Embeddings of Label Components for Sequence Labeling: A Case Study of Fine-grained Named Entity Recognition

Takuma Kato[1] **Kaori Abe**[1,2] **Hiroki Ouchi**[2,1] **Shumpei Miyawaki**[1]
Jun Suzuki[1,2] **Kentaro Inui**[1,2]
[1] Tohoku University [2] RIKEN
{takuma.kato, abe-k, miyawaki.shumpei,
jun.suzuki, inui}@ecei.tohoku.ac.jp
hiroki.ouchi@riken.jp

Abstract

In general, the labels used in sequence labeling consist of different types of elements. For example, IOB-format entity labels, such as `B-Person` and `I-Person`, can be decomposed into span (`B` and `I`) and type information (`Person`). However, while most sequence labeling models do not consider such label components, the shared components across labels, such as `Person`, can be beneficial for label prediction. In this work, we propose to integrate label component information as embeddings into models. Through experiments on English and Japanese fine-grained named entity recognition, we demonstrate that the proposed method improves performance, especially for instances with low-frequency labels.

1 Introduction

Sequence labeling is a problem in which a label is assigned to each word in an input sentence. In many label sets, each label consists of different types of elements. For example, IOB-format entity labels (Ramshaw and Marcus, 1995), such as `B-Person` and `I-Location`, can be decomposed into span (e.g., `B`, `I` and `O`) and type information (e.g., `Person` and `Location`). Also, morphological feature tags (More et al., 2018), such as `Gender=Masc|Number=Sing`, can be decomposed into gender, number and other information.

General sequence labeling models (Ma and Hovy, 2016; Lample et al., 2016; Chiu and Nichols, 2016), however, do not consider such components. Specifically, the probability that each word is assigned a label is computed on the basis of the inner product between word representation and label embedding (see Equation 2 in Section 2.1). Here, the label embedding is associated with each label and independently trained without considering its components. This means that labels are treated as mutually exclusive. In fact, labels often share some components. Consider the labels `B-Person` and `I-Person`. They share the component `Person`, and injecting such component information into the label embeddings can improve the generalization performance.

Motivated by this, we propose a method that shares and learns the embeddings of label components (see details in Section 2.2). Specifically, we first decompose each label into its components. We then assign an embedding to each component and summarize the embeddings of all the components into one as a label embedding used in a model. This component-level operation enables the model to share information on the common components across label embeddings.

To investigate the effectiveness of our method, we take the task of fine-grained Named Entity Recognition (NER) as a case study. Typically, in this task, a large number of entity-type labels are predefined in a hierarchical structure, and intermediate type labels can be used as label components, as well as leaf type labels and B/I-labels. In this sense, the fine-grained NER can be seen as a good example of the potential applications of the proposed method. Furthermore, some entity labels occur more frequently than others. An interesting question is whether our method of label component sharing exhibits an improvement in recognizing entities of infrequent labels. In our experiments, we use the English and Japanese NER corpora with the Extended Named Entity Hierarchy (Sekine et al., 2002) including 200 entity tags. To sum up, our main contributions are as follows: (i) we propose a method that shares and learns label component embeddings, and (ii) through experiments on English and Japanese fine-grained NER, we demonstrate that the proposed method achieves better performance than a standard sequence labeling model, especially for instances with low-frequency labels.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 222–229

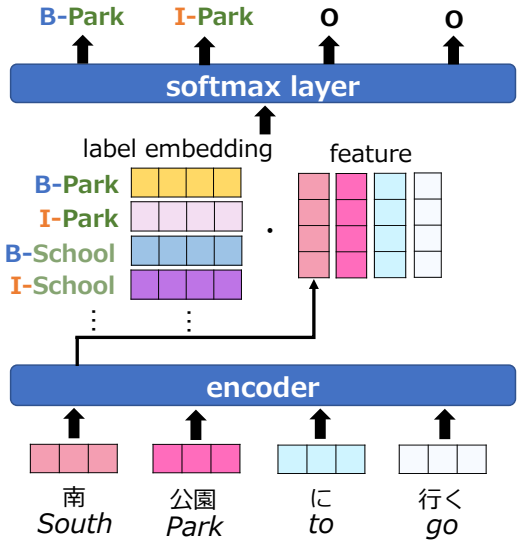


Figure 1: Overview of a standard sequence labelling model. Each label (e.g., B-Park) is annotated as a single unit, disregarding its inner structure ("B" and "Park").

