



# Chinese named entity recognition: The state of the art

Pan Liu<sup>a</sup>, Yanming Guo<sup>a,\*</sup>, Fenglei Wang<sup>b</sup>, Guohui Li<sup>a</sup>

<sup>a</sup> College of Systems Engineering, National University of Defense Technology, Changsha, China

<sup>b</sup> College of Politics, National Defense University, Shanghai, China

## ARTICLE INFO

### Article history:

Received 6 February 2021

Revised 1 September 2021

Accepted 29 October 2021

Available online 9 November 2021

Communicated by Zidong Wang

### Keywords:

CNER

Character representation

Context encoder

Tag decoder

Attention mechanism

Adversarial transfer learning

## ABSTRACT

Named Entity Recognition (NER), one of the most fundamental problems in natural language processing, seeks to identify the boundaries and types of entities with specific meanings in natural language text. As an important international language, Chinese has uniqueness in many aspects, and Chinese NER (CNER) is receiving increasing attention. In this paper, we give a comprehensive survey of recent advances in CNER. We first introduce some preliminary knowledge, including the common datasets, tag schemes, evaluation metrics and difficulties of CNER. Then, we separately describe recent advances in traditional research and deep learning research of CNER, in which the CNER with deep learning is our focus. We summarize related works in a basic three-layer architecture, including character representation, context encoder, and context encoder and tag decoder. Meanwhile, the attention mechanism and adversarial-transfer learning methods based on this architecture are introduced. Finally, we present the future research trends and challenges of CNER.

© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Named Entity Recognition (NER) is a fundamental information extraction task and plays an essential role in natural language processing (NLP) applications such as information retrieval, automatic text summarization, question and answering, machine translation, knowledge graphs. The goal of NER is to extract some predefined specific entities from sentences and identify their correct types, such as person, location, organization. Fig. 1 illustrates that a NER system recognizes two entities from the given sentence.

Early NER methods can be divided into two types: Rule-based methods and statistical-based methods. The rule-based methods refer to match named entities by manually designing massive rules of a specific field according to tasks, which are laborious, and limit their generalization to other fields. The statistical-based methods convert the NER task into a sequence labeling task and use the artificially labeled corpora for training. As the cost of labeling is much lower than the cost of designing rules, statistical-based methods are versatile and do not require too many hand-designed rules and gradually become the mainstream methods before the outbreak of deep learning. In the CoNLL-2003 conference, all of the 16 NER systems that participated in the competition adopt statistical methods [1].

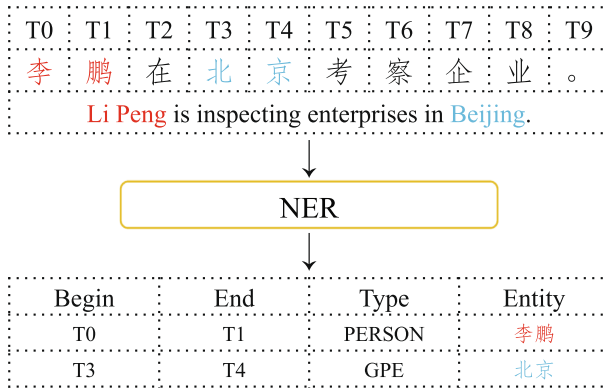
In recent years, deep learning has emerged as a powerful strategy for learning feature representations directly from data and has led to remarkable breakthroughs in the NLP field. When applied in NER, deep learning can learn intricate hidden representations without complex feature engineering and rich domain knowledge. So deep learning-based methods have overwhelmingly surpassed traditional rule-based methods and statistical-based methods in NER.

Due to the differences between languages, there are also many pieces of research on NER methods for specific languages, and many notable NER surveys have been published, as summarized in Table 1, including a wide variety of languages, such as English, Arabic, Hindi and other languages, but not Chinese. Many surveys are universal to every language, but these researches mainly focus on English NER (ENER) and do not report the progress of NER in other languages with their characteristics. Chinese is also an important international language widely used all over the world. Compared with English, there have been many methods specifically applicable to Chinese according to its particularity. Especially in terms of feature representation, the unique radicals, strokes, and glyphs of Chinese characters are introduced into CNER models as input features to improve the performance. As far as we know, there is no review in CNER yet, and our paper is the first one to systematically summarize the latest development of CNER based on some characteristics of Chinese.

The structure of the paper is organized as follows. Section 2 introduces the preliminary knowledge, including datasets, tag

\* Corresponding author.

E-mail addresses: [liupan09@nudt.edu.cn](mailto:liupan09@nudt.edu.cn) (P. Liu), [guoyanming@nudt.edu.cn](mailto:guoyanming@nudt.edu.cn) (Y. Guo), [wangfenglei@nudt.edu.cn](mailto:wangfenglei@nudt.edu.cn) (F. Wang), [guohli@nudt.edu.cn](mailto:guohli@nudt.edu.cn) (G. Li).



**Fig. 1.** An illustration of NER task. The sample sentence is from People's daily dataset, and GPE means Geo-Political Entity.

schemes, evaluation metrics and the difficulties of CNER. Section 3 introduces the traditional methods of CNER. Section 4 summarizes the deep learning-based CNER researches, compares the characteristics, advantages, and disadvantages of different methods, and concludes these researches in a unified table. Section 5 discusses possible future research trends and challenges of CNER. Section 6 concludes this paper.

## 2. Preliminary knowledge

In this section, we introduce some important preliminary knowledge related to CNER, including datasets, tag schemes, evaluation metrics and difficulties of CNER.

### 2.1. Datasets

According to the source and the availability of the datasets, we divide the CNER datasets into three types: public datasets, competition datasets, and private datasets. The public datasets refer to the commonly used academic benchmark datasets, which are publicly available and frequently used in academic researches. The competition datasets mainly include several popular medical datasets, which are generally employed in medical competitions and only accessible to their participants. The private datasets are collected by individuals for their specific applications, and normally not released to the public.

#### 2.1.1. Public Datasets

Table 2 collects some public datasets commonly used in CNER and lists the entity types of each dataset<sup>1</sup>. These public datasets include corpora from different sources such as social media (WEIBO), electronic resumes (RESUME), news (People's Daily), companies (bosonNLP), among which the WEIBO and MSRA [29] are the most widely used in CNER, thus in Section 4, we evaluate recent advances mainly based on these two datasets. Based on the open-source text classification dataset THUCTC of Tsinghua University, the CLUE organization selects some data for NER, then releases the CLUENER2020 [30] dataset. It contains 10 different entity types and has fulfilled multiple baseline model evaluations, and is expected to be a versatile CNER dataset in the future.

Compared with ENER, CNER receives relatively less attention, thus the datasets of CNER are still insufficient. Li et al. [7] lists 22 various public ENER datasets, with the number of entity types ranging from 1 to 505, while the number of entity types in CNER

**Table 1**

Existing NER surveys of languages. 'Universal' means the survey dose not specify a certain language.

