End-to-End Korean Part-of-Speech Tagging Using Copying Mechanism

SANGKEUN JUNG, SK telecom
CHANGKI LEE and HYUNSUN HWANG, Kangwon National University

In this article, we introduce a novel neural architecture for the end-to-end Korean Part-of-Speech (POS) tagging problem. To address the problem, we extend the present recurrent neural network-based sequence-to-sequence models to deal with the key challenges in this task: rare word generation and POS tagging. To overcome these issues, Input-Feeding and Copying mechanism are adopted. Although our approach does not require any manual features or preprocessed pattern matching dictionaries, our best single model achieves an F-score of 97.08. This is competitive with the current state-of-the-art model (F-score 98.03), which requires extensive manual feature processing.

CCS Concepts: • **Computing methodologies** → *Phonology/morphology*;

Additional Key Words and Phrases: Part-of-speech tagging, deep learning, copying mechanism

ACM Reference format:
Sangkeun Jung, Changki Lee, and Hyunsun Hwang. 2018. End-to-End Korean Part-of-Speech Tagging Using Copying Mechanism. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 17, 3, Article 19 (January 2018), 8 pages.
http://dx.doi.org/10.1145/3178458

1 INTRODUCTION

The process of assigning one of the parts of speech to the given words is called Part-of-Speech (POS) tagging. POS tagging is beneficial in many natural language processing applications. For example, in information retrieval, the segmented or tagged tokens are used for indexing, whereas in word sense disambiguation, POS-type information provides discriminative features. Additionally, POS tagging is a useful or mandatory preprocessing step of parsing.

Traditional Korean POS tagging has the following three steps:

(1) Morpheme Segmentation: the input text is divided into morphological tokens suitable for further analysis.
(2) POS Tagging: POS tags are assigned to each token.
(3) Original Form Recovery: Original morpheme forms are recovered.

This work was supported by an Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korean Government (MSIP) (No. 2013-0-00131, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services). This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No. NRF-2016R1C1B1014124).
Authors' addresses: S. Jung, C. Lee (corresponding author), and H. Hwang; emails: hugmanskj@gmail.com, leeck@kangwon.ac.kr, hhs4322@gmail.com.

http://dx.doi.org/10.1145/3178458

Table 1. Example of Korean POS Tagging Process

Input	오늘 경기에서도 정말 잘했다. (You played really well in the game today.)
Tokenization	오늘 경기 에서 도 정말 잘 했다.
POS Tagging	오늘/NNG 경기/NNG 에서/JKB 도/JX 정말/MAG 잘/MAG 했다/XSV+EF ./SF
Original Form Recovery	오늘/NNG 경기/NNG 에서/JKB 도/JX 정말/MAG 잘/MAG 하/XSV 았/EP 다/EF ./SF

Analysis errors can be propagated to next processing modules due to cascade manner of integration.

An example of Korean POS tagging is presented in Table 1. In the tokenization step, the sentence is segmented into morphological units. Note that Korean sentences have space symbols, but, unlike English, the symbols do not guarantee the existence of morphological boundaries (e.g., '경기에서도' is segmented into subunits '경기,' '에서,' and '도'). In the POS tagging step, appropriate POS tags are assigned to each unit according to their context. In the original form recovery step, the original form of each morpheme is recovered (e.g., 했다 → 해/XSV, 았/EP, 다/EF).

Usually, these three processing steps are integrated in a cascade manner. Therefore, any analysis of the quality of each step is totally dependent on the results of the previous step. This means that errors in a previous step are propagated to the following process. Hence, Korean POS tagging is difficult to implement and maintain in production systems because developers must consider three modules at the same time.

In this article, we introduce a novel End-to-End (E2E) approach for Korean POS tagging that allows morphological tokenization, POS tagging, and original form recovery to be processed simultaneously.

The remainder of this article is structured as follows. Section 2 discusses some related work on POS tagging. Section 3 describes the formulation of a POS tagging problem in an E2E fashion and presents the detailed structure of the Copying mechanism used by our model. Section 4 discusses the experimental results. Finally, Section 5 concludes the article.

2 RELATED WORK

The traditional Korean POS tagging problem consists of three sub problems-morphological tokenization, POS tagging, and original form recovery. Dictionary-based heuristic tokenization approaches are used for the tokenization step [3], in which many linguistic rules are considered to determine the boundaries of each token. For POS tagging, machine learning-based approaches such as Conditional Random Fields and Support Vector Machines have been used to determine appropriate tags for each token according to the context [3, 6, 7]. To recover the original form, a lexical-original form pattern dictionary has been used [3, 7].

