Exploiting Hierarchically Structured Categories in Fine-grained Chinese Named Entity Recognition

Jiuding Yang [*1], **Jinwen Luo** [*2], **Weidong Guo** [*2], **Di Niu** [1], **Yu Xu** [2]
[1]University of Alberta
[2]Platform and Content Group, Tencent
[1]{jiuding,dniu}@ualberta.ca
[2]{jamsluo,weidongguo,henrysxu}@tencent.com

Abstract

Chinese Named Entity Recognition (CNER) is a widely used technology in various applications. While recent studies have focused on utilizing additional information of the Chinese language and characters to enhance CNER performance, this paper focuses on a specific aspect of CNER known as fine-grained CNER (FG-CNER). FG-CNER involves the use of hierarchical, fine-grained categories (e.g. Person-MovieStar) to label named entities. To promote research in this area, we introduce the FiNE dataset, a dataset for FG-CNER consisting of 30,000 sentences from various domains and containing 67,651 entities in 54 fine-grained flattened hierarchical categories. Additionally, we propose SoftFiNE, a novel approach for FG-CNER that utilizes a custom-designed relevance scoring function based on label structures to learn the potential relevance between different flattened hierarchical labels. Our experimental results demonstrate that the proposed SoftFiNE method outperforms the state-of-the-art baselines on the FiNE dataset. Furthermore, we conduct extensive experiments on three other datasets, including OntoNotes 4.0, Weibo, and Resume, where SoftFiNE achieved state-of-the-art performance on all three datasets.

1 Introduction

Named Entity Recognition (NER) (Li et al., 2020a; Nasar et al., 2021) is a fundamental component of natural language processing and has been widely studied to overcome the challenges brought by real-world text data. Chinese Named Entity Recognition (CNER) (Liu et al., 2022), as an important subfield of NER, has also drawn wide interests recently (Hao et al., 2013; Zheng et al., 2021; Liu et al., 2021a).

Fine-grained Named Entity Recognition (FG-NER) is a key challenge in current NER research, motivated by the growing demand for more detailed categorization of named entities. Compared to traditional Coarse-grained Named Entity Recognition (CG-NER) (Sun et al., 2002; Mansouri et al., 2008), FG-NER requires identifying the correct label from a larger number of categories for each entity, making it more difficult than CG-NER (Ekbal et al., 2010; Ling and Weld, 2012). These challenges also apply to Fine-grained Chinese NER (FG-CNER). However, most recent research on FG-CNER has focused on addressing the difficulties posed by the nature of Chinese characters (such as word segmentation (Ye et al., 2021)) and leveraging special information that can be extracted from the Chinese language, such as glyphs (Xuan et al., 2020), phonetics (Pinyin) (Sun et al., 2021), and rich lexicons (Zhang and Yang, 2018; Li et al., 2020b). The challenge posed by the increasing amount of corpus lacks attention. In other words, these are language-centric approaches that focus on leveraging special characteristics offered by the language, rather than data-centric approaches that exploit the potential advantages that the data itself can provide.

On the other hand, as the number of named entities increases, there is an increasing need to introduce hierarchical relations between different levels of categories. For example, "Jackie Chan" can be categorized as both a "Person" and a "Movie Star", which can be intuitively represented by a flattened hierarchical label "Person-MovieStar", where "Person" is the parent category and "MovieStar" is the child. Similarly, "Taylor Swift" can be labeled as "Person-Singer". It is clear that "Person-MovieStar" and "Person-Singer" are relevant to each other by sharing the common category of "Person". However, since NER tasks mostly apply sequence labeling to words (characters) (such as BIOES), these flattened hierarchical labels are usually encoded in a one-hot format (Peng and Dredze, 2015), which assumes that labels are mutually exclusive, thus ignoring the potential relevance among labels. A data-centric approach should leverage the relevance

[*]These authors contributed equally to this work.

Findings of the Association for Computational Linguistics: ACL 2023, pages 3407–3421
July 9-14, 2023

between hierarchical labels, such as the fact that "Jackie Chan" as a "Person-MovieStar" is closer to "Person-Singer" than many other categories, and use this knowledge to improve fine-grained NER performance.

Despite the importance of the research agenda outlined above for CNER applications, it has not received much attention in recent CNER literature. We believe this is due to a lack of a high-quality FG-CNER dataset. Existing CNER datasets are either outdated (Levow, 2006; Weischedel et al., 2011) or small-scale (Peng and Dredze, 2015; Zhang and Yang, 2018), and also lack fine-grained hierarchical categories commonly used in CNER applications today. These limitations in datasets have become a bottleneck for current CNER research, making it difficult to develop more advanced approaches that can meet real-world challenges.

To address this bottleneck in Fine-grained Chinese Named Entity Recognition, we introduce the FiNE dataset, a FG-CNER dataset consisting of 30,000 sentences sampled from various domains, containing 67,651 entities falling into 54 fine-grained flattened hierarchical categories. To the best of our knowledge, FiNE is currently the largest fully open-sourced (and test-dataset-included) FG-CNER dataset with hierarchical labels.

To exploit the relatedness of fine-grained labels in FG-CNER and facilitate future research, we propose the SoftFiNE method, a novel approach for solving FG-NER problems based on soft labels. SoftFiNE removes the mutual exclusive assumption and captures the hidden relevance between flattened hierarchical labels. Additionally, we also introduce a properly designed entity augmentation method in SoftFiNE, which can further boost FG-CNER performance by teaching the model to gather more context information about a character.

Experimental results on the FiNE dataset show the effectiveness of SoftFiNE compared to other baselines, such as LEBERT (Liu et al., 2021b), HMCN-F (Sun et al., 2021), LambdaRank (Burges et al., 2006), etc. To address the bottleneck caused by the limited CNER dataset, we conduct extensive experiments on the Weibo (Peng and Dredze, 2015), Resume (Zhang and Yang, 2018) and OntoNotes 4.0 (Weischedel et al., 2011) datasets, which are widely studied CNER datasets released in the past decade. Although SoftFiNE method achieves new state-of-the-art (SOTA) performance, the improvement compared to the previous SOTA is limited in terms of numbers, which also appears in recent CNER research (Mai et al., 2022; Sun et al., 2021; Liu et al., 2021b). This indicates that current datasets may no longer be able to support the design of more advanced CNER techniques due to their limitations, as mentioned earlier. To facilitate future research on FG-CNER, both the FiNE dataset and the source code of the SoftFiNE method are publicly available at https://github.com/XpastaX/SoftFiNE.

