applied sciences

MDPI

Article

Fine-Grained Named Entity Recognition Using a Multi-Stacked Feature Fusion and Dual-Stacked Output in Korean

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

[1] Artificial Intelligence, Konkuk University, 120 Neungdong-ro, Gwangjin-gu, Seoul 05029, Korea; jin3430@konkuk.ac.kr
[2] Computer Science and Engineering, Konkuk University, 120 Neungdong-ro, Gwangjin-gu, Seoul 05029, Korea
* Correspondence: nlpdrkim@konkuk.ac.kr; Tel.: +82-2-450-3499

Abstract: Named entity recognition (NER) is a natural language processing task to identify spans that mention named entities and to annotate them with predefined named entity classes. Although many NER models based on machine learning have been proposed, their performance in terms of processing fine-grained NER tasks was less than acceptable. This is because the training data of a fine-grained NER task is much more unbalanced than those of a coarse-grained NER task. To overcome the problem presented by unbalanced data, we propose a fine-grained NER model that compensates for the sparseness of fine-grained NEs by using the contextual information of coarse-grained NEs. From another viewpoint, many NER models have used different levels of features, such as part-of-speech tags and gazetteer look-up results, in a nonhierarchical manner. Unfortunately, these models experience the feature interference problem. Our solution to this problem is to adopt a multi-stacked feature fusion scheme, which accepts different levels of features as its input. The proposed model is based on multi-stacked long short-term memories (LSTMs) with a multi-stacked feature fusion layer for acquiring multilevel embeddings and a dual-stacked output layer for predicting fine-grained NEs based on the categorical information of coarse-grained NEs. Our experiments indicate that the proposed model is capable of state-of-the-art performance. The results show that the proposed model can effectively alleviate the unbalanced data problem that frequently occurs in a fine-grained NER task. In addition, the multi-stacked feature fusion layer contributes to the improvement of NER performance, confirming that the proposed model can alleviate the feature interference problem. Based on this experimental result, we conclude that the proposed model is well-designed to effectively perform NER tasks.

Keywords: fine-grained named entity recognition; *k*-stacked feature fusion; dual-stacked output; unbalanced data problem

Citation: Kim, H.; Kim, H. Fine-Grained Named Entity Recognition Using a Multi-Stacked Feature Fusion and Dual-Stacked Output in Korean. *Appl. Sci.* **2021**, *11*, 10795. https://doi.org/10.3390/app112210795

Academic Editor: Arturo Montejo-Ráez

Received: 21 September 2021
Accepted: 11 November 2021
Published: 15 November 2021

1. Introduction

Named entity recognition (NER), a well-known task in natural language processing (NLP), identifies word sequences in texts and classifies them into predefined categories. NER was initially studied as a subtask of information extraction, when coarse-grained NER systems that extract the names of people, locations, and organizations from texts were widely used. Growing interest in NLP tasks, such as relation extraction, answering questions, and knowledge base construction, has increased the demand for fine-grained NER systems. Although early NER systems performed well in coarse-grained NER, they often required well-designed features in the form of language-dependent human knowledge. To address this issue, many NER systems have adopted deep learning methods to yield state-of-the-art (SOTA) performance. Although these models based on deep learning delivered good performance, it was restricted to tasks involving coarse-grained classification. In fine-grained NER for English language tasks, certain systems based on deep learning performed satisfactorily and were between 80% and 85% accurate. However, in languages

with a large number of characters that do not use capitalization or word boundary by spacing (e.g., a spacing unit is not a word in Korean), the performance of these systems is unsatisfactory, that is, between 65% and 75% [1]. The main reason for the lower performance is that the training data for fine-grained NER are more unbalanced than those for coarse-grained NER. It is easy to find fine-grained NEs that seldom occur in training data. To alleviate this sparse data problem, we propose an NER model that compensates for the sparseness of fine-grained NEs with the contextual information of coarse-grained NEs that semantically include the fine-grained NEs. Table 1 presents examples of coarse-grained NE categories and their fine-grained NE categories.

Table 1. Example of two-level NE categories.

