applied sciences

MDPI

Article

Syllable-Based Multi-POSMORPH Annotation for Korean Morphological Analysis and Part-of-Speech Tagging †

Hyeong Jin Shin, Jeongyeon Park and Jae Sung Lee *

Department of Computer Science, Chungbuk National University, Cheongju 28644, Republic of Korea
* Correspondence: jasonlee@cbnu.ac.kr
† This paper is an extended version of paper published in Proceedings of the 34th Annual Conference on Human and Language Technology, held in Gyeongju, Republic of Korea, 18–19 October 2022.

Abstract: Various research approaches have attempted to solve the length difference problem between the surface form and the base form of words in the Korean morphological analysis and part-of-speech (POS) tagging task. The compound POS tagging method is a popular approach, which tackles the problem using annotation tags. However, a dictionary is required for the post-processing to recover the base form and to dissolve the ambiguity of compound POS tags, which degrades the system performance. In this study, we propose a novel syllable-based multi-POSMORPH annotation method to solve the length difference problem within one framework, without using a dictionary for the post-processing. A multi-POSMORPH tag is created by combining POS tags and morpheme syllables for the simultaneous POS tagging and morpheme recovery. The model is implemented with a two-layer transformer encoder, which is lighter than the existing models based on large language models. Nonetheless, the experiments demonstrate that the performance of the proposed model is comparable to, or better than, that of previous models.

Keywords: Korean morphological analysis; Korean part-of-speech tagging; transformer encoder; syllable-based annotation

Citation: Shin, H.J.; Park, J.; Lee, J.S. Syllable-Based Multi-POSMORPH Annotation for Korean Morphological Analysis and Part-of-Speech Tagging. *Appl. Sci.* **2023**, *13*, 2892. https://doi.org/10.3390/app13052892

Academic Editors: Ying Shen, Cunhang Fan and Ya Li

Received: 31 January 2023
Revised: 21 February 2023
Accepted: 21 February 2023
Published: 23 February 2023

1. Introduction

Korean is a morphologically rich and agglutinative language. A Korean sentence is composed of word phrases called eojeol, which are composed of one or more morphemes and separated by spaces. Lines (1)–(4) show an example of a POS tagging process for a Korean sentence composed of three eojeols: "고운" (fine), "옷감을" (cloth) and "샀다" (bought). Two of the three eojeols are conjugated and restored to the base form of the morpheme sequences in line (2). The base form sentence is segmented into morphemes, as shown in line (3), and tagged with POS tags (we use the POS tags defined in [1], with the following meanings: VA = adjective, ETM = modifying ending, NNG = generic noun, JKO = objective particle, VV = verb, EP = pre-ending, EF = final ending, and SF = final punctuation mark), as shown in line (4). The morphemes within an eojeol are connected with a plus sign to mark the morpheme boundary. The entire process consists of three steps, originally described by [2]: restoration, segmentation, and POS tagging. These steps can be jointly performed, and the processing order can be reversed in some applications.

고운 옷감을 샀다. ((I) bought a fine cloth.)	Surface	(1)
곱ㄴ 옷감을 사았다.	Restoration	(2)
곱+ㄴ 옷감+을 사+았+다.	Segmentation	(3)
곱/VA+ㄴ/ETM 옷감/NNG+을/JKO 사/VV+았/EP+다/EF+./SF	Tagging	(4)

Note that the base form of the Korean morpheme should be restored for the segmentation and POS tagging of its surface form, and the length of the restored base form is different from (and usually longer than) that of the surface form. In this example, the number of

syllables (a syllable in Korean is a basic writing unit corresponding to a character in other languages) in the base form ("사았다") is larger than that in the surface form ("샀다"). For a Korean morphological analysis and part-of-speech tagging (hereafter KMAP) task, this fact makes it impossible to directly use a character-based n-to-n sequence labeling model, which is empirically proven to be effective for POS tagging in other languages. This is different from Chinese and Japanese word segmentation and POS tagging tasks, where morphemes (or words) are retrieved from a sentence without a morpheme restoration process and length change [3–8].

The difference in the length in the surface and base forms of morphemes has led to various approaches to the KMAP task. A popular approach is the syllable-based compound POS tag annotation method proposed in [9,10]. In this approach, every syllable in the surface sentence is mapped to one or multiple POS tags of the target syllables in the restoration (base form) sentence. If multiple tags are mapped, they are converted into one compound POS tag. Variant syllables and compound POS tags are recorded in a dictionary to restore the base form during the post-processing (see details in Section 2.2).

This annotation method enables the use of efficient n-to-n sequence labeling programs and has been adopted by many researchers [9–11]. However, there are some problems: First, a dictionary must be maintained to restore the morpheme and recover the POS tags, which becomes a burden in terms of time and space as the size of the dictionary increases. Second, for the training corpus construction, annotating a surface syllable to a POS tag or compound POS tag can require linguistic knowledge or heuristics to identify the correct boundary of the corresponding syllables. This can be subjective and prone to error, leading to inconsistent annotations. Third, the sequential numbering of the compound POS tags for ambiguity resolution may increase the complexity of the annotation method and restoration dictionary. Fourth, using the frequency in the restoration dictionary to resolve ambiguities is very limited in context, so it is not as effective as using deep learning models.

To address these issues, we propose a novel syllable-based multi-POSMORPH annotation method, which enables the use of a transformer encoder by providing n-to-n sequence labeling. The multi-POSMORPH tags contain all the information needed to make complete POS tags and base morpheme forms. In addition, the syllable-based annotated corpus is automatically constructed from the eojeol-based POS-tagged corpus. Using the annotation method, we implemented a KMAP program based on a transformer encoder architecture and evaluated its performance.