2 Related Work

CNER datasets. There are many open-source high-quality CNER datasets available. For example, People's Daily is one of the earliest CNER datasets, with over 20,000 samples in four categories (person, location, organization, and date). MSRA (Levow, 2006) is one of the largest CNER datasets, with over 50,000 samples in three categories (person, location, and organization). These datasets have greatly contributed to the early research on CNER, however, their utility in current CNER development is limited due to their outdatedness. Furthermore, real-world applications are increasingly demanding more fine-grained categories to classify named entities.

OntoNotes Release 4.0 (Weischedel et al., 2011) is a large-scale annotated corpus that also includes annotated CNER data with 18 types of categories. However, the commonly used version is processed by Che et al. (2013), where only four categories are available. Resume (Zhang and Yang, 2018) is a smaller dataset with more categories. It contains about 4,500 samples from real resumes and classifies entities into eight categories. Weibo (Peng and Dredze, 2015) has about 2,000 samples with four main categories and two sub-categories for each, indicating whether the entity is a name mention or nominal mention, except "Geo-political", which has no nominal mention. Although it has a hierarchy of labels, the structure is too simple. Thus, Peng and Dredze (2015) treat them as seven different labels while developing their model, which is also followed by later research.

Due to the limitations of existing CNER datasets, we decided to construct the FiNE dataset to facilitate the study of hierarchical FG-CNER.

Recent research on CNER. The development of neural networks has led to a growing interest in combining well-trained word embeddings with deep learning models to support CNER. Various

pictograph information, such as strokes (Sun et al., 2021), radicals (Dong et al., 2016; Xu et al., 2019), and glyphs (Meng et al., 2019), have been utilized to improve the performance of CNER. Additionally, the large lexicons in Chinese, where each word is a combination of characters, have been collected and leveraged to further improve CNER performance (Zhang and Yang, 2018; Li et al., 2020b; Liu et al., 2021b).

Recently, with the success of pre-trained language models in NLP, such as BERT (Devlin et al., 2019), extracting semantic information from target data has become easier, shifting research attention towards utilizing pre-trained knowledge to support advanced studies in CNER (Mai et al., 2022; Zhang et al., 2022a).

There is also research on FG-CNER (Zhang et al., 2020; Liu et al., 2020) that tries to handle more categories, but the number of categories in these works is still relatively small. Additionally, although Gong et al. (2020) utilized hierarchy information to facilitate their CNER task, the hierarchies they used are based on the tree structures of how characters construct a Chinese word. The hidden relevance between flattened hierarchical labels needs to be further studied. Therefore, based on the FiNE dataset, we propose the SoftFiNE method to provide a better solution for FG-CNER tasks.

Figure 1: The hierarchies of all 54 categories in FiNE.

3 Dataset

To facilitate research on Fine-grained Chinese Named Entity Recognition (FG-CNER) with hierarchical structured labels, we construct the FiNE dataset. A brief introduction of the CNER task in the BIOES (begin, inside, outside, singleton, end) labeling scheme is given in Appendix A.

Construction of hierarchical categories. The hierarchies of the categories in FiNE are collected and refined from Chinese Wikipedia[1]. Based on their released knowledge graph, we collect the labels of every entity and keep those with the highest frequencies. However, since the knowledge graph is generated from the open domain, these collected labels often have redundant meanings. We manually fine-grained the categories into at most 3 levels of hierarchies and removed those that are less acknowledged by most people. Then, following the sequence labeling method for the NER task, we flatten the hierarchies and obtain 54 categories with different numbers of hierarchy levels. A list of all categories is provided in Figure 1 which includes 10 sub-labels for the first level, 29 sub-labels for the second level, and 32 sub-levels for the third level.

Annotation. We created a dataset that simulates real-world scenarios by collecting 30,000 passages from QQ Browser, which provides daily content to a general audience of over 300 million people. We selected passages from 27 different domains such as Medicine, Sports and E-Sports based on their click rates and used TextRank4ZH[2] to extract key sentences. Four experts with professional knowledge in NLP were hired to label and classify the entities into hierarchical categories. To ensure accuracy, the 30,000 sentences were divided into three packs and labeled by three annotators, with the fourth double-checking for correctness. Additionally, 100 samples were manually labeled for each pack, and any discrepancies were resolved through re-labeling the whole pack. In the final version of FiNE, the fourth annotator agreed with 89.93 percent of all labels, with the remaining 10.07 percent

[1]https://dumps.wikimedia.org/zhwiki/

[2]https://github.com/letiantian/TextRank4ZH

Dataset	OntoNotes 4.0			Weibo			Resume			FiNE		
	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
#samples	15,724	4,301	4,346	1350	270	270	3,821	463	477	23,000	3,000	4,000
avg. char.	31.3	46.6	47.9	54.7	53.7	55.0	32.5	30.0	31.7	40.5	41.6	41.5
avg. ent.	0.9	1.6	1.8	1.4	1.4	1.5	3.5	3.2	3.4	2.2	2.5	2.6
#categories	4			7			8			54		

Table 1: The statistics of compared datasets. The average number of characters and the average number of entities are reported in "avg. char." and "avg. ent.". OntoNotes 4.0 are used to test coarse-grained CNER, Resume and Weibo are used to test the performance on FG-CNER.

Approaches	miss	wrong	Rank	SoftFiNE
B-Organization-Company-Vehicle&Parts	0	0	0	1
E-Organization-Company-Vehicle&Parts	0	1	-2	0.25
B-Organization-Company	1	0	-1	0.5
E-Organization-Company	1	1	-3	0.125
O	-	-	-inf	0

Table 2: Example of scoring labels, For SoftFiNE, we set $\beta = 2$ and $\gamma = 2$.

removed from the dataset.

As shown in Table 1, our dataset covers all categories in the validation and test sets, and has more fine-grained categories in hierarchical structures compared to existing CNER datasets. Additionally, our dataset is constructed with online passages from multiple domains, making it more representative of real-world FG-CNER implementation. A detailed distribution of all entities is provided in Appendix B.

4 Methodology

In this section, we present SoftFiNE, a solution to the hierarchical labeled fine-grained CNER problem that effectively utilizes the hidden relevance between flattened hierarchical labels to gain a better understanding of them. Before introducing our approach, we will discuss two alternative methods for solving hierarchical labeled FG-CNER.

4.1 Alternative Methods

There are two existing types of approaches to learning from the hierarchies.