Coarse-Grained NE		Fine-Grained NE	
Class	**Example**	**Class**	**Example**
Location	USA, Washington, D.C., Memorial park	Country Province City	Korea Gangwon-do Seoul
Date	Thanksgiving day, 24 April 2020	Year Duration	2020 2019–2021

These examples show that the classes of coarse-grained NEs are supersets that are tightly associated with the classes of fine-grained NEs. If the fine-grained NE "Seoul" in Table 1 does not occur in the training data, the contextual information of the coarse-grained NE "Washington, D.C." could be helpful in that it would enable the NE class "City" of "Seoul" to be inferred because they are both capital cities.

Many NER systems actively use various linguistic and domain-specific features to improve their performance. For example, part-of-speech (POS) tags play an important role in detecting NE boundaries, and domain-specific gazetteers play a decisive role in determining NE categories. Previous NER models based on deep neural networks embedded various types of linguistic and domain-specific knowledge into vector spaces. Then, they used the embedded vectors as nonhierarchical features of input layers, although the embedded vectors imply different levels of knowledge (e.g., POS tags imply a grammatical level of linguistic knowledge, and entities in a gazetteer imply a semantic level of domain knowledge). In addition, some previous works have shown that different layers of deep RNNs encode different types of information [2]. In other words, the embedded vectors are simply concatenated to the word embeddings being used as input, and the concatenated vectors are simply input into the NER models. Therefore, in the case of NER models with deep architecture, such as a multi-stacked recurrent neural network (RNN), different levels of features are mixed and interfere with each other. To alleviate the feature interference problem, we propose a NER model in which a multi-stacked RNN layer hierarchically uses different levels of features.

The remainder of this paper is organized as follows. In Section 2, we review the previous NER models. In Section 3, we describe our model to alleviate the sparse data problem in fine-grained NER. In Section 4, we explain our experimental setup and report some of our experimental results. In Section 5, we provide the conclusion of our study.

2. Previous Studies

NER tasks were previously resolved by considering them as sequence labeling problems. In this regard, most previous NER systems adopted machine learning (ML) models, such as decision trees [3], maximum entropy [4], and conditional random fields [5]. To improve the NER performance, these ML-based systems focused on feature engineering methods, such as word n-grams, part-of-speech n-grams, lexical clues, and knowledge lookup (to determine whether an input word exists in an external knowledge base) [6]. With the recent success of deep learning (DL), many DL-based NER systems have been proposed to

reduce the labor required for feature engineering [7,8]. These DL-based systems performed reasonably by using various distributed representations (e.g., word embeddings and character embeddings) instead of expensive knowledge features. Although many researchers have studied NER, it is not easy to find studies on fine-grained NER with hundreds of NE classes, especially for non-English languages. The authors in [9] presented a fine-grained entity recognizer that solved a multi-label, multi-class classification problem by adapting a Perceptron model. The Ref. [1] conducted an empirical study to develop a fine-grained NER model that is robust across various settings, such as the number of NE classes and the size of the training dataset, by using a bidirectional long short-term memory model with a conditional random field layer (BI-LSTM-CRF) [10]. They reported that a fine-grained NER model that is effective for English is not necessarily effective for Japanese. This showed that a fine-grained NER task has language-dependent characteristics. To overcome a lack of training data, [11] proposed a method using a language model and an expensive knowledge base in a fine-grained NER task. The Ref. [12] proposed a novel adversarial multitask learning framework in which POS tagging is performed together with NER in Chinese. The Ref. [13] proposed a sequence-to-sequence model to consider the entire meaning of an input sentence. They used BI-LSTM as the encoder to equally process the past and future information of an input sentence. Then, they added a self-attention mechanism to address the long-term dependency problem in a long sequence. The Ref. [14] showed that the embeddings of decomposed NE labels can be effectively used to improve the performance of instances with low-frequency NE labels.The Ref. [15] proposed a model that included the initial encoding layer, the enhanced encoding layer, and the decoding layer, combining the advantages of pre-training model encoding, dual bidirectional long short-term memory (BiLSTM) networks, and a residual connection mechanism. The Ref. [16] proposed a Chinese fine-grained NER method based on a language model and model transfer considering active learning (MTAL) to research a few labeled data. The Ref. [17] proposed a Cognitive Impairment model that can filter, study, analyze, and interpret written communications from social media platforms. The Ref. [18] proposed a label attention network (LAN) that captured possible long-term label dependency by utilizing an attention mechanism and label embedding. We adopted this LAN and proposed a dual-stacked LAN, with the lower for coarse-grained NER and the upper for fine-grained NER.

