applied sciences

MDPI

Article

Integrated Model for Morphological Analysis and Named Entity Recognition Based on Label Attention Networks in Korean

Hongjin Kim [1] **and Harksoo Kim** [2,*]

[1] Computer and Communications Engineering, Kangwon National University, Chuncheon 24341, Korea; jin3430@kangwon.ac.kr
[2] Computer Science and Engineering, Konkuk University, Seoul 05029, Korea
* Correspondence: nlpdrkim@konkuk.ac.kr; Tel.: +82-2-450-3499

Received: 23 April 2020; Accepted: 25 May 2020; Published: 28 May 2020

check for updates

Featured Application: Core technology for robust information extraction systems.

Abstract: In well-spaced Korean sentences, morphological analysis is the first step in natural language processing, in which a Korean sentence is segmented into a sequence of morphemes and the parts of speech of the segmented morphemes are determined. Named entity recognition is a natural language processing task carried out to obtain morpheme sequences with specific meanings, such as person, location, and organization names. Although morphological analysis and named entity recognition are closely associated with each other, they have been independently studied and have exhibited the inevitable error propagation problem. Hence, we propose an integrated model based on label attention networks that simultaneously performs morphological analysis and named entity recognition. The proposed model comprises two layers of neural network models that are closely associated with each other. The lower layer performs a morphological analysis, whereas the upper layer performs a named entity recognition. In our experiments using a public gold-labeled dataset, the proposed model outperformed previous state-of-the-art models used for morphological analysis and named entity recognition. Furthermore, the results indicated that the integrated architecture could alleviate the error propagation problem.

Keywords: morphological analysis; named entity recognition; integrated neural network model

1. Introduction

A morpheme refers to the smallest meaningful word in a phrase. In Korean, morphological analysis (MA) is generally performed in the order of morpheme segmentation and part-of-speech (POS) annotation. Based on a Korean sentence, all possible morphemes and their POS tags are suggested through morpheme segmentation. Subsequently, the most suitable morphemes and their POS tags are determined through POS annotation. A named entity (NE) refers to morpheme sequences with specific meanings, such as person, location, and organization names. Named entity recognition (NER) is a subtask of information extraction that identifies NEs in sentences and classifies them into predefined classes. Most NEs are composed of a combination of specific POSs, such as a proper noun, general noun, and number. Therefore, many NER models generally use the results of morphological analysis as informative clues [1,2]. However, this pipeline architecture causes the well-known error propagation problem. In other words, errors of MA directly deteriorate the performances of NER models. MA models for agglutinative languages, such as Korean and Japanese, demonstrate worse performances than those of isolating languages, which significantly affect the performances of the corresponding NER models. Moreover, in languages such as Korean and Japanese that do not use capitalization, detecting NEs without any morphological information such as morpheme boundaries and POS tags is difficult. Table 1 shows an example of named entities affected by MA results in Korean.

Table 1. Example of Korean named entities affected by morphological analysis (MA) results (NP, NNG, JKB, VV, and EF are Korean part-of-speech (POS) tags, and ORG is a Korean named entity (NE) tag).

NLP Step	Incorrect Results
MA	u-ri/NP + eun-haeng/NNG + e/JKB ga/VV + da/EF
NER	N/A
NLP Step	**Correct Results**
MA	u-ri-eun-haeng/NNP + e/JKB ga/VV + da/EF
NER	u-ri-eun-haeng/ORG

In Table 1, to increase readability, we have romanized Korean characters (so-called *Hangeul*) and hyphenated Korean characters (so-called *eumjeols*). The sentence "u-ri-eun-haeng-e ga-da" means "I go to Woori bank." In an incorrect MA result, "u-ri" and "eun-haeng" are incorrectly analyzed as a pronoun (NP) and a general noun (NNG), respectively. This incorrect result yields an incorrect NER result, i.e., "not existing (N/A)" instead of "organization (ORG)." To reduce these error propagation problems, we present an integrated model, in which MA and NER are performed at once.