Hierarchical Multi-label Classification. An alternative approach for solving FG-CNER with hierarchical labels is to treat it as a hierarchical multi-label classification task, where the model makes predictions on each sub-label to determine if it is related to the character and then reconstructs the hierarchies based on the predicted sub-labels (Giunchiglia and Lukasiewicz, 2020; Zhang et al., 2022b). However, this method can lead to accumulation of errors in the predictions of each sub-label, resulting in degraded performance. To evaluate the effectiveness of this approach, we implemented the method introduced by Wehrmann et al. (2018) on the FiNE dataset.

Learning to Rank. Another approach for solving hierarchical FG-CNER is to treat it as a Learning to Rank problem, where the flattened hierarchical labels are ranked by relevance, and the most relevant label is used as the prediction (Cao et al., 2007; Liu et al., 2009). To enable this method, we use the rules in Table 2 to obtain a rank list for each character. For a given ground truth, we calculate its relevance in comparison to other labels by counting the number of missed levels ("miss") and the number of wrong sub-labels ("wrong"). The relevance of the ground truth is set to 0, and for every other label, the relevance is calculated as $1 - \text{miss} - 2 \times \text{wrong}$. The score of the label "O" is set to negative infinity. To leverage the relevant information in the rank list, we implemented RankNet (Burges et al., 2005) and LambdaRank (Burges et al., 2006) on the FiNE dataset. However, if two labels have the same score, they cannot be compared using these methods and such label pairs are excluded during training.

4.2 SoftFiNE

In order to better leverage the relevance between labels, we propose SoftFiNE. Our approach is designed to be simple and focuses on capturing the hidden relevance between flattened hierarchical labels through proper augmentation. The detailed structure of SoftFiNE is illustrated in Figure 2.

Scoring Label. To better represent the relevance between flattened hierarchical labels, we design a label scoring method based on their hierarchical structures. For the i^{th} character t_i in a sentence $T = \{t_i\}_{1 \le i \le M}$, where M is the length of T, denotes its ground truth label with $z_i \in Z$, where Z is the set of all possible flattened hierarchical labels with the total number of N. $\bar{Z}_i = \{z_j\}_{j \neq i, 1 \le j \le N}$ represents all other flattened hierarchical labels that are not the ground truth. z_i has λ_i levels in the

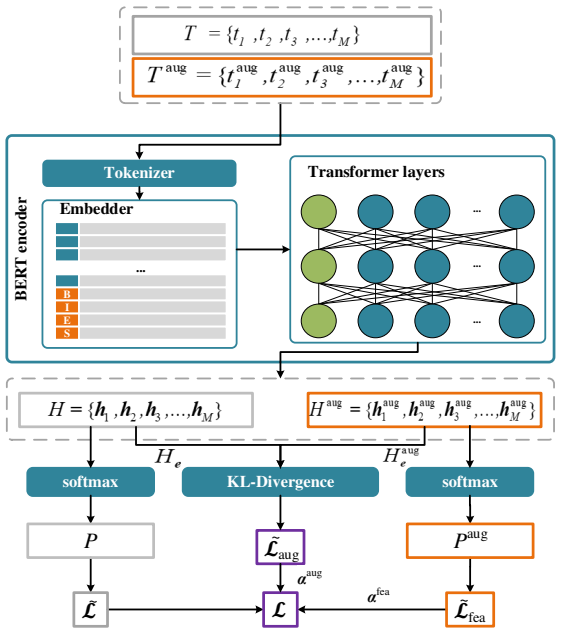

Figure 2: The structure of SoftFiNE. The embedding table of BERT is extended to include the segmentational sub-labels ("B", "I", "E", "S") for augmentation.

format:

$$\begin{cases} \mathrm{O}, & \lambda_i = 0 \\ l_i^{\mathrm{seg}} - l_i^1 - \mathrm{non} - \mathrm{non}, & \lambda_i = 1 \\ l_i^{\mathrm{seg}} - l_i^1 - l_i^2 - \mathrm{non}, & \lambda_i = 2 \\ l_i^{\mathrm{seg}} - l_i^1 - l_i^2 - l_i^3, & \lambda_i = 3 \end{cases},$$

where "non" means no hierarchy at the corresponding level, l_i^1, l_i^2, l_i^3 are the corresponding sub-labels of entity categories and $l_i^{\mathrm{seg}} \in \{B, I, E, S\}$ is the sub-label of BIOES segmentation. Notice characters labeled with "O"have no hierarchy. For each non-"O" label z_x, we use c_x to denote the set of its label components, "non" will not be included.

We first calculate how many levels are missing for each $z_{i,j}$ compared with z_i by

$$\mathrm{miss}_{i,j} = \max(\lambda_j - \lambda_i, 0).$$

Then, we count the number of wrong label components for each z_j compared with z_i:

$$\mathrm{wrong}_{i,j} = \sum_k \mathbf{1}_{i,j}^k,$$

where $\mathbf{1}_{i,j}^k = 1$ if the k^{th} component of c_j not exists in c_i, otherwise is zero.

Finally, we employ the following designed scoring function to obtain the relevance to the ground truth of all labels: $\tilde{y}_i = \{\tilde{y}_{i,j}\}_{1 \le j \le N}$, where

$$\tilde{y}_{i,j} = \begin{cases} \frac{1}{\beta^{(\mathrm{miss}_{i,j} + \gamma \times \mathrm{wrong}_{i,j})}}, & i \neq j, z_j \neq O \\ 0, & z_j = O \\ 1, & \text{otherwise} \end{cases}.$$

$\beta > 1$ and $\gamma > 1$ are two hyperparameters. The larger γ will lead to a harder punishment on wrong-level components when scoring, the larger β will decrease the value of the scores of all flattened hierarchical labels $z_{i,j}$.

The above scoring method can evaluate the degree of how a label is relevant to the ground-truth label, with a maximum degree of 1. The relevance difference is then integrated into those soft labels, which helps the model understand the relations between them. An example of evaluating the relevance between labels is provided in Table 2.

Weighted cross-entropy. To keep the model simple, we employ BERT with a softmax classifier to handle the CNER task.

For a given $t_i \in T$, we encode the sequence with BERT (Devlin et al., 2019):

$$H = \{\boldsymbol{h}_i\}_{1 \le i \le M} = \mathrm{BERT}(T), \qquad (1)$$

where $\boldsymbol{h}_i$ is the feature of t_i. $H = \{\boldsymbol{h}_i\}_{1 \le i \le M}$ is the features of all characters in sentence $T = \{t_i\}_{1 \le i \le M}$.

The result prediction is then

$$\boldsymbol{p}_i = \mathrm{softmax}(\boldsymbol{W}\boldsymbol{h}_i + \boldsymbol{b}),$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are the parameters of a single-layer linear projector. We let $P = \{\boldsymbol{p}_i\}_{1 \le i \le M}$ to represent the prediction of all characters in T, as shown in the bottom-left of Figure 2. The $\mathrm{softmax}(\cdot)$ we used here is the softmax function for normalization.

