Neural Processing Letters (2025) 57:13
https://doi.org/10.1007/s11063-025-11729-x

ECTTLNER: An Effective Cross-Task Transferring Learning Method for Low-Resource Named Entity Recognition

Yiwu Xu[1] · **Yun Chen**[2]

Accepted: 15 January 2025 / Published online: 31 January 2025

Abstract
Named entity recognition is a fundamental task in natural language processing that significantly impacts the performance of its downstream tasks. Cross-task transfer learning methods are more naturally suited for low-resource named entity recognition compared to cross-language and cross-domain transfer learning methods. Existing cross-task transfer learning methods improve the performance of the low-resource named entity recognition by leveraging relevant information from other auxiliary tasks, such as sentence-level and token-level information. However, these methods do not fully exploit token-level information of entities, leaving room for improvement in low-resource named entity recognition. To futher improve the performance of the low-resource named entity recognition, this paper proposes a simple and effective cross-task transfer learning method called ECTTLNER, which introduces Sentence Contains Entities, Sentence Entity Number, Token Is Entity, and Token Boundary Label prediction tasks into named entity recognition and performs multi-task learning together with the main sequence labeling task. Experimental results on three NER datasets demonstrate that ECTTLNER outperforms a set of state-of-the-art baseline models, and achieves more than a 2.6% improvement in F1-score over these baseline models, particularly in low-resource scenarios.

Keywords Named entity recognition · Cross-task transfer learning · Multi-task learning

1 Introduction

Named entity recognition (NER) aims to identify entities from the input text and assign them corresponding labels [1–3] such as person (PER), location (LOC), and organization (ORG). And NER is a fundamental task in Natural Language Processing (NLP) and plays an

✉ Yiwu Xu
953752671@qq.com

Yun Chen
1210437650@qq.com

[1] Guangzhou Institute of Science and Technology, Guangzhou 510540, China

[2] Nanfang College Guangzhou, Guangzhou 510970, China

important role in its downstream tasks such as question answering systems [7], entity linking [5], and relation extraction [6].

With the rapid development of deep learning, neural named entity recognition (NeuralNER) has become a mainstream method owing to its superior performance in recent years [4, 8–14, 35]. However, NeuralNER's performance heavily relies on a large amount of annotated examples, leading to poor performance in low-resource scenarios, which are defined as situations with fewer than 2000 labeled examples available for training. This is primarily because when models are constrained by limited data, they often fail to generalize well to diverse entity types and contexts, resulting in overfitting and reduced accuracy when handling new, unseen data. To improve the NER's performance in low-resource scenarios, some researchers have introduced transfer learning techniques. These studies can be categorized into three areas: cross-language transfer learning (CLTLNER) [15–18], cross-domain transfer learning (CDTLNER) [19–22], and cross-task transfer learning (CTTLNER) [23–27, 34, 40, 41]. Compared to CLTLNER and CDTLNER, CTTLNER has a distinct advantage: the model does not need to learn new tasks from scratch but can instead adapt and optimize quickly based on existing knowledge, which is straightforward for low-resource NER.

The existing CTTLNER methods primarily improve the low-resource NER's performance by leveraging relevant information from other auxiliary tasks, such as sentence-level information [23, 27, 41] and token-level information [26, 27, 34, 40, 41]. In low-resource scenarios, token-level information contributes more to NER than sentence-level information, as confirmed by reference [27]. Additionally, the existing works typically focus on only one type of token-level information, such as the relationships between tokens [26], word spacing information [34], multi-token entity or single-token entity or neither information [27], entity category information [40], and grid-level semantic information [41]. They do not introduce more token-level information regarding entities, which may affect the performance of low-resource NER. Therefore, we attempt to introduce multiple types of token-level information to further enhance the performance of low-resource NER. To this end, we propose a simple and effective cross-task transfer learning method for low-resource NER, namely ECTTLNER. As shown in Fig. 1, besides the main sequence labeling task, we employ four related classification tasks: a binary classification task to predict whether a sentence contains entities, a multi-class classification task to predict the number of entities in a sentence, a binary

Text: Only **France** and **Britain** backed **Fischler** 's proposal .
Sentence Contains Entities(SCE): **True**
Sentence Entity Number(SEN): **3**
Token Is Entity(TIE): 0 1 0 1 0 1 0 0 0
Token Boundary Label(TBL): O B O B O B O O O
NER Label: O B-LOC O B-LOC O B-PER O O O

Fig. 1 An example of automatically generating auxiliary information from our dataset. In our work, we designed four auxiliary tasks to help improve the NER task. Two of them are sentence-level tasks, and the other two are token-level tasks. Specifically, the first sentence-level task predicts whether a sentence contains entities; the second sentence-level task predicts how many entities a sentence contains; the first token-level task predicts whether a given token belongs to an entity; the second token-level task predicts the entity boundary label (B, I, and O) of a given token. Although we added extra labels to the existing data to support the training of auxiliary tasks, these labels can be automatically obtained from the original NER data without additional manual annotation. In summary, we attempt to leverage the multi-level information implicit in the original dataset to enhance the performance of NER

classification task to predict whether a Token belongs to an entity, and a multi-class classification task to predict the entity boundary label (B, I, and O) of a Token. In the remainder of this paper, these four tasks are named Sentence Contains Entities (SCE), Sentence Entity Number (SEN), Token Is Entity (TIE), and Token Boundary Label (TBL), respectively, with the main sequence labeling task named NER. Our goal is to extract useful training signals from these classifications to enhance the robustness of our NER system.

Our contributions can be summarized as follows:

- We propose a simple and effective cross-task transfer learning method, namely ECTTLNER. By introducing SCE, SEN, TIE, and TBL prediction tasks, and performing multi-task learning together with SEQLAB task, this method yields significant benefits in low-resource scenarios. To the best of our knowledge, we are the first to introduce token-level information via Token Is Entity (TIE) and Token Boundary Label (TBL) in the low-resource NER domain.
- Through extensive testing across three NER datasets, ECTTLNER demonstrates superior performance compared to a set of state-of-the-art baseline models, particularly in low-resource scenarios.