Language	Ref.	Year	Topic
Universal	[1]	2007	A survey of named entity recognition and classification
Universal	[2]	2008	Named entity recognition approaches
Universal	[3]	2013	Techniques for named entity recognition: a survey
Universal	[4]	2018	An overview of named entity recognition
Universal	[5]	2018	Recent named entity recognition and classification techniques: a systematic review
Universal	[6]	2019	A survey on named entity recognition
Universal	[7]	2020	A survey on deep learning for named entity recognition
Universal	[8]	2020	A survey of named-entity recognition methods for food information extraction
Arabic	[9]	2009	NERA: Named entity recognition for Arabic
Arabic	[10]	2014	A survey of Arabic named entity recognition and classification
Arabic	[11]	2015	Named entity recognition for Arabic social media
Arabic	[12]	2016	Arabic Named Entity Recognition—A Survey and Analysis
Arabic	[13]	2017	A comparative review of machine learning for Arabic named entity recognition
Arabic	[14]	2019	Arabic named entity recognition using deep learning approach
Arabic	[15]	2019	Arabic named entity recognition: What works and what's next
Indian	[16]	2010	A survey of named entity recognition in English and other Indian languages
Indian	[17]	2011	A survey on named entity recognition in Indian languages with particular reference to Telugu
Indian(Assamese)	[18]	2014	A survey of named entity recognition in Assamese and other Indian languages
Indian(Hindi)	[19]	2016	Survey of named entity recognition systems with respect to Indian and foreign languages
Indian	[20]	2017	Survey of named entity recognition techniques for various Indian regional languages
Indian(Hindi)	[21]	2019	Named entity recognition for Hindi language: A survey
Indian	[22]	2019	Named entity recognition: A survey for Indian languages
Indian(Hindi)	[23]	2020	A survey on various methods used in named entity recognition for hindi language
English	[24]	2013	Named entity recognition in english using hidden markov model
Marathi	[25]	2016	Issues and Challenges in Marathi Named Entity Recognition
Turkish	[26]	2017	Named entity recognition in Turkish: Approaches and issues
Spanish	[27]	2020	Named entity recognition in Spanish biomedical literature: Short review and bert model

<sup>1</sup> Some entity types of OntoNotes Release 5.0 are abbreviations, you can get their exact meaning in reference [28]

**Table 2**  
Public datasets of CNER. ‘#Tags’ refers to the number of entity types.

Corpus	#Tags	Entity types	URL
WEIBO	4	Person, Location, Organization and Geo-political	<a href="https://github.com/hltcoe/golden-horse">https://github.com/hltcoe/golden-horse</a>
MSRA	3	Person, Location, Organization	<a href="https://github.com/InsaneLife/ChineseNLPCorpus/tree/master/NER/MSRA">https://github.com/InsaneLife/ChineseNLPCorpus/tree/master/NER/MSRA</a>
People’s Daily	4	Person, Organization, Geo-political, Date	<a href="https://github.com/Guocail/nlp_corpus/tree/main/open_ner_data/people_daily">https://github.com/Guocail/nlp_corpus/tree/main/open_ner_data/people_daily</a>
bosonNLP	6	Person, Location, Organization, Company, Product, Time	<a href="https://github.com/InsaneLife/ChineseNLPCorpus/tree/master/NER/boson">https://github.com/InsaneLife/ChineseNLPCorpus/tree/master/NER/boson</a>
RESUME	8	Person, Location, Organization, Country, Education, Profession, Race, Title	<a href="https://github.com/Guocail/nlp_corpus/tree/main/open_ner_data/ResumeNER">https://github.com/Guocail/nlp_corpus/tree/main/open_ner_data/ResumeNER</a>
OntoNotes Release 5.0	18	Person, NORP, Facility, Organization, GPE, Location, Product, Event, Work of art, Law, Language, Date, Time, Percent, Money, Quantity, Ordinal, Cardinal	<a href="https://doi.org/10.35111/xmhb-2b84">https://doi.org/10.35111/xmhb-2b84</a>
CLUENER 2020	10	Address, Book, Company, Game, Government, Movie, Name, Organization, Position, Scene	<a href="https://github.com/CLUEbenchmark/CLUENER2020">https://github.com/CLUEbenchmark/CLUENER2020</a>

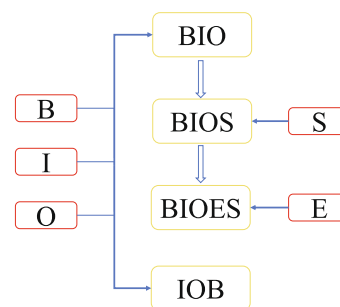
datasets is from 3 to 18. Enlarging the quantity and the diversity of the CNER dataset would greatly facilitate relevant research. In addition, the CNER datasets are also suffer from the labeling ambiguity problem, i.e. the definition of entity types may vary between datasets. For example, the entity ‘Beijing’ belongs to the ‘GPE’ type in WEIBO while the ‘Location’ type in MSRA. Developing a unified annotation scheme is another way to improve the quality of datasets and promote research on CNER.

### 2.1.2. Competition Datasets

The competition datasets are datasets released to participants of some competitions. With the increasing demand for NER in the medical field, there have emerged some related medical competitions and datasets.

Natural language documents in the medical field, such as medical textbooks, medical encyclopedias, clinical cases, medical journals, admission records, test reports, contain much medical expertise and terminology. The core to ‘understand’ medical data is to extract critical information from the medical text accurately. For a given set of pure medical text documents, the goal of the NER is to identify entities related to clinical medical practice and classify them into predefined types, such as recognizing diseases and symptoms. Combining entity recognition technology with medical professional fields and using machines to read medical texts can significantly improve the efficiency and quality of clinical scientific research and serve downstream subtasks. Those Chinese medical competitions demonstrate the urge requirements of NER in medical fields.

However, due to privacy or property rights protection issues, the medical NER corpora, especially the Chinese medical NER corpora, is particularly scarce. Several evaluation competitions only



**Fig. 2.** Evolution of four commonly used tag schemes.

disclose the datasets for the competitions, such as CHIP2020<sup>2</sup> and CCKS2020<sup>3</sup>, and there are no other officially authorized way for these datasets. Specifically, the datasets for the medical NER task of the annual China Conference on Knowledge Graph and Semantic Computing (CCKS) have received much research, and most of the researches on Chinese medical NER are based on CCKS datasets.

### 2.1.3. Special Datasets

Some researchers have constructed their own NER data sets to fulfill some specific tasks. For example, in order to study the NER of Chinese scenic spots, Zhao et al. [31] used crawlers to obtain more than 10,000 travel blogs from the Internet, then cleaned and labeled these data. Authorized from a well-known hospital in Hunan Province, Gao et al. [32] designed a semi-automatic method to construct a Chinese medical NER dataset, including 255 admission records and 9 types of medical entity.

Motivated by the observation that if one entity name is replaced by another entity with the same type, then the new sentence is usually correct in grammar and semantic, Wu et al. [33] proposed a method to generate pseudo labeled data. This approach can act as a general strategy to augment corpora and improve performance.

### 2.2. Tag Schemes

In machine learning, the task of NER is transformed into a task of sequence labeling. A tag scheme is a way to tag tokens (a token is usually a word in English or a character in Chinese), which can uniquely determine the type and position of entities in the sentence. The NER models output the corresponding tag sequence of the input sentence, then the tag sequence can be converted into entities with boundaries and types according to the tag schemes, so the entities are extracted.

In the existing tag schemes, B (begin) represents the first character of the entity, I (inside) represents the middle character of the entity, and E (end) represents the last character of the entity. If the entity is just one character, it is represented by S (singleton). If the character does not belong to any entity, it is represented by O (outside). A tag scheme can include some or all of the tags above. Fig. 2 shows the evolution of four commonly used tag schemes:

- IOB scheme. B is only used when two or more entities of the same type appear consecutively. If an entity is just preceded by another entity of the same type, its first token is tagged with B. Other tokens of entities are all tagged with I, and tokens not belonging to any entities are tagged with O.
- BIO scheme. The first character of all entities is tagged with B, and the subsequent tokens are tagged with I. Those tokens not in entities are tagged with O.