2 Methods

2.1 Baseline model

We describe our baseline model in Figure 1. Given an input sentence, the encoder converts each word into its feature vector. Then, the inner product between each feature vector and label embedding is calculated for computing the label distribution. Finally, the IOB2-format label (Ramshaw and Marcus, 1995) with the highest probability is assigned to each token. The label `B-Park`, indicating the leftmost token of some entity, is assigned to 南 (*South*), and `I-Park`, indicating the token inside some entity, is assigned to 公園 (*Park*). The label `O`, indicating the token outside entities, is assigned to に (*to*) and 行く (*go*).

Formally, for each word x_i in the input sentence $X = (x_1, x_2, \ldots, x_n)$, the model outputs the label $\hat{y}_i$ with the highest probability:

$$\hat{y}_i = \arg\max_{y \in \mathcal{Y}} P(y|x_i, X), \quad (1)$$

where $\mathcal{Y}$ is a label set defined in each data set. The probability distribution is calculated as

$$P(y|x_i, X) = \frac{\exp(\mathbf{W}[y] \cdot \mathbf{f}(x_i, X))}{\sum_{y' \in \mathcal{Y}} \exp(\mathbf{W}[y'] \cdot \mathbf{f}(x_i, X))}, \quad (2)$$

where $\mathbf{W} \in \mathbb{R}^{|\mathcal{Y}| \times D}$ is a weight matrix for the label set $\mathcal{Y}$.[1] Each row of this matrix is associated with each label $y \in \mathcal{Y}$, and $\mathbf{W}[y]$ represents the y-th row vector. $\mathbf{f}(x, X)$ represents the vector encoded by a neural-network-based encoder.

[1] D is the number of dimensions of each weight vector.

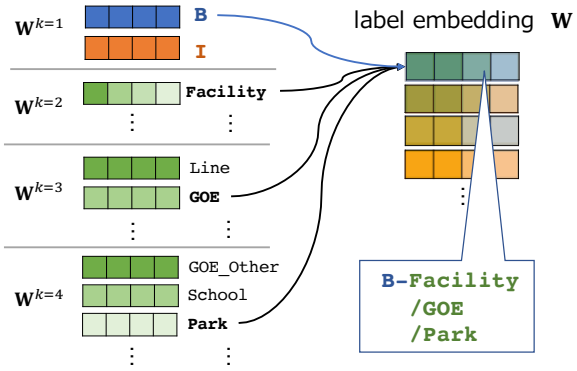


Figure 2: Label embedding calculation. Each label embedding is calculated from its component embeddings.

2.2 Embeddings of label components

We propose to integrate label component information as embeddings into models. This procedure consists of two steps: (i) *label decomposition* and (ii) *label embedding calculation*.

Label decomposition We first decompose each label into its components. Each label consists of multiple types of components. Consider the following example.

$$\texttt{B-Park} = \{\texttt{B}, \texttt{Park}\}$$

The labels defined in a general entity tag set consist of the IOB (e.g., `B`) and entity (e.g., `Park`) component types. Consider another example.

$$\texttt{B-Facility/GOE/Park} = \{\texttt{B}, \texttt{Facility}, \texttt{GOE}, \texttt{Park}\}$$

The labels defined in the Extended Named Entity tag set (Sekine et al., 2002) consist of the four component types: IOB (e.g., `B`), top layer of the entity tag hierarchy (e.g., `Facility`), second layer (e.g., `GOE`) the third layer (e.g., `Park`). In this way, we can regard each label as a set of components (type–value pairs).

Formally, K components of each label y will be denoted by $\mathcal{C}^y = \{c_k\}_{k=1}^K$, where c_k is the index associated with the value of each component type k. The above two examples are represented as $\mathcal{C}^{y=\texttt{B-Park}} = \{c_1 = \texttt{B}, c_2 = \texttt{Park}\}$ and $\mathcal{C}^{y=\texttt{B-Facility/GOE/Park}} = \{c_1 = \texttt{B}, c_2 = \texttt{Facility}, c_3 = \texttt{GOE}, c_4 = \texttt{Park}\}$. This formalization is applicable to arbitrary label sets whose label consists of type-value components.