The remainder of this paper is organized as follows: in Section 2, we summarize previous studies on MA and NER; we propose the integrated model in Section 3; we explain the experimental setup and evaluate the proposed model in Section 4; finally, we conclude our study in Section 5.

2. Previous Studies

MA and NER are considered to be sequence-labeling problems, where POS and NE tags are annotated to a word sequence. For sequence labeling, most previous studies have used statistical-based machine learning (ML) methods, such as structural support vector machine (SVM) [3] and conditional random fields (CRFs) [4]. A method for unknown morpheme estimation using SVM and CRF has been proposed [5]. However, ML models depend on the training corpus size and manually designed features. To resolve these problems, studies based on deep learning have been conducted. Many MA and NER studies have used recurrent neural network (RNN) [6,7]. NER was performed using bidirectional long short-term memory (Bi-LSTM) and CRFs [1]. In another study, an attention mechanism and a gated recurrent unit (GRU) were used, which reduced the number of gates and time complexity of LSTM [8]. An effective method for reflecting external knowledge (i.e., NE dictionary) into Bi-GRU-CRFs was proposed [9]. Additionally, RNNs and CRFs have been used in MA studies [10,11]. To alleviate MA error propagation, an integrated model that simultaneously performs MA and NER has also been studied, which used two layers of Bi-GRU-CRFs [12]. Güngör et al. [13] proposed a model which alleviates morphological ambiguity by jointly learning NER and morphological disambiguation taggers using Bi-LSTM-CRFs for Turkish. As mentioned above, many ML models have used CRFs to obtain optimal paths among all possible label sequences. However, these models did not always yield good performances. Bi-LSTM-Softmax [14] demonstrated better performance than Bi-LSTM-CRFs for POS tagging. To obtain optimal label paths better than those obtained with CRFs, a label attention network (LAN) was proposed, which captured the potential long-term label dependency by providing incrementally refined label distributions with hierarchical attention to each word. Therefore, we adopted this LAN in our integrated model.

3. Integrated Model for MA and NER

For n characters, $C_{1,n}$, in a sentence S, let $M_{1,n}$ and $NE_{1,n}$ denote a morpheme tag sequence and an NE tag sequence in S, respectively. Table 2 shows morpheme tags and NE tags that are defined according to the character-level BIO (beginner–inner–outer) tagging scheme.

Table 2. Morpheme tags and NE tags.

Morpheme Tag	**Descriptions**
B-(NOUN|ADJ|ADV| . . .)	Beginner of a morpheme with the POS following "B-"
I-(NOUN|ADJ|ADV| . . .)	Inner of a morpheme with the POS following "I-"
O	Outer of any morphemes
NE Tag	**Descriptions**
B-(PER|LOC|ORG)	Beginner of an NE with the category following "B-"
I-(PER|LOC|ORG)	Inner of an NE with the category following "I-"
O	Outer of any NEs

The integrated model, known as (MANE), can then be formally expressed using the following equation:

$$MANE(S) \stackrel{def}{=} \underset{M_{1,n},NE_{1,n}}{\arg} \max P(M_{1,n}, NE_{1,n} | C_{1,n}) \tag{1}$$

According to the chain rule, (1) can be rewritten as the following equation:

$$MANE(S) \stackrel{def}{=} \underset{M_{1,n},NE_{1,n}}{\arg} \max P(M_{1,n} | C_{1,n}) P(NE_{1,n} | C_{1,n}, M_{1,n}) \tag{2}$$

To obtain the sequence labels $M_{1,n}$ and $E_{1,n}$ that maximize (2), we adopted a bidirectional long short-term memory with a label attention network (Bi-LSTM-LAN), as shown in Figure 1.

