# Recognizing Nested Entities from Flat Supervision: A New NER Subtask, Feasibility and Challenges

**Anonymous ACL submission**

## Abstract

Many recent named entity recognition (NER) studies criticize flat NER for its non-overlapping assumption, and switch to investigating nested NER. However, existing nested NER models heavily rely on training data annotated with nested entities, while labeling such data is costly. This study proposes a new subtask, *nested-from-flat NER*, which corresponds to a realistic application scenario: given data annotated with flat entities only, one may still desire the trained model capable of recognizing nested entities.

To address this task, we train span-based models and deliberately ignore the spans nested inside labeled entities, since these spans are possibly unlabeled entities. With nested entities removed from the training data, our model achieves 54.8%, 54.2% and 41.1% $F_1$ scores on the subset of spans within entities on ACE 2004, ACE 2005 and GENIA, respectively. This suggests the effectiveness of our approach and the feasibility of the task. In addition, the model's performance on flat entities is entirely unaffected. We further manually annotate the nested entities in the test set of CoNLL 2003, creating a nested-from-flat NER benchmark.[1] Analysis results show that the main challenges stem from the data and annotation inconsistencies between the flat and nested entities.

## 1 Introduction

Named entity recognition (NER) is a fundamental natural language processing (NLP) task that requires detecting text spans of interest, and classifying them into pre-defined entity categories, e.g., Person, Organization, Location. Researchers had been long-term investigating *flat NER* where entity spans are assumed non-overlapping (Collobert et al., 2011; Huang et al., 2015; Lample et al., 2016), while many recent studies criticize such flat setting and switch to *nested NER* that allows an entity to contain other entities inside it (Katiyar and Cardie, 2018; Sohrab and Miwa, 2018; Yu et al., 2020; Yan et al., 2021). For example, a location entity "New York" can be nested in an organization entity "New York University". In nested NER, the two entities are equally considered from annotation through evaluation, while flat NER focuses on the outer entity but ignores the nested one entirely (Tjong Kim Sang and De Meulder, 2003; Finkel and Manning, 2009).[2] Nested NER appears a more general and realistic setting since nested entities are ubiquitous in natural language.

Labeling nested entities is particularly labor-intensive, complicated, and error-prone; for example, Ringland et al. (2019) reported that entities can be nested up to six layers. However, all existing nested NER systems heavily rely on *nested supervision*, namely training on annotated nested NER datasets, such as ACE 2004, ACE 2005 and GENIA (Kim et al., 2003). Directly imposing *flat supervision* would misguide the models to ignore nested structures. This creates an obstacle for them to utilize well-annotated flat NER resources, such as CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) and OntoNotes 5.

This study proposes a new subtask, *nested-from-flat NER*, which asks to train a nested NER model with purely flat supervision. This corresponds to a realistic application scenario: given training data annotated with flat entities only, one may still desire the trained model capable of extracting nested entities from unseen text.

To address this challenging task, we exploit the span-based NER framework which explicitly distinguishes positive samples (i.e., entity spans) from negative samples (i.e., non-entity spans). When training a span-based neural NER model, a stan-

[1] Our code and annotations will be publicly released.

[2] See CoNLL 2003 Annotation Guidelines (https://www-nlpir.nist.gov/related_projects/muc/proceedings/ne_task.html), Subsections 4.3 and A.1.3.

dard protocol regards all unannotated spans as negative samples (Sohrab and Miwa, 2018; Yu et al., 2020; Zhu et al., 2022); however, we deliberately ignore the spans nested in any labeled entities, because these spans are possibly unlabeled nested entities. When the trained model generalizes to all spans, it naturally predicts all possible entities, which may contain nested ones. This is theoretically feasible because the recognizable patterns for flat entities should be to some extent transferable to nested entities.

With nested entities removed from the training splits of ACE 2004, ACE 2005 and GENIA, the nested-from-flat model can achieve 54.8%, 54.2% and 41.1% $F_1$ scores on the subset of spans within entities, respectively. Besides, the overall $F_1$ scores reach 79.2%, 79.3% and 77.3%. Moreover, the model's ability to recognize nested entities does not hurt its performance on flat entities. We further annotate the nested entities in the test split of CoNLL 2003, and analyze the recognition results of our models. We find nested-from-flat NER a challenging task mainly because the annotation standards and data distributions are inconsistent between the flat and nested entities.

This study contributes in threefold:

- We propose nested-from-flat NER, a new subtask with realistic application scenarios. Compatibly, we design a metric – $F_1$ score on the spans within entities, which dedicatedly evaluates how well the model extracts nested entities.
- We provide a solution to nested-from-flat NER, which simply ignores the spans nested in entities during training. Experimental results confirm its effectiveness, as well as the feasibility of this task.
- We manually annotate the nested entities in the test split of CoNLL 2003, resulting in a nested-from-flat NER benchmark named CoNLL 2003 NFF.

## 2 Related Work

The NER task was originally proposed in a context where entities could be regarded as small chunks and thus detected by finite state models (Finkel and Manning, 2009). Hence, in the early years, NER corpus designers chose to annotate only the outermost entities, but ignore/remove the nested ones (Tjong Kim Sang and De Meulder, 2003; Collier and Kim, 2004); and algorithm researchers were focused on using sequence models, such as the conditional random field (CRF) (Lafferty et al., 2001), to recognize flat entities. Facilitated by the deep learning technologies (Krizhevsky et al., 2012; LeCun et al., 2015), neural sequence tagging models with an optional linear-chain CRF became the *de facto* standard solution to flat NER (Collobert et al., 2011; Huang et al., 2015; Lample et al., 2016; Zhang and Yang, 2018; Devlin et al., 2019).

However, nested entities are ubiquitous in natural language. Many recent studies criticize the flat assumption, and switch to a setting that allows nested entities (Finkel and Manning, 2009). This also remarkably facilitates the progress in NER system designs beyond the traditional sequence tagging framework. Hypergraph-based models adopt a tagging scheme that allows multiple tags for a single token and multiple transitions between tags at adjacent positions, and thus complies with nested structures (Lu and Roth, 2015; Katiyar and Cardie, 2018). Span-based methods enumerate or propose candidate spans, and then classify the spans into entity categories (Sohrab and Miwa, 2018; Eberts and Ulges, 2020; Yu et al., 2020; Shen et al., 2021). Other approaches include stacked sequence tagging models (Ju et al., 2018), reformulating NER as a reading comprehension task (Li et al., 2020) or a generation task (Yan et al., 2021), set prediction (Tan et al., 2021; Shen et al., 2022), and word-word relation prediction (Li et al., 2022).

Almost all the existing nested NER models heavily rely on annotated nested NER resources, while labeling nested entities is labor-intensive, complicated and error-prone (Ringland et al., 2019). This study proposes nested-from-flat NER, exploring the possibility of training a nested NER model with flatly annotated data.