<sup>2</sup> <http://www.cips-chip.org.cn>

<sup>3</sup> <http://sigkg.cn/ccks2020/>

**Table 3**  
An example of IOB, BIO, BIOS, BIOES tag schemes.

Token	早	产	儿	肾	功	能	尚	不	成
IOB	O	O	O	I-bod	O	O	O	O	O
BIO	O	O	O	B-bod	O	O	O	O	O
BIOS	O	O	O	S-bod	O	O	O	O	O
BIOES	O	O	O	S-bod	O	O	O	O	O
Token	熟	。	葡	萄	糖	肾	国	较	低
IOB	O	O	I-bod	I-bod	I-bod	B-bod	O	O	O
BIO	O	O	B-bod	I-bod	I-bod	B-bod	O	O	O
BIOS	O	O	B-bod	I-bod	I-bod	S-bod	O	O	O
BIOES	O	O	B-bod	I-bod	E-bod	S-bod	O	O	O
Token	。	易	出	现	糖	尿	。		
IOB	O	O	O	O	I-dis	I-dis	O		
BIO	O	O	O	O	B-dis	I-dis	O		
BIOS	O	O	O	O	B-dis	I-dis	O		
BIOES	O	O	O	O	B-dis	E-dis	O		

- BIOS scheme. If an entity is just one character, it is represented by S (singleton). Other tokens are tagged the same as BIO Scheme.
- BIOES scheme. If an entity includes at least two characters, its last character is tagged with E. Other tokens are tagged the same as BIOS Scheme.

Table 3 uses a sentence in CCKS2020 as an example to show the difference of IOB, BIO, BIOS, BIOES tag schemes. In this example, 'bod' means body and 'dis' means disease, we mark the 'bod' entity in red and 'dis' entity in blue. The number and label between '|||' indicate the location and type of entities in the sentence. For example, '|||11 13 bod|||' means from the start index 11 to end index 13, the characters '葡萄糖' (glucose) form a 'bod' entity. As shown in this table, the sentence is tagged differently with these four tag schemes. It is worth noting that the tag schemes could also affect the NER performance. Reimers et al. [34] compared IOB, BIO, BIOS tag schemes and showed that the IOB scheme performed worse than the BIO and BIOS schemes in NER tasks.

### 2.3. Evaluation Metrics

F-score is a metric used in statistics to measure the accuracy of classification models. For an unbalanced sample distribution, it is insufficient to simply use precision to measure the quality of the model. The F-score jointly considers the precision and recall rate of the classification model and gets their weighted harmonic average. F-score ranges between 0 and 1, and higher score indicates better performance.

Table 4 categorizes the prediction according to whether the entity predicted by the NER model appears in the ground truth. For each predefined entity type  $i$ , the results can be divided into four categories: True Positive ( $TP_i$ ), False Positive ( $FP_i$ ), False Negative ( $FN_i$ ), and True Negative ( $TN_i$ ).

Precision refers to the percentage of named entities correctly identified by the NER system to all entities returned by NER systems. Recall rate refers to the percentage of named entities correctly identified by the NER system to ground truth. For each type  $i$ , the precision and recall of type  $i$  are:

**Table 4**  
Prediction categorizations.

	Entities in ground truth	Entities not in ground truth
Entities returned by NER	True Positives ( $TP_i$ )	False Positives ( $FP_i$ )
Entities not returned by NER	False Negatives ( $FN_i$ )	True Negatives ( $TN_i$ )

**Table 5**  
Calculation of macro-average and micro-average precision, recall and  $F_1$ -score.

Macro average	Micro average
$Precision_{macro} = \frac{1}{n} \sum_{i=1}^n Precision_i$	$Precision_{micro} = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FP_i}$
$Recall_{macro} = \frac{1}{n} \sum_{i=1}^n Recall_i$	$Recall_{micro} = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FN_i}$
$macro - F_1 = 2 \times \frac{Precision_{macro} \times Recall_{macro}}{Precision_{macro} + Recall_{macro}}$	$micro - F_1 = 2 \times \frac{Precision_{micro} \times Recall_{micro}}{Precision_{micro} + Recall_{micro}}$

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad Recall_i = \frac{TP_i}{TP_i + FN_i}$$

The meaning of  $F_\beta$ -score is to combine the two scores of precision and recall into one score. In the process of merging, the weight of the recall is a  $\beta$  times of the precision.

$$F_\beta = (1 + \beta^2) \times \frac{Precision \times Recall}{\beta^2 \times Precision + Recall}$$

$F_1$ -score, also known as balanced F-score, is defined as the harmonic average of precision and recall and widely used in NER systems.

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

In addition to  $F_1$ -score,  $F_{0.5}$ -score and  $F_2$ -score are also widely used in statistics. In the  $F_2$ -score, the weight of recall is higher than precision, and in the  $F_{0.5}$ -score, the weight of precision is higher than recall.

As a multi-classification task, the sequence labeling task transformed by NER usually uses the exact matching macro-average  $F_1$  score and micro-average  $F_1$  score as evaluation metrics. Exact matching means that the boundary and type of the entity are judged correctly at the same time. Macro-average considers each type separately and then performs an arithmetic average of all types to calculate the macro-average precision and recall, and then macro  $F_1$ -score is calculated by macro precision and recall. In comparison, micro-average considers all types at once to calculate the overall micro precision, recall and  $F_1$ -score. Table 5 compares the calculation of macro-average and micro-average precision, recall and  $F_1$ -score. If the distribution of entity types in the dataset is unbalanced, the macro average does not consider the sample size of each type, and the measurement will be biased, so the micro average is more preferred in practical applications.

### 2.4. Difficulties of CNER

Compared with the ENER, CNER suffers tremendous challenges caused by the characteristics of Chinese.

Chinese word boundaries are vague A word is a basic unit with a complete meaning. The most apparent characteristic of Chinese is that the word boundaries are vague, and there is no delimiter to indicate word boundary. In English, there are separators between words to identify boundaries, and each word has a complete meaning. While in Chinese, the character in Chinese can be regarded as a concept between character and word in English. Chinese characters have more semantics than English characters and fewer semantics than words. Some Chinese characters have their independent meanings, but more Chinese characters need to be combined with others to form a meaningful word. Chinese characters are used as basic units in text without clear word separators, vague word boundaries will cause a lot of boundary ambiguity and increase the difficulty of defining the boundaries of Chinese named entities. Therefore, the word boundary information is essential in Chinese, and there are many methods of combining lexicon information in CNER tasks.



Chinese named entities lack obvious morphological features. In English, the first letter of some specified types of entities is usually capitalized, such as the names of specified persons or places. This kind of information is an explicit clue to identify the location and boundary of some named entities. However, it lacks this explicit feature of morphology in Chinese, which increases the difficulty of recognition.

The structure of Chinese named entities is complicated. Chinese named entities are more complicated in composition than English named entities. This complexity is particularly obvious in the names of Chinese persons, places, and organizations. For example, the transliterated persons' names of ethnic minorities and foreigners have different lengths, and there is no uniform word-formation standard, such as '列夫·托尔斯泰' (Lev Tolstoy), '格买提·热合曼' (Nigmat Rheman). Place names have a large number of overlapping inclusions with names of people and organizations, such as '开慧镇' (Kaihui Town), '中山市' (Zhongshan City), in which 'Kaihui' and 'Zhongshan' are people's names. There might exist many nests, aliases, abbreviations and other problems in organization names, such as '湖南驻京办事处' (Hunan Office in Beijing), in which '京' (Jing) is the abbreviation of '北京' (Beijing). To label these named entities correctly, it is often necessary to analyze the semantics of the context.

The rapid emergence of new entities. Many Chinese characters have given rise to new meanings with the rapid growth of Internet information. For example, '阿里巴巴' (Alibaba) originally refers to a character in Arabic stories, but now it can also refer to a listed company. The two Chinese characters '抖' (tremble) and '音' (sound) do not initially constitute a word, but the APP '抖音' (TikTok) is now widely accepted by people. In particular, various Internet terms are emerging in an endless stream, and there are many words that we have never heard. Though new entities emerge in English, it is easy for word segmentation. However, word formation is flexible in Chinese, and it is hard to get the boundaries of new entities, so the emergence of these new entities brings more difficulty for CNER.