Recently, deep neural network-based sequence-to-sequence (seq2seq) models have been applied to many aspects of natural language processing [8]. For Korean, in particular, seq2seq models for POS tagging have been introduced [4], and seq2seq approaches have been employed for structural parsing [2].

Previous seq2seq POS tagging models suffer from out-of-vocabulary (OOV) and lower proper noun probability assignment problems. To overcome these, the proposed approach adopts Input-Feeding and a Copying mechanism for E2E Korean POS tagging. The main contributions of our work are as follows:

- We propose a new E2E Korean POS tagging architecture.
- Input-Feeding and a Copying mechanism are shown to provide effective remedies for the rare word token decoding and tagging problems.
- We report competitive Korean POS tagging performance without any external knowledge.

3 END-TO-END NEURAL ARCHITECTURE FOR KOREAN POS TAGGING

3.1 Sequence-to-Sequence Grammar Modeling

Recently, syntactic constituency parsing was formulated as a seq2seq problem by linearizing the parse tree [8]. In this approach, encoding the network captures the syntactic and semantic representation of a sentence, and decoding the network generates corresponding grammatical symbols.

Similarly, a seq2seq parsing formulation was developed for phrase structure parsing in Korean [2], and the results to date suggest state-of-the-art Korean phrase structure parsing performance.

We can apply a seq2seq parsing formulation to the POS tagging problem (Figure 1(a)). The encoding sentence can be formulated with a Gated Recurrent Unit (GRU) as follows:

$$\begin{aligned}
\overrightarrow{h}_s &= GRU_{forward}(E_{src}(x_s), \overrightarrow{h}_{s-1}) \\
\overleftarrow{h}_s &= GRU_{backward}(E_{src}(x_s), \overleftarrow{h}_{s+1}) \\
\overleftrightarrow{h}_s &= [\overrightarrow{h}_s; \overleftarrow{h}_s],
\end{aligned}$$

where $\overrightarrow{h}_s$ and $\overleftarrow{h}_s$ are forward and backward hidden states over the input sequence, respectively. $E_{src}(x_s)$ is the word embedding function, which returns a distributed vector representation of word x at time s. $\overleftrightarrow{h}_s$ is a concatenation of two vectors, $\overrightarrow{h}_s$ and $\overleftarrow{h}_s$. It contains summaries of both the preceding inputs and the following inputs. GRU is a gating mechanism for recurrent neural networks (RNNs), offering similar performance to long short-term memory with superior computational efficiency.

The encoded sentence representation is directly connected to the first GRU cell in the decoder. The POS tag generated by the decoder is defined as

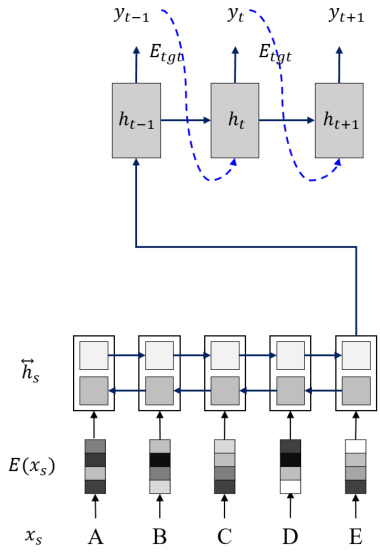
$$\begin{aligned}
c_t &= \overleftrightarrow{h}_T \\
h_t &= GRU_{decoder}(E_{tgt}(y_{t-1}), c_t, h_{t-1}) \\
y_t &= softmax(W_{y,h}h_t + W_{y,y}E_{tgt}(y_{t-1}) + W_{y,c}c_t + b_y),
\end{aligned}$$

where h_t is the hidden state of the decoder, E_{tgt} is the POS tag symbol morpheme embedding, and y_t is a corresponding POS tag and morpheme at decoding time t. The *softmax* function ($e^{z_j} / \sum_{k=1}^{K} e^{z_k}$ for $j = 1, \ldots, K$) is used in the final layer to map a vector and a specific class index to a real value.

