Span-based Unified Named Entity Recognition Framework via Contrastive Learning

Hongli Mao[1], **Xian-Ling Mao**[1*], **Hanlin Tang**[1], **Yu-Ming Shang**[2], **Xiaoyan Gao**[3], **Ao-Jie Ma**[1] and **Heyan Huang**[1]
[1] School of Computer Science & Technology, Beijing Institute of Technology, Beijing, China
[2] Beijing University of Posts and Telecommunications, Beijing, China
[3] Beijing University of Technology, Beijing, China
maohongli.bit@gmail.com, {maoxl, hltang, maaojie, hhy63}@bit.edu.cn, shangym@bupt.edu.cn, gaoxiaoyan@bjut.edu.cn

Abstract

Traditional Named Entity Recognition (NER) models are typically designed for domain-specific datasets and limited to fixed predefined types, resulting in difficulty generalizing to new domains. Recently, prompt-based generative methods attempt to mitigate this constraint by training models jointly on diverse datasets and extract specified entities via prompt instructions. However, due to autoregressive structure, these methods cannot directly model entity span and suffer from slow sequential decoding. To address these issues, we propose a novel **S**pan-based **U**nified **NER** framework via contrastive learning (**SUNER**), which aligns text span and entity type representations in a shared semantic space to extract entities in parallel. Specifically, we first extract mention spans without considering entity types to better generalize across datasets. Then, by leveraging the power of contrastive learning and well-designed entity marker structure, we map candidate spans and their textual type descriptions into the same vector representation space to differentiate entities across domains. Extensive experiments on both supervised and zero/few-shot settings demonstrate that proposed SUNER model achieves better performance and higher efficiency than previous state-of-the-art unified NER models.

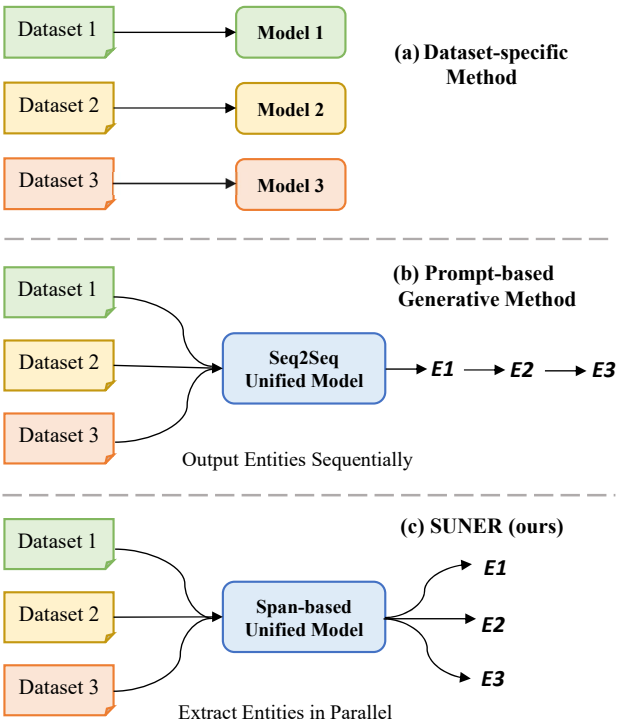

Figure 1: Illustration of 3 NER approaches. (a) Dataset-specific models train individually per corpus. (b) Prompt-based generative models enable joint training but decode entities sequentially. (c) Our unified SUNER method directly models spans for parallel extraction across datasets.

1 Introduction

Named Entity Recognition (NER) is a foundational task for Natural Language Processing (NLP), which aims to extract named entities in the given text and classify them into predefined entity types such as persons, organizations and locations. As a subtask of information extraction, NER serves as a crucial building block in many NLP applications, including entity linking [Ganea and Hofmann, 2017; Le and Titov, 2018], relation extraction [Zhong and Chen, 2021; Rathore *et al.*, 2022] and knowledge graph construction [Sarhan and Spruit, 2021].

The current approaches for NER typically employ pre-trained transformer architectures, like BERT [Devlin *et al.*, 2019] or RoBERTa [Liu *et al.*, 2019], as a base encoding component to perform in-domain supervised learning on individual datasets [Yu *et al.*, 2020; Yan *et al.*, 2023; Zhang *et al.*, 2023], as shown in Figure 1(a). Although they demonstrate strong performance on each dataset, these dataset-specific methods struggle to generalize across new domains or transfer to unfamiliar entity types without costly full retraining. For instance, a NER model trained to identify person and location entities in news text struggles to iden-

*Corresponding author.

tify disease names in medical documents. To enable learned knowledge transfer across domains, recent prompt-based generative methods propose jointly training across diverse datasets with a unified model [Lu *et al.*, 2022; Lu *et al.*, 2023; Wang *et al.*, 2023], as shown in Figure 1(b). To this end, they take prompt instructions and text as input, leveraging a sequence-to-sequence (seq2seq) generative framework to sequentially output entities in an autoregressive manner.

However, due to inherent limitations of autoregressive language models for entity extraction tasks, these prompt-based generative approaches have two major weaknesses. First, compared with span-based methods directly encoding entities at span-level, the sequential entity generation of approaches can only implicitly captures associations between spans and labels, as well as inter-span relationships, making it difficult to accurately predict span boundaries. Second, due to the token-by-token decoding in seq2seq frameworks, these models cannot generate multiple entity predictions in parallel and thus suffer from slower inference speed.