3. Fine-Grained NER Model

To detect word boundaries (i.e., morpheme boundaries) in Korean, many NER models perform morphological analysis in advance. Then, they generally use morphemes and POS tags of an input sentence as inputs. Under this kind of pipeline architecture, errors of morphological analysis directly lead to diminished performance in NER models. To overcome this limitation, we use character n-grams as inputs. Given n characters, $C_{1,n}$, in a sentence S, let $E^c_{1,n}$ and $E^f_{1,n}$ denote sequences of coarse-grained NE tags and fine-grained NE tags in S, respectively. Table 2 presents NE tags that are defined according to the well-known begin-inner-outer (BIO) character-level tagging scheme.

Table 2. Character-unit NE tags.

NE Tag	Description
B-(PER | LOC | ORG | ...)	Beginning character of an NE with the category following "B-"
I-(PER | LOC | ORG | ...)	Inner character of an NE with the category following "I-"
O	Character out of any NE boundary

The fine-grained NER model named FG-NER can then be formally expressed in the following equation.

$$FG - NER(S) \stackrel{\text{def}}{=} argmax P(E^c_{1,n}, E^f_{1,n} | C_{1,n}) \quad (1)$$

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

$$FG-NER(S) \stackrel{\text{def}}{=} argmax P(E^c_{1,n}|C_{1,n})P(E^f_{1,n}|C_{1,n}) \quad (2)$$

As shown in (2), coarse-grained NEs depend on input characters, and fine-grained NEs depend on input characters and the given coarse-grained NEs. To simplify (2), we adopt the following two assumptions: a first-order Markov assumption that a current tag is dependent on the previous tag, and a conditional independent assumption that a current tag is dependent only on its current observational information. Based on these assumptions, we rewrite (2) as the following equation. Note that the reason why we used two assumptions is to simplify Equation (2).

$$FG-NER(S) \stackrel{\text{def}}{=} argmax \prod_{i=1}^{n} \left\{ \frac{P(E^c_i|C_i)P(E^c_i|E^c_{i-1})}{P(E^f_i|C_i,E^c_i)P(E^f_i|E^f_{i-1})} \right\} \quad (3)$$

To obtain the sequence labels, $E^f_{1,n}$, that maximize (3) by using coarse-grained NEs as additional contextual information, we adopt a stacked BI-LSTM-LAN [17]. Figure 1 shows the architecture of the proposed fine-grained NER (FG-NER) model. This model comprises a k-stacked feature fusion layer (shown on the left) and a dual-stacked output layer (shown on the right). The feature fusion layer shown in Figure 1 accepts different levels of input embeddings that are fed into each layer of a three-stack BI-LSTM to yield a sequence of forward hidden and backward hidden states, respectively. Subsequently, these two states of each layer are concatenated to reflect bidirectional contextual information, as shown in the following equation.

$$\begin{aligned} \overrightarrow{h}^k_i &= LSTM(Emb^k_i, \overrightarrow{h}^k_{i-1}) \\ \overleftarrow{h}^k_i &= LSTM(Emb^k_i, \overleftarrow{h}^k_{i-1}) \\ \overleftrightarrow{h}^k_i &= [\overrightarrow{h}^k_i, \overleftarrow{h}^k_i] \\ \overleftrightarrow{H}^k &= \{\overleftrightarrow{h}^k_1, \overleftrightarrow{h}^k_2, \overleftrightarrow{h}^k_3, ..., \overleftrightarrow{h}^k_n\} \end{aligned} \quad (4)$$