Label embedding calculation We then assign an embedding (i.e., trainable vector representation) to each label component and combining the embeddings of all the components within a label into one label embedding. In this study, we investigate two types of typical summarizing techniques: (a) summation and (b) concatenation.

(a) Summation The embedding of each label, $\mathbf{W}[y]$, is calculated by summing the embeddings of its components:

$$\mathbf{W}[y] = \sum_{c_k \in \mathcal{C}^y} \mathbf{W}^k[c_k]. \quad (3)$$

Here, $\mathbf{W}^k$ is an embedding matrix for each component type k, and $\mathbf{W}^k[c_k]$ denotes the c_k-th row vector. Figure 2 illustrates this calculation process. The label `B-Facility/GOE/Park` consists of four components (i.e., `B`, `Facility`, `GOE` and `Park`), each c_k value of which is associated with a row vector of each matrix $\mathbf{W}^k$.

(b) Concatenation The embedding of each label, $\mathbf{W}[y]$, is calculated by concatenating the embeddings of its components:

$$\mathbf{W}[y] = [\mathbf{W}^1[c_1], \ldots, \mathbf{W}^K[c_K]]. \quad (4)$$

Here, similarly to $\mathbf{W}^k$ is an embedding matrix for each component type k Equation 3. Unlike Equation 3, the label component embeddings are concatenated into one embedding. Compared with the summation, one disadvantage of the concatenation is memory efficiency: the number of dimensions of the label embeddings increases according to the number of label components K.

Our label embedding calculation enables models to share the embeddings of label components commonly shared across labels. For example, the embeddings of both `B-Facility/GOE/Park` and `B-Facility/GOE/School` are calculated by adding the embeddings of the shared components (i.e., `B`, `Facility` and `GOE`). Equations 3 and 4 can be regarded as a general form of the hierarchical label matrix proposed by Shimaoka et al. (2017) because our method can treat not only hierarchical structures but also any type of type–value set, such as morphological feature labels (e.g. `Gender=Masc|Number=Sing`).

3 Experiments

3.1 Settings

Dataset We use the Extended Named Entity Corpus for English[2] and Japanese.[3] fine-grained NER (Mai et al., 2018) In this dataset, each NE is assigned one of 200 entity labels defined in the Extended Named Entity Hierarchy (Sekine et al., 2002). For the English dataset, we follow the training/development/test split defined by Mai et al. (2018). For the Japanese dataset, we follow the training/development/test split of Universal Dependencies (UD) Japanese-BCCWJ. (Asahara et al., 2018)[4] Table 1 shows the statistics of the dataset.

Data statistics There is a gap between the frequencies, i.e., how many times each label appears in the training set. We categorize each label into three classes on the basis of its frequency, shown in Table 2. For example, if a label appears 0–100 times in the training set, it is categorized into the "Low" class. Moreover, we denote how many times entities with the labels belonging to each frequency class appear in the development or test set. To better understand the model behavior, we investigate the performance of each frequency class.

Model setup As the encoder $\mathbf{f}(x, X)$ in Equation 2 in Section 2.1, we use BERT[5] (Devlin et al., 2019), which is a state-of-the-art language model.[6] As the baseline model, we use the general label embedding matrix without considering label components, i.e., each label embedding $\mathbf{W}[y]$ in Equation 2 is randomly initialized and independently learned. In contrast, our proposed model calculates the label embedding matrix from label components (Equations 3 and 4). The only difference between these models is the label embedding matrix, so if a performance gap between them is observed, it stems from this point.

Hyperparameters The overall settings of hyperparameters are the same between the baseline and the proposed model. For English, we use the BERT pre-trained on BooksCorpus and English Wikipedia (Devlin et al., 2019). For Japanese, we use the BERT pre-trained on Japanese

[2] We e-mailed the authors of (Mai et al., 2018) and received the English dataset.

[3] https://www.gsk.or.jp/catalog/gsk2014-a/

[4] https://github.com/UniversalDependencies/UD_Japanese-BCCWJ

[5] We use the open-source NER model utilizing BERT: https://github.com/kamalkraj/BERT-NER.