To address the above-mentioned challenges, we draw inspiration from CLIP [Radford *et al.*, 2021] utilizing contrastive learning to enhance model generalizability, and propose a novel **S**pan-based **U**nified **NER** approach via contrastive learning (**SUNER**), which can jointly handle multiple NER datasets, as shown in Figure 1(c). The key idea of our method is to directly align text spans and entity type representations within a shared semantic space, enabling parallel extraction across domains. In this manner, we break through limitations of entity categories by introducing textual descriptions of entity types. Furthermore, by utilizing pretrained language models that have learned general linguistic patterns from large unlabeled corpora, our model achieves enhanced semantic matching and superior generalization capabilities to new domains.

Specifically, the proposed SUNER comprises two modules: span detection and span classification. First, to adapt variations across different domains, detection module focuses solely on extracting mention spans of entities without considering entity types. This approach, separating the identification process from later domain-specific categorization, is more generalized compared to directly classifying entities across diverse datasets. Second, detection module introduces an ingenious entity marker framework that highlight candidate entity spans in the input sentence, effectively capturing entity relationships while retaining the original textual structure to achieve better span representations. Following this, leveraging a BERT-based contrastive learning approach, the module maps candidate spans and textual type descriptions into the same semantic space. This allows flexible extraction per specified category by relying on learned semantic similarities. In experiments, we jointly train SUNER on seven different datasets, supporting a total of 47 entity types. Extensive assessments under supervised conditions as well as zero/few-shot settings validate the effectiveness of our proposed model in achieving better performance and higher efficiency than previous state-of-the-art (SOTA) unified NER models.

In summary, our main contributions are as follows:

- Instead of relying on autoregressive generated models, we introduce a span-based unified NER approach via contrastive learning that can extract entities across domains based on semantic similarities.
- To produce enhanced span representations, we propose a well-designed entity marker that highlights candidate spans, capturing inter-entity relationships while retaining original textual structure.
- In experiments across various settings, our model achieves a 1.9% average improvement under supervised conditions and a 5.9%-6.4% absolute increase for zero/few-shot learning over previous SOTA unified models, while also maintaining higher efficiency.

2 Related Work

2.1 Named Entity Recognition

Dataset-Specific NER Transformer-based pretrained language models [Devlin *et al.*, 2019; Liu *et al.*, 2019; Lee *et al.*, 2020] have made significant achievements in current NER tasks with their strong representational capabilities. For flat NER, classic sequence labeling methods [Souza *et al.*, 2019; Li *et al.*, 2020] combine BERT with Conditional Random Fields (CRF) [Ma and Hovy, 2016] to assign the BIO tag for each token. In a different approach, span-based methods directly model candidate spans and employ BERT to generate span representations for classification, achieving the best performance for both flat and nested NER task [Yu *et al.*, 2020; Li *et al.*, 2022; Zhu and Li, 2022; Yan *et al.*, 2023; Zhang *et al.*, 2023]. However, these methods typically perform in-domain supervised learning on specific datasets and are limited to a set of fixed predefined entity categories. This leads to challenges in adapting to unfamiliar entity types without extensive retraining.

Unified NER To enable cross-domain knowledge transfer, recent prompt-based generative methods propose jointly training on diverse NER datasets and integrating label information into a unified model. Among these, T5-based approaches [Raffel *et al.*, 2020] like PUnifiedNER [Lu *et al.*, 2023] and UIE [Lu *et al.*, 2022] employ a unified structural model using a seq2seq generative framework to sequentially output specified entity types. Meanwhile, InstructUIE [Wang *et al.*, 2023] and UniNER [Zhou *et al.*, 2023] based on Large Language Models (LLM) converts various NER datasets into instruction-following format, subsequently fine-tuning encoder-decoder generative models. However, these generative NER methods struggle with precise entity boundary detection and slow decoding, especially for LLM-based models, whose size and cost limit their use in resource-limited settings. Although USM [Lou *et al.*, 2023] proposes a token-based unified method, it cannot deal with overlapped datasets, and also suffer from inefficient decoding due to semantic linking for each word. In our setting, to improve generalization and time efficiency, we apply a contrastive learning framework for semantic matching between spans and textual type descriptions, enabling parallel entity extraction.

2.2 Contrastive Learning

Self-supervised contrastive learning has been widely utilized in diverse tasks to generate representations [Chuang *et al.*,

2020; Gao *et al.*, 2021; Radford *et al.*, 2021; Han *et al.*, 2022; Tan *et al.*, 2022]. The core concept of contrastive learning aims to pull positive pairs closer while pushing negative pairs apart. SimCSE [Gao *et al.*, 2021] applies twice dropout in the forward process to refine the better sentence representation. The CLIP [Radford *et al.*, 2021] employs a contrastive learning framework for pretraining on semantic matching between image and caption text. This approach aims to transcend the limitations of fixed object categories and improve model generalizability to new categories. In this paper, inspired by the CLIP model, we propose a span-based contrastive learning unified NER approach. Different from some contrastive NER [Das *et al.*, 2022; Huang *et al.*, 2022] methods that focus on token-level contrasts, our approach aligns spans with textual entity types in a shared semantic space. By learning semantic similarities between spans and entity types, our model can better generalize to unseen entity categories.