where Emb^k_i is the i-th input embedding in the k-th stacked LSTM. Then, $\overleftrightarrow{h}^k_i = [\overrightarrow{h}^k_i; \overleftarrow{h}^k_i]$ is the concatenation of the forward hidden state $\overrightarrow{h}^k_i$ and the backward hidden state $\overleftarrow{h}^k_i$ of the i-th input in the k-th stacked LSTM. In the first stacked feature fusion layer (i.e., the lowest layer) of Figure 1, C_0, C_i, and C_{n+1} are a special beginning symbol of a sentence, the i-th one of n input characters, and a special ending symbol of a sentence, respectively. Then, $C_{0,n+1}$ is a randomly-initialized character embedding of each character. A concatenation of three successive character embeddings (i.e., a character tri-gram embedding; $[C_{i-1}; C_i; C_{i+1}]$ is used as an input embedding Emb^1_i for the first stacked feature fusion layer. In the second stacked feature fusion (i.e., the middle layer) of Figure 1, POS_{C_i} is a character-unit POS tag of the i-th input character according to a BIO-tagging scheme similar to Table 1. Then, $Emb(POS_{C_i})$ is a randomly-initialized POS embedding of the i-th character. To enrich input characters with grammatical information, a POS tri-gram embedding, $[Emb(POS_{C_{i-1}}); Emb(POS_{C_i}); Emb(POS_{C_{i+1}})]$, is concatenated with the tri-gram character embedding, Emb^1_i, by using residual connections. The concatenated embedding is used as the input embedding Emb^2_i for the second stacked feature fusion layer. In the last stacked feature fusion layer (i.e., the uppermost layer) of Figure 1, $DIC_{C_{i-1};C_i;C_{i+1}}$ is a dictionary look-up feature on whether a character tri-gram, $[C_{i-1}; C_i; C_{i+1}]$, exists in a dictionary including character trigrams of predefined NE lists (i.e., NE lists in a training data). Then, $Emb(DIC_{C_{i-1};C_i;C_{i+1}})$ is a randomly-initialized dictionary embedding of the i-th character. To enrich input characters with domain knowledge, the dictionary embedding is concatenated with the character trigram embedding, Emb^1_i, by using residual connections. The concatenated embedding is used as the input embedding Emb^3_i for the last stacked feature fusion layer. The hidden states of each stacked layer are

concatenated as $\overleftrightarrow{H} = [\overleftrightarrow{H}^1; \overleftrightarrow{H}^2; \overleftrightarrow{H}^3]$. The lower output layer shown in Figure 1, calculates the degrees of association between H and the coarse-grained NE tag embeddings $Emb(NE^c) = \{Emb(NE_1^c),\ Emb(NE_2^c),\ \ldots,\ Emb(NE_m^c)\}$ based on a multi-head attention mechanism [17], as shown in the following equation.

$$\begin{aligned} head_j &= attention(QW_j^Q, KW_j^K, VW_j^K) = \alpha_j * VW_j^V, \\ WhereQ &= \overleftrightarrow{H}, K = V = Emb(NE^c), \\ \alpha_j &= softmax(\frac{QW_j^Q * (KW_j^k)^T}{\sqrt{d_h}}), \\ A(c_i^c) &= head_1 \oplus head_2 \oplus ... \oplus head_k \end{aligned} \quad , \tag{5}$$

where $W_j^Q \in R^{d_h \times \frac{d_h}{k}}$, $W_j^K \in R^{d_h \times \frac{d_h}{k}}$, and $W_j^V \in R^{d_h \times \frac{d_h}{k}}$ are the weighting parameters of the j-th parameter among k heads to be learned during training. Then, $Emb(NE^c)$ represent the embedding vectors of m coarse-grained NE 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 dimension of H (same as the dimension of Coarse-grained NE embedding). The attention score vector $A(C_i^c)$ represents the degrees of associations between the contextualized input embedding $\overleftrightarrow{h}_i$ of the i-th input character and each coarse-grained NE tag. In other words, the vector can be considered as a potential distribution of coarse-grained tags associated with an input character. In the prediction phase, the lower output layer returns coarse-grained NE tags, as shown in the following equation.

$$\hat{E}_i^c = argmax(\hat{A}_i^1, \hat{A}_i^2, ..., \hat{A}_i^m) \tag{6}$$

where $\hat{A}_i^j$ denotes the j-th one among m attention scores in the trained attention vector $\hat{A}_i$. In the upper output layer in Figure 1, each coarse-grained attention score vector $A(C_i^c)$ is concatenated with the hidden states of each stacked layer H to enrich fine-grained NEs with the contextual information of coarse-grained NEs. Except that the concatenated vector is used as a query vector of the multi-head attention mechanism, the upper output layer follows the same procedure as the lower output layer, as shown in the following equation.