Our contributions are as follows:

1. We propose a novel syllable-based annotation method that enables to perform the joint task of a morphological analysis and POS tagging in one framework and *interact fully between the two tasks*.
2. Our method uses multi-POSMORPH tags, which contain all the information needed to make complete POS tags and base morpheme forms. Therefore, *no dictionary* is required for further post-processing, such as morpheme restoration and ambiguity resolutions in POS tags.
3. We propose an annotation program that *automatically and consistently constructs* a syllable-based annotated corpus from the eojeol-based POS-tagged corpus.
4. We propose a two-layer transformer-based program for the KMAP task, which is *very light and effective* compared to the previous large language-model-based implementations.

This study is an extended version of [12], extending the paper to address POS tagging task and including further investigations of related work and focused analyzes of experimental results. We describe the related work and various approaches in Section 2 and propose the syllable-based multi-POSMORPH annotation method and an implementation model in Section 3. The descriptions of the experiment and discussion follow in Sections 4 and 5, and the conclusion is presented in Section 6.

2. Related Work

2.1. Various Approaches to Korean Morphological Analysis and POS Tagging

The KMAP task has been studied either separately as two tasks or together as one task, using various approaches such as rule-based approaches [13–15], machine learning approaches [9,10,16–19], statistical approaches [2,20–22], and the recently developed deep learning approaches [11,23–29]. The most recent studies using deep learning approaches treat the KMAP task as one task. As mentioned in Section 1, the length difference between the surface form and base form is a major problem when we apply machine learning or deep learning models to our task. Various approaches have been proposed to solve this problem, and we classified these into the following four types. The abstract models are shown in Figure 1:

(A) Sequence-to-sequence (S2S): This approach uses n-to-m sequence-to-sequence (seq2seq) models [30] and performs the restoration, segmentation, and tagging steps simultaneously, taking syllables as input and producing syllables and POS tags as output. This was implemented with various encoder–decoder models: an attention-based encoder–decoder model using a GRU with additional convolution features [23], a statistical machine translation model using a lattice structure as the encoder and a hidden semi-Markov model as the decoder [22], a transformer encoder and decoder model fused with BERT embeddings [31,32], as well as a multitask model using BiLSTM as an encoder and LSTM as a decoder [24].

(B) Generation (Gen): In this approach, the base form is first generated from an input sentence in the surface form using seq2seq deep learning models. Then, the generated sentence is processed for segmentation and POS tagging by a transformer, which maps input syllables to output labels in one-to-one correspondence. This is usually achieved by pipelining the generation (restoration) process with the joint process of segmentation and tagging [25,27,33]. The performance of this approach is usually limited by the propagation errors inherent in the pipeline architecture.

(C) Indirect Generation (IGen): This approach is a pipeline model similar to approach (B) but uses a sequence labeling model at the first stage. The base form is indirectly generated in the first stage by length prediction [29], action commands [34], or concatenated graphemes [33]. The input of the second stage contains intermediate forms or corresponding base forms: the predicted number of surface syllables, latent values, graphemes of the base form, and syllables of the base forms. This is implemented using a non-autoregressive transformer [29] and BiLSTM-CRF [33,34].

(D) Annotation with compound POS tags (CTag): A syllable-based compound POS annotation scheme is used in this approach. Every input syllable is mapped to either a single tag or a compound tag created with the corresponding multiple POS tags. Next, the compound POS tag is translated into normal POS tags with the base form of the surface syllable using a restoration dictionary during the post-processing. This approach must maintain a tag-mapping dictionary, which requires extra time and space. This scheme was initially implemented with a Conditional Random Field model [9,10,35] and later with a Support Vector Machine model [18,19,36] and transformer [11,26].

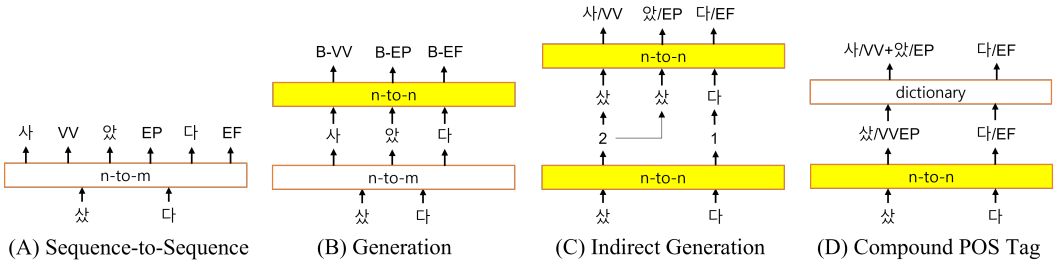

Figure 1. Abstract models of various approaches to solve the length difference between surface and base form. Here, "n-to-m" means seq2seq model and "n-to-n" means sequence labeling model.

2.2. Compound POS Tagging Example

In this section, we describe the compound POS tagging method proposed in [9,10] by showing examples: Line (6) shows an example of the syllable-based compound POS tagging for a sentence in line (5). Line (7) is the result of the restoration. The first eojeol shows two variant syllables: "고" is a part of the base form "곱", and "운" is the combination of a part of the first base syllable "ㅂ" and the second base syllable "ㄴ". In line (6), the two syllables are annotated with each POS tag, even though there is some overlap between the two base form syllables. The base forms were recovered using the restoration dictionary, as shown in (a) and (b) in Table 1.