3 The Proposed Method

The architecture of our proposed SUNER model is shown in Figure 2. It consists of two main modules. First, the span detection module focuses on identifying all mention spans irrespective of entity type. It leverages a BERT-based biaffine model to generate contextual span representations and then applies a sigmoid layer to extract candidate entity spans. Second, the span classification module inserts special marks to highlight these extracted spans in the input text . This marked text, along with entity type descriptions, is then fed into a BERT-based contrastive framework to obtain vector representations of the spans and type texts separately. By maximizing similarity between the span and type text vectors, our model classifies each span into an entity type. In the following, we will first provide a task definition, then present components of our method in detail.

3.1 Task Formulation

The input for our SUNER model includes a sentence $S = [w_1, w_2, \ldots, w_n]$ and entity types T (expressed in natural language), where n denotes the total number of tokens in the sentence. The goal of SUNER is to extract all entities $E = \{(l_i, r_i, t_i)\}_{i=0}^{e_n}$ based on semantic similarity, where e_n is the number of entities and l_i, r_i, t_i represent the left and right boundary indices and type of the i-th entity.

3.2 Span Detection

As discussed in the introduction, we aim to train the model on multiple datasets and learn shared features for domain generalization. To this end, we initially extract entity spans without specifying entity types, a more generalized approach than directly classifying entities across various domains.

Span Representation Encoder

To achieve contextualized span representation for detection, we first augment the input S with the dataset name D during supervised training to harmonize annotations discrepancies, while omitting this field during zero-shot evaluation. Then the augmented $S' = [w_{\text{cls}}, D, w_{\text{sep}}, w_1, w_2, \ldots, w_n]$ is fed into BERT to get token-level representation $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$, where the special tokens w_{cls}, w_{sep} correspond to the start and separator tokens in BERT.

$$\mathbf{H} = \text{BERT}_{\text{d}}(S') \tag{1}$$

Next, we follow [Yu *et al.*, 2020] and employ a biaffine encoder to generate span representation. Given a sentence S with token embeddings $\mathbf{H}$, we process $\mathbf{H}$ through two separate feed-forward networks (FFNs) to obtain the start-of-span representations $\mathbf{H}^s \in \mathbb{R}^{n \times d_h}$ and end-of-span representations $\mathbf{H}^e \in \mathbb{R}^{n \times d_h}$, then the span representation matrix $\mathbf{R}$ is calculated as follows:

$$\begin{aligned} \mathbf{H}^s &= \text{FFN}_{\text{s}}(\mathbf{H}) \\ \mathbf{H}^e &= \text{FFN}_{\text{e}}(\mathbf{H}) \\ \mathbf{R} &= (\mathbf{H}^s)^T \mathbf{U} \mathbf{H}^e + \mathbf{W}(\mathbf{H}^s \oplus \mathbf{H}^e) + \mathbf{b} \end{aligned} \tag{2}$$

where $\mathbf{U} \in \mathbb{R}^{d_h \times d_r \times d_h}$, $\mathbf{W} \in \mathbb{R}^{d_r \times 2d_h}$ and $\mathbf{b} \in \mathbb{R}^{d_r}$ are learnable parameters, d_h and d_r are the hidden size, $\oplus$ denotes element-wise concatenation, and $\mathbf{R} \in \mathbb{R}^{n \times n \times d_r}$. Each cell (i, j) in the matrix $\mathbf{R}$ represents the feature representation of span from the $i-$th token to the $j-$th token.

Utilizing the matrix parallel computation of the biaffine model, our approach can efficiently generate representations of all spans simultaneously.

Span Detection Objective

After getting spans representations from biaffine model, we obtain the span prediction logits $\mathbf{P}$ through a sigmoid layer as follows:

$$\mathbf{P} = \text{Sigmoid}(\mathbf{W}_p \mathbf{R} + \mathbf{b}_p) \tag{3}$$

And then, we use the binary cross entropy to calculate span-based loss as:

$$\mathcal{L}_{span} = -\sum_{0 \le i \le j < n} y_{ij} \log(\mathbf{P}_{ij}) \tag{4}$$

A common issue in pipeline systems is error propagation, where inaccurate mention boundaries will result in incorrect entity type classification. To enhance the precision of span detection in a more fine-grained level, we further introduce an auxiliary boundary detection task to generate high-quality candidate spans. Specifically, we feed the token representations $\mathbf{h}_i$ into two separate multi-layer perceptron (MLP) classifiers, and apply a sigmoid function to obtain the probabilities $\mathbf{P}_i^s$ and $\mathbf{P}_i^e$ of token w_i being the start or end of an entity span respectively:

$$\begin{aligned} \mathbf{P}_i^s &= \text{Sigmoid}(\text{MLP}_{start}(\mathbf{h}_i)) \\ \mathbf{P}_i^e &= \text{Sigmoid}(\text{MLP}_{end}(\mathbf{h}_i)) \end{aligned} \tag{5}$$