### 3. History: Traditional Methods of CNER

Similar to the traditional ENER method, there are mainly two categories of traditional CNER methods: rule-based methods and statistical-based methods.

#### 3.1. Rule-based Methods

Rule-based methods select the matched entities from the text according to some matching rules, which are mainly based on regular expressions or dictionaries. Regular expressions are formed by predefined specific characters and combinations of these specific characters to express a filtering logic for strings or text, and dictionaries are established by collections of entities. The matching rules can be manually designed. For example, exact matching requires that each character of the matched entity meet the rule, and fuzzy matching requires that the similarity between the matched entity and the rule is higher than a certain threshold. Fig. 3 gives an illustration of rule-based methods. To extract the person's name '李鹏' (Lipeng), the method based on regular expression may use the formulation 'Last name + First name', while the method based on dictionary should build a vocabulary which contains the name '李鹏'.

In the early research on NER, it is classic to construct hand-crafted rules and then search for strings matching these rules from

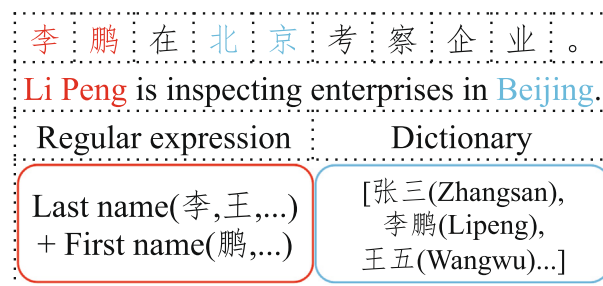


Fig. 3. An illustration of rule-based methods. A person's name is matched by a regular expression and a dictionary.

the text [35]. In this case, the quantity and quality of the matching rules are quite essential for the final performance, and a satisfactory result normally requires thousands of handcrafted rules with specialized knowledge, which is very labor-intensive and time-consuming. To this end, researchers tried to discover and generate rules with the help of machines automatically. The most representative work is the deep learning-CoTrain method proposed by Collins et al. [36]. They first predefined the seed rule set DecisionList, then obtained more rules by applying unsupervised training iterations on the rule set according to the corpus. The final rule set was used for the recognition of named entities. Similarly, Cucerzan et al. [37] proposed a method of automatically generating rules using Bootstrapping. There are also researches attempt to combine the advantages of rules and statistical methods, Mikheev et al. [38] proposed a NER system that combined rules and statistical models and believed that with the addition of statistical models, place names can still be well recognized without using gazetteers.

Rule-based methods can achieve good performance on a specific corpus. However, the better the recognition result, the more rules are needed to be formulated, and the lower feasibility of manually formulating these rules. Moreover, trying to recognize the endlessly changing named entities by formulating limited rules becomes more and more cumbersome, not to mention the extreme dependence of rules on domain knowledge. So that when the domains are very different, the established rules often cannot be transplanted and have to be re-made. These inherent shortcomings led researchers to attempt new research ideas and gradually turn to machine learning for help.

#### 3.2. Statistical-based Methods

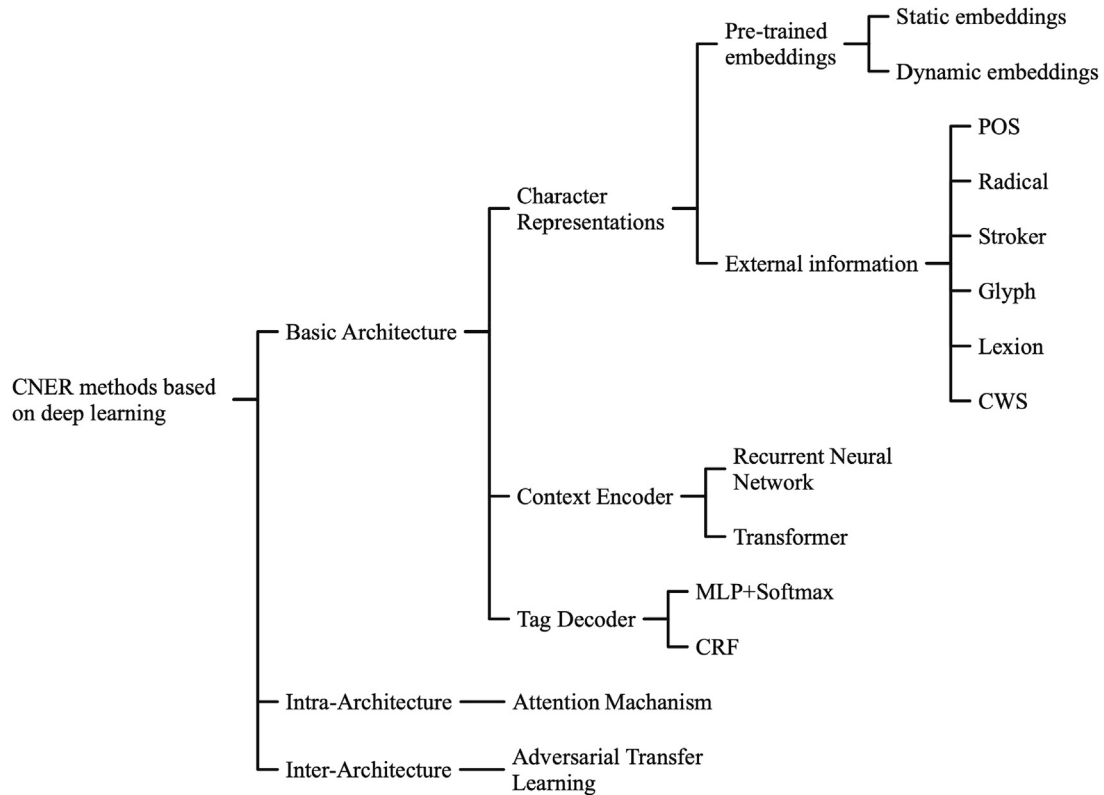
Statistical-based methods convert the CNER task into a sequence labeling task. Given annotated corpora, features are carefully designed to represent each character. By training statistical-based machine learning models on these corpora, each character in the text is serialized and automatically labeled by the trained model. Then the automatically labeled sequence can be decoded according to the tag scheme, and the named entities composed of several characters in the text can be integrated. Statistical-based machine learning models learn to make predictions by training on example inputs and their expected outputs instead of human-created rules.

The classic statistical-based machine learning models have been successfully used to serialize and annotate named entities with good results, including Hidden Markov Model (HMM) [39–41], Maximum Entropy (ME) [42], Conditional Random Field (CRF) [43,44], Support Vector Machine (SVM) [45], and etc. Because the classic NER models are mostly proposed for English texts and are not directly applicable to Chinese, adjusting and optimizing the classic models according to Chinese characteristics can help recognize named entities in Chinese text more effectively. There

**Table 6**

Differences between statistical-based methods and deep learning-based methods.

Statistical-based methods		Deep learning-based methods
Character Representations	Handcrafted features(orthographic, prefixes, suffixes, etc.)	Distributed representations(Word2vec, RNN, ELMo, BERT, etc.)
Machine learning models	Statistical-based models (HMM, ME, CRF, SVM, etc.)	Encoder(LSTM, GRU, Transformer, etc.) Decoder(CRF, Transformer, etc.)



**Fig. 4.** The taxonomy of CNER methods based on deep learning.

are some researches on model improvement in CNER, such as stacked Markov [46], multi-layer conditional random field [47,48].