2 Related Works

Traditional NER methods can be divided into dictionary-based [28], rule-based [29, 30], and machine learning-based [31–33] approaches. With the rapid development of deep learning, neural NER methods—such as LSTM-CRF [8–11, 35] and BERT-LSTM-CRF [12–14]—have achieved promising results. However, these NER methods are not well-suited for low-resource scenarios due to their requirement for large amounts of annotated data. In recent years, various methods have been proposed to address this challenge, with one of the most promising directions being transfer learning. This method can be categorized into three types: cross-lingual transfer learning (CLTLNER), cross-domain transfer learning (CDTLNER), and cross-task transfer learning (CTTLNER).

CLTLNER methods aim to transfer knowledge from rich-resource languages to low-resource languages. For example, Xie et al. [15] proposed a translation approach to enhance cross-lingual text mapping. They utilized a self-attention mechanism as the encoder to improve robustness against word order differences across languages. Ni et al. [16] automatically generated annotated examples for the target language by utilizing label mapping on cross-lingual corpora. Wu et al. [17] proposed a meta-learning algorithm to improve the model's generalization across different languages by fine-tuning a source language model using a small number of test examples from the target language. Bari et al. [18] proposed an unsupervised cross-lingual NER model that incorporates fine-tuning and word-level adversarial learning through feature augmentation and parameter sharing.

In addition to adopting the CLTLNER method, some researchers have also proposed the CDTLNER method. The ideas behind both are similar, focusing on transferring knowledge from rich-resource domains to low-resource domains. For example, Liu et al. [19] proposed a cross-domain NER model to enhance the robustness of zero-resource domain adaptation by employing a hybrid entity expert system and multi-task learning. Wang et al. [22] proposed a cross-domain dual transfer learning system that utilizes label-aware mechanisms, achieving feature and parameter transfer through two label-aware rules. Jia et al. [20] facilitated

knowledge transfer across domains and tasks by designing a novel parameter generation network. Lin and Lu [21] proposed adaptive layers based on existing neural network structures, eliminating the need to retrain the model with source domain data.

Unlike the two methods mentioned above, the CTTLNER method is naturally well-suited for low-resource NER. It improves the low-resource NER's performance by leveraging other related sentence-level and token-level information. Sanh et al. [24] jointly trained tasks for entity mention detection, relation extraction, and named entity recognition using a hierarchical model. Yoon et al. [25] proposed a model that integrates multiple single-task NER models to achieve more accurate predictions in the biomedical domain. However, neither reference [24] nor [25] utilizes additional token-level or sentence-level information. Mo et al. [26] proposed a multitask transformer that integrates entity boundary detection into NER tasks and enhances boundary detection by introducing an auxiliary task to classify the relationships (token-level information) between tokens. Park et al. [34] studied a multitask learning approach that combines NER with word spacing (token-level information) prediction for in-car voice assistants in both Korean and English within the automotive domain. Kruengkrai et al. [23] proposed a multitask learning framework that integrates sentence and token annotation models to improve the low-resource NER's performance. However, they introduced an auxiliary task only for sentence-level information, i.e. product categories are used as sentence-level labels, which required additional manual annotation. Tong et al. [27] introduced various auxiliary tasks that address both sentence-level and token-level information. The auxiliary information for these tasks is generated directly from existing datasets without the need for additional annotation. However, they specifically introduced an auxiliary task for token-level information, namely a multi-token entity classification task. Dai et al. [40] proposed a Category Semantic Enhancement framework for NER (CSE-NER). The model's adaptability was enhanced by introducing additional downstream task information during the fine-tuning stage. Specifically, they integrated a category semantic embedding layer to provide category semantic information (token-level information) distinct from context embeddings. Zhao et al. [41] proposed a Multi-Level Semantic Understanding (MLSU) model for the unified NER task. This model employs an encoder-decoder structure integrated with a sequence-level block for sentence-level information and a grid-level semantic block for token-level information, enhancing its ability to effectively handle complex text structures. In summary, these methods focus solely on introducing one type of token-level information without fully exploring the potential of token-level information for entities. Therefore, we attempt to introduce multiple types of token-level information to further improve the low-resource NER's performance. Moreover, we have compared all the described related works, and the comparison results are presented in Table 1.

3 Methodology

3.1 Model Architecture

The overview of our proposed ECTTLNER method is illustrated in Fig. 2. This method consists of five tasks: the SEQLAB task, the SCE prediction task, the SEN prediction task, the TIE prediction task, and the TBL prediction task. The SEQLAB task aims to predict labels for each token in a given sentence. The SCE prediction task determines whether a given sentence contains entities, the SEN prediction task determines the number of entities a given sentence contains, the TIE prediction task identifies whether each token in a given