The second eojeol shows simple syllable-based tagging: "옷/NNG" and "감/NNG" are tagged separately, and two syllables with the same POS tag are combined to "옷감/NNG" in the restoration (BI tags were adopted for the segmentation of multiple morphemes with the same POS tags in some papers [11,19]; however, here we omit the BI tags for a simple description). The third eojeol includes a compound tag "VVEP", which is used for the one-to-two mapping: from "샀" to "사/VV+았/EP". Table 1 shows the restoration dictionary entries: the (a), (b), and (c) rows are for the taggings in line (6).

An ambiguous case is shown in the (d) and (e) rows, where one compound tag can be restored to two different base forms: "가/VVEC" can be restored to either "가/VV+아/EC" or "그/VV+어/EC", depending on the context. However, it is difficult to find the context information in the dictionary to resolve ambiguities, because the dictionary search occurs during the post-processing time. This problem is partially solved using compound tags that attach sequence numbers for different contexts, or by using the frequency information recorded in the dictionary during the learning process [19].

고운 옷감을 샀다.	Surface	(5)
고/VA+운/ETM 옷/NNG+감/NNG+을/JKO 샀/VVEP+다/EF+./SF	Tagging	(6)
곱/VA+ㄴ/ETM 옷감/NNG+을/JKO 사/VV+았/EP+다/EF+./SF	Restoration	(7)

Table 1. Restoration dictionary example.

	Dictionary Entries	Context Example	
(a)	고/VA→ 곱/VA	고/VA + 운/ETM	(fine)
(b)	운/ETM→ ㄴ/ETM	고/VA + 운/ETM	(fine)
(c)	샀/VVEP→ 사/VV + 았/EP	샀/VVEP + 다/EF	(bought)
(d)	가/VVEC→ 가/VV + 아/EC	가/VVEC + 서/EC	(go)
(e)	가/VVEC→ 그/VV + 어/EC	담/VV + 가/VVEC + 서/EC	(dip)

3. Syllable-Based Multi-POSMORPH Annotation

3.1. Proposed Models

We propose a syllable-based annotation method to solve the length difference problem for KMAP. In this method, every input syllable is mapped to a single tag, called the POSMORPH tag, which is created with the corresponding POS tags and base morph forms (the term "POSMORPH tags" was used in [37,38] to mean both POS and morphological tags; in this paper, we use the same name with slightly different meanings, that is, the POS tag and the morpheme itself). The tag encapsulates all the information needed to generate the results of KMAP, which can be decapsulated by a simple rule-based procedure during post-processing. Therefore, it can utilize whole context information to determine the base forms and POS tags in one deep learning model frame, without a restoration dictionary, which is used in the compound POS tag approach. Our abstract model is shown in Figure 2.

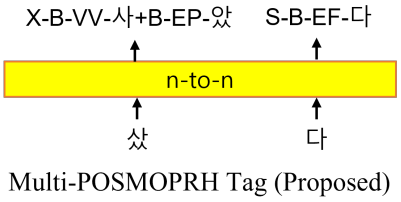

Figure 2. Abstract model for multi-POSMORPH tagging (MTag). Please see Figure 1 for comparison.

The implementation model is composed of a transformer encoder [39] for syllable encoding, with a BiLSTM network [40] on top for sequence labeling. The model learns and produces multi-POSMORPH tags for morpheme restoration, segmentation, and POS tagging. The architecture is shown in Figure 3. The model uses the copying mechanism that copies input syllables into output syllables if they are non-conjugating characters, such as non-Korean characters [41–43].

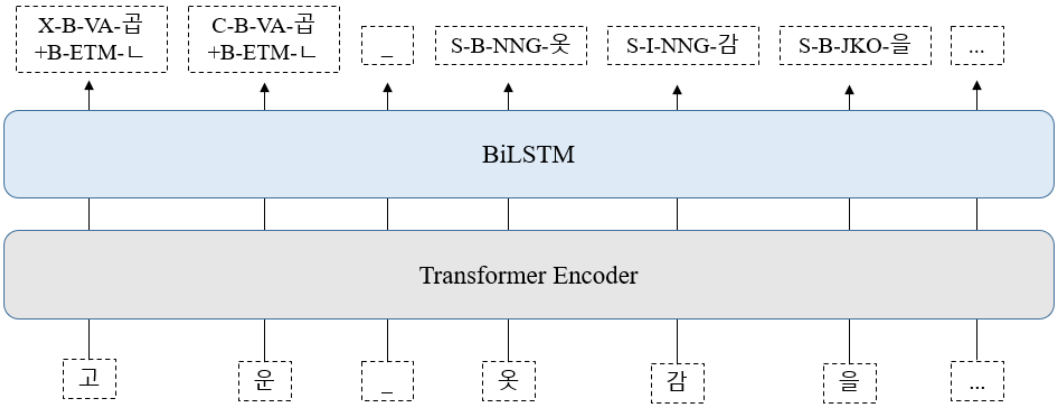

Figure 3. Implementation model for multi-POSMORPH tagging.

3.2. Multi-POSMORPH Annotation Method

This method converts the annotation style from the current per-eojeol style used in POS-tagged corpora to a per-syllable style. First, we retrieve a surface sentence and the corresponding restoration sentence from a POS-tagged corpus. A surface sentence can be aligned with a restoration sentence in the syllable unit using *charAlign*, the modified program from [44]. (See Section 3.3). Next, with the aligned-syllable strings, we can tag the syllable units, as shown in Algorithm 1, that is, if the mapping between surface and restoration sentences is 1-to-1, an "S" tag is prefixed to the restoration sentence. If the mapping is 1-to-n, where n > 1, an "X" tag is prefixed. If the mapping is n-to-m, where n > 1 and m $\geq$ 1, an "X" tag is prefixed for the first syllable of the surface and a "C" tag for the rest of the surface syllables. The "SXC" tags encapsulate the original base forms: a single syllable with "S", and a single or multiple syllable with "X". A "C" tag is used to duplicate the previous "X" tag's contents. This is used for validation only and ignored in decapsulation.