In recent years, deep learning has become a boom in machine learning and has been used in NER systems, with significant performance improvements. The key advantage of deep learning is the capacity to obtain powerful representation learning and semantic analysis through vector representation and neural processing. Compared with linear models, deep learning-based models can learn more complex features from data through nonlinear activation functions and automatically discover the latent representation and processing required for classification or detection [49].

Deep learning-based NER methods also regard NER as a sequence labeling task, which is the subsequent development of statistical-based methods. Different from statistical-based methods, deep learning-based methods use distributed representations instead of handcrafted features to represent characters, and encoder-decoder structures instead of statistics-based learning models for learning. Table 6 compares the difference between statistical-based methods and deep learning-based methods.

#### 4. Nowadays: CNER Methods Based on Deep Learning

CNER methods based on deep learning have gradually become dominant and have achieved continuous performance improve-

ments. Fig. 4 gives the taxonomy of CNER methods based on deep learning. We follow the formulation in [7] and divide the deep learning-based CNER methods into a three-layer basic architecture: character representations, context encoder, and tag decoder. Then based on our observation, we introduce the intra-architecture attention mechanism and inter-architecture adversarial transfer learning. The details will be explained in the following sections.

##### 4.1. Character Representations

The character representations include embeddings and other effective representations of Chinese characters, which is the basic unit in Chinese. Chinese characters are usually treated like English words in natural language processing, and each Chinese character is treated as a token.

The function of the character representations is to map the tokens into a continuous space to facilitate subsequent calculations. Generally speaking, we need to vectorize the input before conducting various machine learning algorithms. One-hot encoding represents every token with a very long vector, and the length of the vector is the size of the dictionary. In the one-hot vector space, two different characters have orthogonal representations and cannot reflect the semantic relationship between tokens. Distributed representation can overcome the shortcomings of one-hot

representation. The basic idea of distributed representation is to map each token into a fixed-length short vector through training. All these vectors form a vector space, and each dimension of the space represents one latent feature, then each token can be regarded as a point in the space. The distributed representation is automatically learned from the text, and it can automatically capture the semantic and syntactic attributes of tokens so that the input characters are often transformed into distributed representations in NER.

#### 4.1.1. Pre-trained character embeddings

The character embeddings from Pre-Trained Models (PTMs) are the fundamental representations for deep learning-based CNER models. Like the word embeddings in English, Chinese character embeddings can be pre-trained on a large number of Chinese corpora. Qiu et al. [50] conducted a comprehensive overview of PTMs for NLP and classified PTMs into pre-trained word embeddings and pre-trained contextual encoders. Based on the classifications of PTMs, the pre-trained character embeddings can be classified into static embeddings and dynamic contextual embeddings.

Static embeddings are trained as lookup tables, and the embeddings of each character are fixed in the table, such as NNLM [51], Word2vec [52], FastText [53], Glove [54], etc. Dong et al. [55] used the CBOW model to train character embeddings on 1.02 GB corpus of Chinese Wikipedia, Wang et al. [56] trained character embeddings on 1.89 GB corpus of Sogou news, and Jia et al. [57] trained character embeddings of Chinese Wikipedia with the Glove model. There are also many works [58–67] used static character embeddings, but no mention of specific models. Static embeddings are non-contextual, and similar words will gather together. So the relationship between characters will also be embedded into the vector space, thus achieve the purpose of mapping from symbol space to vector space.

Dynamic contextual embeddings are also known as pre-trained language models, the representations generated by these models will change according to the context, such as ELMo [68], GPT [69], BERT [70], ERNIE [71], ALBERT [72], NEZHA [73], etc., among which BERT is the most commonly used. For a given character, BERT concatenated its character position embedding, sentence position embedding, and character embedding as input, then used a Masked Language Model (MLM) to perform deep bidirectional representation pre-training on input sentences, which could obtain robust contextual character embeddings. Table 7 compares the performance in some works with BERT. As can be seen, these works





Lexicon	朝阳 (morning sun)	明朝 (Ming Dynasty)	朝鲜半岛 (Korean Peninsula)	朝夕 (morning and evening)
Glyph	 Oracle Bone Script	 Bronze Script	 Clerical Script	 Regular Script
Radical	十 ( ten )	日 ( sun )	十 ( ten )	月 ( moon )
Stoker	一	丿 一 一	一	丿 丿 一 一

Fig. 5. The illustration of some external information of the character ‘朝’.

had brought great improvement of  $F_1$ -score after introducing BERT to their models.

Due to the complexity of models and parameters, pre-trained language models are usually pre-trained by large companies based on a large number of unsupervised corpora and open access to the public. As a result, these models can obtain rich prior information from large-scale corpora so that each character token can generate informative contextual embeddings through them.

There are some other attempts to get better character embeddings. For example, Xu et al. [77] designed another pre-trained language model called Conv-GRU. Conv-GRU passed the original character embeddings through a GRU layer and a convolution layer to obtain local context information and then concatenated the output embeddings with the original character embeddings to combine the semantic knowledge from both local context and long-term dependency together.

#### 4.1.2. External Information

Similar to word representations in ENER, Chinese character representations incorporate lots of external information to CNER, and many of them are particular information like strokes, radicals (roots), and glyphs in Chinese characters. Fig. 5 gives an illustration of some external information of the character Chinese ‘朝’ (morning). This character can be decomposed into 4 radicals that consist of 12 strokes in total. As to glyph information, we illustrate scripts from different historical periods, which are usually very different in shape, and help the model integrate pictographic information from various sources.

Next, we will introduce each external information in detail.

POS information Part of speech (POS) refers to dividing words into nouns, verbs, adjectives, adverbs, etc., according to their char-

Table 7  
Improvement brought by BERT in different works.

Work	Dataset	Model	F1(%)	Improvement(%)	Year
[63]	MSRA	Word2Vec + radical + BGRU-CRF	90.45	4.97	2019
		BERT + radical + BGRU-CRF	95.42		
[74]	MSRA	PLTE	93.26	1.27	2020
		PLTE[BERT]	94.53		
	Ontonotes	PLTE	74.60	6.00	
		PLTE[BERT]	80.60		
	Weibo	PLTE	55.15	14.08	
		PLTE[BERT]	69.23		
	[75]	MSRA	SoftLexicon(LSTM)	93.66	
SoftLexicon(LSTM)+BERT			95.42		
Ontonotes		SoftLexicon(LSTM)	75.64	7.17	
		SoftLexicon(LSTM)+BERT	82.81		
Weibo		SoftLexicon(LSTM)	61.42	9.08	
	SoftLexicon(LSTM)+BERT	70.50			
	[76]	CCKS2018	Word2Vec + CRF		69.01
BERT + CRF			90.54		
Word2Vec + BiLSTM-CRF			75.60	15.83	
BERT + BiLSTM-CRF			91.43		

**Table 8**  
The effect of POS and radical information.

Work	Dataset	Model	F1(%)	Improvement(%)	Year
[55]	MSRA	random + dropout	88.91	0.53	2016
[58]	CCKS2018	random + radical + dropout	89.44		
		LSTM-CRF	67.32	11.62	2019
		POS + LSTM-CRF	78.94		
		SM-LSTM-CRF	69.91	10.16	
		POS + SM-LSTM-CRF	80.07		
[60]	CCKS2017	BILSTM-CRF	88.78	Baseline	2019
		BILSTM-CRF + radical	89.64	0.86	
		BILSTM-CRF + POS	89.06	0.28	
		BILSTM-CRF + radical + POS	90.12	1.34	
		Att-BILSTM-CRF	90.11	Baseline	
		Att-BILSTM-CRF + radical	90.96	0.85	
		Att-BILSTM-CRF + POS	90.81	0.70	
		Att-BILSTM-CRF + radical + POS	91.35	1.24	
[61]	CCKS2017	CRF	85.14	1.87	2019
		POS + CRF	87.01		
		BILSTM-CRF	89.66	-0.11	
		POS + BILSTM-CRF	89.55		
	CCKS2018	CRF	82.49	0.93	
		POS + CRF	83.42		
		BILSTM-CRF	84.13	-0.17	
		POS + BILSTM-CRF	83.96		

acteristics. POS information is of great significance for NER. On the one hand, the entities are nouns, which can help determine whether a phrase is an entity. On the other hand, the POS of a character will also be affected by the POS of contextual characters, which can help indicate where the entity is. It should be noted that the tokens in Chinese are characters, while the POS information is about words. If a word consists of multiple characters, each character must use the same unique POS of this word for representation.