Table 1 Comparison of the related works

Author(s)	Year	Methodology	Key technologies
Savova et al. [28]	2010	Dictionary-based	Dictionary-matching
Etzioni et al. [29]	2005	Rule-based	Predefined rules
Zhang et al. [30]	2013	Rule-based	Predefined rules
Makino et al. [31]	2002	Machine learning-based	HMM-SVM
Krishnan et al. [32]	2006	Machine learning-based	CRFs
He et al. [33]	2015	Machine learning-based	Predefined rules-CRF
Huang et al. [8]	2015	NeuralNER	LSTM-CRF
Lample et al. [9]	2016	NeuralNER	LSTM-CRF
Chiu et al. [10]	2016	NeuralNER	LSTM-CRF
Akbik et al. [11]	2018	NeuralNER	LSTM-CRF
Beltagy et al. [35]	2019	NeuralNER	LSTM-CRF
Devlin et al. [12]	2018	NeuralNER	BERT-LSTM-CRF
Brown et al. [13]	2020	NeuralNER	BERT-LSTM-CRF
Lewis et al. [14]	2019	NeuralNER	BERT-LSTM-CRF
Xie et al. [15]	2018	CLTLNER	Self-attention
Ni et al. [16]	2017	CLTLNER	Corpora-mapping
Wu et al. [17]	2020	CLTLNER	Meta-learning
Bari et al. [18]	2020	CLTLNER	Unsupervised-learning
Liu et al. [19]	2020	CDTLNER	Expert system
Jia et al. [20]	2019	CDTLNER	Parameter-generation
Bill et al. [21]	2018	CDTLNER	Adaptive network
Wang et al. [22]	2018	CDTLNER	Label-aware
Sanh et al. [24]	2019	CTTLNER	Hierarchical model
Yoon et al. [25]	2019	CTTLNER	Multiple identical NER models
Mo et al. [26]	2023	CTTLNER	Multitask learning & A token-level information
Park et al. [34]	2023	CTTLNER	Multitask learning & A token-level information
Kruengkrai et al. [23]	2020	CTTLNER	Multitask learning & A sentence-level information
Tong et al. [27]	2021	CTTLNER	Multitask learning & Two sentence-level information & A token-level information
Dai et al. [40]	2024	CTTLNER	Multitask learning & A token-level information
Zhao et al. [41]	2024	CTTLNER	Multitask learning & A sentence-level information & A token-level information

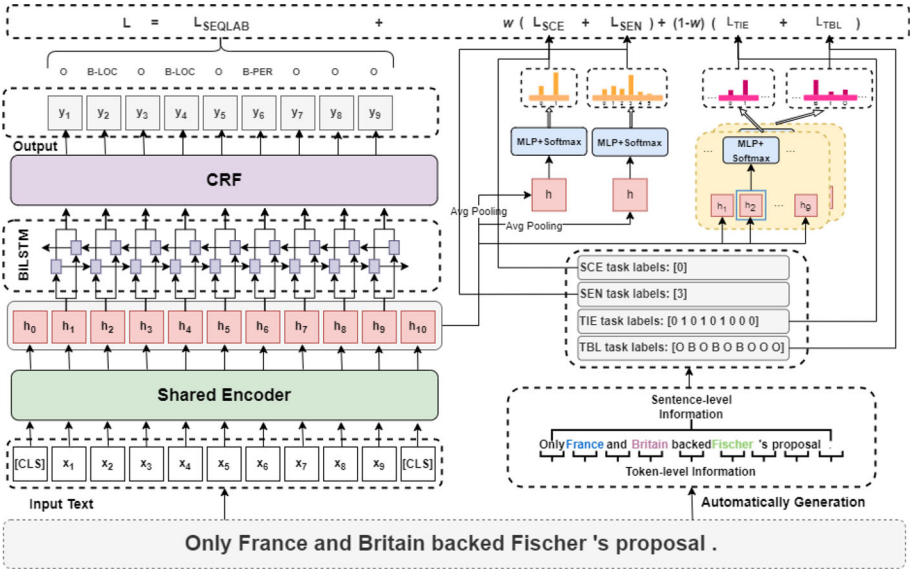

Fig. 2 The overview of our proposed ECTTLNER method

sentence is an entity, and the TBL prediction task predicts the boundary labels for each token in a given sentence. These tasks are jointly trained within a multitask learning framework that utilizes a shared encoder.

3.2 Shared Encoder

The shared encoder is used across all tasks via hard parameter sharing [39]. Consider a sentence S consisting of N tokenized words $\{w_i\}_{i=1}^{N}$. We use the BERT encoder to convert these words into their respective contextual embeddings. We generate an input sequence by concatenating a *[CLS]* token, the tokenized word sequence $\{w_i\}_{i=1}^{N}$, and a *[SEP]* token. We then project the input embeddings into a sequence of context vectors through a series of L stacked Transformer Blocks (TB), as shown in Eq. (1):

$$h_o, ..., h_{n+1} = TB_L([CLS], w_1, ..., w_n, [SEP]) \tag{1}$$

We use the outputs of the shared encoder, denoted as $H = \{h_1,...,h_i,...,h_n\}$, as inputs for both sentence-level and token-level tasks, as described below.

3.3 SEQLAB Task

SEQLAB: It is essentially a sequence-labeling task. In our model, we utilize a BiLSTM to extract global decoding information from long-distance contexts and employ a CRF to determine the globally optimal decoding sequence.

Formally, we assume that the output of the BiLSTM is $r = (r_1,r_2,...,r_n)$ and it has a corresponding general label sequence $y = (y_1,y_2,...,y_n)$. Given an input r, the conditional

probability of the label sequence y is calculated as shown in Eq. (2):

$$p(y|r;\theta) = \frac{\prod_{i=1}^{n} \psi(r_i, y_i, y_{i-1})}{\sum_{y' \in Y(S)} \prod_{i=1}^{n} \psi(r_i, y'_i, y'_{i-1})} \tag{2}$$

where *Y(S)* is all possible label sequences for the sentence set S, θ is the parameter set, and $\varphi(r_i, y_i, y_{i-1})$ is the potential function, calculated as in Eq. (3):

$$\psi(r_i, y_i, y_{i-1}) = \exp(y_i^T W^T r_i + y_{i-1}^T T y_i) \tag{3}$$

where W and T are parameters.

The loss function for the SEQLAB task is calculated as shown in Eq. (4):

$$L_{SEQLAB} = -\sum_{s \in S} \log(p(y_s|r_s;\theta)) \tag{4}$$

where, S represents the set of sentences in the training set, r_s denotes the output of the BiLSTM for sentence s, and y_s denotes the corresponding label sequence for sentence s.

3.4 SCE and SEN Prediction Tasks

Both the SCE and SEN prediction tasks are sentence-level classification tasks. Our model aims to obtain the globally optimal representation of words using sentence-level labels.

SCE: It is a sentence-level binary classification task. Its goal is to predict whether a given sentence S contains entities. A value of 0 indicates no entities are present, while a value of 1 indicates entities are present. We utilize this global guidance information at the sentence-level to prevent the model from mistakenly predicting the presence of entities in sentences that do not contain them, and vice versa.