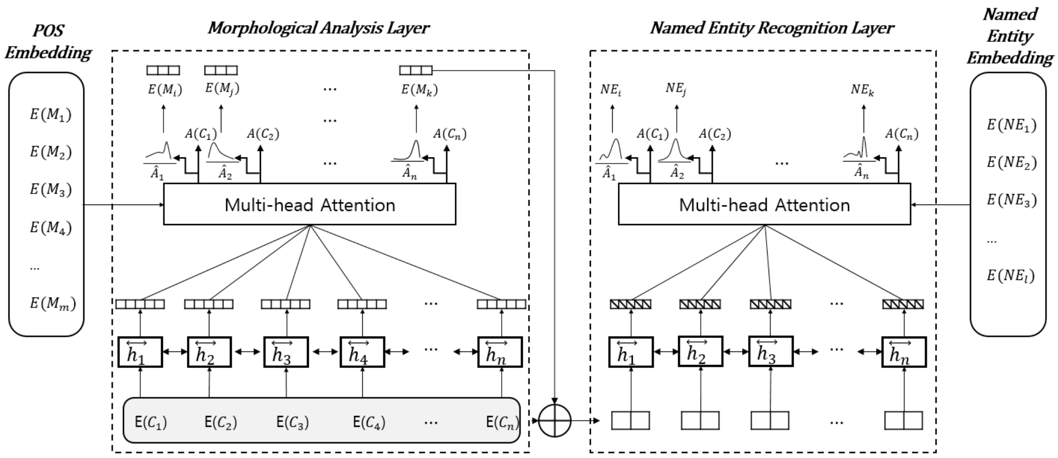

Figure 1. Overall architecture of MANE.

MANE comprises two layers of Bi-LSTM-LAN: an MA layer, shown on the left, and an NER layer, shown on the right. The input unit of the MA layer is a character, and each character is represented by a concatenation of three types of embeddings: character, alphabet, and feature embeddings, as shown in Figure 2.

In Figure 2, C_i is the i-th character in a sentence, and $E_c(C_i)$ is a character embedding of C_i. Each character embedding is represented by a randomly initialized n-dimensional vector and fine-tuned during training. To render MANE robust to typographical errors, we additionally represent each character through an alphabet embedding. A Korean character consists of a first consonant called *chosung*, a vowel called *joongsung*, and a final consonant called *jongsung* that can be omitted. For example, in the word "hak-kyo (school)", the first character "hak" comprises three alphabets; "h" called *chosung*, "a" called *joongsung*, and "k" called *jongsung*. On the other hand, the second character "kyo" comprises two alphabets; "k" called *chosung* and "yo" called *joongsung*. In Figure 2, $E_a\left(C_i^j\right)$ is an alphabet embedding of the j-th alphabet in C_i that comprises the maximum of three alphabets in

Korean, and each alphabet embedding is represented in the same manner as the character embeddings. The maximum three alphabet embeddings are passed into a convolutional neural network (CNN) with 100 filters (filter widths: 1, 2, and 3) [15]. In NER, dictionary look-up features— which are used to check whether there is an input word in a preconstructed NE dictionary—significantly affect the performance. Based on Kim's study [9], in which effective dictionary look-up features have been proposed for Korean NER, we adopted the same dictionary look-up features in MANE. In Figure 2, $E_f(C_i)$ is a feature embedding of C_i based on looking up a predefined NE dictionary. Subsequently, the character, alphabet, and feature embeddings are concatenated into the input embedding $E(C_i)$, as shown in Figure 1.

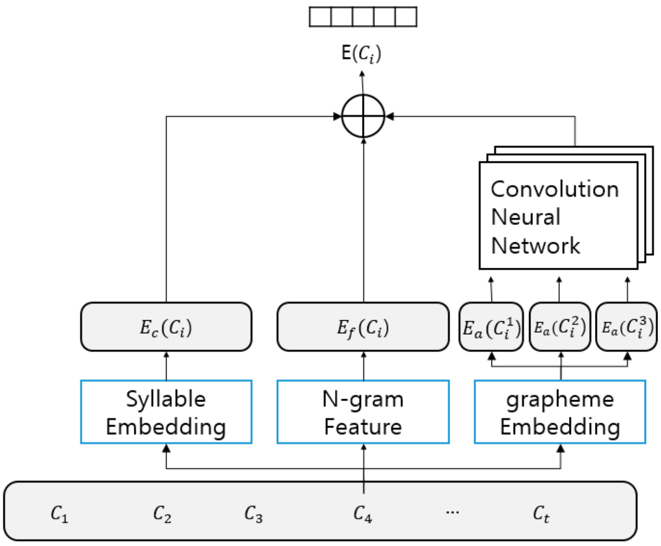

Figure 2. Input unit of MANE.