## 3 Method

**Span-based NER.** Given a $T$-length sentence, a span-based neural NER model enumerates all possible spans, and builds a span representation $\boldsymbol{z}_{ij} \in \mathbb{R}^d$ for each span $(i, j)$, typically based on the contextualized embeddings from a pretrained language model (PLM). The span representations are then fed into a classifier:

$$\hat{\boldsymbol{y}}_{ij} = \mathrm{softmax}(\mathbf{W}\boldsymbol{z}_{ij} + \boldsymbol{b}), \qquad (1)$$

where $\mathbf{W} \in \mathbb{R}^{c \times d}$ and $\boldsymbol{b} \in \mathbb{R}^c$ are learnable parameters, and $\hat{\boldsymbol{y}}_{ij} \in \mathbb{R}^c$ is the estimated posterior probabilities over entity types (including an additional "non-entity" type).

Given the one-hot encoded ground truth $\boldsymbol{y}_{ij} \in \mathbb{R}^c$, the model can be trained by optimizing the cross-entropy loss for all spans:

$$\mathcal{L} = -\sum_{0 \le i \le j < T} \boldsymbol{y}_{ij}^{\mathsf{T}} \log(\hat{\boldsymbol{y}}_{ij}). \tag{2}$$

In the inference time, the spans predicted to be "non-entity" are discarded; while the remaining ones, together with their predicted types, are output as recognized entities.

**Nested-from-Flat NER.** We start from an example sentence: "Mr. John Smith graduated from New York University last year". In a typical nested NER annotation scheme, "John Smith", "New York University" and "New York" should be labeled as Person, Organization and Location entities, respectively; while flat NER ignores any nested entities, namely "New York" in this example. Nested-from-flat NER asks: if only flat entities are available in the training data, how to develop a model that recognizes nested entities in unseen sentences?

Formally, given a sentence annotated with flat entities, denote all the spans as a set $\mathcal{A}$, and the entity spans as a set $\mathcal{E}$. A standard span-based NER modeling protocol regards $\mathcal{E}$ as positive samples, and its complement $\mathcal{A} \setminus \mathcal{E}$ as negative samples. However, unlabeled nested entities may exist in $\mathcal{A} \setminus \mathcal{E}$ and thus be incorrectly treated as negative samples.

To address this issue, we define the set of *within-entity spans*:

$$\mathcal{I} = \{(s, e) \mid \exists\, (s', e') \in \mathcal{E}, \text{s.t. } s' \le s \le e < e' \text{ or } s' < s \le e \le e'\}, \tag{3}$$

and the set of *out-of-entity spans*:

$$\mathcal{O} = \mathcal{A} \setminus \mathcal{I}. \tag{4}$$

Note that the two sets are mutually exclusive; and the entity spans belong to the out-of-entity spans, i.e., $\mathcal{E} \subseteq \mathcal{O}$. Figure 1 visualizes the two sets of spans of the aforementioned 10-token sentence, which cover the upper triangular area of the resulting 10x10 matrix.

Clearly, unlabeled nested entities can only appear in the within-entity spans $\mathcal{I}$, rather than the out-of-entity spans $\mathcal{O}$. Therefore, the supervisory signals (positive vs. negative samples) are reliable in $\mathcal{O}$, but of high risk in $\mathcal{I}$.

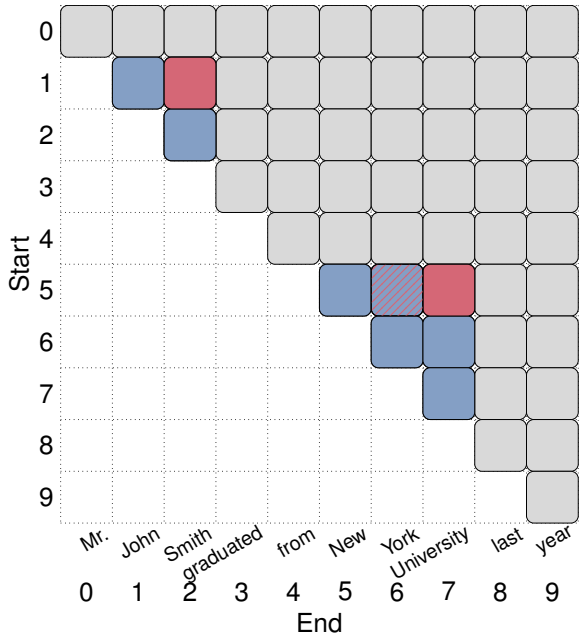



Figure 1: Within-entity and out-of-entity spans for sentence "Mr. John Smith graduated from New York University last year". Within-entity spans are colored in blue. Out-of-entity spans include the entity spans colored in red, and those colored in gray. The span colored in blue but hatched in red is an unlabeled nested entity.

This leads to the key ingredient of our solution. In the training time, we train the model with samples from $\mathcal{O}$ while ignore $\mathcal{I}$:

$$\mathcal{L} = -\sum_{(i,j) \in \mathcal{O}} \boldsymbol{y}_{ij}^{\mathsf{T}} \log(\hat{\boldsymbol{y}}_{ij}). \tag{5}$$

Empirically, $\mathcal{O}$ contains substantially more span-level samples than $\mathcal{I}$. For example, the out-of-entity spans are over 100 times more than the within-entity spans in the CoNLL 2003 training split. Hence, the out-of-entity spans are sufficient for training the model.

In the inference time, we let the model generalize to all spans of test sentences, predicting all possible entities, regardless nested or not. If the model is well-trained, it is able to recognize entities nested within others.

**Negative Sampling on Within-Entity Spans.** As aforementioned, in a standard protocol, the within-entity spans $\mathcal{I}$ are all regarded as negative samples because they are unlabeled. While in our method for nested-from-flat NER, the spans in $\mathcal{I}$ are entirely ignored in training.

Inspired by Li et al. (2021), we find it sometimes beneficial to additionally sample a few spans from $\mathcal{I}$ and use them as negative samples. To formulate this trick, we introduce a hyperparameter $\gamma$, which represents the negative sampling rate for $\mathcal{I}$. Thus, we have three schemes for the within-entity spans:

- *Full Negative* ($\gamma = 1$): Using all spans in $\mathcal{I}$ as negative samples; this corresponds to the standard span-based NER training protocol.
- *Sampling* ($0 < \gamma < 1$): Randomly sampling spans with probability $\gamma$ from $\mathcal{I}$ as negative samples. Empirically, a relatively small sampling rate (e.g., $\gamma = 0.01$, our default) works much better than large rates.
- *Full Ignoring* ($\gamma = 0$): Ignoring $\mathcal{I}$ in training.

## 4 Experimental Settings

**Datasets.** Although our model can be trained on flatly annotated data, it relies on test data with nested entities to evaluate the trained model. Hence, we first perform experiments on some nested NER benchmarks, i.e., ACE 2004[3], ACE 2005[4] and GENIA (Kim et al., 2003). Before training, we deliberately remove all the nested entities but keep the outermost ones in the training and development splits (Finkel and Manning, 2009), which thus satisfies the nested-from-flat setting.

In addition, we manually annotate the nested entities in the test split of CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003). Three human experts were hired for this project, and they were asked to strictly follow the original CoNLL 2003 Annotation Guidelines. The final annotation results are based on an additional round of manual validation that resolves the disagreements between the annotators. The resulting dataset is named *CoNLL 2003 NFF*, which dedicatedly serves as a nested-from-flat NER benchmark. More details on the data annotation, processing and descriptive statistics can be found in Appendix A.

**Evaluation.** Same as in the standard NER, an entity is evaluated to be correct if both its predicted boundaries and category exactly match the ground truth. The evaluation metric is the micro $F_1$ score on the test split. Unless otherwise noted, we run each experiment for 10 times and report the average $F_1$ score with corresponding standard deviation.