This is achieved by applying average pooling to H, which we represent as h. Finally, we use a multilayer perceptron (MLP) classifier and a Softmax layer to predict the probability of whether a sentence contains entities, as calculated in Eq. (5):

$$P(n|H)_{SCE} = Soft\max(MLP_{SCE}(h)) \tag{5}$$

where, $h \in R^d$ represents the pooling output of the model, d is the dimension of the shared feature vector, n is the number of labels, and for the SCE prediction task, n is 2.

SEN: It is a sentence-level multi-classification task. Its goal is to predict the number of entities in a given sentence. This task is more challenging than SCE because accurately predicting the number of entities is difficult, and there are often issues with under- or over-identifying entities. To address this issue, this paper limits the possible number of entities to six cases. If a sentence contains 0, 1, 2, 3, or 4 entities, the corresponding classification result will be 0, 1, 2, 3, or 4, respectively. If there are 5 or more entities, the classification result will be set to 5.

Similarly, we applied an MLP classifier and a Softmax layer to predict the probability of the number of entities contained in a sentence, as calculated in Eq. (6):

$$P(n|H)_{SEN} = Soft\max(MLP_{SEN}(h)) \tag{6}$$

In the SEN task, the value of n is set to 6.

The loss for our sentence-level task is calculated as in Eq. (7)–(9):

$$L_{SBE} = -\sum_{k} \sigma\left(y_k^{SBE} = \hat{y}^{SBE}\right) \log\left(P(n|H)_{SBE}\right) \tag{7}$$

$$L_{SEN} = -\sum_{k} \sigma\left(y_k^{SEN} = \hat{y}^{SEN}\right) \log\left(P(n|H)_{SEN}\right) \tag{8}$$

$$L_S = L_{SBE} + L_{SEN} \tag{9}$$

where, if class $\hat{y}$ of H is the correct ground-truth label of class k, then $\sigma(y_k = \hat{y}) = 1$. Otherwise, $\sigma(y_k = \hat{y}) = 0$.

3.5 TIE and TBL Prediction Tasks

TIE and TBL prediction tasks are two token-level classification tasks. These tasks aim to enrich the local word representation of the shared encoder. We use two classifiers to determine whether each token is an entity and to predict its boundary label.

TIE: It is a token-level binary classification task. The goal is to determine whether a given token encoding h_i belongs to an entity. However, in some cases, the model may mistakenly classify a non-entity token as an entity or vice versa. Therefore, we introduce token-level local guidance information to address this issue.

Specifically, we input the tokenized representation h_i into an MLP classifier and apply a Softmax layer to obtain the probability P_{TIE}^i of whether the tokenized representation h_i is an entity, as calculated in Eq. (10):

$$P_{TIE}^i = Soft\max(MLP_{TIE}(h_i)) \tag{10}$$

TBL: It is a token-level multi-classification task. Given a h_i in H, the goal is to predict its boundary label in *{B, I, O}*, where *B, I,* and *O* represent the start, internal, and other positions of the entity, respectively. By introducing this task, the model will understand the specific boundary labels of h_i, thus alleviating the problem of entity boundary ambiguity.

We use the same method to obtain the probability P_{TBL}^i of the boundary label for the tokenized representation h_i, as calculated in Eq. (11):

$$P_{TBL}^i = soft\max(MLP_{TBL}(h_i)) \tag{11}$$

The loss calculation for our token-level classification task is given by Eq. (12)–(14):

$$L_{TIE} = -\sum_{i=1}^{N} [y_{TIE}^i \log P_{TIE}^i + (1 - y_{TIE}^i) \log(1 - P_{TIE}^i)] \tag{12}$$

$$L_{TBL} = -\sum_{i=1}^{N} \left[y_{TBL}^i \log P_{TBL}^i + (1 - y_{TBL}^i) \log(1 - P_{TBL}^i)\right] \tag{13}$$

$$L_T = L_{TIE} + L_{TBL} \tag{14}$$

where, y_{TIE}^i and y_{TBL}^i represent the labels for whether the token h_i is an entity and its boundary label, respectively.

Finally, we calculate the final loss of the model by weighting and summing the losses from different tasks, as shown in Eq. (15):

$$L = L_{SEQLAB} + wL_S + (1 - w)L_T \tag{15}$$

where w is a hyperparameter that balances the sentence-level and token-level tasks.

Table 2 The statistics of three datasets

Dataset	Type	Train	Dev	Test
ConLL-2003[36]	#sentence	14,041	3250	3453
	#word	23,623	9966	9488
NCBI-disease[37]	#sentence	5432	923	940
	#word	9284	3478	3569
JNLPBA[38]	#sentence	18,607	1939	4260
	#word	20,712	5679	9624

4 Experiments

4.1 Datesets

We used three NER datasets in our experiments. The ConLL-2003 dataset [36], published by the European Association for Computational Linguistics in 2003, is an English named entity recognition dataset that contains four types of named entities: PER, LOC, ORG, and MISC. The NCBI-disease dataset [37] and the JNLPBA dataset [38] are two public datasets in the biomedical domain focused on NER. NCBI-disease includes four entity types: SpecificDisease, CompositeMention, DiseaseClass, and Modifier. JNLPBA encompasses five entity types: protein, cell_type, DNA, RNA, and cell_line. Table 2 summarizes the statistics of these datasets. Furthermore, to ensure the accuracy and reliability of the labels automatically generated from the original dataset, we have manually inspected the dataset for errors or anomalies based on domain knowledge and experience.