In the MA layer, the input embeddings $E(C) = \{E(C_1), E(C_2), \ldots, E(C_n)\}$ of the n characters in a sentence are fed into a Bi-LSTM to yield a sequence of forward-hidden and backward-hidden states, respectively. Subsequently, these two states are concatenated to reflect bidirectional contextual information, as shown in the following equation:

$$\begin{aligned} \overrightarrow{h}_i &= LSTM\left\{E(C_i),\ \overrightarrow{h}_{i-1}\right\}, \\ \overleftarrow{h}_i &= LSTM\left\{E(C_i),\ \overleftarrow{h}_{i-1}\right\}, \\ \overleftrightarrow{h}_i &= \left[\overrightarrow{h}_i; \overleftarrow{h}_i\right], \\ \overleftrightarrow{H} &= \left\{\overleftrightarrow{h}_1,\ \overleftrightarrow{h}_2,\ \ldots,\ \overleftrightarrow{h}_n\right\}, \end{aligned} \tag{3}$$

where $\left[\overrightarrow{h}_i; \overleftarrow{h}_i\right]$ is the concatenation of forward hidden state $\overrightarrow{h}_i$ and backward hidden state $\overleftarrow{h}_i$ of the i-th character in a sentence. Next, the degrees of association between the contextualized input embeddings $\overleftrightarrow{H} = \left\{\overleftrightarrow{h}_1,\ \overleftrightarrow{h}_2, \ldots\ \overleftrightarrow{h}_n\right\}$ and the morpheme tag embeddings $E(M) = \{E(M_1), E(M_2), \ldots, E(M_m)\}$ are calculated based on a multihead attention mechanism [14], as shown in the following equation:

$$\begin{aligned} head_j &= attention\left\{QW_j^Q,\ KW_j^K,\ VW_j^V\right\} = \alpha_j * VW_j^V, \text{ where } \mathrm{Q} = \overleftrightarrow{H},\ K = V = E(M), \\ \alpha_j &= softmax\left(\frac{QW_j^Q * \left(KW_j^K\right)^T}{\sqrt{d_h}}\right), \\ A(C_i) &= head_1 \bigoplus head_2 \bigoplus \cdots \bigoplus head_k, \end{aligned} \tag{4}$$

where W_j^Q, W_j^K, and W_j^V are the weighting parameters of the j-th parameter among k heads to be learned during training. The morpheme tag embeddings $E(M)$ represent the embedding vectors of the m morpheme tags that are randomly initialized and fine-tuned during training. The attention score α_j is calculated using a scaled-dot product, where d_h is a normalization factor and denotes that the hidden size of Bi-LSTM is the same as the dimension of the morpheme tag embeddings. The attention score vector $A(C_i)$ represents the degrees of association between the contextualized input embedding $\overleftrightarrow{h}_i$ of the i-th input character and each morpheme tag. In other words, it can be considered as a potential distribution of morpheme tags associated with an input character. In the prediction phase, the MA layer outputs the morpheme tags, as shown in the following equation:

$$M_i = argmax\left(\hat{A}_i^1,\ \hat{A}_i^2,\ \ldots,\hat{A}_i^m\right), \tag{5}$$

where $\hat{A}_i^j$ denotes the j-th one among m attention scores in the trained attention vector $\hat{A}_i$.

In the NER layer, the i-th input embedding $E(C_i)$ is concatenated to the embedding of the morpheme tag with a maximum attention score, $E(M_i)$. Subsequently, the concatenated vectors are fed into a Bi-LSTM in the same manner that is used for the MA layer, as shown in the following equation:

$$\begin{gathered} \overrightarrow{h}_i = LSTM\left\{[E(C_i); E(M_i)],\ \overrightarrow{h}_{i-1}\right\}, \\ \overleftarrow{h}_i = LSTM\left\{[E(C_i); E(M_i)],\ \overleftarrow{h}_{i-1}\right\}, \\ \overleftrightarrow{h}_i = \left[\overrightarrow{h}_i; \overleftarrow{h}_i\right], \\ \overleftrightarrow{H} = \left\{\overleftrightarrow{h}_1,\ \overleftrightarrow{h}_2,\ \ldots\ \overleftrightarrow{h}_n\right\} \end{gathered} \tag{6}$$