Then, the boundary-based loss can be calculated as:

$$\begin{aligned} \mathcal{L}_{bdr}^s &= -\sum_{0 \le i < n} y_i^s \log(\mathbf{P}_i^s) \\ \mathcal{L}_{bdr}^e &= -\sum_{0 \le i < n} y_i^e \log(\mathbf{P}_i^e) \\ \mathcal{L}_{bdr} &= \mathcal{L}_{bdr}^s + \mathcal{L}_{bdr}^e \end{aligned} \tag{6}$$

Finally, we combine span-based loss $\mathcal{L}_{span}$ with the auxiliary boundary-based loss $\mathcal{L}_{bdr}$ using a weight λ to produce the overall training objective for the span detection module:

$$\mathcal{L}_{dection} = (1 - \lambda)\mathcal{L}_{span} + \lambda \mathcal{L}_{bdr} \tag{7}$$

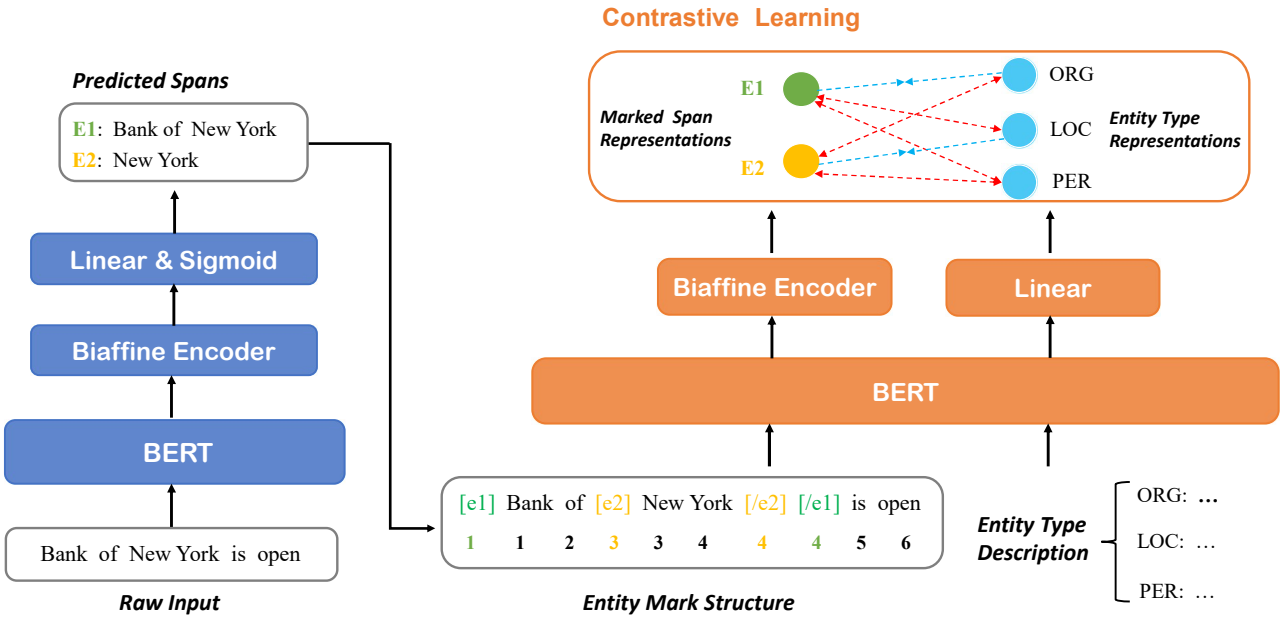

Figure 2: The overall architecture of our proposed model.

3.3 Span Classification

To further improve generalization across domains, the span classification module employs an entity marker structure for representing entity spans and utilizes contrastive semantic matching with textual type descriptions for effective entity categorization.

Entity Mark Structure

In order to effectively leverage boundary information for span classification, we do not directly employ the biaffine encoder representations. This method simply concatenates the start and end tokens, only capturing contextual information around each individual entity. It fails to capture inter-dependencies between the spans, which are crucial for entity type confirmation, especially for nested entities.

Instead of processing each span at token level, we insert specialized entity markers at the input layer to highlight spans in the original text. This marker representation allows us to capture dependencies between entities at full span level. Specifically, given an input sentence S and candidate entity spans $C = \{c_1, c_2, \ldots, c_m\}$, we insert text markers $[e_i]$ and $[/e_i]$ before and after each span c_i to construct a hierarchically marked sentence $\hat{S}$:

$$\begin{aligned}\hat{S} = \ldots, [e_1], w_{\text{start}(1)}, \ldots, w_{\text{end}(1)}, [/e_1], \\ \ldots, [e_m], w_{\text{start}(m)}, \ldots, w_{\text{end}(m)}, [/e_m], \ldots\end{aligned} \tag{8}$$

where $w_{\text{start}(i)}$ and $w_{\text{end}(i)}$ denote the start and end tokens of span c_i respectively.

In addition, to preserve original textual structure of the input for the pretrained language encoder, we tie the positional embeddings of the markers with the start and end tokens of the corresponding span. For instance, the marker $[e_k]$ and the start token $w_{start(k)}$, as well as the closing marker $[/e_k]$ and the end token $w_{end(k)}$ share the same position embedding. In this manner, we keep the position id of original text tokens unchanged. Furthermore, we impose a constrained attention mask to the attention layers. Text tokens only attend to text tokens and are isolated from entity marker tokens. Meanwhile, entity marker tokens can attend to all text tokens to aggregate information for their associated spans, as well as other marker tokens to capture inter-span dependencies.