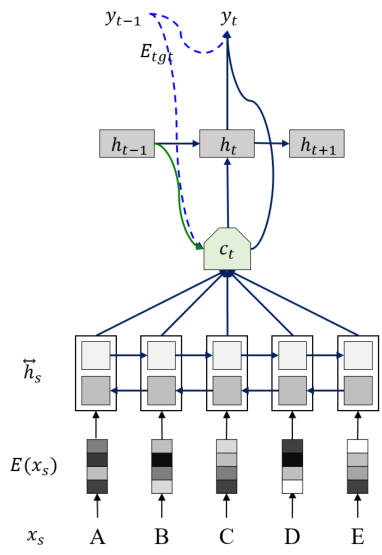
3.2 Attention Mechanism with Input-Feeding

An important extension of the seq2seq model is the addition of an attention mechanism. As sentence encoding is temporarily conducted by only the RNN units, traditional seq2seq models perform poorly with long sentences. The attention mechanism tackles this problem by blending multiple vectors globally. We can extend the seq2seq model by modifying the decoding process (Figure 1(b)) as follows:

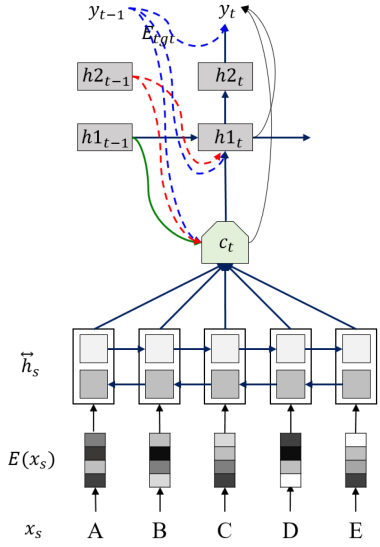
$$e_s^t = V_{att} \cdot \tanh(W_{att}[E_{tgt}(y_{t-1}); h_{t-1}; \overleftrightarrow{h}_s])$$



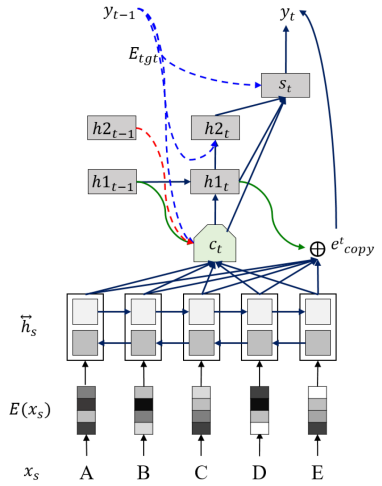
(a) Basic seq2seq POS tagging model



(b) Seq2seq POS tagging model with attention mechanism



(c) Seq2seq POS tagging model with attention mechanism and Input-Feeding



(d) Seq2seq POS tagging model with attention, Input-Feeding, and Copying mechanism

Fig. 1. Comparisons of E2E POS tagging neural architectures.

$$a_s^t = \frac{\exp(e_s^t)}{\sum_{i=1}^{T} \exp(e_i^t)}$$

$$c_t = \sum_{i=1}^{T} a_i^t \overleftrightarrow{h}_i$$

$$h_t = GRU_{decoder}(E_{tgt}(y_{t-1}), c_t, h_{t-1})$$
$$y_t = softmax(W_{y,h}h_t + W_{y,y}E_{tgt}(y_{t-1}) + W_{y,c}c_t + b_y).$$

The decoding process can be improved by feeding the previous context input to the current decoding network, thus ensuring that it is sensitive to past tagging decisions. This technique is called Input-Feeding [5]. Input-Feeding can be defined (Figure 1(c)) as follows:

$$e_s^t = V_{att} \cdot \tanh(W_{att}[E_{tgt}(y_{t-1}); h1_{t-1}; h2_{t-1}; \overleftrightarrow{h}_s])$$
$$a_s^t = \frac{\exp(e_s^t)}{\sum_{i=1}^{T}\exp(e_i^t)}$$
$$c_t = \sum_{i=1}^{T} a_i^t \overleftrightarrow{h}_i$$
$$h1_t = GRU_{decoder}(E_{tgt}(y_{t-1}), c_t, h1_{t-1}, h2_{t-1})$$
$$h2_t = RELU(W_f h1_t + b_f)$$
$$y_t = softmax(W_{y,h1}h1_t + W_{y,h2}h2_t + W_{y,y}E_{tgt}(y_{t-1}) + W_{y,c}c_t + b_y),$$

where $h1$ and $h2$ denote the output from the first and second decoding hidden layers. RELU is a rectified linear unit, defined as $max(0, x)$, where x is the input to a neuron.

3.3 Copying Mechanism for POS Tagging

Traditional seq2seq grammar tagging struggles to assign a lower proper noun probability and to decode rare words. To overcome these problems, we apply a copying mechanism that assigns higher probabilities to rare or OOV words, resulting in better sequence generation during decoding.

The Copying mechanism has two modes: *generative* and *copy*. In generative mode, we assume a vocabulary $v = \{v_1, v_2, \ldots, v_N\}$ and assign a value of UNK to any OOV word. In copy mode, we have another set of words for all the unique words in the source sentence $\chi = \{x_1, \ldots, x_T\}$, where T is the sentence length. As χ may contain words that are not in v, copying a sub-sequence in χ enables the network to output some OOV words [1].