4.2 Experimental Settings

We implemented the ECTTLNER method based on the source code by Tong et al. (2021) [27], utilizing the pre-trained model *bert-base-uncased* as the shared encoder. For the low-resource scenarios, we obtained the training datasets by randomly deleting sentences from the entire training set and extracting 1000 and 2000 sentences. All datasets were trained with a batch size of 16, a maximum sentence length of 256, and a dropout rate of 0.1 following the shared encoder. The training epoch number was set to 20. During training, all models were optimized using Adam with a learning rate of 5e−5 and a linear warm-up scheduler. The models were trained on NVIDIA RTX 2080 Ti GPUs and evaluated using standard metrics: Precision, Recall, and F1-score, to assess overall performance.

4.3 Baseline Models

We compared our proposed model with the following baseline models:

BiLSTM-CRF(2015) [8]**:** This employs BiLSTM to effectively learn features before and after the word and utilizes CRF to obtain dependencies among labels.

BiLSTM-CNNs(2016) [10]**:** This utilizes a CNN and pretrained word embeddings to extract word-level and character-level embeddings, and partial vocabulary matches are encoded in the neural network.

NeuralNER(2016) [9]**:** This divides words into a series of characters and employs BiLSTM instead of a CNN to learn character-level features.

CS Embeddings(2018) [11]**:** This obtains context embeddings for all characters in the word; these embeddings are then concatenated with pretrained word embeddings to form the final word representation.

SciBERT(2019) [35]**:** This introduces a BERT-based scientific text contextual embedding model that achieves state-of-the-art performance in multiple tasks.

Sanh et al.(2019) [24]**:** This uses a hierarchical model to jointly train the tasks of entity mention detection, relation extraction, and named entity recognition.

CollaboNet(2019) [25]**:** This demonstrates that multiple single-task NER models can be used to improve prediction accuracy in the biomedical domain.

Mo et al.(2023) [26]**:** This proposes a multi-task transformer that incorporates the entity boundary detection task the named entity recognition task.

MT-BioBERT(2021) [27]**:** This proposes a multi-task learning framework that combines several sentence-level information and a token-level information to improve the low-resource NER's performance.

ADMit(2023) [34]**:** This explores a multi-task learning method that combines NER and word spacing prediction in the automotive domain.

CSE-NER(2024) [40]**:** This explores a multi-task learning method that combines NER and Entity Type Classification in the general domain.

MLSU(2024) [41]**:** This leverages an Encoder-Decoder structure that integrates sequence and grid-level semantic blocks, enhancing the model's ability to process complex text structures effectively.

4.4 Main Results

We compared our model with several state-of-the-art NeuralNER methods [8–11, 35] and CTTLNER methods [24–27, 34, 40, 41]. Table 3 reports the performance comparison between our model and these methods across three datasets, including percentage values of precision (P), recall (R), and F1-score (F). The results in bold indicate the best performance for F1-score.

There are several key observations from Table 3. First, it is evident that our model achieves the best performance among all compared methods on all datasets. For instance, on the ConLL-2003 dataset, our model achieves a 3.45% increase in F1-score compared to the best baseline model [11] of the NeuralNER methods, and a 2.21% increase in F1-score compared to the best baseline model [27] of the CTTLNER methods. This improvement is attributed to the fact that [27] utilizes only one type of token-level information, which determines whether each word belongs to a multi-token entity or a single-token entity. In contrast, we employ two distinct types of token-level information, leading to better results. This indicates that the token-level information we use is more effective and validates the progressiveness of our multi-task learning strategy. The reasons why the token-level tasks used in our method are more effective than those used in existing methods are primarily as follows: The "Token is Entity" prediction task allows the model to more accurately predict the masked entity tokens by explicitly injecting entity labels into the sentence context. The "Token Boundary Label" prediction task enables the model to clearly identify the start and end positions of entities using B, I, O labels, which is crucial for accurately extracting entity information. Another interesting observation is that our model also achieves higher recall scores on all three datasets than other methods, with an average increase of more than 2.88%, 4.77%, and

Table 3 Model performance comparison on the three datasets

Models		ConLL-2003			NCBI-Disease			JNLPBA		
		P	R	F	P	R	F	P	R	F
Neural-NER	BiLSTM-CRF(2015) [8]	92.78	87.43	90.02	85.47	74.32	79.51	73.47	68.27	70.77
	BiLSTM-CNNs(2016) [10]	91.35	91.06	91.21	82.61	76.67	79.52	73.96	70.52	72.20
	NeuralNER(2016) [9]	90.88	90.62	90.75	85.67	64.30	73.46	73.08	71.56	72.31
	CS Embeddings(2018) [11]	92.37	93.12	92.74	85.02	87.33	86.16	71.18	77.68	74.29
	SciBERT(2019) [35]	88.46	89.13	88.79	84.32	89.06	86.63	70.73	80.36	75.24
CTTL-NER	Sanh et al. (2019) [24]	–	–	91.63	–	–	–	–	–	–
	CollaboNet(2019) [25]	93.31	87.47	90.29	85.48	87.27	86.36	78.92	88.42	83.38
	Mo et al. (2023) [26]	93.11	93.77	93.44	–	–	–	–	–	–
	MT-BioBERT(2021) [27]	94.23	90.22	93.98	88.90	90.94	89.91	83.90	85.94	84.19
	ADMit(2023) [34]	92.97	93.24	93.10	–	–	–	–	–	–
	CSE-NER(2024) [40]	–	–	93.46	–	–	89.90	–	–	–
	MLSU(2024) [41]	93.44	94.33	93.89	–	–	–	–	–	–
ECTTLNER	96.16	96.32	**96.19**	95.19	94.77	**94.68**	89.78	90.18	**89.75**	

The significance bold indicates the lowest F1-score value

5.79%, respectively. This suggests that the introduction of token-level and sentence-level tasks aids the model in predicting more positive results. These results demonstrate that our method effectively harnesses relevant information from auxiliary tasks to enhance entity boundary detection and more accurately identify entities. Furthermore, the entity boundary information embedded in our model further enhances the model's performance.

To verify the model's performance in low-resource scenarios, we simulated such scenarios and conducted experiments. Initially, we randomly deleted sentences from the training set to create low-resource scenarios of varying sizes, while keeping the development and test sets unchanged. Subsequently, we used these datasets to evaluate the model's performance.