$$\begin{aligned} head_j &= attention(QW_j^Q, KW_j^K, VW_j^V) = \alpha_j * VW_j^V, \\ WhereQ &= [LSTM(A(C^c); \overleftrightarrow{H})], K = V = Emb(NE^f), \\ \alpha_j &= softmax(\frac{QW_j^Q * (KW_j^k)^T}{\sqrt{d_h}}), \\ A(c_i^f) &= head_1 \oplus head_2 \oplus ... \oplus head_k \end{aligned} \quad , \tag{7}$$

where $A(C^c)$, $Emb(NE^f)$, and $A(C_i^f)$ are the coarse-grained attention score vectors of n input characters, fine-grained NE tag embeddings, and a fine-grained attention score vector of the i-th input character, respectively. In the prediction phase, the upper output layer follows the same process as the lower output layer, as shown in the following equation.

$$\hat{E}_i^f = argmax(\hat{A}_i^1, \hat{A}_i^2, ..., \hat{A}_i^l), \tag{8}$$

where $\hat{A}_i^j$ denotes the j-th attention score of l fine-grained NE categories.

In general, coarse-grained training data are less unbalanced than fine-grained training data because coarse-grained NEs constitute a superset of fine-grained NEs. This led us to use a two-phase training scheme to optimize the weighting parameters of the proposed model. We first train the lower output layer to minimize the cross-entropy between the

correct coarse-grained NE tags, E_i^c, and the outputs of the lower output layer, $\hat{E}_i^c$, as shown in the following equation.

$$H_{\hat{E}^c}(E^c) = -\sum_i \hat{E}_i^c log(E_i^c). \quad (9)$$

In this phase, the weighting parameters in the lower output layer are considered to be pre-trained because they were trained by using less unbalanced training data. Then, we train the upper output layer to minimize the cross-entropy between the correct fine-grained NE tags, E_i^f, and the outputs of the upper output layer, $\hat{E}_i^f$, as shown in the following equation.

$$H_{\hat{E}^f}(E^f) = -\sum_i \hat{E}_i^f log(E_i^f). \quad (10)$$

In this second phase, we expect the weighting parameters in the lower output layer to be fine-tuned to specific values associated with the fine-grained NE tags.

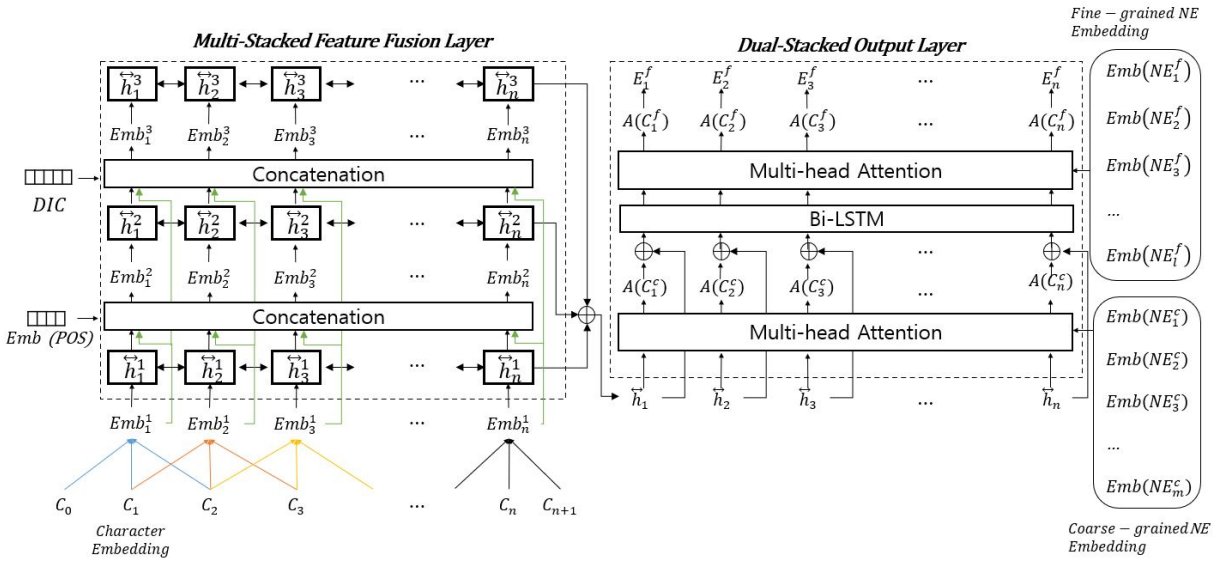

Figure 1. Overall architecture of FG-NER.