The probability of generating target word y_t given input sentence $X = \{x_1, x_2, \ldots, x_T\}$ is calculated by the mixture of probabilities as follows:

$$e_s^t = V_{att} \cdot \tanh(W_{att}[E_{tgt}(y_{t-1}); h1_{t-1}; h2_{t-1}; \overleftrightarrow{h}_s])$$
$$a_s^t = \frac{\exp(e_s^t)}{\sum_{i=1}^{T}\exp(e_i^t)}$$
$$c_t = \sum_{i=1}^{T} a_i^t \overleftrightarrow{h}_i$$
$$s_t = W_{y,h1}h1_t + W_{y,h2}h2_t + W_{y,y}E_{tgt}(y_{t-1}) + W_{y,c}c_t + b_y$$
$$e_{copy_s}^t = \tanh(W_{copy}\overleftrightarrow{h}_s) \cdot h1_t$$
$$p(y_t|X) = \begin{cases} \frac{1}{Z}(\exp(s_t) + \sum_{j:x_j=y_t}\exp(e_{copy_j}^t) & , y_t \in \chi \\ \frac{1}{Z}\exp(s_t) & , otherwise \end{cases}.$$

Comparisons of traditional seq2seq models, attention-based models, and the proposed Input-Feeding and Copying model are shown in Figure 1(d).

Table 2. Example of Network Input and Output

Text	오른쪽으로 가세요 (Go to the right)
Network Input	오 른 쪽 으 로 <sp> 가 세 요
Network Output	오 른 쪽 <NNG> <sp> 으 로 <JKB> 가 <VV> 시 <EP> 어요 <EF>
Tagged Result	오른쪽/NNG 으로/JKB 가/VV 시/EP 어요/EF

Table 3. Comparison of F-score Given by Previous Approaches and the Proposed Method (*Denotes a Different Test Dataset)

	dev	test
CRF [7] with EX knowledge	—	97.65*
S-SVM [3] with EX knowledge	—	98.03
Sequence-to-Sequence [4]	—	97.15*
Our (sequence-to-sequence + attention)	96.56	95.92
Our + Input-Feeding	97.50	96.87
Our + Input-Feeding + Copying	**97.64**	**97.08**

3.4 Input and Output Design

Table 2 presents example input and output for the proposed model. The input text is split character by character, with spaces replaced by the <sp> symbol. The processed input is fed into the embedding layer of the encoder, and the corresponding POS-tagged results are generated by the decoder. Sequentially generated tokens before <tag> (e.g., 오 른 쪽) are interpreted as a morpheme, and subsequential tag symbols (e.g., <NNG>) are interpreted as POS tags.

4 EXPERIMENTS

The Sejong Korean POS tagging dataset[1] was used to test our proposed model. This dataset has 97,410 sentences, and these were split into training, development, and test sets containing 88,225, 1,000, and 8,185 sentences, respectively. Both the source and target embedding used 200-dimensional vectors, and the hidden layers used 1,000-dimensional vectors.

The F1-score was used to measure the performance of POS tagging. This metric can be interpreted as a weighted average of the precision and recall. The precision is the ratio $tp/(tp + fp)$, where tp is the number of true positives and fp is the number of false positives. The recall is the ratio $tp/(tp + fn)$, where fn is the number of false negatives. The formula for the F1-score is

$$F1 = 2 * \frac{precision * recall}{precision + recall}.$$

Table 3 compares the scores achieved by the proposed model with those of other methods. The S-SVM method [3] produces the best performance on the same test set, but this approach uses a large verb-recovery dictionary to find the original morpheme form and requires intensive feature engineering, whereas our method does not need a dictionary or handcrafted features. Our model

[1] http://www.sejong.or.kr.

Table 4. Example of POS Tagging Results with and without Copying Mechanism

		POS tagging Result
Ex.1	Reference	..... 도쿄/NNP ./SP **삿포로/NNP** 등/NNB 의/JKG ...
	With Copying	..... 도쿄/NNP ./SP **삿포로/NNP** 등/NNB 의/JKG ...
	Without Copying	..... 도쿄/NNP ./SP **엑포로/NNP** 등/NNB 의/JKG ...
Ex. 2	Reference	..... 윈도/NNG 2000/SN 이나/JC **윈도/NNG 엑스피/NNG** (/SS XP/SL)/SS ...
	With Copying	..... 윈도/NNG 2000/SN 이나/JC **윈도엑스피/NNP** (/SS XP/SL)/SS ...
	Without Copying	..... 윈도/NNG 2000/SN 이나/JC **윈도셰스피/NNP** (/SS XP/SL)/SS ...
Ex. 3	Reference	..... 업체/NNG 들/XSN 은/JX 신종/NNG 웜/NNG 의/JKG 진단/NNG ...
	With Copying	..... 업체/NNG 들/XSN 은/JX 신종/NNG 웜/NNG 의/JKG 진단/NNG ...
	Without Copying	..... 업체/NNG 들/XSN 은/JX 신종/NNG 늪/NNG 의/JKG 진단/NNG ...