Given the soft label $\tilde{y}_i$, an intuitive way is to use the cross-entropy to build the training object:

$$\ell_i = -\sum_{j=1}^{N} \tilde{p}_{i,j} \log p_{i,j},$$

where

$$\tilde{p}_{i,j} = \frac{\tilde{y}_{i,j}}{\sum_{k=1}^{N} \tilde{y}_{i,k}}, \qquad (2)$$

$\tilde{p}_{i,j}$ represents the normalized ground truth relevance score between t_i and j^{th} hierarchical label, while $p_{i,j} \in \boldsymbol{p}_i$ is the corresponding predicted normalized relevance score. However, the ground-truth label of different characters may have different numbers of hierarchical levels (λ), which

Sentence	薯	条	这	样	做	真	的	很	好	吃
Augmented	B	E	这	样	做	真	的	很	好	吃
Flattened Hierarchical Label	B-food-dish	E-food-dish	O	O	O	O	O	O	O	O
Multi-label	B,food,dish	E,food,dish	O	O	O	O	O	O	O	O

Table 3: An example of entity augmentation. The sentence means "it is really delicious if making fries like this" in English. For the flattened hierarchical label, each character will be classified into a single class. For multi-label classification, each character will have multiple sub-labels.

leads to a different scaling factor $\sum_{k=1}^{N} \tilde{y}_{i,k}$. After the normalization in Equation (2), the value of the ground truth may be too low if the scaling factor is large, which makes it harder to learn from the sample. Thus, we simply use the scaling factor to weight the loss term of t_i:

$$\begin{aligned}\tilde{\ell}_i &= -(\sum_{k=1}^{N} \tilde{y}_{i,k}) \sum_{j=1}^{N} \frac{\tilde{y}_{i,j}}{\sum_{k=1}^{N} \tilde{y}_{i,k}} \log p_{i,j} \\ &= -\sum_{j=1}^{N} \tilde{y}_{i,j} \log p_{i,j}.\end{aligned} \tag{3}$$

The loss $\tilde{\mathcal{L}}$ for the whole sentence is then

$$\begin{aligned}\tilde{\mathcal{L}} &= \sum_{i=1}^{M} \tilde{\ell}_i \\ &= -\sum_{i=1}^{M} \sum_{j=1}^{N} \tilde{y}_{i,j} \log p_{i,j}.\end{aligned} \tag{4}$$

Augmentation. The above approach helps the model understand the relation between labels from the hierarchies. To better leverage the information provided by the input samples, we propose a novel augmentation method that can help the model better understand the entity classification of CNER.

Table 3 gives an example of our augmentation method. For a given sentence $T = \{t_i\}_{1 \le i \le M}$, we generate its augmentation $T^{\text{aug}} = \{t_i^{\text{aug}}\}_{1 \le i \le M}$. Recall the segmentation label types in BIOES are begin (B), inside (I), outside (O), singleton (S), end (E), denote the segmentation label of t_i as seg_i, we have

$$t_i^{\text{aug}} = \begin{cases} \text{seg}_i, & \text{seg}_i \in \{B, I, E, S\} \\ t_i, & \text{seg}_i = O \end{cases},$$

Following Equation (3), we obtain the loss term of the augmented sentence T^{aug} by

$$\begin{aligned}\tilde{\mathcal{L}}^{\text{aug}} &= \sum_{i=1}^{M} \tilde{\ell}_i^{\text{aug}} \\ &= -\sum_{i=1}^{M} \sum_{j=1}^{N} \tilde{y}_{i,j} \log p_{i,j}^{\text{aug}}\end{aligned}$$

where

$$\boldsymbol{p}_i^{\text{aug}} = \text{softmax}(\boldsymbol{W}\boldsymbol{h}_i^{\text{aug}} + \boldsymbol{b}),$$

$$H^{\text{aug}} = \{\boldsymbol{h}_i^{\text{aug}}\}_{1 \le i \le M} = \text{BERT}(T^{\text{aug}}).$$

We use $P^{\text{aug}} = \{\boldsymbol{p}_i^{\text{aug}}\}_{1 \le i \le M}$ to represent the predictions of all characters in the augmented sentence T^{aug}. The segmentational labels are added to the vocabulary of the tokenizer of BERT. By doing this, our model can learn a general embedding of those labels, which grants a better understanding of the segmentation.

To teach BERT to gather more context information when predicting the labels of the character in entities for a given sentence, denote the features of all characters in all entities of T as H_e, and H_e^{aug} are those for the augmented sentence, we employ KL-divergence to close the distributions of $\hat{\boldsymbol{h}}_x^{\text{aug}} \in H_e^{\text{aug}}$ to $\hat{\boldsymbol{h}}_x \in H_e$, where $1 \le x \le \hat{N}$, and $\hat{N}$ is the number of characters of all entities in the sentence T. Denote the distribution of $\hat{\boldsymbol{h}}_i$ as $\hat{\boldsymbol{d}}_i = \text{softmax}(\hat{\boldsymbol{h}}_i)$, and the distribution of $\hat{\boldsymbol{h}}_i^{\text{aug}}$ as $\hat{\boldsymbol{d}}_i^{\text{aug}} = \text{softmax}(\hat{\boldsymbol{h}}_i^{\text{aug}})$, we can get the feature loss term for t_i:

$$\tilde{\ell}_i^{\text{fea}} = \sum_{j=1}^{\hat{N}} \hat{\boldsymbol{d}}_{i,j} \log \frac{\hat{\boldsymbol{d}}_{i,j}}{\hat{\boldsymbol{d}}_{i,j}^{\text{aug}}},$$

where $\hat{\boldsymbol{d}}_{i,j}$ and $\hat{\boldsymbol{d}}_{i,j}^{\text{aug}}$ are the j^{th} element in $\hat{\boldsymbol{d}}_i$ and $\hat{\boldsymbol{d}}_i^{\text{aug}}$. The feature loss for the whole sequence is then:

$$\begin{aligned}\tilde{\mathcal{L}}^{\text{fea}} &= \sum_{i=1}^{M} \tilde{\ell}_i^{\text{fea}} \\ &= \sum_{i=1}^{M} \sum_{j=1}^{\hat{N}} \hat{\boldsymbol{d}}_{i,j} \log \frac{\hat{\boldsymbol{d}}_{i,j}}{\hat{\boldsymbol{d}}_{i,j}^{\text{aug}}}.\end{aligned}$$

By performing KL-divergence on the original feature and the augmentation feature, BERT learns to generate more robust outputs of the characters of the entities, thus making the classification much easier. Also, when encoding a segmentation label

in the augmented sentence, the model is taught to generate the feature that is close to the original one and make the correct prediction, which forces BERT to integrate the information of the whole sentence better to understand what to generate during the feature extraction.