As shown in Table 4, we documented the performances of the best baseline (SOTA-N) from the NeuralNER methods, the best baseline (SOTA-C) from the CTTLNER methods, and our method across different resource scenarios. The F1-score results in bold indicate the highest performance.

From the results on 1000, 2000, and 5000 sentences, it is evident that our method significantly outperforms SOTA-C on all three datasets. This is especially true when the training set consists of only 1000 sentences, where our method shows the most significant improvement in F1-score compared to SOTA-C. This indicates that our proposed method is more robust in low-resource scenarios. For the three complete datasets, our proposed model improves the F1-score by 0.51%, 0.35%, and 0.56% compared to SOTA-C, with the largest improvement observed on JNLPBA. Additionally, when comparing our method with SOTA-N on the three datasets, it is clear that our method is superior to SOTA-N in all low-resource scenarios. It is also worth noting that our method is more effective than SOTA-N on all complete datasets. Therefore, by introducing SBE, SEN, TIE, and TBL prediction tasks, we not only enhance the performance of NER models on the entire dataset but also significantly boost their robustness in low-resource scenarios.

4.5 Ablation Study

To measure the impact of different auxiliary tasks in our model, we conducted a series of ablation studies, each removing a single component: SCE, SEN, TIE, and TBL. Specifically, we utilized the NeuralNER model BERT-BiLSTM-CRF without these components as our 'baseline'. The results of the ablation studies are presented in Table 5. The results in bold indicate the best performance.

From these results, it is evident that the multi-task model, which includes SCE, SEN, TIE, and TBL, shows significant improvement over the baseline model in all low-resource scenarios and across the entire data of all three datasets. However, compared to TIE and TBL, SCE and SEN provide less assistance, suggesting that token-level information plays a more significant role in multi-task learning than sentence-level information. Additionally, when compared to SCE and TIE, SEN and TBL offer less enhancement. This indicates that the effectiveness of auxiliary tasks in multi-task learning is closely related to their individual performance. In fact, due to the inherent difficulty of multi-classification tasks, their performance tends to be lower than that of binary classification tasks. Nevertheless, by integrating multi-task learning with these four components, as proposed in our method, we can achieve the optimal performance.

Table 4 Model performance comparison under different resource scenarios

#Sentences	Methods	ConLL-2003			NCBI-disease			JNLPBA		
		P	R	F	P	R	F	P	R	F
1000	SOTA-N	83.17	84.83	83.01	80.09	86.54	84.55	78.46	79.12	78.32
	SOTA-C	86.21	86.86	86.03	83.99	91.64	87.65	81.53	82.18	81.35
	ECTTLNER	89.04	89.44	**88.88**	89.87	92.29	**89.94**	84.37	84.81	**84.14**
2000	SOTA-N	88.13	88.51	87.62	82.65	88.84	87.26	84.98	82.66	82.02
	SOTA-C	91.15	91.52	90.74	85.84	91.88	90.45	88.56	85.91	85.12
	ECTTLNER	93.11	92.42	**91.53**	92.33	93.51	**91.86**	86.95	86.25	**86.34**
5000	SOTA-N	90.54	90.41	90.28	90.02	91.15	90.22	85.04	84.2	84.66
	SOTA-C	93.65	93.49	93.26	93.08	93.49	92.9	88.1	87.94	87.71
	ECTTLNER	93.97	93.82	**93.88**	94.21	94.76	**93.41**	88.81	88.51	**88.54**
All	SOTA-N	94.83	94.12	94.48	89.6	94.76	92.11	84.99	85.63	86.08
	SOTA-C	95.93	95.22	95.68	94.37	94.64	94.33	88.06	90.94	89.19
	ECTTLNER	96.16	96.32	**96.19**	95.19	94.77	**94.68**	89.78	90.18	**89.75**

The significance bold indicates the lowest F1-score value

Table 5 The results of ablation study under different resource scenarios

#Sentences	Model	ConLL-2003	NCBI-Disease	JNLPBA
1000	Baseline	86.38	85.08	78.03
	+ SCE	87.67	87.92	81.05
	+ SEN	87.49	87.44	80.90
	+ TIE	88.50	89.19	82.25
	+ TBL	87.11	88.57	82.14
	+ SCE & SEN & TIE & TBL	**88.88**	**89.94**	**84.14**
2000	Baseline	88.96	86.90	80.07
	+ SCE	90.29	89.80	83.17
	+ SEN	90.10	89.31	83.01
	+ TIE	91.15	91.10	84.40
	+ TBL	89.71	90.46	84.29
	+ SCE & SEN & TIE & TBL	**91.53**	**91.86**	**86.34**
5000	Baseline	91.24	88.37	82.11
	+ SCE	92.60	91.32	85.29
	+ SEN	92.41	90.82	85.13
	+ TIE	93.48	92.64	86.54
	+ TBL	92.01	91.99	86.44
	+ SCE & SEN & TIE & TBL	**93.88**	**93.41**	**88.54**
All	Baseline	93.49	89.57	83.23
	+ SCE	94.88	92.56	86.45
	+ SEN	94.69	92.05	86.29
	+ TIE	95.79	93.90	87.73
	+ TBL	94.28	93.24	87.62
	+ SCE & SEN & TIE & TBL	**96.19**	**94.68**	**89.75**

The significance bold indicates the lowest F1-score value

4.6 Hyperparameter Impact Analysis

To analyze the impact of the hyperparameter on model performance, we conducted a series of experiments on the ConLL-2003, NCBI-disease, and JNLPBA datasets. In each experiment, we varied the hyperparameter w from 0.1 to 0.9 in steps of 0.1, while keeping other parameters fixed at the values described in our experimental settings. Essentially, the value of w controls the importance of the two types of tasks: sentence-level and token-level classification tasks. The model performance under different task weights is depicted in Fig. 3. We found that when w is between 0.2 and 0.4, our model achieved the best performance across all datasets. This suggests that token-level prediction tasks play a more significant role than sentence-level prediction tasks. Additionally, we observed that the model's performance decreases as w increases. Therefore, finding a good balance between these two tasks is crucial. Moreover, the sensitivity of ECTTLNER to the parameter w decreases with the increase in training data.

