively.

In the two tasks on the HSD dataset, the three tokenizers show larger differences in performance, where Mecab-ko, the upper bound model, reports the best accuracies. While the WordPiece tokenizer generally obtained higher accuracies than K-MT, K-MT performed better than WordPiece in hate speech detection.

In addition, we performed a Named Entity Recognition task with the three tokenizers above. We used the publicly released training set[7] from the Naver NLP Challenge 2018, dividing it into 76,949, 4,050, and 8,999 sentences for each of the training set, validation set, and test set. The named entity labels include 14 types of named entities: person, field, organization, location, artificial objects, cultural terms, date, time, number, event, animal, plant, material, other terms, and none.

The NER dataset includes various named entity words and rarely includes other types of noises like

[7] http://air.changwon.ac.kr/?pageid=10

spacing errors. We included this experiment since our K-MT is expected to work well in separating and preserving proper nouns and named entities which are likely to be newly-encountered domain-specific words in texts.

We applied a simple 2-layer RNN model. We used a batch size of 64, an embedding dimension of 100, a hidden dimension of 128, and a dropout of 0.25. We used an Adam optimizer and Cross-Entropy loss and trained the model for 10 epochs. Table 3 shows the classification accuracies of the three tokenizers on the task.

Tokenizer	25K tokens valid	25K tokens test	all tokens valid	all tokens test
Mecab-ko (upper bound)	67.41	67.61	70.36	70.64
WordPiece	**67.57**	**68.02**	-	-
K-MT (ours)	63.02	63.37	**72.78**	**73.05**

Table 3: Performances on the NER Dataset

When we first limited the size of vocabulary for all the tokenizers as 25,000, we observed the best accuracies of 67.57 and 68.02 for validation and test set with WordPiece tokenization, and similar performance with Mecab-ko. Our K-MT had marginally lower accuracies of 63.02 and 63.37 for the validation and the test set, respectively.

On the other hand, we used unlimited tokens for the models' vocabulary with Mecab-ko and K-MT to remove the effect of the out-of-vocabulary problem. Then we obtained higher accuracies; especially, K-MT reported a validation accuracy of 72.78 and a test accuracy of 73.05, which are higher than Mecab-ko. This indicates the potential of our tokenizer to be robust to newly-listed terms, proper nouns, and named entities in user-generated texts.

To analyze the performance of K-MT, we took 500 random samples from each of the NSMC, HSD, and NER datasets for tokenization. First, we quantitively compared the tokenization of the different tokenizers. Table 4 represents the ratio of samples that were tokenized identically by all the tokenizers. In general, the agreement between tokenizations is very low, which represents the how differently each token splits the sentences. However, our K-MT and Mecab-ko agree in 3.2% of the example tokenizations, which is higher than other tokenizer pairs. This indicates that our purpose of imitating the morpheme-based tokenizer by K-MT is effective to some degree.

Table 5 shows the average length of tokenized sentences (the number of tokens) of each tokenizer.

Tokenizer Pair	Agreement Ratio
Mecab-ko & WordPiece	1.0%
Mecab-ko & K-MT	**3.2%**
WordPiece & K-MT	1.0%
overall	0.73%

Table 4: Agreement Ratio of Tokenization

Here, the tokenized sentences of K-MT had the shortest average sentence length, with the average number of tokens being 53.094. The short length indicates that K-MT avoids oversegmenting words, possibly because its tokens may be longer than that of other tokenizers.

Tokenizer	Average Number of Tokens
Mecab-ko	56.997
WordPiece	65.447
K-MT	53.094

Table 5: Average Number of Tokens Tokenized by each Tokenizer

In general, our K-MT is good at tokenizing words in Korean including:

- Proper nouns or named entities: even in cases of very unusual names or names with typos where WordPiece and Mecab-ko failed to separate correctly.
  - 정기순 cengkiswun (name of a person) 까지 kkaci "even"
  - 천년유혼 chennyenyuhon (error of 천녀유혼 chennyeyuhon (name of a film))
- Internet slang: K-MT works well similarly to WordPiece and Mecab-ko.
  - 이뻐 ippe "(a variated form of) pretty" 용 yong (a variated form of final ending)

However, K-MT does not work well with:

- Cases with spacing errors: K-MT is based on word segments separated by whitespaces. If multiple words are adjoined without spaces, it cannot properly separate suffixes, while WordPiece and Mecab-ko worked to some degree.
- Cases with new suffixes not included in the pre-defined list
- Since we simply match all the suffixes in the pre-defined list, some of them may be incorrectly matched and separated from the word segment.