The annotation tags are created according to the three steps described in Section 1: restoration, segmentation, and tagging. Following that step, we can extend annotations on restoration sentences to handle morpheme segmentation and POS tagging. By consulting the eojeol-based POS-tagged corpus, we identify the B (beginning) or the I (inside) of a morpheme for segmentation, and the POS tag of the morpheme to which the current syllable belongs for tagging. Using this information, each syllable in the restoration sentence is annotated using BI tags and POS tags for segmentation and POS tagging.

Table 2 presents each step of an example of syllable annotation. Columns 2 and 3 present the alignment between the surface sentence and the restoration (base) sentence. Columns 4, 5, and 6 list the syllable-based POSMORPH tags for restoration, segmentation, and tagging, respectively, corresponding to the surface syllables in the same row in column 1. For example, "C-곱ㄴ" is the

syllable-based restoration tag for "운," and "X-B-VV-사+B-EP-았" is the syllable-based tagging tag (POSMORPH tag) for surface syllable "샀".

Algorithm 1 Syllable-based multi-POSMORPH Annotation for Restoration

```
Input: a pair of sentences (SurfaceSent, RestorationSent),
       a corresponding eojeol-based POS-tagged sentence
Output: list of aligned (syllable, multi-POSMORPH tag) pairs
1. AlignedPairs = charAlign (SurfaceSent, RestorationSent)
2. for (Surface, Restoration) in AlignedPairs:
3.     if len(Surface) == 1 and len(Restoration) == 1:
4.         yield Surface, "S-" + Restoration
5.     else if len(Surface) == 1 and len(Restoration) > 1:
6.         yield Surface, "X-" + Restoration
7.     else if len(Surface) > 1:
8.         yield Surface[0], "X-" + Restoration
9.         for syl in Surface[1:]:
10.            yield syl, "C-" + Restoration
```

Table 2. Example of syllable-based multi-POSMORPH annotation.

1. Surface (Syllable)	2. Surface (Aligned)	3. Restoration (Aligned)	4. Restoration (Syllable)	5. Segmentation (Syllable)	6. Tagging (Syllable)
고	고운	곱ㄴ	X-곱ㄴ	X-B-곱-B-ㄴ	X-B-VA-곱 + B-ETM-ㄴ
운			C-곱ㄴ	C-B-곱-B-ㄴ	C-B-VA-곱 + B-ETM-ㄴ
_	_	_	_	_	_
옷	옷	옷	S-옷	S-B-옷	S-B-NNG-옷
감	감	감	S-감	S-I-감	S-I-NNG-감
을	을	을	S-을	S-B-을	S-B-JKO-을
_	_	_	_	_	_
샀	샀	사았	X-사았	X-B-사 + B-았	X-B-VV-사 + B-EP-았
다	다	다	S-다	S-B-다	S-B-EF-다
.	.	.	S-.	S-B-.	S-B-SF-.

Reference sentence in Sejong POS-tagged form is 곱/VA+ㄴ/ETM 옷감/NNG+을/JKO 사/VV+았/EP+다/EF+./SF. Underbar (_) represents a space token.

3.3. Syllable Alignment

The algorithm of *charAlign* for syllable alignment can be briefly explained by the following formula [44]:

$$m[i,j] = min(m[i-d_i, j-d_j] + acost(x[i-d_i:i], y[j-d_j:j])) \quad \forall(d_i, d_j). \tag{8}$$

Here, m is the cost matrix used to find the minimum cost needed to align the source sentence x and restoration sentence y. At position (i, j), every possible alignment category (d_i, d_j) is used to find the minimum cost. The possible alignment categories used in this paper are presented in Table 3. In addition, *acost* is the cost required to align the sub-string of x and y (represented in Python list style), which is calculated using the corpus statistics of co-occurrence and character code similarities. Using the dynamic programming-based *charAlign*, we can obtain surface and restoration sub-string pairs that are aligned according to the minimum alignment cost.

An analysis revealed that the usage of 94.9% of the syllables in the Sejong corpus does not vary. This means that KMAP programs cannot achieve an accuracy higher than 94.9% at syllable level without handling variant types. Table 3 presents the category of variant

types. Note that, for efficient processing, alignment category 0:1 was changed to category 1:2 by including one more syllable in the context of the alignment pair.

Table 3. Syllable alignment category of variations (%, Surface:Restoration).

1:1	1:2	1:3	1:4	2:1	2:2	2:3
5.21	90.56	0.78	0.01	0.03	2.40	1.02

4. Experiments

4.1. Dataset and Hyperparameters

We used the Sejong POS-tagged corpus [1], in which apparent errors such as raw text input errors, discrepancies between the surface form and the analyzed form, and format errors were corrected. We did not correct the errors that might lead to extensive modifications, such as inconsistent annotation errors. We normalized some characters with glyphs that look similar but have different unicodes. As a result, we used about 854,000 sentences, which were divided using a ratio of 8:1:1 into the training, validation, and test datasets. The statistics of the dataset are shown in Table 4.

Table 4. Statistics of the Sejong POS-tagged corpus.

Types	Values
Sentences	854,481
Eojeols	10,052,682
Morphemes	22,918,094
Syllables	32,447,285
POS tag types	42
Eojeols per sent. (avg.)	11.76
Morp. per eojeol (avg.)	2.28
Syl. per eojeol (avg.)	3.23
Syl. per morp. (avg.)	1.42

The hyperparameters of our implementation model are shown in Table 5. Note that the number of transformer layers was chosen to be two, and the best-performing number of layers we tested (1, 2, 4, 8, and 12 layers were evaluated). This model is relatively light when compared with BERT-based models using 12 transformer layers [11,26].