Contrastive Learning Framework

Our contrastive learning framework comprises two modules: an entity type encoder and a span representation encoder. To conserve memory and enable more efficient training, both encoders utilize a shared BERT base model. In this task, we process two types of inputs: entity type descriptions and the modified sentence $\hat{S}$ with marked candidate entities.

Entity Type Encoder First, we leverage the entity type encoder to generate representations for each targeted entity. In this study, we describe each entity type with a sequence of natural language tokens. Specifically, for a given textual entity type E_k, we feed E_k into BERT and pass the resulting [CLS] representation through a MLP layer to obtain the projected type embedding $\mathbf{e}_k$:

$$\mathbf{h}^{E_k}_{[\text{CLS}]} = \text{BERT}_c(E_k) \tag{9}$$

$$\mathbf{e}_k = \text{MLP}_{ety}(\mathbf{h}^{E_k}_{[\text{CLS}]}) \tag{10}$$

By leveraging descriptive entity text rather than discrete labels, our model can fully utilize the robust representational power of pre-trained language models which are trained on vast unlabeled corpora. This approach significantly boosts the model's ability to generalize across new domains. Even for previously unseen entities, the use of explanatory type texts with BERT embeddings allows model to better understand and categorize these entities.

Span Representation Encoder Similar to the span representation encoder used in the span detection module, we feed the marked sentence $\hat{S}$ containing candidate entity spans into a BERT-based biaffine model to produce vector representations $\hat{\mathbf{r}}_i$ for each span c_i, formulated as:

$$\hat{\mathbf{H}} = \mathrm{BERT_c}(\hat{S}) \tag{11}$$

$$\hat{\mathbf{r}}_i = \mathrm{Biaffine}\left(\hat{\mathbf{h}}_{\widehat{\mathrm{start}}(\mathrm{i})}; \hat{\mathbf{h}}_{\widehat{\mathrm{end}}(\mathrm{i})}\right) \tag{12}$$

where $\widehat{\mathrm{start}}(\mathrm{i})$ and $\widehat{\mathrm{end}}(\mathrm{i})$ are the indices of $[e_i]$ and $[/e_i]$ in $\hat{S}$. With this entity marker framework, we effectively capture entity relationships while preserving the original textual structure, thereby achieving better span representations.

Contrastive Object Based on the entity type embeddings and entity span embeddings discussed above, we calculate the supervised contrastive loss via the InfoNCE [Oord *et al.*, 2018] formulation as follows:

$$\mathcal{L}_{\text{class}} = -\log \frac{\exp\left(\cos\left(\hat{\mathbf{r}}_i, \mathbf{e}_{t_i}\right)/\tau\right)}{\sum_{k=1}^{e_n} \mathbb{1}_{[k \neq t_i]} \exp\left(\cos\left(\hat{\mathbf{r}}_i, \mathbf{e}_k\right)/\tau\right)} \tag{13}$$

where t_i denotes the target type for span c_i, $\mathbb{1}_{[k \neq t_i]}$ indicates whether category k matches this label, cos calculates cosine similarity and τ refers to a tuned temperature parameter. Through this objective, we pull span representations closer to their corresponding type vector (positive pair) while pushing farther from unrelated categories (negative pairs) to distinguish entities across domains.

3.4 Training and Inference

Training For the span detection and classification modules, we fine-tune two separate pretrained BERT models, $\mathrm{BERT_d}$ and $\mathrm{BERT_c}$, using the task-specific losses $\mathcal{L}_{dection}$ and $\mathcal{L}_{class}$ respectively. When training the classification module, we only consider gold entity spans in each sentence as supervision signals rather than predicted entities with potential noise.

Inference During inference, for span detection, we first prune out non-entity spans by setting a threshold θ_1, then greedily select the highest probability span proposals while ignoring conflicting proposals to produce candidate entity spans. For span classification, we categorize each span c_i by selecting the type embedding $\mathbf{e}_k$ with maximum cosine similarity $\cos\left(\hat{\mathbf{r}}_i, \mathbf{e}_k\right)/\tau$ to its span representation $\hat{\mathbf{r}}_i$, and only retain entities with matching scores higher than threshold θ_2.

4 Experiments

4.1 Experimental Setting

Datasets

We train and evaluate our model on seven existing public NER benchmarks including diverse domains such as news, biomedicine, movie, and restaurant, etc. The used datasets include three nested NER datasets: ACE 2004[1], ACE 2005[2] and GENIA [Kim *et al.*, 2003]; along with four flat NER datasets: CoNLL 2003 [Sang and De Meulder, 2003], OntoNotes 5[3], MIT Restaurant and MIT Movie [Liu and Lane, 2017]. We use MIT Restaurant and MIT Movie datasets with standard train, dev, and test splits, while adopting the splits of Yu et al. [2020] for the remaining datasets. After unifying the labels with identical semantics across different datasets, we ultimately construct a unified corpus encompassing 47 entity categories.