Word	Tokenization
차광남 chakwangnam personal name	(Mecab-ko) 차광chakwang 남nam (WordPiece) 차cha 광kwang 남nam (K-MT) 차광남chakwangnam
김유미의 kimyumiuy personal name (kimyumi) - genitive (uy)	(Mecab-ko) 김유미kimyumi 의uy (WordPiece) 김유kimyu 미의miuyi (K-MT) 김유미kimyumi 의uy
갑산공원 kapsankongwen location (kapsan) - "park" (kongwen)	(Mecab-ko) 갑산kapsan 공원kongwen (WordPiece) 갑kap 산san 공원kongwen (K-MT) 갑산공원kapsankongwen

Table 6: Tokenization of Proper Nouns

Word	Tokenization
국제농구연맹 kwukceynongkwuyenmayng International Basketball Federation	(Mecab-ko) 국제kwukce 농구nongkwu 연맹yenmayng (WordPiece) 국제kwukce 농구nongkwu 연맹yenmayng (K-MT) 국제농구연맹kwukceynongkwuyenmayng
대학교육협의회 tayhakkyoyukhyepuyhoy University Education Council	(Mecab-ko) 대학tayhak 교육kyoyuk 협의회hyepuyhoy (WordPiece) 대학교tayhakkyo 육yuk 협의회hyepuyhoy (K-MT) 대학교육협의회tayhakkyoyukhyepuyhoy
참교육학부모회 chamkyoyukhakpwumohoy Parents' Association for True Education	(Mecab-ko) 참cham 교육kyoyuk 학부모회hakpwumohoy (WordPiece) 참cham 교육kyoyuk 학부hakpwu 모mo 회hoy (K-MT) 참교육학부모회chamkyoyukhakpwumohoy

Table 7: Tokenization of Compositional Named Entities

Since NSMC and HSD datasets include various types of noise, especially spacing errors, there may have been examples where our K-MT could not accurately tokenize text. Moreover, K-MT is not advantageous when dealing with texts of various domains because it uses a limited size of vocabulary without subword tokens, unlike WordPiece.

However, K-MT works well with newly-listed words and terms as in the NER dataset, which existing language models based on formal texts cannot properly tokenize. For example, Table 6 shows some examples, including proper nouns, where K-MT correctly preserves the meaningful form of tokens.

Table 7 includes examples of named entities that are composed of multiple common nouns but are used as proper nouns. While K-MT does not oversegment them, Mecab-ko splits them into common nouns, and WordPiece sometimes splits them into meaningless subword tokens.

Moreover, K-MT is based on a simpler algorithm than Mecab-ko and other morpheme analyzers, making it easier to combine with other kinds of recent tokenization models such as the WordPiece model.

5 Conclusion

We implemented a simple Korean morpheme-like tokenizer, K-MT, to deal especially with proper nouns, coinages, internet slangs, and newly-listed domain-specific words in user-generated texts. To evaluate the tokenizer, we performed classification tasks on movie reviews, comments on news articles, and a Named Entity Recognition dataset, and we compared the results with those of a morpheme analyzer and character-based WordPiece model. From the results, we observed that our K-MT works well with proper nouns, coinages, and internet slang, while it still needs to be improved for cases with spacing errors.

For future work, we will improve our tokenizer by considering context information beyond the word segment boundary and additional methods to deal with the spacing errors. Also, we plan to construct a better tokenization algorithm by combining K-MT with the WordPiece model as a hybrid method.

Acknowledgements

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2019S1A5A2A01036576).

References

Dolly Agarwal, Somya Gupta, and Nishant Baghel. 2020. Automated assessment of noisy crowdsourced free-text answers for Hindi in low resource setting. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 122–131.

Seok-jae Choi. 2012. The study of combined endings' condition. *Journal of Korean Linguistics*, 63:275–311.

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.

Jiyoung Han, Youngin Lee, Junbum Lee, and Meeyoung Cha. 2019. The fallacy of echo chambers: Analyzing the political slants of user-generated news comments in Korean media. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 370–374.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47.

Ankit Kumar, Piyush Makhija, and Anuj Gupta. 2020. Noisy text data: Achilles' heel of bert. In *Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text*, pages 16–21.

Jae Jun Lee, Suhn Beom Kwon, and Sung Mahn Ahn. 2018. Sentiment analysis using deep learning model based on phoneme-level korean. *Journal of Information Technology Services*, 17:79–89.

Junbum Lee. 2020. Kcbert: Korean comments bert. In *Proceedings of the 32nd Annual Conference on Human and Cognitive Language Technology*, pages 437–440.

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.

Sangah Lee and Hyopil Shin. 2021. Combining sentiment-combined model with pre-trained bert models for sentiment analysis. *Journal of KIISE*, 48:815–824.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553.