[3] https://catalog.ldc.upenn.edu/LDC2005T09.
[4] https://catalog.ldc.upenn.edu/LDC2006T06.

In addition to the overall $F_1$ score that considers all spans, we separately evaluate the trained model on the within-entity spans $\mathcal{I}$ and the out-of-entity spans $\mathcal{O}$, yielding within-entity and out-of-entity $F_1$ scores, respectively. In this study, the within-entity $F_1$ score is the core metric, which reflects how well the model recognizes nested entities.

**Hyperparameters.** In all experiments, we use RoBERTa (Liu et al., 2019) of the base size (12 layers, 768 hidden size) as the PLM, followed by a single-layer 400-dimensional LSTM (Hochreiter and Schmidhuber, 1997). We choose three representative span-based NER decoders, i.e., SpERT (Eberts and Ulges, 2020), biaffine (Yu et al., 2020) and DSpERT (Zhu et al., 2022), where the DSpERT is specified with a 6-layered span Transformer. In addition, boundary smoothing regularization (Zhu and Li, 2022) is applied with $\epsilon = 0.1$.

The models are trained by the AdamW optimizer (Loshchilov and Hutter, 2018) for 20 epochs with batch size 48. Gradients are clipped at $\ell_2$-norm of 5 (Pascanu et al., 2013). The learning rates are 1.5e-5 and 2.5e-3 for pretrained weights and randomly initialized weights, respectively; a scheduler of linear warmup is applied in the first 20% epochs followed by linear decay.

**Computational Cost.** Based on the above configurations, a DSpERT consists of 170.4M parameters; it takes about 5.6 hours to train a DSpERT on ACE 2004/2005 and GENIA, and 1.3 hours on CoNLL 2003 NFF. A SpERT/biaffine model has 126M parameters, and the training time is 1/5–1/3 of that for DSpERT. All the experiments are run on NVIDIA RTX A6000 GPUs.

## 5 Results on Nested NER Datasets

Table 1 presents the evaluation results, i.e., within-entity, out-of-entity and overall $F_1$ scores of three span-based models on ACE 2004, ACE 2005 and GENIA. *Full Negative*, *Sampling* and *Full Ignoring* are three training schemes described above, which perform nested-from-flat experiments where the nested entities are removed from the training and development splits. *Sampling* uses a fixed rate $\gamma = 0.01$. An exception is *Gold Superv.*, which retains and uses the ground-truth nested entities for training; this serves as an empirical upper bound for the nested-from-flat results.

DSpERT+*Sampling* appears the best configuration for recognizing nested entities, achieving

| | ACE 2004 | | | ACE 2005 | | | GENIA | | |
|---|---|---|---|---|---|---|---|---|---|
| | Within | Out | Overall | Within | Out | Overall | Within | Out | Overall |
| SpERT | | | | | | | | | |
| + Full Negative | $7.7_{\pm 0.4}$ | $84.1_{\pm 0.9}$ | $69.6_{\pm 0.8}$ | $7.6_{\pm 1.6}$ | $82.1_{\pm 0.7}$ | $71.0_{\pm 0.6}$ | $7.5_{\pm 0.9}$ | $79.6_{\pm 0.7}$ | $74.3_{\pm 0.7}$ |
| + Sampling | $17.3_{\pm 1.1}$ | $\mathbf{86.3}_{\pm 0.4}$ | $\mathbf{71.4}_{\pm 0.6}$ | $23.8_{\pm 2.0}$ | $\mathbf{84.8}_{\pm 0.2}$ | $\mathbf{72.9}_{\pm 0.5}$ | $26.7_{\pm 1.8}$ | $\mathbf{80.8}_{\pm 0.5}$ | $\mathbf{74.8}_{\pm 0.5}$ |
| + Full Ignoring | $\mathbf{21.2}_{\pm 1.1}$ | $\mathbf{86.4}_{\pm 0.5}$ | $65.0_{\pm 1.1}$ | $\mathbf{27.2}_{\pm 0.7}$ | $\mathbf{84.7}_{\pm 0.4}$ | $69.1_{\pm 0.8}$ | $\mathbf{28.2}_{\pm 1.8}$ | $\mathbf{80.5}_{\pm 0.3}$ | $74.4_{\pm 0.3}$ |
| + *Gold Superv.* | *77.2*$_{\pm 1.3}$ | *84.5*$_{\pm 0.8}$ | *82.3*$_{\pm 0.7}$ | *73.8*$_{\pm 0.9}$ | *83.2*$_{\pm 0.2}$ | *80.9*$_{\pm 0.3}$ | *51.9*$_{\pm 0.5}$ | *81.0*$_{\pm 0.5}$ | *77.7*$_{\pm 0.4}$ |
| Biaffine | | | | | | | | | |
| + Full Negative | $9.1_{\pm 0.4}$ | $86.9_{\pm 0.3}$ | $72.3_{\pm 0.2}$ | $9.9_{\pm 1.3}$ | $84.4_{\pm 0.4}$ | $73.5_{\pm 0.3}$ | $15.4_{\pm 1.1}$ | $82.6_{\pm 0.2}$ | $\mathbf{77.1}_{\pm 0.2}$ |
| + Sampling | $34.0_{\pm 2.1}$ | $\mathbf{88.0}_{\pm 0.3}$ | $\mathbf{74.9}_{\pm 0.5}$ | $41.2_{\pm 1.3}$ | $\mathbf{86.1}_{\pm 0.3}$ | $\mathbf{77.0}_{\pm 0.3}$ | $\mathbf{39.4}_{\pm 1.1}$ | $\mathbf{83.7}_{\pm 0.3}$ | $76.6_{\pm 0.2}$ |
| + Full Ignoring | $\mathbf{40.9}_{\pm 1.3}$ | $\mathbf{88.1}_{\pm 0.2}$ | $74.4_{\pm 0.4}$ | $\mathbf{45.4}_{\pm 1.1}$ | $\mathbf{86.3}_{\pm 0.4}$ | $76.7_{\pm 0.4}$ | $\mathbf{39.1}_{\pm 1.0}$ | $\mathbf{83.8}_{\pm 0.3}$ | $76.2_{\pm 0.3}$ |
| + *Gold Superv.* | *86.2*$_{\pm 0.2}$ | *87.5*$_{\pm 0.2}$ | *87.1*$_{\pm 0.2}$ | *83.7*$_{\pm 0.6}$ | *85.9*$_{\pm 0.2}$ | *85.4*$_{\pm 0.3}$ | *54.2*$_{\pm 0.6}$ | *83.4*$_{\pm 0.2}$ | *79.6*$_{\pm 0.1}$ |
| DSpERT | | | | | | | | | |
| + Full Negative | $9.4_{\pm 0.7}$ | $86.9_{\pm 0.2}$ | $72.3_{\pm 0.2}$ | $11.1_{\pm 1.6}$ | $84.5_{\pm 0.2}$ | $73.7_{\pm 0.2}$ | $10.3_{\pm 0.7}$ | $82.9_{\pm 0.3}$ | $\mathbf{77.3}_{\pm 0.3}$ |
| + Sampling | $\mathbf{54.8}_{\pm 1.3}$ | $\mathbf{88.6}_{\pm 0.1}$ | $\mathbf{79.2}_{\pm 0.4}$ | $\mathbf{54.2}_{\pm 1.2}$ | $\mathbf{86.7}_{\pm 0.2}$ | $\mathbf{79.3}_{\pm 0.2}$ | $\mathbf{41.1}_{\pm 0.9}$ | $\mathbf{83.7}_{\pm 0.6}$ | $76.6_{\pm 0.5}$ |
| + Full Ignoring | $39.4_{\pm 2.5}$ | $85.9_{\pm 1.4}$ | $65.6_{\pm 2.0}$ | $39.6_{\pm 3.6}$ | $84.9_{\pm 1.3}$ | $68.5_{\pm 2.6}$ | $\mathbf{40.9}_{\pm 1.1}$ | $\mathbf{83.3}_{\pm 0.7}$ | $76.0_{\pm 0.4}$ |
| + *Gold Superv.* | *87.0*$_{\pm 0.3}$ | *88.0*$_{\pm 0.3}$ | *87.7*$_{\pm 0.2}$ | *85.6*$_{\pm 0.6}$ | *86.0*$_{\pm 0.2}$ | *85.9*$_{\pm 0.2}$ | *55.9*$_{\pm 0.8}$ | *83.7*$_{\pm 0.1}$ | *80.3*$_{\pm 0.1}$ |