Implementation Detail

To ensure a fair comparison with previous SOTA works [Zhu and Li, 2022; Yan *et al.*, 2023], we utilize RoBERTa-base [Liu *et al.*, 2019] as the pretrained transformer encoder for both the span detection and classification modules, resulting in a total parameter size of 220M for our SUNER model. In the span detection module, we set the auxiliary loss weight λ as 0.7, the biaffine encoder hidden size is 300. The filtering thresholds θ_1 and θ_2 are set to 0.5 and 0.4, respectively. During training, all parameters are optimized using Adam with a peak learning rate of $1.5e-5$, while Hyper-parameter tuning is performed based on validation set.

Evaluation

We use span-level precision, recall and F1 score to evaluate the performance. A predicted entity is confirmed correct only when its label and boundaries exactly match the ground truth. The reported results are averaged over three runs with different random seeds.

4.2 Results on Supervised Settings

Compared with Unified Methods

For supervised experiments, we first compare SUNER against three unified NER approaches:

- USM [Lou *et al.*, 2023] decouples NER into two token-linking tasks using RoBERTa-Large encoder. However, it cannot jointly train on overlapped datasets.
- PUnifiedNER [Lu *et al.*, 2023], utilizing a T5-based model, employs a prompt-based generative framework to sequentially output entities. We reproduce PUnifiedNER under the same setting.
- One-stage SUNER directly employs the span-based contrastive approach for classification without a separate span detection step.

The results are presented in Table 1, the following observations can be found: (1) SUNER significantly outperforms the generative PUnifiedNER method, achieving an average F1 score improvement of +1.90% across all datasets. Due to token-by-token decoding limitations, the generative approach struggles for direct span modeling and interaction. This limitation is particularly evident on datasets with intricate nested structures, such as Genia, where PUnifiedNER underperforms SUNER by a substantial margin of 4.14%. (2) When separating span detection from domain-specific entity categorization, SUNER on average surpasses the one-stage method by +0.56%. It illustrates that span detection is a more

[1]https://catalog.ldc.upenn.edu/LDC2005T09

[2]https://catalog.ldc.upenn.edu/LDC2006T06

[3]https://catalog.ldc.upenn.edu/LDC2013T19

Method	ACE 04	ACE 05	GENIA	CoNLL 03	OntoNotes 5	Restaurant	Movie	Avg.
USM	87.34	-	-	92.97	-	-	-	-
PUnifiedNER	84.75	85.34	74.61	91.82	87.85	82.17	88.54	85.01
One-stage SUNER	87.04	86.76	77.32	92.83	89.82	81.75	**88.95**	86.35
SUNER(Ours)	**87.75**	**87.21**	**78.75**	**93.34**	**90.54**	**82.41**	88.39	**86.91**

Table 1: The supervised results on unified methods. Avg. indicates the average F1 score across all datasets. The best F1 scores are in **Bold**.

Method	ACE 04	ACE 05	GENIA	CoNLL 03	OntoNotes 5	# P
Biaffine [Yu *et al.*, 2020]	86.70	85.40	80.50	93.50	91.30	N*330M
Seq2Seq [Yan *et al.*, 2021]	86.84	84.74	79.23	93.24	90.28	N*140M
W^2NER [Li *et al.*, 2022]	87.52	86.79	81.39	93.07	90.50	N*110M
BS [Zhu and Li, 2022]	87.98	87.15	-	**93.65**	**91.74**	N*110M
Biaffine-CNN[Yan *et al.*, 2023]	87.31	**87.42**	80.33	-	-	N*110M
DiffusionNER[Shen *et al.*, 2023]	**88.39**	86.93	**81.53**	92.78	90.66	N*330M
SUNER(Ours)	87.75	87.21	78.75	93.34	90.54	1*220M

Table 2: The supervised results on dataset-specific methods. # P refers to the estimated number of parameters required for real-world deployment across N datasets.

general task, enabling the model to capture shared features across domains and consequently enhance its performance.

Compared with Dataset-Specific Methods

We further compare the performance of the proposed model against recent dataset-specific SOTA methods, as shown in Table 2. Different from other models that train separate models for each dataset, SUNER adopts a more challenging strategy by using a single model for joint training across diverse datasets. This unified strategy not only improves the model's ability to generalize but also optimizes storage efficiency. Additionally, our model achieves competitive performance to SOTA dataset-specific methods. Notably, due to benefit from handling of similar entity types in ACE, our model lags behind the SOTA by only 0.64% on ACE 2004 and 0.21% on ACE 2005. While there is a 2.78% performance gap on Genia, this can be attributed to their use of BioBERT [Lee *et al.*, 2020], a specialized model for biomedical texts.

4.3 Results on Zero-shot/Few-shot Settings

Zero-shot Settings

For zero-shot experiments, we focus on in-domain CoNLL 2003 dataset and out-of-domain Restaurant and Movie datasets. In this setting, we first pretrain SUNER on six other corpora before applying model to the new target datasets.

As shown in Table 3, SUNER substantially exceeds the prompt-based PUnifiedNER across all datasets, achieving an impressive average F1 increase of +6.37%. Especially for out-of-domain generalization, SUNER demonstrates considerable gains of +8.13% and +7.09% F1 on Restaurant and Movie datasets respectively. Our approach even averages higher than the UniNER, which is built on a costly 7B parameter LLM that may be impractical in resource-limited settings. By aligning unseen entities to explanatory type texts under our span-based contrastive framework, SUNER better recognizes these unfamiliar categories.