4. Evaluation

4.1. Datasets and and Experimental Settings

In our experiments, we used a gold-labeled corpus annotated with 14 coarse-grained NE tags and 147 fine-grained NE tags. This corpus was constructed by ETRI (Electronics and Telecommunications Research Institute, https://www.etri.re.kr/eng/main/main.etri, accessed on 15 September 2021). This corpus has been tagged with the coarse-grained named entity and fine-grained named entity in the sentences in the encyclopedia. In addition, it is the only training data for fine-grained NER in Korean. Table 3 presents the distribution of NE tags found in the gold-labeled corpus.

We converted the gold-labeled corpus into an NE dataset in which each character was annotated with the NE tags in Table 2. Then, we divided the NE dataset into training, validation, and test datasets, respectively, to obtain a ratio of 8:1:1. Finally, we evaluated the proposed model using the following evaluation measures: precision, recall rate, and F1-score.

$$Precision = \frac{\#\ of\ correct\ NE's}{\#\ of\ NE's\ returned\ by\ a\ system}. \quad (11)$$

$$Recall = \frac{\#\ of\ correct\ NE's\ returned\ by\ asystem}{\#\ of\ correct\ NE's\ in\ a\ test\ data}. \quad (12)$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}. \quad (13)$$

To calculate the precision, recall rate, and F1-score, the proposed model automatically generates NE sequences by concatenating the characters with B tags and successive I tags.

Table 3. Distribution of NE tags which mainly occurred in the corpus.

Coarse-Grained NE Tag	Description	Percent
QT	Quantity	15.9%
DT	Date	14.0%
OG	Organization	12.6%
TR	Theroy	8.7%
CV	Civilization	8.2%
Fine-Grained NE Tag	**Description**	**Percent**
TR-Technology	The technology of Theory	7.6%
DT-Year	The Year of Date	5.3%
DT-Month	The month of Date	4.8%
PS-Name	The name of Person	4.7%
OG-Business	The business of Organization	3.9%

4.2. Implementation

We implemented the proposed model using the Pytorch [19]. Training and prediction occurred on a per-sentence level. Table 4 lists the parameter settings we used to train the model.

Table 4. Model parameters.

Parameter	Value
The dimension of character embedding	50
The dimension of POS embedding	16
The dimension of hidden node in the feature fusion layer	128
The dimension of hidden node in the output layer	256
The dimension of Coarse-grained NE embedding	768
The dimension of Fine-grained NE embedding	512
Batch size	64
Learning rate	0.001
Epoch	100

4.3. Experimental Results

First, we evaluated the effectiveness of the k-stacked feature fusion layer and the dual-stacked output layer; the results are summarized in Table 5.

Table 5. Performance comparison depending on changes in the architecture.

Model	Precision	Recall	F1-Score
1-In+1-Out	0.783	0.681	0.728
3-In+1-Out	0.831	0.730	0.777
3-In(H)+1-Out	0.844	0.752	0.795
1-In+2-Out	0.808	0.701	0.750
3-In+2-Out	0.849	0.760	0.801
3-In(H)+2-Out	0.865	0.769	0.814

In Table 5, "k-In", "k-In(H)", "1-Out", and "2-Out" denote a k-stacked feature fusion layer in which a flat concatenation of all input embeddings (i.e., a character trigram embedding, a POS trigram embedding, and a dictionary embedding) is fed into the first layer, the proposed feature fusion layer in which different levels of input embeddings are hierarchically fed into the k-stacked LSTMs, a single output layer (i.e., only the upper output layer), and a dual-stacked output layer, respectively. The results in Table 5 show that the models with a dual-stacked output layer always outperformed the models with a single output layer. This reveals that the proposed dual-stacked output layer contributes to alleviate the problem of unbalanced training data. In addition, "3-In(H)+2-Out" delivered the best performance. This reveals that the hierarchical feature embeddings in a stacked feature fusion layer are able to effectively hand over different levels of linguistic features to an output layer.