Table 1: Results of nested-from-flat experiments on nested NER datasets. Reported are average $F_1$ scores with corresponding standard deviations of 10 independent runs. The normally styled rows are results by a nested-from-flat setting where nested entities are removed from the training and development splits. The *gray italicized* rows (i.e., "*Gold Superv.*") are results with nested entities retained and used in training; these serve as an empirical upper bound for the nested-from-flat experiments. The best $F_1$ scores are in bold for each model.

within-entity $F_1$ scores of 54.8%, 54.2% and 41.1% on ACE 2004, ACE 2005 and GENIA, respectively. These scores are largely 2/3 – 3/4 of the corresponding "oracle" results (i.e., 87.0%, 85.6% and 55.9%) by gold supervision. Considering the unavailability of nested supervision, such performance is very encouraging, suggesting the feasibility of the nested-from-flat NER task. In addition, *Full Ignoring* and *Sampling* significantly outperform the standard span-based NER training protocol, i.e., *Full Negative*, across all models and datasets; this suggests the effectiveness of our proposed approach. As previously analyzed, the within-entity spans probably contain unlabeled nested entities, so treating them all as negative samples strongly biases the model's behavior in recognizing nested entities.

For each model, the out-of-entity $F_1$ scores are in general of similar magnitudes across different schemes. This means that the additional ability for recognizing nested entities is obtained for free, without any performance sacrifice on the flat (outermost) entities.

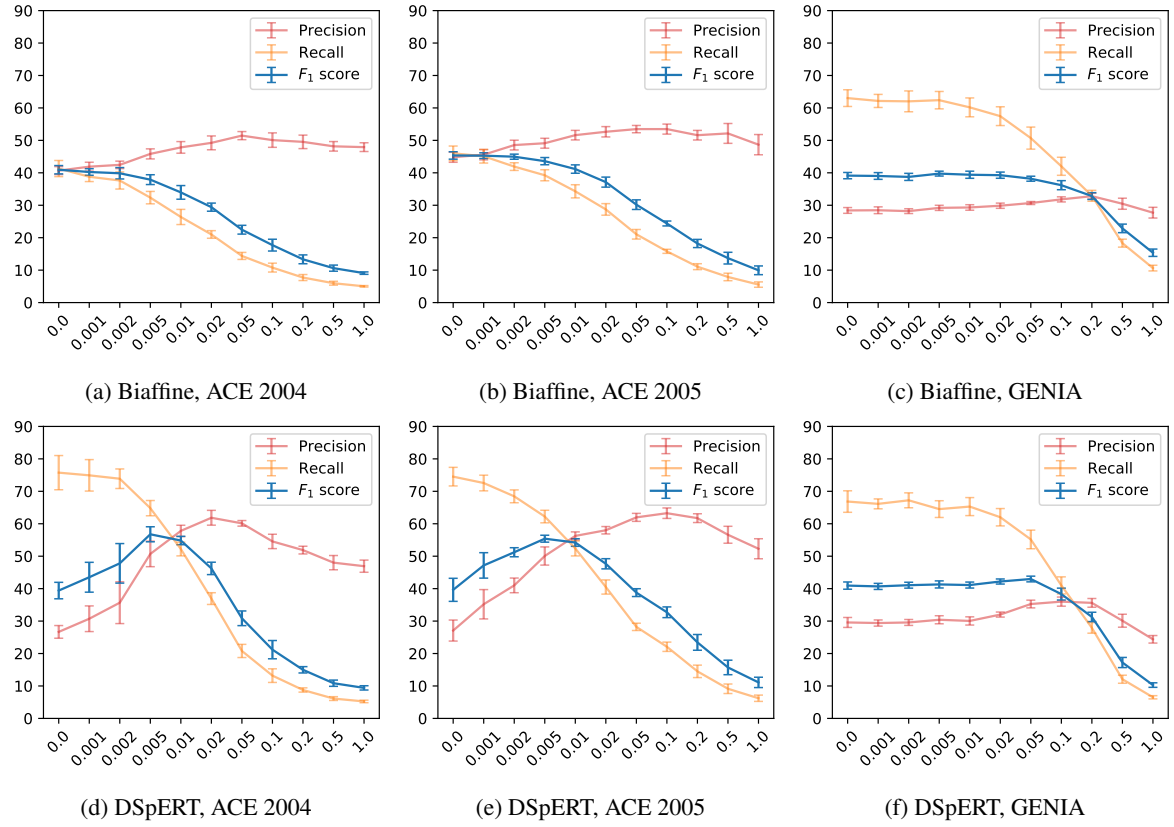
The best overall $F_1$ scores by the nested-from-flat models are 79.2%, 79.3% and 77.3% on ACE 2004, ACE 2005 and GENIA, respectively. Such performance is also competitive, with 3.0 – 8.5 percentage gaps to the upper bounds. Note that our results without any nested supervision are even comparable to the state-of-the-art nested NER performance reported several years ago (e.g., Katiyar and Cardie, 2018; Wang and Lu, 2018).

**Effect of Negative Sampling.** According to the experimental results, DSpERT perform best with *Sampling* (i.e., $\gamma = 0.01$), while SpERT and biaffine are more compatible with *Full Ignoring* (i.e., $\gamma = 0$). To investigate the effect of negative sampling, we plot the within-entity precision rates, recall rates and $F_1$ scores for different negative sampling rates in Figure 2. It shows that the resulting patterns and thus the optimal values of $\gamma$ significantly differ across models and datasets.

As shown in Figures 2c–2f, without negative sampling, the trained model may produce a recall rate much higher than the precision rate. Negative sampling dynamically rebalances the precision and recall rates. In general, a higher negative sampling rate $\gamma$ guides the model to classify the within-entity spans more likely as negative samples, which results in a higher precision but a lower recall.[5] Hence, the precision-recall balance can be achieved by setting a good value for $\gamma$. For example, DSpERT finds the optimal $\gamma = 0.005$ on ACE 2004/2005 (Figures 2d and 2e), and the optimal $\gamma = 0.05$ on GENIA (Figure 2f). On the other hand, in case that the precision and recall are balanced when $\gamma = 0$ (Figures 2a and 2b), negative sampling is unnecessary.