Table 5. Hyperparameters

Parameters	Values
No. transformer layers	2
No. BiLSTM layers	2
Hidden-layer size	768
Learning rate	1×10^{-5}
Batch size	64
Dropout rate	0.3

4.2. Evaluation Metrics

Evaluation was performed at three levels: syllable, morpheme, and eojeol. The performance at the syllable and eojeol levels was measured using accuracy without considering

spacing units. This formula is presented in (9). Morpheme-level performance was measured using the F1-score, which is based on precision and recall. Because we count the number of matching morphemes per sentence, finding the correct corresponding target is challenging in long sequences if some morphemes are omitted or incorrectly inserted. To mitigate this problem, we used the Levenshtein distance (denoted as *dist* in the formula) to calculate precision and recall, as (10) and (11) indicate. The F1-score is calculated from the precision and recall, as shown in (12).

$$Accuracy = \frac{num.\ of\ correctly\ predicted\ syllables\ (eojeols)}{total\ num.\ of\ syllables\ (eojeols)\ in\ test\ set} \tag{9}$$

$$Precision = \frac{max(|gt|, |out|) - dist(gt, out)}{|out|} \tag{10}$$

$$Recall = \frac{max(|gt|, |out|) - dist(gt, out)}{|gt|} \tag{11}$$

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

4.3. Results

We evaluated the three syllable-based multi-POSMORPH annotation methods using the proposed end-to-end model illustrated in Figure 3. The evaluation results are listed in Table 6. As the process became more complex in the higher stages of the annotation, the performance in terms of the eojeol F1-score reflected this complexity by increasing in the order of restoration (99.41%), segmentation (98.02%), and tagging (96.12%).

The Sejong corpus is generally used to evaluate KMAP tasks because it is publicly available and large. However, there are some annotation errors, such as inconsistent annotations that could degrade the performance of the system. Some researchers used a subset of the corpus and/or modified it for the evaluation, which results in Sejong corpus evaluation sets with various sizes. In contrast, we used the full set of the corpus for the evaluation, with minor corrections or deletions, as described in Section 4.1.

Table 7 presents the tagging performance of previous models on the various evaluation sets. Because the models were evaluated on different subsets of the corpus with some corrected data, the performance comparison is not rigorous but approximate. Most of the models [10,11,19] that used comparatively smaller test datasets with screened or corrected data achieved better results than the others. However, among the models evaluated on test datasets of more than 40K sentences, our proposed model performed best at the eojeol level, with the exception of Matteson et al.'s eojeol-level result, which was obtained on correctly annotated test data. If we were to use correctly annotated test data in the Sejong corpus, we would expect an increase in the performance. This is discussed in Section 5.4.

Table 6. Evaluation result of three syllable-based annotation methods.

	Syllable	**Morpheme**			**Eojeol**
	Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**
Restoration	99.83 ± 0.01	-	-	-	99.41 ± 0.02
Segmentation	99.28 ± 0.02	98.94	98.82	98.88 ± 0.01	98.02 ± 0.02
Tagging	98.41 ± 0.02	97.78	97.70	97.74 ± 0.02	96.12 ± 0.03

Table 7. Performance comparison with previous work (morpheme F1, syllable, and eojeol accuracy, %).

Authors	Model	Approach	Data (K sents.)	Tagging Performance		
			Training/Test	Syllable	Morpheme	Eojeol
Shim, 2013 [10]	CRF	CTag	Konan 33/3 *	97.99	-	**96.47**
Lee, 2013 [18]	Structural SVM	CTag	Sejong 133/13 *	-	96.99	-
Lee et al., 2016 [19]	Structural SVM	CTag	Sejong 56/6 *	-	-	96.41 §
Li et al., 2017 [23]	GRU	S2S	Sejong 90/10	-	97.27	95.28
Park et al., 2019 [11]	BERT + BiLSTM + CRF	CTag	Sejong 753/9	-	**98.74**	-
Choi et al., 2016 [33]	BiLSTM + CRF	IGen	Sejong 783/87	-	-	94.89
Na and Kim., 2017 [22]	Lattice + HMM	S2S	Sejong 200/50	-	96.49	94.77
Matteson et al., 2018 [34]	BiLSTM + CRF	IGen	Sejong 807/43	-	-	96.20 †
Min et al., 2019 [26]	BERT + BiLSTM	CTag ‡	Sejong 200/50	-	95.22	93.90
Song and Park, 2019 [24]	BiLSTM + LSTM + PGN + CRF	S2S	Sejong 200/50	-	97.43	95.68
Song and Park, 2020 [25]	BiLSTM + CRF	Gen	Sejong 200/50	-	97.27	95.28
Jin and Yu, 2021 [28]	BiLSTM + CRF + Conv.	S2S	Sejong 200/50	-	98.07	95.92
Cho and Song, 2022 [29]	Transformer(EnDe) + CRF	IGen	Sejong 200/50	-	97.59	95.83
Proposed	Transformer(En) + BiLSTM	MTag	Sejong 769/87	**98.41** $\pm$ 0.02	97.74 $\pm$ 0.02	96.12 $\pm$ 0.03

* Estimated from the original number of eojeols (calculated using 12 eojeols per sentence). § Accuracy without using a pre-analyzed dictionary. † Accuracy for correctly annotated test data. ‡ Subwords were used instead of syllables as the annotation units.

5. Discussion

5.1. Architecture Comparison of KMAP Approaches

As described in Section 2.1, various approaches have been tackled for the KMAP task because of the length difference problem. We compared the approaches with an averaged eojeol accuracy. The approaches we chose for the comparison are the ones tested with large test data, as shown in the bottom nine rows of Table 7. We omitted [26] the approach of CTag using word pieces because it is different from the original CTag approaches [9,10].