Training object. Combing the above components, we form the training object of SoftFiNE with

$$\mathcal{L} = \tilde{\mathcal{L}} + \alpha^{\text{aug}}\tilde{\mathcal{L}}^{\text{aug}} + \alpha^{\text{fea}}\tilde{\mathcal{L}}^{\text{fea}},$$

where α^{aug} and α^{fea} are two hyper-parameters we utilize to control the contribution of each kind of loss.

Combining the soft labels with the proposed augmentation approach, SoftFiNE can better understand the flattened hierarchical labels by learning the hidden relevance between them. The augmentation further grants a better understanding of sequence labeling by learning the embeddings of segmentation labels and using them to predict the categories. Moreover, our method has simple model architecture, and it only uses a BERT model with a linear projector for prediction, which grants a fast inference speed.

5 Experiments

We carry out an extensive set of experiments to investigate the effectiveness of SoftFiNE on FiNE and three other commonly used CNER datasets: OntoNotes 4.0, Weibo, and Resume. The statistics of those datasets are given in Table 1. Following previous literature on CNER, we report standard precision (P), recall (R), and micro F1-score (F1) to evaluate the performance. We compare SoftFiNE with baselines having valid official code on the FiNE dataset. Detailed experimental settings are given in Appendix C.

5.1 Results

Table 4 gives the results of SoftFiNE and selected baselines on our FiNE dataset. From the table, we can conclude some interesting findings: the BERT baseline outperforms all other baselines on FiNE; methods with a CRF decoder (Liu et al., 2021b) usually have lower performance than their implementation without CRF. We believe these are caused by the hierarchical structures of the labels, which are much harder compared with coarse-grain labels. Those flattened hierarchical labels are very likely to be relevant to each other, which challenges the models more in distinguishing different categories, thus causing the degradation of the performance of CRF. Instead, by employing softmax normalization as decoders, BERT learns more directly from the samples and their labels. As a transformer-based PLM, its ability to understand tasks is much higher than most designed decoders on our FiNE, which leads to better performance. The hierarchical multi-label classification and learning-to-rank methods underperform the BERT baseline, proving our assumptions above. Moreover, we can observe that BERT does not outperform all baselines on the other three datasets, which have lower number of categories. Also, by comparing the results of RankNet and LambdaRank, we can conclude that it is efficient to include global relevance (ranking) information while training. By keeping the model in a simple structure and leveraging a well-designed supervise method, our method achieves the best overall performance (F1) compared to all other models by at least 0.78.

	P	R	F1
BERT	73.88	70.71	72.26
BERT+CRF	74.77	69.32	71.94
Li et al. (2020b)	63.44	62.72	63.08
Li et al. (2020b)+BERT	70.51	71.68	71.09
Liu et al. (2021b)	71.64	46.41	56.33
Liu et al. (2021b)-CRF	66.85	70.32	68.54
Yu et al. (2020)	71.29	66.63	68.89
Sun et al. (2021)	70.86	72.39	71.62
Shen et al. (2021)	70.08	**73.72**	71.85
Wehrmann et al. (2018)	73.13	69.67	71.36
Burges et al. (2005)	31.48	29.34	30.37
Burges et al. (2006)	70.49	67.86	69.15
SoftFiNE	75.61	70.64	**73.04**
SoftFiNE-soft	**76.73**	68.61	72.45
SoftFiNE-aug.	73.98	70.95	72.43

Table 4: The experimental results on FiNE. "+BERT" mean with BERT, "+CRF" means with CRF, "-CRF" means without CRF.

With the supervision of generated soft labels, SoftFiNE can better understand the relevance of hierarchies to distinguish different flattened hierarchical labels. Moreover, the augmentation method also grants SoftFiNE a better vision of word segmentation and classification.

We implement our method on the other three commonly used datasets to further test the capability of SoftFiNE, and address the bottleneck of existing CNER datasets. The experimental results are shown in Table 5, from where SoftFiNE achieves better overall performance than all other state-of-

	OntoNotes 4.0			Weibo			Resume		
	P	R	F1	P	R	F1	P	R	F1
BERT	82.81	81.05	81.92	72.78	64.90	68.61	95.55	96.01	96.28
BERT+CRF	81.99	81.65	81.82	-	-	67.33	95.75	95.28	95.51
Zhang and Yang (2018)	76.35	71.56	73.88	-	-	58.79	94.81	94.11	94.46
Li et al. (2020b)	-	-	76.45	-	-	63.42	-	-	95.45
Li et al. (2020b)+BERT	-	-	81.82	-	-	68.55	-	-	95.86
Liu et al. (2021b)	-	-	82.08	-	-	70.75	-	-	96.08
Ma et al. (2020)	83.41	82.21	82.81	-	-	70.50	96.08	96.13	96.11
Meng et al. (2019)	81.87	81.40	81.63	67.68	67.71	67.60	96.62	96.48	96.54
Sun et al. (2021)	80.77	83.65	82.18	68.75	72.97	70.80	-	-	-
Mai et al. (2022)	84.30	80.33	83.21	72.98	71.12	72.04	97.12	95.34	96.22
Zhu and Li (2022)	81.65	**84.03**	82.83	70.16	**75.36**	72.66	96.63	**96.69**	96.66
SoftFiNE	83.53	83.11	**83.32**	**75.32**	71.15	**73.18**	**97.40**	96.38	**96.89**
SoftFiNE-soft	**84.35**	81.36	82.83	74.05	69.95	71.94	97.27	96.26	96.76
SoftFiNE-aug	82.69	82.63	82.66	72.46	72.12	72.29	96.03	96.38	96.20

Table 5: The experimental results on OntoNotes 4.0, Weibo and Resume.

the-art baselines. This is because the BIOES sequence labeling method brings hierarchy information into the labels, which SoftFiNE can utilize to understand different labels better. However, the numerical improvement in performance is relatively small, which also commonly exists on other baselines. For example, SoftFiNE outperforms Mai et al. (2022) by 0.11 on OntoNotes 4.0, Mai et al. (2022) outperforms Ma et al. (2020) by 0.11 on Resume, and Sun et al. (2021) outperforms Ma et al. (2020) by 0.30 on Weibo. Although the improvements are numerically small, it does not mean the methods are not good. Instead, this is caused by the limitation of the current CNER dataset: they might be too easy or too small, while the models today are becoming more and more advanced in understanding languages and making precise predictions.