Method	CoNLL	Restaurant	Movie	Avg.
Zero-shot				
ChatGPT	-	37.76	41.00	-
USM	-	23.51	42.11	-
UniNER	-	36.20	**48.70**	-
PUnifiedNER	66.53	35.62	39.27	47.14
SUNER(Ours)	**70.43**	**43.75**	46.36	**53.51**
Few-shot				
Biaffine	71.94	48.53	53.30	57.92
PUnifiedNER	74.85	54.75	55.96	61.86
SUNER(Ours)	**79.74**	**62.67**	**60.89**	**67.77**

Table 3: The zero-shot / few-shot results on three datasets. Results of ChatGPT and UniNER are reported from [Wang *et al.*, 2023] and [Zhou *et al.*, 2023], respectively.

Few-shot Settings

To further analyze model generalization with limited supervision, we conduct additional few-shot experiments based on the pretrained zero-shot model. Specifically, we sample just 20 training sentences per target dataset and repeat each experiment three times to avoid the effect of random sampling.

The results of few-shot learning are shown in Table 3. Again, our proposed SUNER model outperforms PUnifiedNER with +5.91% higher average F1, demonstrating the benefits of modeling entities at the span level. Although span-based Biaffine approaches achieve high supervised performance in dataset-specific setting, directly classifying entities via MLP layer struggles to transfer across unfamiliar domains. In contrast, SUNER exhibits strong adaptation capabilities, improving average zero-shot scores by 14.26% using just 20 examples per dataset.

Method	GENIA			
	# P	F1	Sent/s	SpeedUP
PUnifiedNER	220M	74.61	26.82	1.00×
One-stage SUNER	110M	77.32	**374.21**	**13.95**×
SUNER(Ours)	220M	**78.75**	159.65	5.96×

Table 4: Comparison in terms of parameters, performance, and inference speed on the test set of GENIA. # P means the number of parameters. All experiments are conducted on a single GeForce RTX 3090 with the same setting.

5 Analysis

5.1 Analysis of Inference Efficiency

In this section, we conduct experiments to compare the inference efficiency between SUNER and other baseline models. As illustrated in Table 4, compared to the generative PUnifiedNER, SUNER not only obtains higher accuracy under similar parameter settings but also achieves a notable 5.96× speed increase. Due to seq2seq decoding limitations, PUnifiedNER can only output entities step by step in an autoregressive manner. In contrast, our span-based contrastive framework enables parallel extraction, significantly boosting processing speed. When compared to the One-stage SUNER, our model exhibits the stronger generalization, improving performance on the GENIA dataset by 1.43%. Overall, SUNER achieves satisfactory accuracy at affordable computational cost, proving its effectiveness for unified NER task in practice applications.

	GENIA	CoNLL 03
Default	**78.75**	**93.34**
Mark w/o attention mask	78.14	93.01
Mark w/o position tie	78.42	93.13
w/o Mark	77.93	92.89

Table 5: A comparison of different entity mark structure. (1) **Mark w/o attention mask**: Remove the constrained attention mask and all tokens can attend to each other. (2) **Mark w/o position tie**: Detach marker position embeddings from span boundary embeddings. (3) **w/o Mark** : Remove mark structure entirely, using the raw text as input.

5.2 Analysis of Entity Mark Structure

To analysis the effect of entity mark structure, we compare our approach with three other variant methods. The results are shown in Table 5. All marker-augmented configurations outperform the raw text input setting, validating the importance of modeling span interactions in NER Task. Especially for nested task like GENIA, nested dependency relationships are crucial for entity type confirmation. In addition, removing either the constrained attention mask or position embedding tying leads to varying degrees of performance drops, indicating both are critical for effectively incorporating markers while preserving original textual structure.

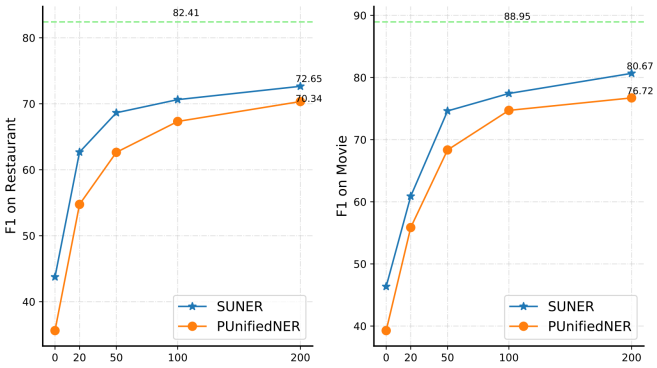

Figure 3: Comparisons of performance with varying number of training example on out-of-domain Restaurant and Movie datasets.

We achieve this balance by isolating text tokens from entity marker tokens through the attention mask, and ensuring position embedding tying to preserve the position embeddings of the original tokens. As a result, the BERT context vectors of the text tokens remain unchanged. Yet, by integrating span vectors using entity markers, we facilitate interactions between spans, leading to superior performance.