However, a higher within-entity $F_1$ score from

[5] Empirically, the precision rate also turns to decrease after $\gamma$ exceeds a relatively large value, e.g., 0.1.




(a) Biaffine, ACE 2004 (b) Biaffine, ACE 2005 (c) Biaffine, GENIA

(d) DSpERT, ACE 2004 (e) DSpERT, ACE 2005 (f) DSpERT, GENIA

Figure 2: Precision, recall and $F_1$ scores on within-entity spans by different sampling rates. All the results are average scores of 10 independent runs; the error bars represent the corresponding standard deviations.

the precision-recall rebalance may not necessarily leads to a higher overall $F_1$ score. Note that (1) $\mathcal{I}$ contains much less ground-truth entities than $\mathcal{O}$, and (2) the predicted entities in $\mathcal{I}$ are always much less precise than those in $\mathcal{O}$. Hence, a relatively high recall in $\mathcal{I}$ has very limited contribution to the overall recall, but yields many false positive samples and thus results in a large drop in the overall precision. On the contrary, high-precision low-recall predicted entities in $\mathcal{I}$ would be a safe and preferred choice. This also explains why *Full Negative* can sometimes achieve high overall $F_1$ scores (e.g., DSpERT on GENIA, Table 1).

Appendices B and C provide category-specific results and span representation visualizations, respectively.

## 6 Results on CoNLL 2003 NFF

**Case Study.** We start from a case study to intuitively demonstrate the results of nested-from-flat NER on the well-known CoNLL 2003 dataset. Specifically, we train DSpERT with *Sampling* ($\gamma = 0.01$) on the training split, and use the trained model to predict entities on the test split. As aforementioned, we have also annotated the nested entities in the test split.

Table 2 shows 10 example test sentences, marked with the ground-truth and predicted entities. There exist some successful cases that nested entities are correctly recognized. For example, in Sentences 1–4, "U.S.", "Singapore", "Melbourne" and "Zimbabwe" are correctly predicted as `LOC` entities, each within another `ORG`, `LOC` or `MISC` entity; in Sentences 5 and 6, "Albanian" and "Asian" are correctly recognized as `MISC` entities, each within another `ORG` or `MISC` entity.

The incorrect recognition results contain the following typical scenarios:

- The first or last name within a full person name, as a false positive `PER` entity (e.g., Sentences 2, 3, 7).
- A geopolitical concept within a specific location name, as a false positive `LOC` or `MISC` entity (e.g., Sentence 4, 6).
- The anchor word of an event/organization name, as a false positive `MISC`/`ORG` entity (e.g., Sentences 4, 6, 10).

| |
|---|
| 1. [Mills]$_{\text{PER}}$ is the 38th person to die in [Florida]$_{\text{LOC}}$ 's electric chair since the [[U.S.]$_{\text{LOC}}$ Supreme Court]$_{\text{ORG}}$ reversed itself in 1976 and legalised the death penalty . |
| 2. There is the international prestige [Singapore]$_{\text{LOC}}$ would enjoy , but " more importantly there is a genuine national interest in fostering better global free trade and an open market " , said [[Tan]$_{\text{PER}}$ Kong Yam]$_{\text{PER}}$ , head of Business Policy at the [National University of [Singapore]$_{\text{LOC}}$]$_{\text{ORG}}$ . |
| 3. [West Indies]$_{\text{LOC}}$ were 53 for two in 15 overs when rain stopped play at the [[Melbourne]$_{\text{LOC}}$ Cricket Ground]$_{\text{LOC}}$ after captain [Courtney [Walsh]$_{\text{PER}}$]$_{\text{PER}}$ won the toss and elected to bat . |
| 4. [[Zimbabwe]$_{\text{LOC}}$ [Open]$_{\text{MISC}}$]$_{\text{MISC}}$ on Saturday ( [South [African]$_{\text{MISC}}$]$_{\text{MISC}}$ unless stated ) |
| 5. [FIFA]$_{\text{ORG}}$ had banned [Albania]$_{\text{LOC}}$ indefinitely after its sports ministry had ordered the suspension of [[Albanian]$_{\text{MISC}}$ Football Association]$_{\text{ORG}}$ general secretary [Eduard Dervishi]$_{\text{PER}}$ and dissolved the executive committee . |
| 6. [South [Korea]$_{\text{LOC}}$]$_{\text{LOC}}$ made virtually certain of an [[Asian]$_{\text{MISC}}$ [Cup]$_{\text{MISC}}$]$_{\text{MISC}}$ quarter-final spot with a 4-2 win over [Indonesia]$_{\text{LOC}}$ in a Group A match on Saturday . |
| 7. [Dutch]$_{\text{MISC}}$ forward [Reggie [Blinker]$_{\text{PER}}$]$_{\text{PER}}$ had his indefinite suspension lifted by [FIFA]$_{\text{ORG}}$ on Friday and was set to make his [[[Sheffield]$_{\text{LOC}}$]$_{\text{ORG}}$ Wednesday]$_{\text{ORG}}$ comeback against [Liverpool]$_{\text{ORG}}$ on Saturday . |
| 8. [[Bayer]$_{\text{ORG}}$ [Leverkusen]$_{\text{LOC}}$]$_{\text{ORG}}$ ( [Germany]$_{\text{LOC}}$ ) |
| 9. Corrects headline from [NBA]$_{\text{ORG}}$ to [NHL]$_{\text{ORG}}$ and corrects team name in second result from [[La]$_{\text{LOC}}$ Clippers]$_{\text{ORG}}$ to [[[Ny]$_{\text{LOC}}$]$_{\text{ORG}}$ Islanders]$_{\text{ORG}}$ . |
| 10. [Philadelphia]$_{\text{LOC}}$ , which fell from an [[NFC]$_{\text{MISC}}$ [East]$_{\text{MISC}}$]$_{\text{MISC}}$ tie with the [[[Dallas]$_{\text{LOC}}$]$_{\text{ORG}}$ [Cowboys]$_{\text{ORG}}$]$_{\text{ORG}}$ and [[[Washington]$_{\text{LOC}}$]$_{\text{ORG}}$ [Redskins]$_{\text{ORG}}$]$_{\text{ORG}}$ , go on the road against the [[New York]$_{\text{LOC}}$ [Jets]$_{\text{ORG}}$]$_{\text{ORG}}$ and then entertain [[Arizona]$_{\text{ORG}}$]$_{\text{LOC}}$ . |

Table 2: Example sentences with ground-truth and predicted entities from the test split of CoNLL 2003 NFF. The green entities are true positive samples, the red ones are false negative, and the orange ones are false positive.

- A LOC entity mislabeled as ORG within an organization name (e.g., Sentences 7, 9, 10).

- Nested entities that rarely appear independently at the topmost level in the corpus (e.g., Sentences 8, 10), especially for abbreviations (e.g., Sentence 9).