5.2 Ablation Studies

To examine the effectiveness of each component of SoftFiNE, we conduct ablation studies on the four datasets. The results are shown in the corresponding tables, where "-soft" represents SoftFiNE without using generated soft labels, and "-aug" represents SoftFiNE without augmentation.

From the results, we can conclude that both the soft label and the entity augmentation are keys for SoftFiNE to achieve better performance. The augmentation method can greatly boost the precision of the prediction, and it outperforms all baselines on the four datasets on precision. With such an augmentation method, SoftFiNE can learn the embedding of the segmentation labels by using their encoded features to classify recognized entities. Combined with the design of closing their encoded features to those of their original characters, the BERT encoder in SoftFiNE is forced to gather more information from the context of an entity to make correct classification, which grants a high performance on precision.

The soft label method can outperform most baselines by leveraging the hierarchy information between flattened labels. SoftFiNE can leverage the hierarchy information brought by the BIOES sequence labeling method to find the hidden relevance between the different characters of an entity (e.g., "B-person" and "E-person"), and between the characters from different entities that have the same segmentation labels (e.g., "B-person" and "B-location"), which can help the BERT encoder achieve better performance even without fine-grained hierarchical categories.

Combining the advantages of those two novel designs, our SoftFiNE can better understand both CNER and FG-CNER, and achieves the new state-of-the-art of the four datasets.

6 Conclusion

In this paper, to facilitate research on Fine-Grained Chinese Named Entity Recognition (FG-CNER) with hierarchical labels, we construct the FiNE dataset, which contains 30,000 high-quality samples with 54 inter-related categories organized in a hierarchical structure. The data are sampled from various domains of online passages and are closer to the requirements of real-world CNER implementation today. To exploit the hierarchical information present in fine-grained NER labels, we further propose the SoftFiNE method, a novel method aiming at enhancing the performance of FG-CNER by leveraging the hidden relevance

between flattened hierarchical labels and learning with relevance-aware soft labels. The proposed SoftFiNE method also uses entity augmentation to understand samples better. Experimental results show that SoftFiNE has achieved the best performance on four datasets, including FiNE, Weibo, Resume, and OntoNotes 4.0.

Limitations

Our constructed dataset, FiNE, has limitations in terms of entity category balance. Some categories have a lower number of online passages and less user attention, resulting in an unbalanced distribution of entities across categories. We aimed to simulate a real-world situation by sampling passages based on their click rates. But this may have contributed to the imbalance.

Additionally, our proposed method, SoftFiNE, is specifically designed for fine-grained Chinese named entity recognition with hierarchical categories, and thus has its own limitations. The model is kept simple in structure, with most efforts focused on developing supervision methods, which result in more hyperparameters and require a grid search to find the optimal hyperparameters. This can be resource-intensive. However, it has fast inference speed that is comparable to the BERT baseline in real-world applications. To address these limitations, future research could explore methods to automatically balance the loss ratio and dynamically score relevance between flattened hierarchical labels.

Ethics Statement

We sample all data in FiNE from QQ Browser, while the platform has already censored all samples. Therefore, the sensitive or personal information is neither present to the annotator nor included in the released dataset. The annotation of FiNE cost over 25,000 CNY. We hired four experts in NLP data annotation (through a third party), each of whom has at least one year of working experience in NLP data annotation. It took 672 work hours (8 hours per day for 21 days) to complete the annotation process. The hourly rate is about 37.20 CNY, higher than the standard local rate of 25.30 CNY.

References

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96.

Christopher Burges, Robert Ragno, and Quoc Le. 2006. Learning to rank with nonsmooth cost functions. *Advances in neural information processing systems*, 19.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

Wanxiang Che, Mengqiu Wang, Christopher D Manning, and Ting Liu. 2013. Named entity recognition with bilingual constraints. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 52–62.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. 2016. Character-based lstm-crf with radical-level features for chinese named entity recognition. In *Natural Language Understanding and Intelligent Applications*, pages 239–250. Springer.

Asif Ekbal, Eva Sourjikova, Anette Frank, and Simone Paolo Ponzetto. 2010. Assessing the challenge of fine-grained named entity recognition and classification. In *proceedings of the 2010 Named Entities Workshop*, pages 93–101.

Eleonora Giunchiglia and Thomas Lukasiewicz. 2020. Coherent hierarchical multi-label classification networks. *Advances in Neural Information Processing Systems*, 33:9662–9673.

Chen Gong, Zhenghua Li, Qingrong Xia, Wenliang Chen, and Min Zhang. 2020. Hierarchical lstm with char-subword-word tree-structure representation for chinese named entity recognition. *Science China Information Sciences*, 63(10):1–15.

Zhifeng Hao, Hongfei Wang, Ruichu Cai, and Wen Wen. 2013. Product named entity recognition for chinese query questions based on a skip-chain crf model. *Neural Computing and Applications*, 23(2):371–379.

Gina-Anne Levow. 2006. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020a. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):50–70.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuan-Jing Huang. 2020b. Flat: Chinese ner using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6836–6842.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Chenguang Liu, Yongli Yu, Xingxin Li, and Peng Wang. 2021a. Named entity recognition in equipment support field using tri-training algorithm and text information extraction technology. *IEEE Access*, 9:126728–126734.

Jingang Liu, Chunhe Xia, Haihua Yan, and Wenjing Xu. 2020. Innovative deep neural network modeling for fine-grained chinese entity recognition. *Electronics*, 9(6):1001.

Pan Liu, Yanming Guo, Fenglei Wang, and Guohui Li. 2022. Chinese named entity recognition: The state of the art. *Neurocomputing*, 473:37–53.

Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331.

Wei Liu, Xiyan Fu, Yue Zhang, and Wenming Xiao. 2021b. Lexicon enhanced chinese sequence labeling using bert adapter. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5847–5858.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuan-Jing Huang. 2020. Simplify the usage of lexicon in chinese ner. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5951–5960.

Chengcheng Mai, Mengchuan Qiu, Kaiwen Luo, Ziyan Peng, Jian Liu, Chunfeng Yuan, and Yihua Huang. 2022. Pretraining multi-modal representations for chinese ner task with cross-modality attention. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 726–734.

Alireza Mansouri, Lilly Suriani Affendey, and Ali Mamat. 2008. Named entity recognition approaches. *International Journal of Computer Science and Network Security*, 8(2):339–344.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. *Advances in Neural Information Processing Systems*, 32.