5.3 Analysis of Generalization Capability

As previously highlighted, our proposed SUNER aims to improve model generalization to new domains through joint training across different datasets. To validate this point, we evaluate SUNER's performance against the baseline models with different training sizes, particularly focusing on out-of-domain Restaurant and Movie datasets. As shown in Figure 3, even with just 50 training examples, SUNER achieves approximately 80% of the optimal performance, which demonstrates its quick adaptation ability to new domains. When increasing the dataset size to 200 examples, SUNER's performance continues to improve, lagging behind the SOTA by reasonable margins of 9.76% on Restaurant and 8.28% on Movie. By incorporating a semantic matching-based contrastive framework, SUNER can perform well with limited labeled data, highlighting its potential in practical applications where annotation scarcity hinders model development.

6 Conclusion

In this paper, we present a novel method SUNER that leverages span-based contrastive framework to extract entities in parallel for unified NER. To improve model generalization, we divide SUNER into two stages, separating span detection from domain-specific span classification. In the span classification stage, we utilize descriptive entity text instead of discrete labels to help model better understand and categorize entities. We conduct extensive experiments in various settings, including supervised and zero/few-shot conditions. The results demonstrate that our approach achieves better performance and higher efficiency compared to previous state-of-the-art methods.

Acknowledgments

The work is supported by National Natural Science Foundation of China (No. 62172039, U21B2009 and 62276110) and MIIT Program(CEIEC-2022-ZM02-0247).

References

[Chuang *et al.*, 2020] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

[Das *et al.*, 2022] Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca Passonneau, and Rui Zhang. CONTaiNER: Few-shot named entity recognition via contrastive learning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6338–6353, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

[Ganea and Hofmann, 2017] Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[Gao *et al.*, 2021] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6894–6910. Association for Computational Linguistics (ACL), 2021.

[Han *et al.*, 2022] Wei Han, Hui Chen, Zhen Hai, Soujanya Poria, and Lidong Bing. Sancl: Multimodal review helpfulness prediction with selective attention and natural contrastive learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5666–5677, 2022.

[Huang *et al.*, 2022] Yucheng Huang, Kai He, Yige Wang, Xianli Zhang, Tieliang Gong, Rui Mao, and Chen Li. COPNER: Contrastive learning with prompt guiding for few-shot named entity recognition. In Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na, editors, *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2515–2527, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

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

[Le and Titov, 2018] Phong Le and Ivan Titov. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1595–1604, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[Lee *et al.*, 2020] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[Li *et al.*, 2020] Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. Flat: Chinese ner using flat-lattice transformer. *arXiv preprint arXiv:2004.11795*, 2020.

[Li *et al.*, 2022] Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10965–10973, 2022.

[Liu and Lane, 2017] Bing Liu and Ian Lane. Multi-domain adversarial learning for slot filling in spoken language understanding. *arXiv preprint arXiv:1711.11310*, 2017.

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

[Lou *et al.*, 2023] Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Universal information extraction as unified semantic matching. 2023.

[Lu *et al.*, 2022] Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. Unified structure generation for universal information extraction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[Lu *et al.*, 2023] Jinghui Lu, Rui Zhao, Brian Mac Namee, and Fei Tan. Punifiedner: a prompting-based unified ner system for diverse datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13327–13335, 2023.

[Ma and Hovy, 2016] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1064–1074, 2016.

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[Rathore *et al.*, 2022] Vipul Rathore, Kartikeya Badola, Parag Singla, et al. Pare: A simple and strong baseline for monolingual and multilingual distantly supervised relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 340–354, 2022.

[Sang and De Meulder, 2003] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.

[Sarhan and Spruit, 2021] Injy Sarhan and Marco Spruit. Open-cykg: An open cyber threat intelligence knowledge graph. *Knowledge-Based Systems*, 233:107524, 2021.

[Shen *et al.*, 2023] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. DiffusionNER: Boundary diffusion for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 3875–3890, Toronto, Canada, July 2023. Association for Computational Linguistics.

[Souza *et al.*, 2019] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*, 2019.

[Tan *et al.*, 2022] Qingyu Tan, Ruidan He, Lidong Bing, and Hwee Tou Ng. Domain generalization for text classification with memory-based supervised contrastive learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6916–6926, 2022.

[Wang *et al.*, 2023] Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*, 2023.

[Yan *et al.*, 2021] Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. A unified generative framework for various NER subtasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 5808–5822. Association for Computational Linguistics, August 2021.

[Yan *et al.*, 2023] Hang Yan, Yu Sun, Xiaonan Li, and Xipeng Qiu. An embarrassingly easy but strong baseline for nested named entity recognition. 2023.

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

[Zhang *et al.*, 2023] Sheng Zhang, Hao Cheng, Jianfeng Gao, and Hoifung Poon. Optimizing bi-encoder for named entity recognition via contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[Zhong and Chen, 2021] Zexuan Zhong and Danqi Chen. A frustratingly easy approach for entity and relation extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 50–61, 2021.

[Zhou *et al.*, 2023] Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. Universalner: Targeted distillation from large language models for open named entity recognition. *arXiv preprint arXiv:2308.03279*, 2023.

[Zhu and Li, 2022] Enwei Zhu and Jinpeng Li. Boundary smoothing for named entity recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 7096–7108. Association for Computational Linguistics, May 2022.