Radical Information Radicals of Chinese characters are the most basic semantics unit and also called roots in Chinese. The same radical in different characters usually have similar or identical meanings. For example, in the characters ‘抱’ (hug), ‘推’ (push), ‘打’ (hit), ‘扔’ (throw), they share the radical ‘扌’ (hand), which means that this character is related to the hand or hand movement. Therefore, the introduction of radicals can provide much semantic information for the model and help entity recognition. Shi et al. [78] presented the first piece of evidence on the feasibility and utility of radical embeddings for Chinese language processing. For one character with multiple radicals, Dong et al. [55] used bidirectional LSTM to extract the radical embeddings and then concatenated it with the character embeddings as the final character representation. Table 8 compares the effect of POS and radical information in different works. In most cases, POS and radical information could bring F1 improvement. For example, [58] conducted a deep learning model incorporating POS and self-matching attention for named entity recognition on CCKS2018 and got more than 10% improvement to the LSTM-CRF model and SM-LSTM-CRF model after introducing POS information. [60] jointly evaluated the effect

of radical and POS information, and found these external information could brought consistent improvement over the baseline. These improvement indicate that the POS and radical information is effective for CNER.

Stroke Information The strokes of Chinese characters are the most basic unit of writing. Although a single stroke has no clear meaning, the characters are written in sequences of strokes, containing some useful information. Stroke information can be extracted through CNN or RNN structure, just like the character-level information in English NER [79]. Luo et al. [80] proposed a Chinese electronic medical record entity recognition method based on stroke and ELMo [68]. Their results indicated that stroke ELMo can learn a lot of internal structure information of Chinese characters by pre-training the language model on large-scale data and get better results than random character ELMo.

Glyph Information As hieroglyphs, Chinese characters are developed step by step from oracle bone scripts. The original oracle bone scripts imitated the shape of things so that rich pictographic information can be obtained from the character image to help CNER. Many literatures [81,57,82,83] incorporated glyph information into the character representation. These methods treated Chinese characters as images and used convolutional neural networks to extract information and semantics of font images. Table 9 shows the performance in some works using glyph information, and glyph information can bring some improvements in the case of F1-scores having reached a high level. Among them, Meng et al. [81] proposed Glyce: Glyph-vectors for Chinese character representations, which is a versatile character presentation for logo-

**Table 9**  
Improvement brought by Glyph information.

Work	Dataset	Model	F1(%)	Improvement(%)	Year
[81]	MSRA	BERT	94.80	0.74	2019
		BERT + Glyce	95.54		
		Lattice-LSTM	93.18	0.71	
		Lattice-LSTM + Glyce	93.89		
[57]	MSRA	BILSTM-CRF	89.94	1.14	2019
		BILSTM-CRF + glyph embeddings	91.08		
[83]	MSRA	BERT + BILSTM-CRF	95.30	1.19	2019
		BERT + BILSTM-CRF + GLYNN	96.49		



graphic and can be integrated into existing deep learning systems like word embeddings. In addition, Chen et al. [84] presented a multi-modal model, Glyph2Vec, to tackle Chinese out-of-vocabulary word embedding problem. Glyph2Vec extracts visual features from character glyphs to expand current word embedding space for out-of-vocabulary word embedding, without the need of accessing any corpus, which is useful for improving Chinese NLP systems, especially for low-resource scenarios.

Lexicon Information Lexicon information is also named as dictionary or gazetteer information. Words composed of multiple Chinese characters in lexicons are rich in information. Chinese text has no spaces, and its basic unit is characters, so CNER is much more difficult than ENER in determining the boundary of entities. The introduction of lexicon information can bring a lot of boundary information to the model and improve the accuracy of entity segmentation. In recent years, adding lexicon information to CNER models has been proved very effective.

There are two ways to bring in lexicon information. One is Lattice model, which forms a new embedding for each word node, and sends it to the context encoder together with the character embedding, and modifies the context encoder accordingly so that it can encode this structure. The Lattice model was first proposed by Zhang et al. [85], known as the Lattice-LSTM model. In the Lattice-LSTM model, each character was treated as a lattice, the information of each word was added to its last character to form a lattice. If a character got no word end up with it, the character was a lattice itself. Then Lattice-LSTM used adjusted LSTM as a context encoder to make it accept Lattice as input. For lattice sequence, the update strategy of Lattice LSTM was similar to LSTM, and for external word nodes, Lattice LSTM would add extra edge skips to the update path. WC-LSTM [86] made a slight improvement to Lattice-LSTM, used four different strategies to encode word information into a fixed-size vector so that it could be trained in batches. PLTE [74] augmented self-attention with positional relation representations to incorporate lattice structure and introduced a porous mechanism to augment localness modeling and maintain the strength of capturing the rich long-term dependencies. It integrated the word information into the porous lattice Transformer encoder, enabled it to be processed in batches to capture the dependence between characters and words. FLAT [87] proposed a new position encoding method for Lattice, which converted the lattice structure into a flat structure consisting of spans. Each span corresponded to a character or latent word and its position in the original Lattice. This method gave each character or word two position labels, head and tail, respectively. Then it used the head and tail information to calculate the 4 relative distances between every 2 nodes. This way of relative position coding could make a node get better attention relative to other nodes to achieve better coding of the whole structure. With the power of the Transformer and well-designed position encoding, FLAT can fully leverage the lattice information and has an excellent parallelization ability.

Another way is to add the lexicon matching information to the character representations directly, which is equivalent to only modifying the embedding layer. Wang et al. [88] designed three types of features to extract lexicon information and used two fusion methods of lexicon information and character information. One fusion method is to concatenate these two pieces of information together directly and send them to the context encoder. The other is to send the lexicon information and character information separately to their respective encoders and then concatenate them together and send to the decoder. Ding et al. [67] proposed an Adapted Gated Graph sequence Neural Network (Adapted GGNN) and combined a variety of dictionaries to allow the model to learn meaningful features from these dictionaries automatically. Ma et al. [75] borrowed from the lattice models and proposed the

SoftLexicon model, which added lexicon information to the character representations with a fixed-length vector without modifying the context encoder. SoftLexicon matched the sentence with the dictionary. For each character, SoftLexicon found all the words containing it, divided them into four categories, mapped them into four categories of vectors, and then concatenated these four vectors with the character representations. So that the boundary information and word meaning information are added to the input presentation layer simultaneously, this method avoided designing a complicated sequence modeling architecture, achieved an inference speed up than those of SOTA methods, along with a better performance. Fig. 6 compares the performance in above works based on Lattice model. It can be seen that recent Lattice-based models got better results than the original Lattice model, and different models proposed in 2020 got comparable results.