Next, the attention scores between the contextualized input embeddings $\overleftrightarrow{H} = \left\{\overleftrightarrow{h}_1,\ \overleftrightarrow{h}_2, \ldots\ \overleftrightarrow{h}_n\right\}$ and the NE tag embeddings $E(NE) = \{E(NE_1), E(NE_2), \ldots, E(NE_l)\}$ are calculated using the same multihead attention mechanism as the MA layer. The attention score vector $A(C_i)$ represents the degrees of association between the contextualized input embedding $\overleftrightarrow{h}_i$ and each NE tag.

Generally, open datasets for training MA models are larger than those for training NER models. Thus, we use a two-phase training scheme in order to optimize the hyperparameters of MANE using different sizes of training data; large POS-tagged data and small NE-tagged data. We first train the MA layer based on the cross-entropy between the correct POS tags, M_i, and the outputs of the MA layer, $\hat{M}_i$, as shown in the following equation:

$$H_{\hat{M}}(M) = -\sum_i \hat{M}_i \log(M_i) \tag{7}$$

In other words, the outputs of the NER layer do not take part in the first training phase. Subsequently, we train all layers based on the cross-entropy between the correct NE tags, NE_i, and the outputs of the NER layer, $\widehat{\mathrm{NE}}_i$, as shown in the following equation:

$$H_{\widehat{\mathrm{NE}}}(NE) = -\sum_i \widehat{\mathrm{NE}}_i \log(NE_i) \tag{8}$$

The outputs of the MA layer do not take part in the second training phase. We expect the hyperparameters in the MA layer to be fine-tuned to the values associated with the correct NE tags in the second training phase.

4. Evaluation

4.1. Datasets and Experimental Setups

For our experiments, we used two gold-labeled corpora: one for evaluating MA models, and the other for evaluating NER models. The first corpus was the 21st century Sejong POS-tagged corpus [16], as shown in Table 3.

Table 3. Summary of the 21st century Sejong corpus.

Description	Numbers
Sentences	833,386
Morphemes	21,034,232
Tags	46

The second corpus was the public NE-tagged corpus (5000 sentences) used in the 2016 Korean Information Processing System Competition [17], as shown in Table 4.

Table 4. Summary of the public NE-tagged corpus.

Description	Numbers
Person	3416
Location	2611
Organization	4010
Date	2688
Time	388

We converted the POS-tagged and NE-tagged corpora into a morpheme dataset and an NE dataset, in which the characters were annotated with morpheme tags and NE tags, as shown in Table 2. Subsequently, we divided the morpheme datasets and the NE datasets into training datasets and test datasets, respectively, at a ratio of 9:1. Finally, we evaluated MANE in terms of the following evaluation measures:

$$Accuracy = \frac{\#\,of\ correct\ morpheme\ tags\ or\ NE\ tags}{\#\,of\ morpheme\ tags\ or\ NE\ tags\ returned\ by\ a\ system} \tag{9}$$

$$Precision = \frac{\#\,of\ correct\ morpheme\ or\ NE}{\#\,of\ morpheme\ or\ NE\ returned\ by\ a\ system} \tag{10}$$

$$Recall = \frac{\#\,of\ correct\ morpheme\ or\ NE\ returned\ by\ a\ system}{\#\,of\ correct\ morpheme\ or\ NE\ in\ a\ test\ data} \tag{11}$$

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

4.2. Implementation

We implemented MANE using PyTorch 0.3.1. Training and prediction were performed on a per-sentence level. We set the sizes of the character, morpheme tag, and NE tag embeddings to 50, 128, and 128, respectively. Subsequently, we randomly initialized and fine-tuned these embeddings. For alphabet embedding, we set the number and sizes of the CNN filters to 100 and 1, 2, 3, respectively. Next, we set the number of attention heads to 4. The training required 100 epochs and was performed by mini-batch stochastic gradient descent, based on the Adam optimizer, with a fixed learning rate of 0.001. Each mini-batch comprised 32 sentences due to our hardware limitation. The length of each sentence was fixed to 200 which was the maximum length of a sentence in the training data. For short sentences, the remainder of the input units were filled with padding.