Most scenarios are largely attributable to the *annotation inconsistency*, i.e., the inconsistency of the annotation standards between the nested and flat entities. For example, (1) if an entity mention is nested within its full name in text, nested NER annotation guidelines (e.g., ACE; Doddington et al., 2004) typically label the full name only, but ignore the substring mention. This avoids redundancy, since the two mentions refer to a same entity concept. However, the same substring should be annotated if it appears at the topmost level in text. (2) CoNLL 2003 contains a large amount of sports news, where city/country names are ubiquitously used to refer to team names; such mentions should be annotated as ORG entities according to the guidelines (Tjong Kim Sang and De Meulder, 2003). However, the same city/country mentions should be annotated as LOC entities if they appear within the full team names. Given such inconsistencies, a model trained by flat supervision plausibly learns patterns inapplicable to the nested entities. This results in redundant or mislabeled entities, although some of them might be acceptable in practice.

Some scenarios are associated with the *data inconsistency*, i.e., the inconsistency of the data distributions between the within-entity and out-of-entity spans. Some nested entity mentions almost never appear independently at the topmost level in text. For example, some location abbreviations (e.g., "NY" or "LA") are always nested within other entities in the corpus. This poses a very challenging case for the nested-from-flat NER task, due to the lack of supervision. Actually, a nested-from-flat NER model may never succeed in that case unless sufficient external knowledge is introduced and utilized, such as knowledge databases (Wang et al., 2021; Geng et al., 2022) or more powerful PLMs.

**Post Processing.** Based on the above analysis, we propose two post-processing operations on the predicted entity set:

- If a PER entity is nested within another PER entity, remove the nested one; because it is probably a first/last name inside the full name.

- If an ORG entity is nested within another entity, change the entity label to LOC; because it is probably a location name used to refer to team names in other context in the corpus.

| | Within | Out | Overall |
|---|---|---|---|
| Full Negative | $14.6_{\pm 0.2}$ | $\mathbf{93.4}_{\pm 0.1}$ | $\mathbf{89.8}_{\pm 0.1}$ |
| w/ Post Processing | $14.6_{\pm 0.2}$ | – | $\mathbf{89.7}_{\pm 0.1}$ |
| Sampling | $16.6_{\pm 1.1}$ | $\mathbf{93.5}_{\pm 0.2}$ | $84.3_{\pm 0.6}$ |
| w/ Post Processing | $\mathbf{33.2}_{\pm 1.0}$ | – | $88.6_{\pm 0.3}$ |
| Full Ignoring | $8.7_{\pm 0.4}$ | $92.9_{\pm 0.4}$ | $70.3_{\pm 0.9}$ |
| w/ Post Processing | $31.8_{\pm 1.2}$ | – | $82.4_{\pm 0.8}$ |

Table 3: Results of nested-from-flat experiments by DSpERT on CoNLL 2003 NFF. Reported are average $F_1$ scores with corresponding standard deviations of 10 independent runs. The best $F_1$ scores are in bold.

Table 3 lists the evaluation results of DSpERT on CoNLL 2003 NFF. With the help of post processing, DSpERT+*Sampling* achieves the best within-entity $F_1$ score of 33.3%. This score seems low, relative to those on ACE 2004/2005 (i.e., 54%+). The main reason is that CoNLL 2003 specifies that named entities should be unique identifiers like proper names or acronyms (Tjong Kim Sang and De Meulder, 2003), while ACE additionally includes pronouns or descriptions that refer to entities (Doddington et al., 2004). Note that the pronouns and descriptions can be labeled more consistently between the within-entity and out-of-entity spans, which lowers the difficulty of nested-from-flat NER on ACE 2004/2005. In other words, CoNLL 2003 NFF poses a more strict and challenging benchmark of nested-from-flat NER.

Similar to the results on other datasets, our model is trained to recognize nested entities without affecting the out-of-entity performance. Hence, compared to a model that predicts flat entities only, our method always has merit for the additional ability of nested entity recognition.

## 7 Discussion and Conclusion

Although the NLP community has undertaken increasing efforts to investigate and develop nested NER models, many existing NER resources are flatly designed and annotated, especially in languages other than English. For example, the widely-used Chinese NER benchmarks, e.g., OntoNotes 4, MSRA (Levow, 2006), Weibo NER (Peng and Dredze, 2015) and Resume NER (Zhang and Yang, 2018), are all flat; similar situation holds for Japanese (Iwakura et al., 2016), Korean (Jeong et al., 2020), Vietnamese (Truong et al., 2021), etc. Most domain-specific entity recognition datasets are also designed in a flat scheme (Uzuner et al., 2011; Albright et al., 2013; Jeong et al., 2020).

Nested-from-flat NER corresponds to a realistic application scenario: given training data annotated with flat entities only, one may still desire the trained model capable of recognizing nested entities. This task is theoretically feasible, because the recognizable patterns for outermost entities should be, at least partially, transferable to nested entities. To the best of our knowledge, this study is the first to validate and investigate this mechanism.

On the other hand, nested-from-flat NER is a challenging setting because of the data and annotation inconsistencies between the within-entity and out-of-entity spans. Hence, the models may learn inapplicable or insufficient patterns when transferred to recognizing nested entities.

We choose the span-based NER framework because it explicitly distinguishes between positive and negative spans, which allows us to flexibly manipulate the negative samples in the within-entity area. Since the within-entity spans probably contain unlabeled nested entities, it is straightforward to ignore these spans in loss computation; while we empirically find it beneficial to apply negative sampling with a very small rate (i.e., 0.01). The negative sampling is inspired by Li et al. (2021)'s solution for unlabeled entity problem, but their optimal sampling rate is much larger (0.3 – 0.4).

In conclusion, this study proposes nested-from-flat NER, a new subtask that asks to train a nested NER model with flatly annotated data. We find a simple but effective solution to this task. With nested entities removed from the training data, our model can achieve 54.8%, 54.2% and 41.1% within-entity $F_1$ scores on ACE 2004, ACE 2005 and GENIA, respectively. Moreover, the model's performance on flat entity recognition is completely unaffected by its additional ability to recognize nested entities. We further propose a nested-from-flat NER benchmark, CoNLL 2003 NFF, which consists of CoNLL 2003 and our annotations of nested entities in the test set. With in-depth case study, we find that the main challenges stem from the data and annotation inconsistencies between the flat and nested entities.

## 8 Limitations

We acknowledge that our modeling techniques *per se* are simple, and the issue of data and annotation inconsistencies between the within-entity and out-of-entity spans has not be fully addressed in this

study. In particular, the post processing used for CoNLL 2003 NFF is rule-based, and thus inapplicable to other corpora. However, simple approaches are straightforward, which allow us to clearly verify the feasibility of nested-from-flat NER and ensure the reproducibility of our results. The nested-from-flat NER performance can be promisingly improved by utilizing more external knowledge, either explicitly via knowledge databases (Wang et al., 2021; Geng et al., 2022) or implicitly with more powerful PLMs.

In addition, the negative sampling rate $\gamma$ significantly affects the performance, while its optimal value differs across datasets. Hence, one has to tune this hyperparameter when she applies our approach to a new dataset.

## References


Daniel Albright, Arrick Lanfranchi, Anwen Fredriksen, William F Styler IV, Colin Warner, Jena D Hwang, Jinho D Choi, Dmitriy Dligach, Rodney D Nielsen, James Martin, et al. 2013. Towards comprehensive syntactic and semantic annotations of the clinical narrative. *Journal of the American Medical Informatics Association*, 20(5):922–930.