Table 8 shows the result in the order of the better performing approach types: MTag, IGEN, S2S, and GEN. From the result, we can conjecture that the n-to-n model performs better than the n-to-m model, and the end-to-end model performs better than the pipeline model. Therefore, we can conjecture that our proposed method is the best approach type for the KMAP task because it uses end-to-end and n-to-n models.

Table 8. Performance comparison of architectures.

Approaches	Avg. eojeol acc.	End-to-End	n-to-n	Pipeline	n-to-m
MTag	96.12	✓	✓		
IGEN	95.54		✓	✓	
S2S	95.46	✓			✓
GEN	95.28		✓	✓	✓

5.2. Copying Mechanism

The restoration process is the first stage in the KMAP task. This is an important step that strongly affects the performance of the following stages, which are segmentation and tagging. As we mentioned in Section 3.3, 94.9% of the syllables were unchanged during the restoration process because most syllables or characters are not the targets of the restoration. These include non-Korean characters, such as numerical, English, Chinese, and special-symbol characters, which can be easily identified by their character codes.

We adopted a copying mechanism [41–43] to ensure they remained unchanged during not just the restoration process but also the segmentation and tagging processes. An ablation

test showed that the copying mechanism improved the F1-score performance of the POS tagging per eojeol by 0.03% points.

5.3. Out-of-Vocabulary Labels

The size of the tag sets increases as more analytic features, such as the base form, segmentation, and POS tags, are included. Moreover, not all possible combinations of the features appeared in the training data, which might cause a data sparseness problem. We measured the size of the training data and the ratio of the out-of-vocabulary (OOV) labels in the test data for the three annotation methods. The statistics were measured 10 times and averaged from data that were randomly sampled from the whole dataset (32.4M syllable annotation instances), where 80% were allocated to the training dataset and 20% were allocated to the test dataset.

Table 9 presents the statistics. The size of the set of the output labels increases by 1.5 times when segmentation labels are added to the restoration labels and doubles when tagging labels are added. However, the ratio of the OOV types did not change much, even though the difference in the size of the output label set is large. Moreover, the OOV instance ratio increases as the size of the output label set increases. Nevertheless, the absolute value of the ratio was very small, ranging from 0% to 0.002%, which does not substantially affect the performance of our proposed annotation methods when large output label sets are used.

Table 9. Size of the output label set and OOV ratio for the three annotation methods.

	Label Set Size	OOV Ratio (%)	
		Type	Instance
Restoration	8039 ± 17	5.764 ± 0.256	0.006 ± 0.000
Segmentation	12,508 ± 18	6.236 ± 0.192	0.010 ± 0.000
Tagging	25,079 ± 32	6.293 ± 0.174	0.020 ± 0.001

5.4. Error Analysis

We sampled and analyzed 300 errors, which comprise about 1% of the total errors. We categorized five main types of error: restoration, segmentation, tagging, annotation, and variant. The procedural type errors implicate errors in the subsequent procedures. For example, a restoration error usually implies both a segmentation error and tagging error. A segmentation error usually implies a tagging error. The annotation error type includes annotation mistakes and data input mistakes in the test data. Inconsistent annotations cause inconsistent learning, which will produce variable analysis results. We considered any inconsistent annotation cases that we found in the training corpus and that helped output the variant to be variant annotation errors.

The loose definition of the corpus markup principle allows for dual standards [45,46], which cause inconsistent annotations in the training data. A typical case is compound words that are considered to be one unit but allowed to be segmented into smaller units in the analysis, for instance, “혈액형/NNG” (blood type) vs. “혈액/NNG+형/XSN”, “어떻든/MAG” (anyhow) vs. “어떻/VA+든/EC”, and “잦아지/VV+ㄴ/ETM” (frequent) vs. “잦/VA+아/EC+지/VX+ㄴ/ETM”.

In the evaluation, we measured the performance using the test data as golden labels. However, this approach is unfair in the following cases:

1. When the golden label is incorrect and the prediction is correct (annotation errors);
2. When the golden label has dual standards and the prediction is the other standard that is not the golden label (variant errors).

For a fair evaluation, we adjusted the numbers by reclassifying the error types of the above two unfair cases. Figure 4 shows the percentage of the measured and adjusted error types. A segmentation error is the most common type of error, comprising 44% of all errors

in the measured case, and tagging errors are ranked second. After the adjustment, 40% of the adjusted errors are not errors (Ann 18% + Var 22%) and mostly originate from measured errors of segmentation errors (19% = 44% − 25%) and tagging errors (17% = 40% − 23%). This makes us believe that the evaluation result would improve if we were to adjust the test dataset for a fair evaluation.

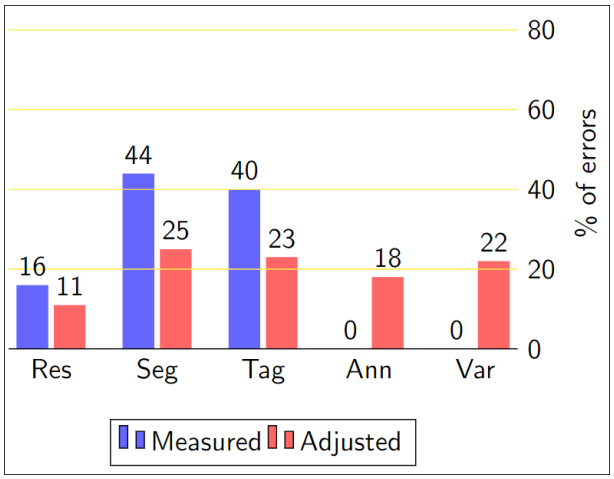

Figure 4. Error Analysis: 40% of the identified errors were found to be non-errors after adjustment (Ann 18% and Var 22%).