CWS Information Chinese word segmentation (CWS) information comes from word segmentation tools, which segments words first and then uses the results for subsequent research. CWS information can also bring boundary information to the NER model, but the model results are easily affected by the quality of the word segmentation results. Li et al. [89] proposed a novel word-aligned attention to exploiting explicit word information, which was complementary to various character-based Chinese pre-trained language models. Through careful consideration of multi-granularity segmentation results, it could implicitly reduce the error caused by automatic annotation. Among those studies on CCKS datasets without the use of pre-trained language models, Luo et al. [80] fused CWS, radical, lexicon, stroke information and achieved the best results on the datasets of CCKS2017 ( $F_1$ :91.75%) and CCKS2018 ( $F_1$ :90.05%).

#### 4.1.3. Discussion of Character Representations

The pre-trained character embeddings are necessary for NER models, and adding external information may lead to improved performance, but at the cost of losing the versatility of NER systems. Fig. 7 gives an illustration of the character representations and their methods. The Character embeddings and external information representations are often used in any combination so that the characteristics of the characters can be expressed well. Duan et al. [90] conducted a study on features of the CRF-based CNER and believed that better results could be achieved by combining the commonly used representations.

As shown in Table 7–9, adding more external information can usually improve performance. Song et al. [66] combined word embeddings, character embeddings, radical embeddings and achieved the highest  $F_1$ -score of 71.86% on the WEIBO dataset. Sehanobish et al. [83] combined BERT and glyph information and achieved the highest  $F_1$ -score of 96.49% on the MSRA dataset, and 71.81% on the WEIBO dataset, which was almost the same as

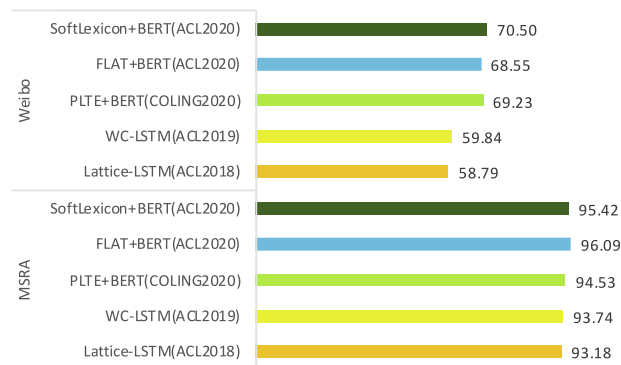


Fig. 6. Performance( $F_1$ ) in works based on Lattice model.



	Pre-trained character embeddings		External Information					
	Lookup tables	Pre-trained language models	POS	Radical Information	Stroke Information	Glyph Information	Lexicon Information	Chinese Word Segmentation
Illustrations	Word2vec, Glove, FastText, etc.	BERT, ELMo, NEZHA, etc.	Noun	十日十月			朝阳, 明朝, 朝夕, 朝鲜	明朝/的/皇帝
methods	Static embeddings	Contextual embeddings	embeddings	embeddings and RNN	RNN	CNN	Lattice	CWS tools

Fig. 7. The illustration of representations of the character ‘朝’.

the first place of 71.86%. Wu et al. [76] used the pre-trained language model NEZHA to achieve the highest  $F_1$ -scores in CCKS2017 and CCKS2018, which were 93.58% and 95.08%, respectively.

#### 4.2. Context Encoder

The context encoder uses recurrent neural networks, Transformer models, or other networks to capture the contextual correlation between tokens. The input characters get independent character representations through the embedding layer, but characters may have different meanings in different sentences. The role of the Context Encoder is to model the contextual semantics of the characters in the sentence. It encodes the character representations and obtains their contextual representation in the current sentence for each character by analyzing the position and dependency of the characters. Then input characters are encoded from independent character representations into contextual representations.

##### 4.2.1. Recurrent Neural Network

The most classic context encoder is based on Recurrent Neural Networks (RNN) and its variants. RNN is a type of neural network that takes sequence data as input, recurses in the evolution direction of the sequence, and all nodes (recurrent units) are connected in a chain [91]. The RNN designs a recurrent structure that does not change with the length and position of the sequence data, and the parameters are shared. The state of each recurrent unit at the current time step is determined by the input of the time step and the state of the previous time step. Furthermore, its weight is shared; that is, the recurrent unit uses the same weight to process all time steps. Compared with feedforward neural networks, weight sharing reduces the total parameter amount of RNN and so that RNN can extract features that change over time in the sequence and deal with sequences with different lengths.

Due to gradient disappearance, the original RNN assigns more weight to the nearest node and cannot learn long-distance dependence. To handle this problem, gated recurrent networks give the RNN the ability to control its internal information accumulation through the gated unit, which can grasp the long-distance dependence and selectively forget the information to prevent overload when learning. Mainstream gated recurrent networks include Gated Recurrent Units (GRU [92]) and Long and Short-Term Memory networks (LSTM [93]).

The RNN-based models have achieved remarkable results in modeling sequence data. In particular, the bidirectional RNN can effectively use the forward and backward information of the data and model the dependency between characters better. Most of the CNER works (see Table 11) use the RNN-based structure (LSTM, GRU) as the encoder, and LSTM has received much more attention than GRU. Moreover, when some Lattice-based methods [85,86,74]

introduced lexicon information, some changes were made to LSTM to adapt the structure of the input data.

##### 4.2.2. Transformer

The Transformer is a neural network model proposed by the Google team in 2017 [94]. It is based only on the attention mechanism and completely avoids loops and convolutions. The Transformer follows the common overall framework of the encoder-decoder structure. Both the encoder and the decoder are stacked using self-attention and fully connected layers. The self-attention mechanism is just a mechanism that redistributes the weight for each token through a calculation based on all tokens in the context. The attention mechanism can directly establish long-distance dependence, so it can also effectively encode contextual information. Due to the simple structure, the calculation is simple and easy to parallel. Like the encoder and decoder functions in the NER method, the encoder is responsible for encoding its contextual information for each input token. The decoder obtains an output according to the information. The difference is that the decoder in NER outputs the type while the Transformer outputs a vector representation.

Li et al. [87] transformed Transformer so that it can accept Lattice as input and get higher  $F_1$ -score on MSRA dataset than other Lattice-based methods. But Transformer is not always better, both Meng et al. [81] and Sehanobish et al. [83] used BERT and glyph information, the former adopted Transformer as the context encoder while the latter used BiLSTM, and the latter (WEIBO: 71.81%, MSRA: 96.49%) performed better than the former (WEIBO: 67.60%, MSRA: 95.54%).

##### 4.2.3. Discussion of Context Encoder

The RNN-based models usually calculate the input and output sequences in order. They generate the hidden state sequence of the position according to the hidden state of the previous step and the input. This inherent sequential property hinders the parallelization of sample training, which becomes crucial for longer sequence lengths because the size of memory limits the batch size of samples.

The Transformer uses a self-attention mechanism to assign weights to each token dynamically. The self-attention mechanism makes each input token be treated equally. The distance between any two tokens is equal so that there will be no errors due to the influence of distance and length and no gradient disappearance. However, the position information lost by the Transformer is crucial in NER, and adding a Position Embedding to the feature vector is only a stopgap. It does not change the inherent defects of the Transformer structure.