4.3. Experimental Results

Our first experiment was to compare the MA performances of MANE with those of the previous state-of-the-art MA models, as shown in Table 5.

Table 5. Performance comparison on morphological analysis.

Model	Accuracy	F1-Score
Structural SVMs [18]	0.9802	0.9803
Bi-LSTM-CRFs-MA [10]	0.9780	0.9877 (0.9750)
Stacked Bi-GRU-CRFs-MA [19]	0.9840 (0.9590)	-
Seq2Seq [11]	-	0.9793
MANE-MA	0.9793	0.9789
MANE	0.9886	0.9880

In Table 5, MANE-MA is an independent model with the same architecture as that of the MA layer in Figure 1. Structural SVMs [18] constitute an integrated model for automatic word spacing and morphological analysis in Korean. Bi-LSTM-CRFs-MA [10] and stacked Bi-GRU-CRFs-MA [19] are integrated deep learning models for the task described in [18]. For a fair comparison, we used correctly spaced input sentences in these previous integrated models. In addition, we showed the performances of a modified Bi-LSTM-CRFs-MA [10] and a modified stacked Bi-GRU-CRFs-MA [19] in which additional linguistic features, such as morpheme dictionary look-ups and pre-analysis dictionary look-ups, were excluded. The parenthesized scores denote the performances of the modified versions reported in their papers [10,19]. Seq2Seq [11] is a generative MA model based on a sequence-to-sequence network. As shown in Table 5, MANE outperformed all the comparison models. When MANE was compared to the modified versions, the performance differences were even larger. To verify the performance differences between MANE and the comparison models, we repeated a performance evaluation of MANE five times. In the repeated evaluations on MA, the performance variations of MANE were ± 0.0003 in accuracies and ± 0.0001 in F1-scores. As a result, MANE always showed higher performances than all of the previous MA models. The p-values of F1-scores between MANE and the comparison models were from 5.19×10^{-9} to 0.00537. This implies that the performance differences are statistically significant at the 0.05 level. Moreover, it showed higher performances than MANE-MA. This reveals that the NER layer can be useful in improving the performance of the MA layer.

In our second experiment, we compared the NER performances of MANE with those of the previous state-of-the-art NER models, as shown in Table 6.

Table 6. Performance comparison on named entity recognition.

Model	F1-Score
Bi-GRU-CRFs-NE [20]	0.8022
Bi-LSTM-CRFs-NE [1]	0.8549
Stacked Bi-GRU-CRFs-NE [9]	0.8576
MorpheNE [12] (MA and NER integrated)	0.8566
Attention-CRFs [8]	0.8188
MANE-NE	0.8583
MANE	0.8597

In Table 6, MANE-NE is an independent model with the same architecture as that of the NE layer in Figure 1. MANE-NE uses a pretrained MA layer for POS information. Subsequently, the parameter of the MA layer is frozen to block tuning when training the NE layer. Bi-GRU-CRFs-NE [20] constitutes a baseline NER model based on GRUs with a CRF layer. Bi-LSTM-CRFs-NE [1] represents an NER model, in which a word representation is expanded using word, POS, and syllable embeddings, as well as dictionary look-up features. Stacked Bi-GRU-CRFs-NE [9] constitutes an NER model with two layers of Bi-GRU-CRFs, in which effective dictionary look-up features were used. Attention-CRFs [8] perform NER based on the attention mechanism and the CRFs. MANE-NE is the independent NER model in Table 5. Therefore, correctly POS-tagged sentences were used as input. In Table 6, MorpheNE [12] is an integrated model for MA and NER based on Bi-GRU-CRFs. MorpheNE did not use correctly POS-tagged sentences as inputs. As shown in Table 6, the performances of Bi-GRU-CRFs-NE [20] and attention-CRFs [8] are inferior to those of others because these models did not use any dictionary look-up features. This reveals that the dictionary look-up features have a significant effect on the improvement of NER performances in Korean. As shown in Table 6, MANE outperformed all of the comparison models, although it did not use correctly POS-tagged sentences as inputs. In the five repeated evaluations of NER, the performance variations of MANE were ±0.0012 in F1-scores. As a result, MANE always showed higher F1-scores than all of the previous NER models. The p-values of F1-scores between MANE and the comparison models were from 1.67×10^{-8} to 0.007594. This implies that the performance differences are statistically significant at the 0.05 level. In particular, MANE performed better than MorpheNE. Moreover, MANE greatly outperformed MorpheNE in memory consumption and prediction time, as shown in Table 7.