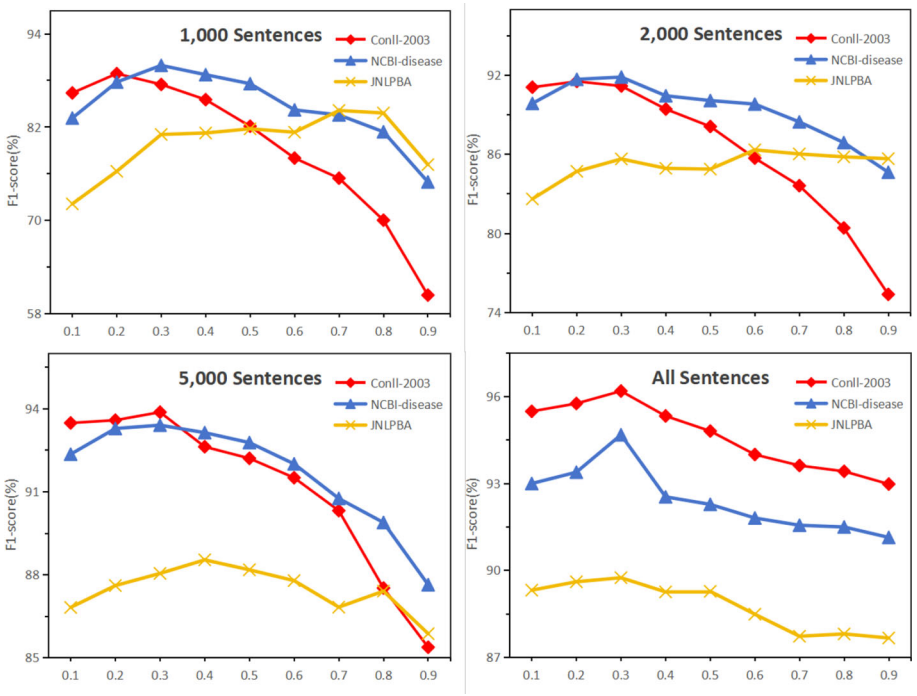

Fig. 3 The impact of hyperparameter under different resource scenarios

4.7 Error Analysis

Although ECTTLNER has demonstrated consistent top performance across different datasets, it is not immune to misclassification errors. This section offers an error analysis of our method, with the aim of investigating the types of errors and their prevalence in our model's output. Similar to the approach in [42], we categorize the errors into five types:

No annotation: The error where the model predicted tokens as entities, even though the tokens were not annotated as such.

No extraction: The error where the model fails to predict entities that are annotated.

Wrong range: The error where the model incorrectly predicts the spans of entities.

Wrong tag: The error where the model predicted tokens as an entity with the right span but the wrong tag type.

Wrong range and tag: The error where the predicted tokens were wrong in both span and tag type.

To conduct an error analysis of our model, we utilized the test set from ConLL-2003, which consists of 3453 sentences. Table 6 illustrates the distribution of these different types of errors.

From Table 6, it is evident that the error type with the highest rate observed is *Wrong tag*. This indicates that the model extracts tokens with the correct span as entities, but the tag type is incorrect. Furthermore, we also observed that the error type with the lowest rate is *Wrong range*, signifying the model's capability in detecting entities with the correct span. These two observations can be explained by the fact that token-level boundary labels from

Table 6 Summary of errors

Error type	Number	Rate (%)
No annotation	120	11.67
No extraction	198	19.26
Wrong range	84	8.17
Wrong tag	374	36.38
Wrong range and tag	252	24.52
All errors	1028	100

TBL incorporate the correct span information, while the semantic information of words may be more vague in the representation of the shared encoder.

To reduce the error rate of *Wrong tag*, we believe that character embeddings may be the solution. These embeddings can enhance the model's expressive power, especially when dealing with morphologically rich languages, enabling better recognition of complex entities and thereby helping to improve the model's performance.

5 Conclusion

We proposed a simple and effective cross-task transfer learning NER method called ECTTL-NER, which introduces SBE, SEN, TIE, and TBL prediction tasks in a multi-task learning framework. The experimental results on the ConLL-2003, NCBI-disease, and JNLPBA datasets show that the ECTTLNER method is more effective than a set of state-of-the-art baseline models and is more robust in low-resource scenarios. In addition, the performance of each auxiliary task and their importance were analyzed in depth across the three datasets. Furthermore, an analysis of the influence of hyperparameter was conducted on these three datasets. Finally, we conducted an error analysis of our model. Our work provides a new perspective on advancing cross-task transfer learning in NER, emphasizing the importance of leveraging multiple token-level information in low-resource scenarios. By introducing data augmentation methods, we demonstrate how to effectively enhance the performance and robustness of NER models. We believe that these methods not only improve the effectiveness of low-resource NER but also offer new ideas for future research, propelling further advancements in the field of NER. In future work, will explore effective methods for generating vectorized representations to jointly handle multiple natural language processing tasks.

Acknowledgements This work was conducted without any financial support.

Author Contributions Yiwu Xu: Conceptualization, methodology, software, Writing–original draft. Yun Chen: Validation, writing–review & editing, supervision.