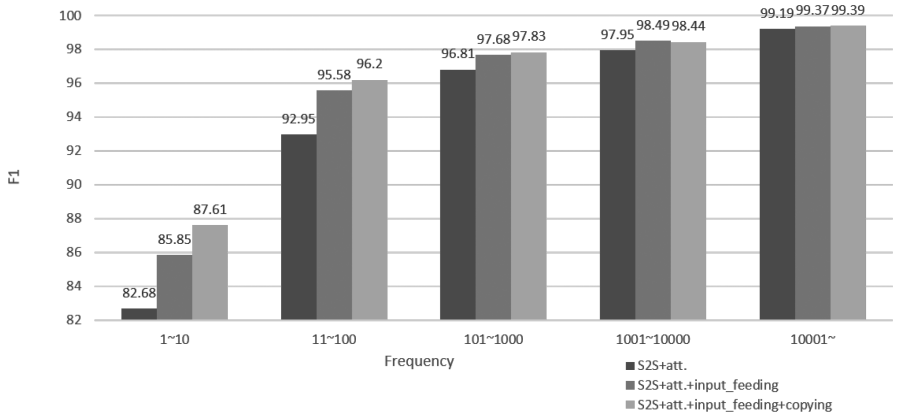


Fig. 2. Performance over frequency of input token. For example, 11~100 stands for a collection of sentences that contain a rare word whose frequency is between 11 and 100. The frequency refers to the training data.

exhibits competitive E2E performance without any external knowledge or manual effort by the developer.

Table 4 presents examples of POS tagging results with/without the Copying mechanism. The proper noun 삿포로, which is a very rare word in the training data (occurring only 3 times), is wrongly generated as 엑포로 (where 엑스포 occurs 47 times in the training data) by the baseline model, but it is correctly tagged with the Copying mechanism (Ex. 1). In Ex. 2, the proper noun 윈도엑스피 is tagged correctly by the Copying mechanism, but the segmentation is wrong.

We can see from Figure 2 that the proposed model is more robust to rare words. In the case of very rare words, the model with attention, Input-Feeding, and the Copying mechanism clearly

achieves better results. Additionally, we can see that the Input-Feeding enhanced attention model is superior to the pure attention RNN decoder.

5 CONCLUSION

In this article, we have shown that a character-level seq2seq model with Input-Feeding and a Copying mechanism can achieve competitive performance with state-of-the-art approaches, including carefully tuned SVM models. The improvement in F-score from 95.92 to 97.08 is a result of several key components described in this article to deal with the generation of rare words and POS tagging. Unlike previous studies, our model does not require either manual feature selection or knowledge such as a lexical-original form pattern dictionary. We believe that the proposed architecture could be successfully applied to POS tagging in other languages, and such applications are planned for future work.

REFERENCES

[1] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* 1 (2016), 1631–1640.

[2] Hyunsun Hwang and Changki Lee. 2016. Korean phrase structure parsing using sequence-to-sequence learning. In *HCLT*.

[3] Changki Lee, Junseok Kim, Jeonghee Kim, and Hyunki Kim. 2013. Joint models for korean word spacing and POS tagging using structural SVM. *J. KIISE: Softw. Appl.* 40, 12 (2013), 826–832.

[4] Jianri Li, EuiHyeon Lee, and Jong-Hyeok Lee. 2017. Sequence-to-sequence based morphological analysis and part-of-speech tagging for korean language with convolutional features. *Journal of KIISE* 44, 1 (2017), 57–62.

[5] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv:1508.04025* (2015).

[6] Seung-Hoon Na. 2015. Conditional random fields for korean morpheme segmentation and POS tagging. *ACM Trans. Asian Low-Resource Lang. Inf. Process.* 14, 3 (2015), 10.

[7] Seung-Hoon Na and Young-Kil Kim. 2014. Phrase-based statistical model for korean morpheme segmentation and POS tagging. *Korea Computer Congress.* 571–573.

[8] Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems.* 2773–2781.

Received May 2017; revised September 2017; accepted December 2017