Zara Nasar, Syed Waqar Jaffry, and Muhammad Kamran Malik. 2021. Named entity recognition and relation extraction: State-of-the-art. *ACM Computing Surveys (CSUR)*, 54(1):1–39.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 548–554.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2782–2794.

Jian Sun, Jianfeng Gao, Lei Zhang, Ming Zhou, and Changning Huang. 2002. Chinese named entity identification using class-based language model. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021. Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2065–2075.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical multi-label classification networks. In *International conference on machine learning*, pages 5075–5084. PMLR.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. 2011. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*.

Canwen Xu, Feiyang Wang, Jialong Han, and Chenliang Li. 2019. Exploiting multiple embeddings for chinese named entity recognition. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2269–2272.

Zhenyu Xuan, Rui Bao, and Shengyi Jiang. 2020. Fgn: Fusion glyph network for chinese named entity recognition. In *China Conference on Knowledge Graph and Semantic Computing*, pages 28–40. Springer.

Yuan-Nong Ye, Liu-Feng Zheng, Meng-Ya Huang, Tao Liu, and Zhu Zeng. 2021. The algorithms for word segmentation and named entity recognition of chinese medical records. In *International Conference on Artificial Intelligence and Security*, pages 397–405. Springer.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Min Zhang, Bicheng Li, Qilong Liu, and Jing Wu. 2022a. Chinese named entity recognition fusing lexical and syntactic information. In *2022 the 6th International Conference on Innovation in Artificial Intelligence (ICIAI)*, pages 69–77.

Tingting Zhang, Yaqiang Wang, Xiaofeng Wang, Yafei Yang, and Ying Ye. 2020. Constructing fine-grained entity recognition corpora based on clinical records of traditional chinese medicine. *BMC medical informatics and decision making*, 20(1):1–17.

Xinyi Zhang, Jiahao Xu, Charlie Soh, and Lihui Chen. 2022b. La-hcn: label-based attention for hierarchical multi-label text classification neural network. *Expert Systems with Applications*, 187:115922.

Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. *arXiv preprint arXiv:1805.02023*.

Hengyi Zheng, Bin Qin, and Ming Xu. 2021. Chinese medical named entity recognition using crf-mt-adapt and ner-mrc. In *2021 2nd International Conference on Computing and Data Science (CDS)*, pages 362–365. IEEE.

Enwei Zhu and Jinpeng Li. 2022. Boundary smoothing for named entity recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7096–7108.

A Introduction to BIOES

Here we briefly introduce the CNER task in BIOES (begin, inside, outside, singleton, end) labeling scheme. For a given sentence, each character will be tagged by a combination of its segmentational sub-label (B, I, E, S) and its categorical sub-label (e.g., Time-Festival, Event-Historical) if it is a component of a named entity, while the rest characters are labeled with "O" to represents they are not a part of any entity. Table 3 gives a real case of a sentence and its labels. The goal is to find out the correct label for every character in the sentence, from which we can extract the named entities and their categories with respect to those two types of sub-labels. A detailed introduction of CNER and different labeling schemes are provided by Liu et al. (2022).

B Distribution of FiNE

Table 6 gives the distribution of all 67,651 sample in FiNE. We ensure there are at least 100 entities for each flat categories. Notice the number of entities in "Person" is not the summation of all entities whose flat categories has "Person" as their first level sub-label. Instead, they are the entities that can not be classified into any sub-level of "Person". Same for "Location-Site" and "Organization-Company". The categories in FiNE are designed to cover general online passages. Three levels of categories are enough to cover most topics without increasing the annotation cost or making it difficult to distinguish between very similar sub-categories.

C Experimental Setting

For FiNE and OntoNotes 4.0, we set the batch size to 16, the learning rate to $2e{-}5$, dropout rate to 0.5 for all models. We train the models for 20 epochs for FiNE and 5 epochs for OntoNotes 4.0 and use 5% training steps to warn up. For Weibo and Resume, we set the batch size to 4. The decay rates are set to 0.5 and 0.1 for Resume and Weibo, with the learning rates of $3e{-}5$ and $2e{-}5$ correspondingly. The dropout rates are 0.1 and 0.5. We train 12 epochs for those two datasets and utilize 10% of training steps to warm up.

For the hyper-parameters of SoftFiNE, we perform grid search to find the best combination. Specifically, for all datasets, we search $(\alpha^{\text{aug}}, \alpha^{\text{fea}})$ among (0.1,0.1), (0.1,0.05), (0.01,0.01) and (0.01,0.005), β in $\{2, 3, 4\}$ and γ in $\{1.5, 2, 2.5\}$. The result combination of $(\alpha^{\text{aug}}, \alpha^{\text{fea}}, \beta, \gamma)$ are $(0.1, 0.1, 4, 2)$ on FiNE and OntoNotes 4.0, $(0.01, 0.01, 3, 2)$ on Weibo and $(0.01, 0.005, 2, 2)$ on Resume.

For reproduced baselines in Table 4, we only use "bert-base-chinese"[3] to extract the embeddings of tokens for all methods for a fair comparison, other experimental settings are same with their original paper.

We train all methods on a single NVIDIA Tesla A100 GPU. Similar to BERT, our parameter amount is less than 120M. Each epoch of FiNE takes 6 minutes to train, for the other datasets, it takes at most 4 minutes to train an epoch.

Flat Hierarchical Label	Amount	Flat Hierarchical Label	Amount
Person	3299	Work-App-Software	919
Person-Entertainment	4561	Organization-Company	2282
Person-Esport	700	Organization-Company-VehicleBrand	1106
Person-Virtual	3645	Organization-Company-3CBrand	657
Person-Sport	2275	Organization-Company-FashionBrand	396
Person-Economy	1413	Organization-Team-SportTeam	1588
Person-History	2002	Organization-Team-EsportTeam	504
Person-Politics	2723	Organization-Team-EntertainmentTeam	303
Location-Site	1272	Organization-SocialInstitution-Public	1235
Location-Site-City	4660	Organization-SocialInstitution-Government	2009
Location-Site-Scenery	514	Organization-SocialInstitution-Financing	652
Food-Material	2068	Organization-SocialInstitution-Association	542
Food-Dish	1742	Event-Historical	568
Creature-Pet	417	Event-SocialEvent-SportActivity	973
Creature-PlantAnimal	2277	Event-SocialEvent-EntertainmentEvent	410
Time-DataTime	4462	Event-SocialEvent-SocialActivity	410
Time-Festival	591	Event-SocialEvent-Exhibition	135
Time-Dynasty	509	Product-Consumer-Technology	441
Work-Video-Movie	1006	Product-Consumer-DailyNecessity	275
Work-Video-TVSeries	553	Product-Consumer-FinancialProduct	245
Work-Video-VarietyShow	263	Product-VirtualGoods	720
Work-Video-Animation	426	Product-Tool-VehicleParts	1209
Work-Video-TVShow	553	Product-Tool-Transportation	116
Work-Art-Literature	701	Product-Tool-Weaponry	627
Work-Art-Documentary	620	MedicalCare-Treatment	343
Work-Art-Music	581	MedicalCare-Drug	829
Work-App-Game	1303	MedicalCare-Disease	2231

Table 6: The distribution of all 67,651 entities in FiNE.