Table 7. Comparisons of memory consumption and prediction time.

Model	**Average Memory Usage (Megabyte)**	**Average Prediction Time (Millisecond)**
MANE	4276	4.8
MorpheNE	4946	8.0

This indicates that the LAN of MANE is more effective and efficient than the CRF of MorpheNE in alleviating error propagation problems. In addition, MANE demonstrated higher performances than MANE-NE. This reveals that the proposed architecture may be a good solution to the error propagation problem.

The last experiment demonstrated the effectiveness of pretraining the MA layer using different training data sizes, as shown in Table 8.

Table 8. Performances of NER according to different training data sizes.

Size of Dataset	**Parameters**	**F1-Score**
50% of the morpheme dataset + the NE dataset	Static	0.8476
	Fine-tuned	0.8497
70% of the morpheme dataset + the NE dataset	Static	0.8531
	Fine-tuned	0.8554
90% of the morpheme dataset + the NE dataset	Static	0.8583
	Fine-tuned	0.8597

In Table 8, "static" means that the parameters in the MA layer were frozen after pretraining using the morpheme dataset, and "fine-tuned" means that the parameters in the MA layer were fine-tuned during the second training phase, in which the MA and NER layers were trained using the NE dataset. As shown in Table 8, the more training data learned in the MA layer, the better the performance was in the NE layer. In addition, Table 7 shows that the second training phase affected the improvement in the NER performances.

5. Conclusions

We proposed an integrated model based on label attention networks that simultaneously performed MA and NER. The proposed model comprised two layers of Bi-LSTM-LAN that were closely associated with each other. The lower layer performed MA, whereas the upper layer performed NER. To optimize the weighting parameters of the proposed model, we used a two-phase training scheme: in the first phase, the lower layer was trained for MA, whereas in the second phase, all layers were trained for NER. In our experiments using public datasets, the proposed model outperformed all of the previous state-of-the-art models in Korean. Moreover, the proposed integrated model demonstrated greater performances than the independent MA (i.e., the lower layer) and the independent NER models (i.e., the upper layer). Based on these experiments, we conclude that the proposed model can effectively reduce the error propagation problem caused by a pipeline architecture. Moreover, we conclude that the proposed model can provide important feedback information from the upper layer (the NER model) to the lower layer (the MA model).

Author Contributions: Conceptualization, H.K. (Harksoo Kim); methodology, H.K. (Harksoo Kim); software, H.K. (Hongjin Kim); validation, H.K. (Hongjin Kim); formal analysis, H.K. (Harksoo Kim); investigation, H.K. (Harksoo Kim); resources, H.K. (Hongjin Kim); data curation, H.K. (Hongjin Kim); writing—original draft preparation, H.K. (Hongjin Kim); writing—review and editing, H.K. (Harksoo Kim); visualization, H.K. (Harksoo Kim); supervision, H.K. (Harksoo Kim); project administration, H.K. (Harksoo Kim); funding acquisition, H.K. (Harksoo Kim). Both authors have read and agree to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant, funded by the Korean government (MSIP) (No. 2016R1A2B4007732). This work was also supported by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2013-0-00131, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services).

Acknowledgments: We thank the members of the NLP laboratory at Kangwon National University and Konkuk University for their technical support.