Jihyung Moon, Won Ik Cho, and Junbum Lee. 2020. BEEP! Korean corpus of online news comments for toxic speech detection. In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media*, pages 25–31.

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.

Cheoneum Park, Dongheon Lee, Kihoon Kim, Changki Lee, and Hyunki Kim. 2019. Korean movie review sentiment analysis using self-attention and contextualized embedding. *Journal of KIISE*, 46:901–908.

Hyun-jung Park, Min-chae Song, and Kyung-shik Shin. 2018. Sentiment analysis of korean reviews using cnn: Focusing on morpheme embedding. *Journal of Intelligence and Information Systems*, 24:59–83.

Kyubyong Park, Joohong Lee, Seongbo Jang, and Dawoon Jung. 2020. 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.

Eetu Sjöblom, Mathias Creutz, and Mikko Aulamo. 2018. Paraphrase detection on noisy subtitles in six languages. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 64–73.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

A List of Grammatical Morphemes Used for K-MT

We provide an unexhaustive list of the grammatical morphemes we used for out K-MT in Table 8. The full list of morphemes are on the documentations of Kkma morpheme analyzer[8] and Mecab-ko morpheme analyzer[9]. We also provide the variated forms of grammatical morphemes used in noisy texts.

[8] http://kkma.snu.ac.kr/statistic?submenu=morp

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

Type	Morphemes
Word-Final Connective	고ko, 어e, 아a, 게key, 지ji, 면myen, 아서ase, 며mye, 다ta, 지만jiman, 면서myense, 는데nuntey, 다고tako, 어서ese, 라고lako, 라la, 아야aya, 어야eya, 으며umye, 다가taka
Word-Final Suffix	다ta, 습니다supnita, 어e, 어요eyo, 야ya, 지ci, 는다nunta, 아a, 죠cyo, 아요ayo, 지요ciyo, 는가nunka, 에요eyyo, 자ca, 을까ulkka, 냐nya, 오o, 니ni, 나na, 네ne (stylistic variated forms) 당tang, 습니당supnitang, 어용eyong, 어여eye, 예여yeyye, 아용ayong, 아여aye, 지용ciyong, 지여ciye, 에용eyyong, 에여eyye, 넹neng, 거든용ketunyong, 거든여ketunye, 구용kwuyong, 구여kwuye, 군용kwunyong, 군여kwunye, 나용nayong, 나여naye, 네용neyyong, 네여neyye, 용yong, 애여ayye, 더군용tekwunyong, 더군여tekwunye, 는데용nunteyyong, 는데여nunteyye, 잖아용canhayong, 잖아여canhaye, 답니당tapnitang, 세용seyyong, 세여seyye, 을까용ulkkayong, 을까여ulkkaye, 니까용nikkayong, 니까여nikkaye, 애용ayyong, 대용tayyong, 대여tayye, 으니까용unikkayong, 으니까여unikkaye, 외당oytang, 더라구용telakwuyong, 더라구여telakwuye, 예용yeyyong, 고용koyong, 고여koye, 라구용lakwuyong, 라구여lakwuye, 는군용nunkwunyong, 는군여nunkwunye, 다구용takwuyong, 다구여takwuye, 랍니당lapnitang, 래용layyong, 래여layye, 소이당soitang, 드라구용tulakwuyong, 드라구여tulakwuye, 던가용tenkayong, 던가여tenkaye, 은데용unteyyong, 은데여unteyye, 어야지용eyaciyong, 어야지여eyaciye, 던데용tunteyyong, 던데여tunteyye, 라고용lakoyong, 라고여lakoye, 아서용aseyong, 아서여aseye, 아야지용ayaciyong, 아야지여ayaciye, 입니당ipnitang, 습니다용supnitayong, 습니다여supnitaye, 아서당asetang, 합니당hapnitang, 는당nuntang, 단당tantang, 읍니당upnitang
Pre-final Suffix	갔kass, 것kes, 겄kess, 겟keys, 겠keyss, 긋kus, 댔tayss, 더te, 데tey, 드tu, 랬layss, 렷lyes, 리li, 사오sa, 사옵saop, 샤sya, 시si, 시겠sikeyss, 씨ssi, 아시asi, 앗as (stylistic variated forms) 이i, 아니ani, 했hayss, 햇hays, 셨syess, 셧syes, 엇es, 한han, 하ha, 씨ssi, 들tul
Postposition	가ka, 거나kena, 고ko, 과kwa, 까지kkaci, 나na, 다가taka, 대로taylo, 도to, 두twu, 든tun, 랑lang, 로lo, 를lul, 루lwu, 마냥manyang, 마다mata, 마저mace, 만man

Table 8: Grammatical Morphemes Used for K-MT