5.5. Further Improvement

We have empirically proved that our method is effective, but we list a few points for future work that could further improve our method:

- Multi-POSMORPH tags are made of a combination of POS tags and morpheme syllables. Therefore, the tag set size is very large, with 25,000 tags in the current model. This could be reduced, for example, if we borrow the idea of action commands that generate base forms with a few commands [34].
- Multiplying the combined POSMORPH tags can easily cause OOV problems due to data sparseness in the training corpus. Dividing alignment pairs into more smaller pairs would mitigate the OOV problem. For example, a 2:2 alignment can be divided into two 1:1 alignments if this does not hurt the performance.
- Conjugation rules for irregular verbs and adjectives are well-defined in Korean grammar. The rules may help to process untrained conjugated cases if we can merge them into our model.

6. Conclusions

We proposed a novel annotation method to conduct the KMAP task in one step of an n-to-n sequence labeling process. The method uses a POSMORPH tag combined with the POS tag and syllable of the base form, which eliminates post-processing with a dictionary that causes performance degradations in the previous compound POS tagging method. The KMAP program using our annotation method can be implemented with a two-layer transformer encoder and a two-layer BiLSTM model, which is a relatively light model. Nonetheless, the experiments proved that our proposed model is highly effective, achieving a 96.12% F1-score in the eojeol-level evaluation, which was the highest score among the models tested on the large unfiltered test dataset. However, our proposed method still needs to be improved due to limitations such as the large size of the multi-POSMORPH tag set and the out-of-vocabulary problem due to data sparseness in the training corpus. We leave these issues for future work.

Author Contributions: Conceptualization, J.S.L.; methodology, H.J.S. and J.S.L.; software, H.J.S. and J.P.; validation, H.J.S. and J.P.; formal analysis, H.J.S., J.P. and J.S.L.; investigation, H.J.S. and J.S.L.; resources, H.J.S., J.P. and J.S.L.; data curation, H.J.S., J.P. and J.S.L.; writing—original draft preparation, H.J.S. and J.S.L.; writing—review and editing, H.J.S., J.P. and J.S.L.; visualization, H.J.S. and J.S.L.; supervision, J.S.L.; project administration, J.S.L.; funding acquisition, J.S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1I1A3059545). This work was conducted during the research year of Chungbuk National University in 2020.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. The National Institute of Korean Language. *21st Century Sejong Project Final Result, Revised Edition*; The National Institute of the Korean Language: Seoul, Republic of Korea, 2014. (In Korean)
2. Lee, J. Three-step probabilistic model for Korean morphological analysis. *J. KIISE Softw. Appl.* **2011**, *38*, 257–268. (In Korean)
3. Tseng, H.; Chen, K. Design of Chinese morphological analyzer. In Proceedings of the COLING-02: The First SIGHAN Workshop on Chinese Language Processing, Taipei, Taiwan, 1 September 2002; Volume 18, pp. 1–7.
4. Xue, N. Chinese word segmentation as character tagging. *Int. J. Comput. Linguist. Chin. Lang. Process.* **2003**, *8*, 29–48.
5. Ng, H.T.; Jin, K. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 277–284.
6. Peng, F.; Feng, F.; McCallum, A. Chinese segmentation and new word detection using conditional random fields. In Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004), Geneva, Switzerland, 23–27 August 2004; pp. 562–568.
7. Kudo, T.; Yamamoto, K.; Matsumoto, Y. Applying conditional random fields to Japanese morphological analysis. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 25–26 July 2004; pp. 230–237.
8. Nagata, M. A stochastic Japanese morphological analyzer using a forward-DP backward-A* N-best search algorithm. In Proceedings of the 15th International Conference on Computational Linguistics, Kyoto, Japan, 5–9 August 1994; Volume 1, pp. 201–207.
9. Shim, K. Syllable based Korean POS tagging without morphological analysis. *Korean J. Cogn. Sci.* **2011**, *22*, 327–345. (In Korean) [CrossRef]
10. Shim, K. Morpheme restoration for syllable-based Korean POS tagging. *J. KIISE Softw. Appl.* **2013**, *40*, 182–189. (In Korean)
11. Park, C.; Lee, C.; Kim, H. Korean morphological analysis and part-of-speech tagging with LSTM-CRF based on BERT. In Proceedings of the 31st Annual Conference on Human and Language Technology, Daejeon, Republic of Korea, 11–12 October 2019; pp. 34–36. (In Korean)
12. Shin, H.J.; Park, J.; Lee, J.S. Korean morpheme restoration and segmentation based on transformer. In Proceedings of the 34th Annual Conference on Human and Language Technology, Gyeongju, Republic of Korea, 18–19 October 2022; pp. 403–406. (In Korean)
13. Kim, Y. *Natural Language Processing*, 2nd ed.; Life & Power Press Co., Ltd.: Paju-si, Republic of Korea, 2003; pp. 69–111. (In Korean)
14. Kang, S. *Korean Morphological Analysis and Information Retrieval*, 2nd ed.; Hongreung Publishing Company: Seoul, Republic of Korea, 2003; pp. 73–440. (In Korean)
15. Shim, K.; Yang, J. MACH: A supersonic Korean morphological analyzer. In Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002), Taipei, Taiwan, 24 August–1 September 2002.
16. Shim, K. Auto spacing in Korean using CRF. *Korean J. Cogn. Sci.* **2011**, *22*, 217–233. (In Korean)
17. Na, S.H.; Yang, S.I.; Kim, C.H.; Kwon, O.W.; Kim, Y.K. CRFs for Korean morpheme segmentation and POS tagging. In Proceedings of the 24th Annual Conference on Human and Language Technology, Pusan, Republic of Korea, 11–13 October 2012; pp. 12–15. (In Korean)
18. Lee, C. Joint models for Korean word spacing and POS tagging using structural SVM. *J. KIISE Softw. Appl.* **2013**, *40*, 604–606. (In Korean)
19. Lee, C.; Lim, J.; Lim, S.; Kim, H. Syllable-based Korean POS tagging based on combining a pre-analyzed dictionary with machine learning. *J. KIISE Softw. Appl.* **2016**, *43*, 362–369. (In Korean) [CrossRef]
20. Lee, D.; Rim, H. Probabilistic models for Korean morphological analysis. In Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts (IJCNLP2005), Jeju Island, Republic of Korea, 11–13 October 2005.
21. Lee, S.; Lee, D.; Rim, H. A corpus-based hybrid model for morphological analysis and part-of-speech tagging. *J. Korean Assoc. Comput. Educ.* **2008**, *13*, 11–18. (In Korean)
22. Na, S.; Kim, Y. Phrase-based statistical model for Korean morpheme segmentation and POS tagging. *IEICE Trans. Inf. Syst.* **2017**, *101*, 512–522. [CrossRef]
23. Li, J.; Lee, E.; Lee, J. Sequence-to-sequence-based morphological analysis and part-of-speech tagging for Korean language with convolutional features. *J. KIISE Softw. Appl.* **2017**, *40*, 57–62. (In Korean) [CrossRef]