	P	R	F1
SoftFiNE$_{\text{level-1}}$	84.84	81.83	83.03
SoftFiNE$_{\text{level-2}}$	77.27	72.80	74.97
SoftFiNE$_{\text{level-3}}$	75.61	70.64	73.04

Table 7: The performance of SoftFiNE on different label levels of FiNE.

D Additional Experimental Results

Table 7 gives the test performance of SoftFiNE on FiNE with different label levels. "SoftFiNE$_{\text{level-1}}$" gives the performance of only checking whether the first level is correctly predicted, "SoftFiNE$_{\text{level-2}}$" only checks the first two levels, and "SoftFiNE$_{\text{level-3}}$" checks all levels. For example, if a character has the true label "B-Work-App-Software" but is wrongly predicted as "B-Work-APP-Game", it will be seen as a correct prediction by "SoftFiNE$_{\text{level-1}}$" since the first levels of the true label and the prediction are all "B-Work". Similarly, it will also be seen as correct by "SoftFiNE$_{\text{level-2}}$" with both having "B-Work-App". However, the prediction is false for "SoftFiNE$_{\text{level-3}}$", since the third levels of the two labels are different.

From Table 7, we can observe an apparent degradation in performance when considering more levels. This demonstrates the challenge brought by having hierarchical labels in the CNER task, which commonly exist in CNER applications today. We believe FiNE can greatly support future CNER research by providing entities with hierarchical structured fine-grained labels.

E Case Study

Table 8 gives a real-case example in FiNE, together with the ground truth and predicted (by SoftFiNE) relevance scores of some hierarchical labels. From the table, we can observe that SoftFiNE can successfully predict the relevance between the ground truth label and other labels. For example, for character "我", SoftFiNE can find the degree of relevance to "S-Work-Video-TVShow", which has the same categorical label but different segmentational sub-label to its ground truth label "B-Work-Video-TVShow". Such an ability to judge relevance between flattened hierarchical labels can help SoftFiNE better understand CNER tasks and perform better.

From Table 8, we can also find an interesting phenomenon that our model thinks labels with wrong segmentational sub-labels (i.e., B, I, O, E, S) are more relevant to the true label than labels with

[3]https://huggingface.co/bert-base-chinese

参加的是第一届《我型我秀》,他是那一年的冠军。 Participating in the first My Show, he took the crown.				
	我	型	我	秀
B-Work-Video-TVShow	1.000/1.000	0.062/0.054	0.062/0.045	0.062/0.056
I-Work-Video-TVShow	0.062/0.032	1.000/1.000	1.000/1.000	0.062/0.062
E-Work-Video-TVShow	0.062/0.037	0.062/0.054	0.062/0.051	1.000/1.000
S-Work-Video-TVShow	0.062/0.049	0.062/0.066	0.062/0.068	0.062/0.053
B-Work-Video-VarietyShow	0.062/0.029	0.004/0.002	0.004/0.002	0.004/0.002
I-Work-Video-VarietyShow	0.004/0.003	0.062/0.028	0.062/0.028	0.004/0.004
E-Work-Video-VarietyShow	0.004/0.002	0.004/0.002	0.004/0.002	0.062/0.028
S-Work-Video-VarietyShow	0.004/0.003	0.004/0.003	0.004/0.003	0.004/0.004

Table 8: A good case example from the test set of FiNE, where "我型我秀(My Show)" is the named entity. The table gives the relevance scores of some representative labels. In each cell, the left score is the true relevance score calculated with $\beta = 4$ and $\omega = 2$, and the right score is the prediction made by SoftFiNE.

用沙琪玛拍少年的你。 Filming Better Days by using sachima.				
	少	年	的	你
B-Work-Video-Movie	1.000/0.000	0.062/0.000	0.062/0.000	0.062/0.000
I-Work-Video-Movie	0.062/0.000	1.000/0.000	1.000/0.000	0.062/0.000
E-Work-Video-Movie	0.062/0.000	0.062/0.000	0.062/0.000	1.000/0.000
S-Work-Video-Movie	0.062/0.000	0.062/0.000	0.062/0.000	0.062/0.000

Table 9: A bad case example from the test set of FiNE similar to Table 8, where both "沙琪玛(sachima)" and "少年的你(Better Days)" are named entities. The table presents the predictions of "少年的你(Better Days)".

wrong categorical sub-labels. For example, for character "我", "B-Work-Video-VarietyShow" is less relevant to "B-Work-Video-TVShow" compared with "S-Work-Video-TVShow". This may suggest that the segmentational sub-labels and the categorical sub-labels should have different weights when scoring the relevance between flattened hierarchical labels. Future research on FG-CNER with hierarchical labels could follow the suggestions and either design a better relevance scoring function or iteratively refine relevance based on the predictions of their models.

Table 9 gives a bad case that commonly exists on SoftFiNE and all baselines. In the table, SoftFiNE predicts the relevance of all related flattened labels as zero, where, intuitively, it should at least give some scores greater than zero. One reason may be the polysemy in the Chinese language. While the original meaning of the sentence is "filming (the poster of) Better Days by using sachima (as background with photo editing techniques)", in Chinese, it can also mean "slap the young you with sachima", since "拍" can either mean "film" or "slap" and "少年的你" means "young you" if directly translated into English. The models may incorrectly understand the sentence and thus fail to recognize "Better Days" even if the entity indeed exists in the training set of FiNE. Moreover, the sentence itself is short, which contains less information to help the model correctly understand its meaning.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitation

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☑ A4. Have you used AI writing assistants when working on this paper?
use Grammarly to check grammar

B ☑ Did you use or create scientific artifacts?

3,5

☑ B1. Did you cite the creators of artifacts you used?
3,5

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
3

C ☑ Did you run computational experiments?

5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
3

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Ethic

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
1

☑ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Ethic statement

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.