Data Availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Li Q, Ji H (2014) Incremental joint extraction of entity mentions and relations. In: Proceedings of the 52nd annual meeting of the association for computational linguistics, (ACL 2014), pp 402–412
2. Zhong Z, Chen D (2010) A frustratingly easy approach for entity and relation extraction. ar**v preprint ar**v, 2010.12812
3. Li X, Feng J, Meng Y et al (2020)A unified MRC framework for named entity recognition. In: Meeting of the association for computational linguistics, (ACL 2020)
4. Ma X, Hovy E (2016) End-to-end sequence labeling via bi-directional lstm-cnns-crf. ar**v preprint ar**v, 1603.01354
5. Han X, Sun L, Zhao J (2011) Collective entity linking in web text: a graph-based method. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, (SIGIR 2011), pp 765–774
6. Lin Y, Shen S, Liu Z, et al (2016) Neural relation extraction with selective attention over instances. In: Proceedings of the 54th annual meeting of the association for computational linguistics, (ACL 2016), pp 2124–2133
7. Dong L, Wei F, Zhou M, et al (2015) Question answering over freebase with multi-column convolutional neural networks. In: Proceedings of the 53rd Annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (ACL-ICNLP 2015), pp 260–269
8. Huang Z, Xu W, Yu K (2015) Bidirectional LSTM-CRF models for sequence tagging. ar**v preprint ar**v, 1508.01991
9. Lample G, Ballesteros M, Subramanian S, et al (2016) Neural architectures for named entity recognition. ar**v preprint ar**v, 1603.01360
10. Chiu JPC, Nichols E (2016) Named entity recognition with bidirectional LSTM-CNNs. Trans Assoc Comput Linguist 4:357–370
11. Akbik A, Blythe D, Vollgraf R (2018) Contextual string embeddings for sequence labeling. In: Proceedings of the 27th international conference on computational linguistics, (COLING 2018), pp 1638–1649
12. Devlin J et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arxiv preprint arxiv, 1810.04805
13. Brown T et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901
14. Lewis M et al (2019) Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arxiv preprint arxiv, 1910.13461
15. Xie J, Yang Z, Neubig G et al (2018) Neural cross-lingual named entity recognition with minimal resources. In: Proceedings of EMNLP, pp 369–379
16. Ni J, Dinu G, Florian R (2017) Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. In: Proceedings of ACL, pp 1470–1480
17. Wu Q, Lin Z, Wang G et al (2020) Enhanced meta-learning for cross-lingual named entity recognition with minimal resources. Proc AAAI 34:9274–9281
18. Bari MS, Joty S, Jwalapuram P (2020) Zero-resource cross-lingual named entity recognition. Proc AAAI 34:7415–7423
19. Liu Z, Winata GI, Fung P (2020) Zero-resource cross-domain named entity recognition. In: Proceedings of the 5th workshop on representation learning for NLP, pp 1–6
20. Jia C, Liang X, Zhang Y (2019) Cross-domain NER using cross-domain language modeling. In: Proceedings of ACL, pp 2464–2474
21. Lin BY, Lu W (2018) Neural adaptation layers for cross-domain named entity recognition. In: Proceedings of EMNLP, pp 2012–2022
22. Wang Z, Qu Y, Chen L, et al (2018) Label-aware double transfer learning for cross-specialty medical named entity recognition. In: Proceedings of NAACL, pp 1–15

23. Kruengkrai C, Nguyen TH, Aljunied SM, et al (2020) Improving low-resource named entity recognition using joint sentence and token labeling. In: Proceedings of ACL, pp 5898–5905
24. Sanh V, Wolf T, Ruder S (2019) A Hierarchical multi-task approach for learning embeddings from semantic tasks. In: Proceedings of AAAI
25. Yoon W, So CH, Lee J et al (2019) Collabonet: collaboration of deep neural networks for biomedical named entity recognition. BMC Bioinform 20(10):55–65
26. Mo Y et al (2023) Multi-task transformer with relation-attention and type-attention for named entity recognition. In: Proceedings of ICASS
27. Tong Y, Chen Y, Shi X (2021) A multi-task approach for improving biomedical named entity recognition by incorporating multi-granularity information. In: Findings of the association for computational linguistics: ACL-IJCNLP 2021, pp 4804–4813
28. Savova GK et al (2010) Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. J Am Medical Inform Assoc 17(5):507–513
29. Etzioni O et al (2005) Unsupervised named-entity extraction from the web: an experimental study. Artif Intell 165(1):91–134
30. Zhang S, Elhadad N (2013) Unsupervised biomedical named entity recognition: experiments with clinical and biological texts. J Biomed Inform 46(6):1088–1098
31. Makino T, Yoshihiro O, Jun'ichi T (2002) Tuning support vector machines for biomedical named entity recognition. In: Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain
32. Krishnan V, Manning CD (2006) An effective two-stage model for exploiting non-local dependencies in named entity recognition. In: Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics
33. He Y, Luo C, Binyao Hu (2015) A geographic named entity recognition method based on the combination of CRF and rules. Comput Appl Softw 32:179–185
34. Park C, Jeong S, Kim J (2023) ADMit: improving NER in automotive domain with domain adversarial training and multi-task learning. Expert Syst Appl 225:120007
35. Beltagy I, Kyle L, Arman C (2019) SciBERT: a pretrained language model for scientific text. arxiv preprint arxiv, 1903.10676
36. Sang EF, De Meulder F (2003) Introduction to the ConLL-2003 shared task: Language-independent named entity recognition. arxiv preprint cs/0306050
37. Doğan RI, Leaman R, Lu Z (2014) NCBI disease corpus: a resource for disease name recognition and concept normalization. J Biomed Inform 47:1–10
38. Collier N, Ohta T, Tsuruoka Y, et al (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). pp 73–78
39. Ruder S (2017) An overview of multi-task learning in deep neural networks. arxiv preprint arxiv:1706.05098
40. Dai D, Zhang G, Li S, et al (2024) CSE-NER: A Category Semantic Enhanced Multi-Task Learning Framework for Named Entity Recognition. In: 2024 5th international seminar on artificial intelligence, networking and information technology (AINIT). IEEE, pp 1730–1735
41. Zhao Y, Ren J (2024) leveraging multi-level semantic understanding in a unified NER model. IEEE Access
42. Ichihara M, Komiya K, Iwakura T et al (2015) Error analysis of named entity recognition in bccwj. ReCALL 61:2641