24. Song, H.J.; Park, S.B. Korean morphological analysis with tied sequence-to-sequence multi-task model. 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), Hong Kong, China, 3–7 November 2019; pp. 1436–1441.
25. Song, H.J.; Park, S.B. Korean part-of-speech tagging based on morpheme generation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process. (TALLIP)* **2020**, *19*, 1–10. [CrossRef]
26. Min, J.; Na, S.; Sin, J.; Kim, Y. BERT with subword units for Korean morphological analysis. In Proceedings of the 31st Annual Conference on Human and Language Technology, Daejeon, Republic of Korea, 11–12 October 2019; pp. 37–40. (In Korean)
27. Youn, J.; Lee, J. A deep learning-based two-steps pipeline model for Korean morphological analysis and part-of-speech tagging. *J. KIISE Softw. Appl.* **2021**, *48*, 444–452. (In Korean) [CrossRef]
28. Jin, G.; Yu, Z. A hierarchical sequence-to-sequence model for Korean POS tagging. *Trans. Asian Low-Resour. Lang. Inf. Process.* **2021**, *20*, 1–13. [CrossRef]
29. Cho, S.; Song, H.J. Non-autoregressive multi decoders for Korean morphological analysis. In Proceedings of the 34th Annual Conference on Human and Language Technology, Gyeongju, Republic of Korea, 18–19 October 2022; pp. 418–423. (In Korean)
30. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS'14), Montreal, QC, Canada, 8–13 December 2014; Volume 2, pp. 3104–3112.
31. Lee, C.; Ra, D. Korean morphological analysis method based on BERT-fused transformer model. *KIPS Trans. Softw. Data Eng.* **2021**, *11*, 169–178. (In Korean)
32. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. 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), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
33. Choi, J.; Youn, J.; Lee, S. A grapheme-level approach for constructing a Korean morphological analyzer without linguistic knowledge. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 3872–3879.
34. Matteson, A.; Lee, C.; Lim, H.; Kim, Y. Rich character-level information for Korean morphological analysis and part-of-speech tagging. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 2482–2492.
35. Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML'01), San Francisco, CA, USA, 28 June–1 July 2001; pp. 282–289.
36. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Their Appl.* **1998**, *13*, 18–28. [CrossRef]
37. Müller, T.; Schmid, H.; Schütze, H. Efficient higher-order CRFs for morphological tagging. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Washington, CA, USA, 18–21 October 2013; pp. 322–332.
38. Heigold, G.; Neumann, G.; Genabith, J.V. Neural morphological tagging from characters for morphologically rich languages. *arXiv* **2016**, arXiv:1606.06640.
39. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *ADvances Neural Inf. Process. Syst.* **2017**, *30*, 1–13.
40. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
41. Luong, M.T.; Sutskever, I.; Le, Q.V.; Vinyals, O.; Zaremba, W. Addressing the rare word problem in neural machine translation. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 27–31 July 2015; pp. 11–19.
42. Hwang, H.; Lee, C. Korean Morphological Analysis using Sequence-to-sequence learning with copying mechanism. In Proceedings of the Korean Information Science Society 2016 Winter Conference, Pyeongchang, Republic of Korea, 21–22 December 2016; pp. 443–445. (In Korean)
43. Jung, S.; Lee, C.; Hwang, H. End-to-end Korean part-of-speech tagging using copying mechanism. *ACM Trans. Asian Low-Resour. Lang. Inf. Process. (TALLIP)* **2018**, *17*, 1–8. [CrossRef]
44. Gale, W.A.; Church, K.W. A program for aligning sentences in bilingual corpora. *Comput. Linguist.* **1994**, *19*, 75–102.
45. Kim, E.; Choi, K. On correction guideline of tagged corpus. In Proceedings of the 12th Annual Conference on Human and Language Technology, Seoul, Republic of Korea, 13–14 October 2000. (In Korean)
46. Kim, I. *Conducting Korean POS Tagged Corpus*; The National Institute of the Korean Language: Seoul, Republic of Korea, 2019. (In Korean)