The second experiment was conducted to compare the performance of the proposed model with those of the previous fine-grained NER models; the results are summarized in Table 6. In this table, "KoELECTRA-NER" is an NER model in which the character-based ELECTRA model [20] in Korean is fine-tuned to a sequence-labeling task. ELECTRA is a pre-trained language model with SOTA performance in many downstream NLP tasks, such as span prediction, sequence labeling, and text classification. We carried out a Korean NER task by pre-training KoELECTRA by using 96M sentences with 2.6B tokens. Then, we fine-tuned KoELECTRA by using the training NE dataset. "BI-LSTM-LAN" [21] is an NER model that simultaneously performs morphological analysis and coarse-grained NER in Korean. This model achieved SOTA performance by outperforming Korean NER models that did not use large pre-trained language models, such as BERT [22], ALBERT [23], and ELECTRA.

Table 6. Performance comparison with the previous models.

Task	Model	Precision	Recall	F1-Score
Fine-grained NER	KoELECTRA-NER	0.855	0.757	0.802
	3-In(H)+2-Out	0.865	0.769	0.814
Coarse-grained NER	Bi-LSTM-LAN	0.855	0.813	0.833
	KoELECTRA-NER	0.879	0.838	0.857
	3-In(H)-1-Out	0.861	0.831	0.845

The results presented in Table 6 show that "3-In(H)+2-Out" outperformed "KoELECTRA-NER" in the fine-grained NER task. We attribute the improved performance to the well-formed neural network architecture with the stacked feature fusion layer and its ability to effectively reflect contextual information and features. On the coarse-grained NER task, the performance of "3-In(H)+2-Out" was slightly less accurate than that of "KoELECTRA-NER". However, "3-In(H)+2-Out" was 25 times lighter than "KoELECTRA-NER".

4.4. Discussion

In Table 5, "3-In+1-Out" and "3-In+2-Out" were significantly outperformed by "1-In+1-Out" and "1-In+2-Out", respectively. This suggests that the tri-gram character vector that concatenates three uni-gram character vectors has much more enriched information than the uni-gram character vector. In addition, "3-In(H)+1-Out" and "3-In(H)+2-Out" were outperformed by "3-In+1-Out" and "3-In+2-Out", respectively. This suggests that our hierarchical feature embeddings in a stacked feature fusion layer are more practical to capture POS and dictionary look-up features. Finally, "3-In(H)+2-Out" was outperformed by "3-In(H)+1-Out". This suggests that using coarse-grained NE information contributes to the recognization of fine-grained NE.

5. Conclusions

We proposed a fine-grained NER model that compensates for the sparseness of fine-grained NEs by using the contextual information of coarse-grained NEs. In addition, the model uses a hierarchical approach to alleviate the interference of features at different levels with each other. The proposed model consists of a multi-stacked feature fusion layer and a dual-stacked output layer. The feature fusion layer generates multiple levels of sentence representations and word representations by using multi-stacked BI-LSTMs. Based on the multilevel representations, the output layer returns fine-grained NE tags by using dual-stacked BI-LSTMs in which the lower layer is trained for coarse-grained NER. In the experiments, the proposed model delivered SOTA performance. Based on the experimental results, we concluded that the proposed model can effectively alleviate the problem caused by unbalanced data in fine-grained NER tasks. In addition, we concluded that the feature fusion architecture of the proposed model can contribute to the alleviation of the feature interference problem.

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). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-00368, A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques). Also, This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2020R1F1A1069737).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: We thank the menbers of the NLP laboratory at Konkuk University for their technical support.