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

References

1. Na, S.-H.; Kim, H.; Min, J.; Kim, K. Improving LSTM CRFs Using Character-Based Compositions for Korean Named Entity Recognition. *Comput. Speech Lang.* **2019**, *54*, 106–121. [CrossRef]
2. Yu, H.; Ko, Y. Expansion of Word Representation for Named Entity Recognition Based on Bidirectional LSTM CRFs. *J. KIISE* **2017**, *44*, 306–313. (In Korean) [CrossRef]
3. Lee, C.; Jang, M. Named Entity Recognition with Structural SVMs and Pegasos Algorithm. *Korean J. Cogn. Sci.* **2010**, *21*, 655–667. (In Korean)
4. Lee, C.; Hwang, Y.-G.; Oh, H.-J.; Lim, S.; Heo, J.; Lee, C.-H.; Kim, H.-J.; Wang, J.-H.; Jang, M.-G. Fine-Grained Named Entity Recognition Using Conditional Random Fields for Question Answering. In *Proceedings of the Asia Information Retrieval Symposium, Singapore, 16–18 October 2006*; Springer: Singapore; pp. 581–587.
5. Choi, M.; Kim, H. Part-Of-Speech Tagging and the Recognition of the Korean Unknown-words Based on Machine Learning. *KTSDE* **2011**, *18*, 45–50. (In Korean)
6. Chiu, J.P.; Nichols, E. Named Entity Recognition with Bidirectional LSTM-CNNs. *Trans. Assoc. Comput. Linguist.* **2016**, *4*, 357–370. [CrossRef]
7. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural Architectures for Named Entity Recognition. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, June 2016; pp. 260–270.
8. Park, S.; Kim, H. Named Entity Recognition using Attention Mechanism. In Proceedings of the HCLT, Seoul, Korea, 12–13 October 2018; pp. 678–680. (In Korean).
9. Kim, H.; Kim, H. How to Use Effective Dictionary Feature for Deep Learning based Named Entity Recognition. In Proceedings of the HCLT, Daejeon, Korea, 11–12 October 2019; pp. 319–321. (In Korean).
10. Kim, S.-W.; Choi, S.-P. Research on Joint Models for Korean Word Spacing and POS (Part-Of-Speech) Tagging based on Bidirectional LSTMCRF. *J. KIISE* **2018**, *45*, 792–800. (In Korean) [CrossRef]
11. Choe, B.; Lee, I.; Lee, S. Korean Morphological Analyzer for Neologism and Spacing Error based on Sequence-to-Sequence. *J. KIISE* **2020**, *47*, 70–77. (In Korean) [CrossRef]
12. Lee, H.; Park, G.; Kim, H. Effective Integration of Morphological Analysis and Named Entity Recognition Based on a Recurrent Neural Network. *Pattern Recognit. Lett.* **2018**, *112*, 361–365. [CrossRef]

13. Güngör, O.; Uskudarli, S.; Güngör, T. Improving Named Entity Recognition by Jointly Learning to Disambiguate Morphological Tags. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, August 2018; pp. 2082–2092.
14. Cui, L.; Zhang, Y. Hierarchically-Refined Label Attention Network for Sequence Labeling. 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. 4106–4119.
15. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
16. Corpus, S. *21st Century Sejong Project. The National Institute of the Korean Language*; National Institute of Korean Language: Seoul, Korea, 2010.
17. Corpus. *The National Institute of the Korean Language*; National Institute of Korean Language: Seoul, Korea, 2016.
18. Lee, C. Joint Models for Korean Word Spacing and POS Tagging Using Structural SVM. *J. KIISE* **2013**, *40*, 826–832. (In Korean)
19. Kim, H.; Kim, H. Effective Integration of Automatic Word Spacing and Morphological Analysis in Korean. In Proceedings of the 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), Pusan, Korea, 19–22 February 2020; pp. 275–278.
20. Kim, S.; Choi, S. Research on the Various Neural Network Topologies for Korean NER Based on Bidirectional GRU-CRF applying Lexicon Features. *KIISE Trans. Comput. Pract.* **2019**, *25*, 99–105. (In Korean) [CrossRef]