Nigel Collier and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 73–78.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(ARTICLE):2493–2537.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Markus Eberts and Adrian Ulges. 2020. Span-based joint entity and relation extraction with Transformer pre-training. In *Proceedings of the 24th European Conference on Artificial Intelligence*, Santiago de Compostela, Spain.

Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Singapore. Association for Computational Linguistics.

Zhichao Geng, Hang Yan, Zhangyue Yin, Chenxin An, Xipeng Qiu, and Xuanjing Huang. 2022. TURNER: The uncertainty-based retrieval framework for Chinese NER. *arXiv preprint arXiv:2202.09022*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Tomoya Iwakura, Kanako Komiya, and Ryuichi Tachibana. 2016. Constructing a Japanese basic named entity corpus of various genres. In *Proceedings of the Sixth Named Entity Workshop*, pages 41–46, Berlin, Germany. Association for Computational Linguistics.

Dong-Ho Jeong, Min-Kang Heo, Hyung-Chul Kim, and Sang-Won Park. 2020. Constructing a Korean named entity recognition dataset for the financial domain using active learning. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 208–212, Indian Institute of Technology Patna, Patna, India. NLP Association of India (NLPAI).

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459, New Orleans, Louisiana. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus – a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25.

John D Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 8th International Conference on Machine Learning*, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

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, Sydney, Australia. Association for Computational Linguistics.

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022. Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Yangming Li, Shuming Shi, et al. 2021. Empirical analysis of unlabeled entity problem in named entity recognition. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867, Lisbon, Portugal. Association for Computational Linguistics.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318. PMLR.

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, Lisbon, Portugal. Association for Computational Linguistics.

Nicky Ringland, Xiang Dai, Ben Hachey, Sarvnaz Karimi, Cecile Paris, and James R. Curran. 2019. NNE: A dataset for nested named entity recognition in English newswire. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5176–5181, Florence, Italy. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Yongliang Shen, Xiaobin Wang, Zeqi Tan, Guangwei Xu, Pengjun Xie, Fei Huang, Weiming Lu, and Yueting Zhuang. 2022. Parallel instance query network for named entity recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 947–961, Dublin, Ireland. Association for Computational Linguistics.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Zeqi Tan, Yongliang Shen, Shuai Zhang, Weiming Lu, and Yueting Zhuang. 2021. A sequence-to-set network for nested named entity recognition. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 3936–3942.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Thinh Hung Truong, Mai Hoang Dao, and Dat Quoc Nguyen. 2021. COVID-19 named entity recognition for Vietnamese. In *Proceedings of the 2021*

*Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2146–2153, Online. Association for Computational Linguistics.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11).

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 204–214, Brussels, Belgium. Association for Computational Linguistics.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2021. Improving named entity recognition by external context retrieving and cooperative learning. 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 1800–1812, Online. Association for Computational Linguistics.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5154–5163, Hong Kong, China. Association for Computational Linguistics.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various NER subtasks. 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 5808–5822, Online. Association for Computational Linguistics.

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.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564, Melbourne, Australia. Association for Computational Linguistics.

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, Dublin, Ireland. Association for Computational Linguistics.

Enwei Zhu, Yiyang Liu, and Jinpeng Li. 2022. Deep span representations for named entity recognition. *arXiv preprint arXiv:2210.04182*.

## A Datasets

**ACE 2004 and ACE 2005** are two English nested NER datasets created by the Automatic Content Extraction (ACE) Program (Doddington et al., 2004). The corpus consists of broadcast transcripts, newswire and newspaper data; the entity types include Person (PER), Organization (ORG), Facility (FAC), Location (LOC), Geo-political Entity (GPE), Vehicle (VEH), and Weapon (WEA). Our data processing and splits follow Lu and Roth (2015).

As indicated by the annotation guidelines, ACE aims to recognize all mentions of entities, not just names. In other words, an entity mention can be a name, a description, or a pronoun, as long as it clearly refers to the entity.

**GENIA** is a nested NER dataset on English biological articles (Kim et al., 2003). There are five entity categories, i.e., DNA, RNA, Protein, Cell Line, and Cell Type. Our data processing follows Lu and Roth (2015), and data splits follow Yan et al. (2021) and Li et al. (2022).

**CoNLL 2003 NFF** is a nested-from-flat NER benchmark that consists of the text data and flat NER annotations of CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003), and our annotations of nested entities in the test split. The corpus consists of Reuters news stories in 1996 and 1997; the entity categories are Person (PER), Organization (ORG), Location (LOC), and Miscellaneous (MISC). We use the original data splits for experiments.

In CoNLL 2003, named entities are limited to unique identifiers, such as proper names and acronyms. It excludes pronouns and descriptions that refer to entities.

We hired three NLP experts to additionally annotate the nested entities in the test split. BRAT Rapid Annotation Tool (Stenetorp et al., 2012) was deployed to provide the annotation user interface. The annotators were asked to carefully read through the CoNLL 2003 Annotation Guidelines, and strictly follow the guidelines when manually labeling the nested entities that had been ignored in the original data. All the original annotations, even a few incorrect ones (Wang et al., 2019), are retained without modification. Each annotator labeled all documents in the test set, and the final results are based on an additional round of manual validation that resolves the inter-annotator disagreements.

Table 4 presents the descriptive statistics of the datasets.

## B Category-Specific Results

Table 5 lists the categorical results of within-entity $F_1$ scores on ACE 2004, ACE 2005 and GENIA.

The performance significantly varies across categories. Specifically, the categorical $F_1$ scores range from 38% to 60% on ACE 2004/2005, from 18% to 51% on GENIA. In general, the model performs better on categories that contain more entities, such as PER and GPE in ACE 2004/2005, and the Protein type in GENIA. This is reasonable, since such categories have more positive samples in the training data, which enable the model to learn more accurate decision boundaries.

The categorical $F_1$ scores by nested-from-flat models are consistently lower than, and positively correlated with the corresponding scores by gold supervision. The Pearson correlation coefficients between the categorical $F_1$ scores are positive for all the three datasets. One exception is the Cell Type category in GENIA, where the nested-from-flat model surprisingly outperforms its counterpart with gold supervision; we conjecture that some Cell Type entities are incorrectly annotated in the training data and thus misguide the trained model.

## C Visualization of Span Representations

Figure 3 presents the t-SNE visualizations (Van der Maaten and Hinton, 2008) of the pre-logit span representations. The representations are constructed by DSpERT on the test sentences of ACE 2004.

For the model trained by flat supervision, the within-entity span representations are largely clustered by categories, but a part of negative samples are mixed into the positive clusters, resulting in unclear and ambiguous decision boundaries (Figure 3a). In contrast, the out-of-entity span representations form clear and tight categorical clusters (Figure 3b). However, if the model is trained on data with nested annotations, both the within-entity and out-of-entity representations are clearly clustered by categories (Figures 3c, 3d).

Hence, the spans mixed across positive and negative clusters in Figure 3a lack supervision. As suggested by the case study on CoNLL 2003 NFF, these span samples probably correspond to the data and annotation inconsistencies between the within-entity and out-of-entity spans. They are particularly difficult to discriminate in the nested-from-flat setting.