[6] The state of the art model on the Extended Named Entity Corpus is the LSTM + CNN + CRF model that uses dictionary information (Mai et al., 2018)

Dataset	English		Japanese	
	# of Sentences	# of Entities	# of Sentences	# of Entities
train	14176	27686	34784	72318
dev	1573	3032	7009	11954
test	3942	7682	6783	11669

Table 1: Statistics of the datasets.

Frequency Classes		English		Japanese	
		Dev	Test	Dev	Test
Low	(0∼100)	1125	2798	666	619
Middle	(101∼500)	1224	3128	2,875	2,531
High	(501∼)	683	1756	8,413	8,519

Table 2: Details of frequency classes.

Wikipedia (Shibata et al., 2019). We fine-tune them on the Extended NER corpus for solving fine-grained NER. We set the training epochs to 20 in fine-tuning. Both the baseline and the proposed models are trained to minimize cross-entropy loss during training. We set a batch size of 32 and a learning rate of 5.0×10^{-5} using Adam (Kingma and Ba, 2015) for the optimizer. We choose the dropout rate from among $\{0.1, 0.3, 0.5\}$ on the basis of the F_1 scores in each development set.[7] We set the number of dimensions of the hidden states in BERT. In the baseline model, we set the number of dimensions of the label embedding $\mathbf{W}$ in Equation 2 to 768. In the proposed models, we also use the same dimension size 768 for $\mathbf{W}$ in Equations 3 and 4.

3.2 Results

We report averaged F_1 scores across five different runs of the model training with random seeds. Table 3 shows F_1 scores for overall classes and each label frequency class on each test set.

Overall performance For the overall labels, the proposed models (PROPOSED:SUM and PROPOSED:CONCAT) outperformed the baseline model on English and Japanese datasets. These results suggest the effectiveness of our proposed method for calculating the label embeddings from label components.

Performance for each frequency class For all the label frequency classes, the proposed model with summation (PROPOSED:SUM) yielded the best results among the three models. In particular, for low-frequency labels, the proposed model with summation (PROPOSED:SUM) achieved a remarkable improvement of F_1 compared with the baseline model. Also, the proposed model with concatenation (PROPOSED:CONCAT) achieved an improvement of F_1. These results suggest that exploiting label embeddings of the components shared across labels improves the generalization performance, especially for low-frequency labels.

3.3 Analysis

Recall that the entity tag set used in the datasets has a hierarchical structure. This means that label components at higher layers appear more frequently than those at lower layers and are shared across many labels. As shown in Table 3, the proposed models achieve performance improvements for low-frequency labels. Here, we can expect that the embeddings of high-frequency shared label components help the model correctly predict the low-frequency labels. To verify this hypothesis, we compare between F_1 scores of the baseline and proposed models, shown in Table 4. Here, the targets to investigate are the three-layered, low-frequency labels[8] that have a high-frequency, second layer component.[9] As shown in Table 4, the PROPOSED:SUM model outperformed the baseline model. This indicates that for predicting low-frequency labels, it is effective for the model to use shared components. On the other hand, the PROPOSED:CONCAT model underperformed the baseline model. One possible reason is that the model obtains less information by concatenating label embeddings than by summing them.

[7] In our experiments, we found that the models trained with the dropout rate of 0.1 achieved the best performance on each development set.

[8] We exclude the labels that consist of only two layers, such as `Timex/Date`.

[9] In this paper, we also regard the second-layer components appearing over 100 times in the training set as high-frequency.

	Low	Middle	High	Overall
		English		
BASELINE	79.83±0.27	80.29±0.46	90.82±0.32	84.99±0.27
PROPOSED:SUM	**81.15**±0.24	**80.99**±0.27	**90.87**±0.26	**85.67**±0.13
PROPOSED:CONCAT	80.40±0.38	80.31±0.28	90.75±0.23	85.20±0.16
		Japanese		
BASELINE	44.39±0.29	51.73±0.50	70.82±0.32	68.06±0.27
PROPOSED:SUM	**45.34**±0.91	**51.93**±0.66	**71.04**±0.49	**68.34**±0.41
PROPOSED:CONCAT	44.76±1.12	51.45±0.40	70.52±0.29	67.77±0.23

Table 3: Comparison between the baseline and proposed models. Cells show the F_1 scores and standard deviations on each test set.