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

References

1. Mai, K.; Pham, T.H.; Nguyen, M.T.; Nguyen, T.D.; Bollegala, D.; Sasano, R.; Sekine, S. An empirical study on fine-grained named entity recognition. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 711–722.
2. Peters, E.M.; Neumann, M.; Iyyer, M.; Gradner, M. Deep contexualized word representation. In Proceedings of the NAACL-HLT 2018, New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237.
3. Sekine, S.; Grishman, R.; Shinnou, H. A decision tree method for finding and classifying names in Japanese texts. In Proceedings of the 6th Workshop on Vary Large Corpora, Montreal, QC, Canada, 15–16 August 1998; pp. 171–178.
4. Borthwick, A.; Sterling, J.; Agichtein, E.; Grishman, R. NYU: Description of the MENE named entity system as used in MUC-7. In Proceedings of the Seventh Message Understanding Conference, Fairfaxm VA, USA, 29 April–1 May 1998.
5. Cohen, W.W.; Sarawagi, S. Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. In Proceedings of the KDD 2004, Seattle, WA, USA, 22–25 August 2004; pp. 89–98.
6. Nadeau, D.; Sekine, S. A survey of named entity recognition and classification. *J. Linguist. Investig.* **2007**, *30*, 3–26. [CrossRef]
7. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural architectures for named entity recognition. In Proceedings of the NAACL-HLT 2016, San Diego, CA, USA, 12–17 June 2016; pp. 260–270.
8. Chiu, J.P.; Nichols, E. Named entity recognition with bidirectional LSTM-CNNs. *Trans. Assoc. Comput. Linguist.* **2016**, *40*, 357–370. [CrossRef]
9. Ling, X.; Weld, D.S. Fined-Grained Entity Recognition. In Proceedings of the 26th AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; pp. 94–100.
10. Ma, X.; Hovy, E. End-to-end sequence labeling via bi-directional lstmcnns-crf. In Proceedings of the Association Computational Linguistics, Berlin, Germany, 18–24 May 2016; pp. 1064–1074.

11. Dogan, C.; Dutra, A.; Gara, A.; Gemma, A.; Shi, L.; Sigamani, M.; Walters, E. Fine-grained named entity recognition using elmo and wikidata. *arXiv* **2019**, arXiv:1904.10503.
12. Man, X.;Yang, P. Fine-grained Chinese Named Entity Recognition in Entertainment News Using Adversarial Multi-task Learning. In Proceedings of the 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, 6–9 December 2019; pp. 1671–1675.
13. Zhu, H.; He, C.; Fang, Y.; Xiao, W. Fine Grained Named Entity Recognition via Seq2seq Framework. *IEEE Access* **2020**, *8*, 53953–53961. [CrossRef]
14. Kato, T.; Abe, K.; Ouchi, H.; Miyawaki, S.; Suzuki, J.; Inui, K. Embeddings of Label Components for Sequence Labeling: A Case Study of Fine-grained Named Entity Recognition. In Proceedings of the Association Computational Linguistics, Virtual, 5–10 July 2020; pp. 222–229.
15. Liu, J.; Xia, C.; Yan, H.; Xu, W. Innovative Deep Neural Network Modeling for Fine-Grained Chinese Entity Recognition. *Electronics* **2020**, *9*, 1001. [CrossRef]
16. Yao, L.; Huang, H.; Wang, K.-W.; Chen, S.-H.; Xiong, Q. Fine-Grained Mechanical Chinese Named Entity Recognition Based on ALBERT-AttBiLSTM-CRF and Transfer Learning. *Symmetry* **2020**, *12*, 1986. [CrossRef]
17. Thakur, N.; Han, C.Y. A Multimodal Approach for Early Detection of Cognitive Impairment from Tweets. In *Human Interaction, Emerging Technologies and Future Systems V. IHIET 2021. Lecture Notes in Networks and Systems*; Ahram, T., Taiar, R., Eds.; Springer: Cham, Switzerland, 2021; Volume 319._2. [CrossRef]
18. Cui, L.; Zhang, Y. Hierarchically-refined label attention network for sequence labeling. In Proceedings of the EMNLP, Hong Kong, China, 3–7 November 2019; pp. 4113–4126.
19. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8026–8037.
20. Clark, K.; Luong, M.-T.; Le, Q.V.; Manning, C.D. ELECTRA: Pre-training text encoders as discriminators rather than generators. In Proceedings of the ICLR, Virtual, 26 April–1 May 2020; pp. 1–14.
21. Kim, H.; Kim, H. Integrated Model for Morphological Analysis and Named Entity Recognition Based on Label Attention Networks in Korean. *Appl. Sci.* **2020**, *10*, 3740. [CrossRef]
22. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
23. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A lite BERT for self-supervised learning of language representations. In Proceedings of the ICLR, Virtual, 26 April–1 May 2020.