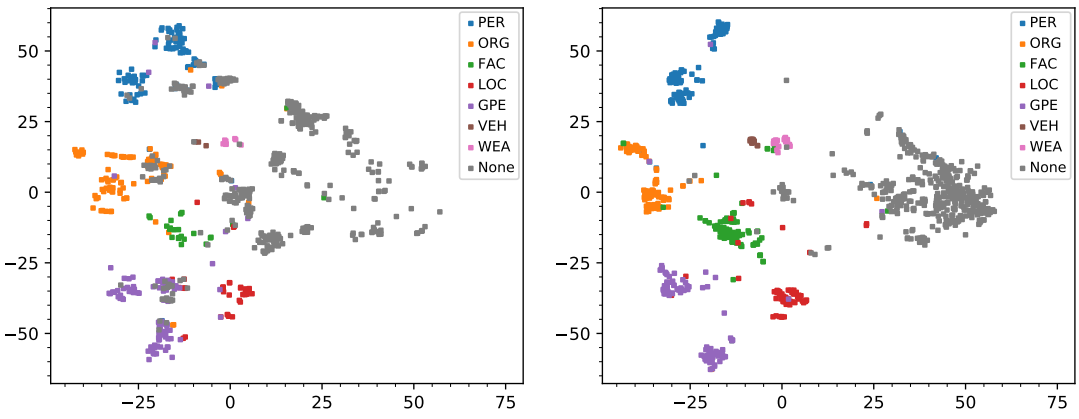
| | ACE 2004 | | | ACE 2005 | | | GENIA | | | CoNLL 2003 NFF | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Train | Dev. | Test | Train | Dev. | Test | Train | Dev. | Test | Train | Dev. | Test |
| #Sentence | 6,799 | 829 | 879 | 7,336 | 958 | 1,047 | 15,023 | 1,669 | 1,854 | 14,987 | 3,466 | 3,684 |
| Nested (%) | 39.5 | 35.3 | 42.4 | 36.6 | 35.6 | 31.5 | 21.3 | 19.5 | 24.1 | – | – | 11.3 |
| #Entity | 22,207 | 2,511 | 3,031 | 24,687 | 3,217 | 3,027 | 46,164 | 4,371 | 5,511 | 23,499 | 5,942 | 5,648 |
| Nested (%) | 28.2 | 27.2 | 29.1 | 24.4 | 22.2 | 23.8 | 9.5 | 9.6 | 11.4 | – | – | 7.9 |
| Ave. Len. | 2.5 | 2.6 | 2.5 | 2.3 | 2.1 | 2.3 | 1.9 | 2.1 | 2.1 | 1.4 | 1.4 | 1.4 |
| Max. Len. | 57 | 35 | 43 | 49 | 30 | 27 | 17 | 18 | 15 | 10 | 10 | 6 |

Table 4: Descriptive statistics of datasets. "#Sentence" denotes the number of sentences, under which "Nested (%)" denotes the proportion of sentences with nested entities. "#Entity" denotes the number of entities, under which "Nested (%)" denotes the proportion of nested entities, "Ave. Len." and "Max. Len." denote the average and maximum lengths of entities, respectively.

| | ACE 2004 | | ACE 2005 | |
|---|---|---|---|---|
| | Sampling | *Gold S.* | Sampling | *Gold S.* |
| PER | $55.7_{\pm 0.8}$ | $90.0_{\pm 0.3}$ | $57.9_{\pm 0.8}$ | $89.6_{\pm 0.4}$ |
| ORG | $48.9_{\pm 1.6}$ | $81.8_{\pm 0.5}$ | $49.5_{\pm 1.6}$ | $80.7_{\pm 1.2}$ |
| FAC | $51.0_{\pm 3.3}$ | $82.4_{\pm 1.9}$ | $52.9_{\pm 4.5}$ | $78.8_{\pm 2.1}$ |
| LOC | $43.7_{\pm 3.6}$ | $77.4_{\pm 1.5}$ | $37.8_{\pm 3.2}$ | $75.0_{\pm 3.5}$ |
| GPE | $59.4_{\pm 2.6}$ | $88.6_{\pm 0.5}$ | $57.9_{\pm 4.3}$ | $88.4_{\pm 1.0}$ |
| VEH | $37.9_{\pm 7.3}$ | $85.7_{\pm 0.0}$ | $46.7_{\pm 2.0}$ | $81.4_{\pm 3.2}$ |
| WEA | $55.8_{\pm 8.2}$ | $75.2_{\pm 2.4}$ | $38.4_{\pm 1.7}$ | $69.6_{\pm 3.0}$ |
| Correlation | 0.203 | | 0.909 | |

| | GENIA | |
|---|---|---|
| | Sampling | *Gold S.* |
| DNA | $22.9_{\pm 1.4}$ | $33.0_{\pm 1.4}$ |
| RNA | – | – |
| Protein | $51.0_{\pm 0.7}$ | $65.3_{\pm 0.8}$ |
| Cell Line | $17.9_{\pm 2.5}$ | $28.3_{\pm 2.3}$ |
| Cell Type | $28.1_{\pm 1.2}$ | $11.6_{\pm 1.3}$ |
| Correlation | 0.788 | |

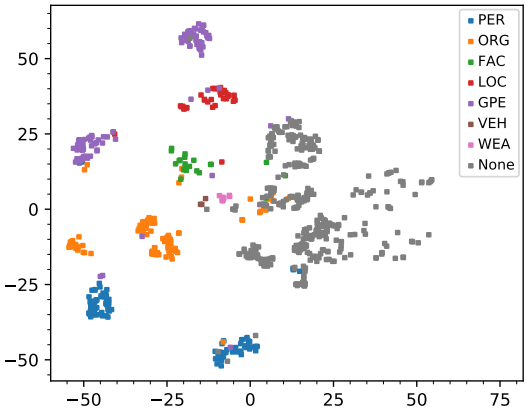
Table 5: Categorical within-entity $F_1$ scores by DSpERT on nested NER datasets. Reported are average $F_1$ scores with corresponding standard deviations of 10 independent runs. "*Gold S.*" indicates results with gold supervision. "–" means no ground-truth nested entities in the test set.



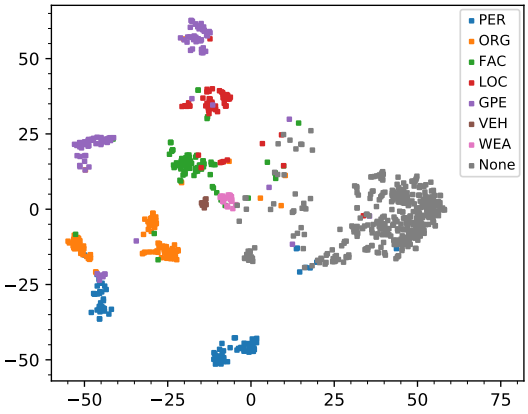

(a) Sampling, Within-entity




(b) Sampling, Out-of-entity




(c) *Gold Superv.*, Within-entity




(d) *Gold Superv.*, Out-of-entity

Figure 3: t-SNE visualization of pre-logit span representations by DSpERT on ACE 2004 test sentences. Each row compares the within-entity and out-of-entity span representations from a same model, visualized by a shared t-SNE.