	English	Japanese
Baseline	76.58±0.26	49.66±0.68
Proposed:Sum	**77.76**±0.30	**50.05**±1.19
Proposed:Concat	76.77±0.71	49.31±1.12

Table 4: Comparison between the baseline and the proposed models in the Low frequency class.

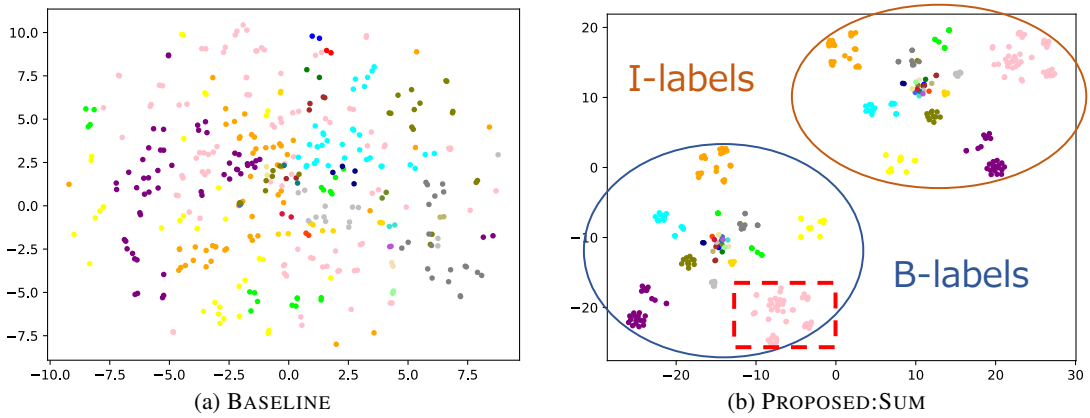
3.4 Visualization of label embedding spaces

To better understand the label embeddings created from the label components by our proposed method, we visualize the learned label embeddings. Specifically, we hypothesize that the embeddings of the labels sharing label components are close to each other and form clusters in the embedding space if they successfully encode the shared label component information. To verify this hypothesis, we use the t-SNE algorithm (van der Maaten and Hinton, 2008) to map the label embeddings learned by the baseline and proposed models onto the two-dimensional space, shown in Figure 3. As we expected, some clusters were formed in the label embedding space learned by the proposed model, shown in Figure 3b, while there is no distinct cluster in the one learned by the baseline, shown in Figure 3a. By looking at them in detail, we obtained two findings. First, in the embedding space learned by the proposed model, we found that two distinct clusters were formed corresponding to the two span labels (i.e. `B` and `I`). Second, the labels that have the same top layer label (represented in the same color) also formed some smaller clusters within the `B` and `I`-label clusters. For example, Figure 3c shows the `Product` cluster whose members are the labels sharing the top layer label `Product`. From these figures, we could confirm that the embeddings of the labels sharing label components (span and upper-layer type labels) form the clusters.

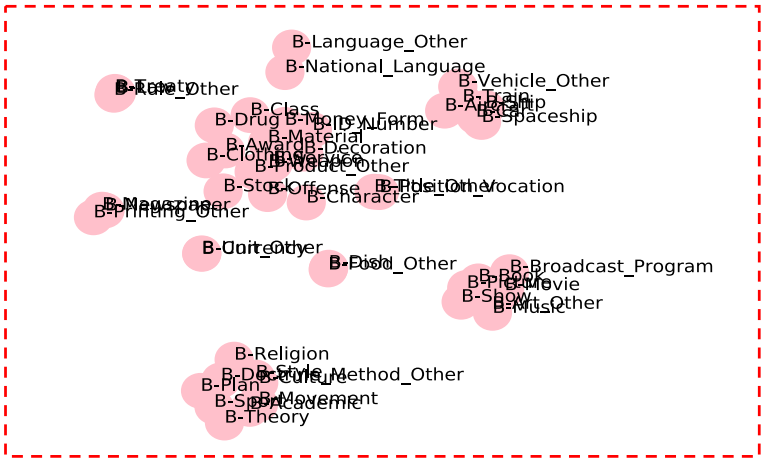
4 Related work

Sequence labeling has been widely studied and applied to many tasks, such as Chunking (Ramshaw and Marcus, 1995; Hashimoto et al., 2017), NER (Ma and Hovy, 2016; Chiu and Nichols, 2016) and Semantic Role Labeling (SRL) (Zhou and Xu, 2015; He et al., 2017). In English fine-grained entity recognition, Ling and Weld (2012) created a standard fine-grained entity typing dataset with multi-class, multi-label annotations. Ringland et al. (2019) developed a dataset for nested NER dataset. These datasets independently handle each label without considering label components. In Japanese NER, Misawa et al. (2017) combined word and character information to improve performance. Mai et al. (2018) reported that dictionary information improves the performance of fine-grained NER. Their methods do not consider label components and are orthogonal to our method.

Some existing studies take shared components (or information) across labels into account. In Entity Typing, Ma et al. (2016) and Shimaoka et al. (2017) proposed to calculate entity label embeddings by considering a label hierarchical structure. While their method is limited to only a hierarchical structure, our method can be applied to any set of components and can be regarded as a general form of their method. In multi-label classification, Zhong et al. (2018) assumed that the labels co-occurring in many instances are correlated with each other and share some common features, and proposed a method that learns a feature (label em-



(a) BASELINE (b) PROPOSED:SUM



(c) Enlarged view of a cluster in (b). The embeddings of the labels sharing the top layer label `Product` form this cluster.

Figure 3: Visualization of the label embedding space. The same color represents the labels that have the same hierarchical top layer label.

bedding) space where such co-occurring labels are close to each other. The work of Matsubayashi et al. (2009) is the closest to ours in terms of decomposing the features of labels. They regard an original label comprising a mixture of components as a set of multiple labels and made models that are able to exploit the multiple components to effectively learn in the SRL task.

5 Conclusion

We proposed a method that shares and learns the embeddings of label components. Through experiments on English and Japanese fine-grained NER, we demonstrated that our proposed method improves the performance, especially for instances with low-frequency labels. For future work, we envision to apply our method to other tasks and datasets and investigate the effectiveness. Also, we plan to extend the simple label embedding calculation methods to more sophisticated ones.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number JP19H04162 and JP19K20351. This work was also partially supported by a Bilateral Joint Research Program between RIKEN AIP Center and Tohoku University. We would like to thank the members of Tohoku NLP Laboratory, the anonymous reviewers, and the SRW mentor Gabriel Stanovsky for their insightful comments. We also appreciate Alt inc. for providing the corpus of English extended named entity data.

References

Masayuki Asahara, Hiroshi Kanayama, Takaaki Tanaka, Yusuke Miyao, Sumire Uematsu, Shinsuke Mori, Yuji Matsumoto, Mai Omura, and Yugo Murawaki. 2018. Universal dependencies version 2 for Japanese. In *Proceedings of LREC*, pages 1824–1831.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of EMNLP*, pages 1923–1933.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of ACL*, pages 473–483.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of ACL*, pages 1064–1074.

Yukun Ma, Erik Cambria, and Sa Gao. 2016. Label embedding for zero-shot fine-grained named entity typing. In *Proceedings of COLING*, pages 171–180.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Khai Mai, Thai-Hoang Pham, Minh Trung Nguyen, Tuan Duc Nguyen, Danushka Bollegala, Ryohei Sasano, and Satoshi Sekine. 2018. An empirical study on fine-grained named entity recognition. In *Proceedings of COLING*, pages 711–722.

Yuichiroh Matsubayashi, Naoaki Okazaki, and Jun'ichi Tsujii. 2009. A comparative study on generalization of semantic roles in FrameNet. In *Proceedings of ACL and AFNLP*, pages 19–27.

Shotaro Misawa, Motoki Taniguchi, Yasuhide Miura, and Tomoko Ohkuma. 2017. Character-based bidirectional LSTM-CRF with words and characters for Japanese named entity recognition. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 97–102.

Amir More, Özlem Çetinoğlu, Çağrı Çöltekin, Nizar Habash, Benoît Sagot, Djamé Seddah, Dima Taji, and Reut Tsarfaty. 2018. CoNLL-UL: Universal morphological lattices for universal dependency parsing. In *Proceedings of LREC*, pages 3847–3853.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of Third Workshop on Very Large Corpora*, pages 82–94.

Nicky Ringland, Xiang Dai, Ben Hachey, Sarvnaz Karimi, Cécile Paris, and James R. Curran. 2019. NNE: A dataset for nested named entity recognition in english newswire. In *Proceedings of ACL*, pages 5176–5181.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended named entity hierarchy. In *Proceedings of LREC*, pages 1818–1824.

Tomohide Shibata, Daisuke Kawahara, and Sadao Kurohashi. 2019. Improving japanese syntax parsing with bert. In *Natural Language Processing*, pages 205–208.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural architectures for fine-grained entity type classification. In *Proceedings of EACL*, pages 1271–1280.

Yongjian Zhong, Chang Xu, Bo Du, and Lefei Zhang. 2018. Independent feature and label components for multi-label classification. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pages 827–836.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of ACL*, pages 1127–1137.

A Appendices

A.1 Additional results

	Top	Second	Third
	English		
BASELINE	90.01±0.27	86.69±0.32	83.22±0.28
PROPOSED:SUM	**90.53**±0.06	**87.53**±0.11	**83.87**±0.20
PROPOSED:CONCAT	90.28±0.09	87.04±0.13	83.18±0.30
	Japanese		
BASELINE	72.68±0.20	66.22±0.36	66.84±0.34
PROPOSED:SUM	**73.13**±0.43	**66.37**±0.42	**67.00**±0.59
PROPOSED:CONCAT	72.50±0.30	66.19±0.24	66.42±0.49

Table 5: Comparison between the baseline and proposed models for the labels at each hierarchical layer.

	English	Japanese
BASELINE	96.32±0.10	84.74±0.18
PROPOSED:SUM	96.31±0.11	85.01±0.15
PROPOSED:CONCAT	96.27±0.07	84.83±0.11

Table 6: Comparison between the baseline and the proposed models in span (only considering B, I labels).

Performance for each hierarchical category Table 5 shows F_1 scores for each hierarchical category. The proposed model with summation (PROPOSED:SUM) outperformed the other models in all the hierarchical categories. For the labels at the top layer, in particular, PROPOSED:SUM achieved an improvement of the F_1 scores by a large margin on the Japanese dataset.

Performance for entity span boundary match Table 6 shows F_1 scores for entity span boundary match, where we regard a predicted boundary (i.e., B and I) as correct if it matches the gold annotation regardless of its entity type label. The performance of the proposed models was comparable to the baseline model. This indicates that there is a performance difference not in identification of entity spans (entity detection) but in identification of entity types (entity typing).

A.2 Case study

We observe actual examples predicted by the proposed model with summation, shown in Table 7.

In Example (a) and (b), Both models succeeded to recognize the entity span. However, only the proposed model also correctly predicted the type label. Note that the entities `Location/Spa` and `Natural_Object/Living_Thing/Living_Thing_Other` appear rarely, but rather to the extent of the top layer components `Location` and `Natural_Object` that appear frequently in the training set. Therefore, these examples suggest that the proposed model effectively exploits shared information of label components, especially in terms of the hierarchical layer.

Example (a)	下呂 温泉 発祥 の 地・・・ (The birthplace of Gero Spa ...)	
ENTITY	下呂 (*Gero*)	温泉 (*Spa*)
GOLD	B-Location/Spa	I-Location/Spa
BASELINE	B-Facility/Facility_Other	I-Facility/Facility_Other
PROPOSED:SUM	B-Location/Spa	I-Location/Spa
Example (b)	・・・ where clavaviridae derives from .	
ENTITY	clavaviridae	
GOLD	B-Natural_Object/Living_Thing/Living_Thing_Other	
BASELINE	B-Location/Astral_Body/Constellation	
PROPOSED:SUM	B-Natural_Object/Living_Thing/Living_Thing_Other	
Example (c)	・・・あお白い 日 の 光・・・ (... the pale sunlight ...)	
ENTITY	あお白い (*pale*)	
GOLD	B-Color/Color_Other	
BASELINE	O	
PROPOSED:SUM	B-Color/Nature_Color	

Table 7: Examples of both model outputs in fine-grained NER.

Although, we found that the proposed model predicts partially correct labels even though it is not totally correct in some cases. In Example (c), あ お 白 い (*pale*) is categorized into `Color/Color_Other`, the proposed model also predicted the wrong label `Color/Nature_Color`. However, interestingly, the proposed model correctly recognized the top layer of the type label as `Color`, which is in contrast to the completely wrong prediction of the baseline model.