**Table 11**  
Summary of recent work in CNER.

Work	Character representation		Attention		Attention		Performance (F1-score)	Year
	Character embeddings	External Information	Input -> Encoder	Context Encoder	Encoder -> Decoder	Tag Decoder		
[55]	Word2vec	Radical	✓	LSTM	✓	CRF	MSRA:89.78% MSRA:90.95%	2016
[58]		POS		LSTM		CRF	CCKS2018:78.94% CCKS2018:80.07%	2019
[59]				LSTM		CRF	CCKS2018:86.68% CCKS2018:87.26%	2019
[60]		POS, Radical		LSTM		CRF	CCKS2017:90.12% CCKS2017:91.35%	2019
[61]	✓	POS, Dictionary		LSTM	✓	CRF	CCKS2017:90.48% CCKS2018:86.11%	2019
[80]	Word2vec	CWS, Radical, Lexicon, Stroker		LSTM		CRF	CCKS2017:91.75% CCKS2018:90.05%	2020
[76]	Word2vec BERT ERNIE ALBERT NEZHA Word2vec BERT ERNIE ALBERT NEZHA			O		CRF	CCKS2018:69.01% CCKS2018:90.54% CCKS2018:93.37% CCKS2018:87.68% CCKS2018:93.58% CCKS2018:75.60% CCKS2018:91.43% CCKS2018:93.11% CCKS2018:90.12% CCKS2018:95.08%	2020
[56]	Sogou news Word2vec	Radical		LSTM		CRF	Peoples'Daily:92.06% Peoples'Daily:94.37%	2019
[62]	✓	Position, segmentation	Concolution - attention	GRU	✓	CRF	WEIBO:53.80% MSRA:90.32% WEIBO:55.91% MSRA:92.34% WEIBO:59.31% MSRA:92.97%	2019
[63]	Word2vec BERT	Radical		GRU	✓	CRF	MSRA:90.45% MSRA:95.42%	2019
[77]	Conv-GRU Embedding	Word, Radical		GRU		CRF	WEIBO:68.93% MSRA:91.45%	2019
[88]	✓	Dictionary		LSTM		CRF	CCKS2017:91.24%	2019
[64]	✓	Lexicon, Word		LSTM		CRF	WEIBO:63.09% MSRA:93.47%	2019
[65]	✓	Word, Position		LSTM	✓	CRF	WEIBO:59.5% MSRA:92.99%	2020
[81]	BERT	Glyph		Transformer		CRF	WEIBO:67.60% MSRA:95.54%	2019
[57]	Wikipedia GloVe	Glyph		LSTM		CRF	MSRA:91.11%	2019
[82]	BERT	Radical, Glyph		LSTM		CRF	WEIBO:70.01% MSRA:95.51%	2020
[83]	BERT	Glyph		LSTM		CRF	WEIBO:71.81% MSRA:96.49%	2019
[66]	✓	Radical, Word	✓	GRU		CRF	WEIBO:71.86% MSRA:92.71%	2020
[67]	✓	Adapted GGNN Gazetteers		LSTM		CRF	WEIBO:59.5% MSRA:94.4%	2020
[85]	✓	Lexicon		Lattice-LSTM		CRF	WEIBO:58.79% MSRA:93.18%	2018
[86]	✓	Lexicon		WC-LSTM		CRF	WEIBO:59.84% MSRA:93.36%	2019
[74]	✓	Lexicon		PLTE		CRF	WEIBO:55.15% MSRA:93.26%	2019
[87]	BERT BERT ✓ BERT	Lexicon		MLP FLAT		CRF	WEIBO:69.23% MSRA:94.53% WEIBO:68.20% MSRA:94.95% WEIBO:63.42% MSRA:94.35% WEIBO:68.55% MSRA:96.09%	2020
[75]	✓ BERT	SoftLexicon		LSTM		CRF	WEIBO:61.42% MSRA:93.66% WEIBO:70.50% MSRA:95.42%	2020
[99]	BERT	Lexicon, radical		Transformer	✓	CRF	WEIBO:70.43% MSRA:96.24%	2021

✓ in the Character embeddings column means Pre-trained character embeddings is used, but no specific model is mentioned. ✓ in Attention columns means self-attention. 'O' in Context Encoder column means no use of Encoder. CWS:Chinese Word Segmentation, LSTM: Long Short-Term Memory, GRU: Gated Recurrent Unit, MLP: Multi-Layer Perceptron, CRF: Conditional Random Field.

#### 4.3. Tag Decoder

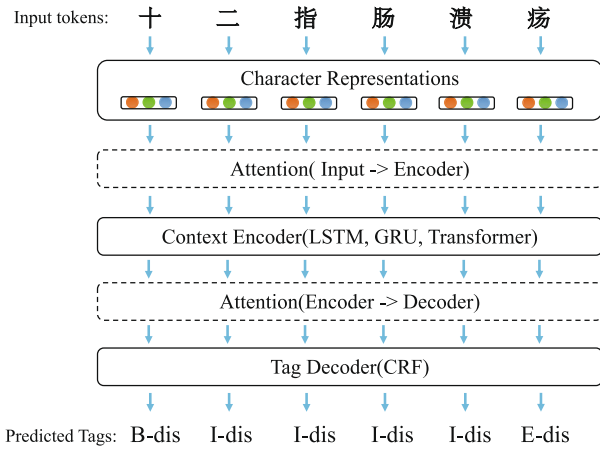
The tag decoder uses the encoded context information to predict the tag of the token and is the final stage of the NER model. It takes contextual representations as input and generates a tag sequence corresponding to the input sequence. There are currently two main forms of implementation.

(1) MLP + Softmax. The sequence labeling task will be converted into a multi-class classification task. After obtaining the representation of each character, a fully connected layer is directly used to obtain the score of each label corresponding

to the character. In this structure, the label of each token is independently predicted by the decoder based on its contextual representation, regardless of its surrounding tokens.

(2) Conditional Random Field (CRF [95]). CRF can model the internal dependencies of the tag sequence. For example, in the BIOES tag scheme, only I or E can be connected after B, but not S. The CRF can gradually learn this inter-label dependence during training, thereby avoiding some grammar errors. It is usually used as the decoder in most CNER researches.

CRF is the Markov random field of random dependent variable  $Y$  under the condition of given random independent variable  $X$ .



**Fig. 8.** The overall framework of the deep learning-based CNER methods with attention layers. The basic layers in solid lines, and the optional attention layers in dotted lines.

The linear chain CRF is mainly used in the sequence labeling task. In the conditional random field model  $P(Y|X)$ ,  $Y$  is the output variable, which represents the labeled sequence, and  $X$  is the input variable, which represents the observation sequence that needs to be labeled. When learning, we use the training data to obtain the conditional probability model  $\hat{P}(Y|X)$  through maximum likelihood estimation or regularised maximum likelihood estimation; when predicting, for a given input sequence  $x$ , we find the output  $\hat{y}$  with the largest conditional probability  $\hat{P}(y|x)$ .

Given the conditional random field  $\hat{P}(Y|X)$ , the input sequence  $x$  and the output sequence  $y$ , the problem of calculating the probability of a conditional random field is to calculate the conditional probabilities  $P(Y_i = y_i|x)$ ,  $P(Y_{i-1} = y_{i-1}, Y_i = y_i|x_i)$  and the corresponding mathematical expectation.

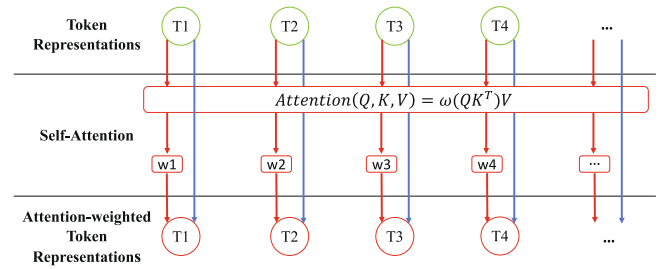
Given the conditional random field  $P(Y|X)$  and the input sequence  $x$  (observation sequence), the prediction of the CRF is to find the output sequence  $y^*$  (label sequence) with the largest conditional probability. The prediction algorithm of the CRF is the well-known Viterbi algorithm, which uses dynamic programming to solve the optimal path efficiently, and gets the label sequence with the highest probability.

#### 4.4. Attention Mechanism

Many studies have added attention between the representation layer and the encoder layer or between the encoder layer and the decoder layer to improve the representation for downstream tasks. The overall framework of the deep learning-based CNER methods with attention layers is shown in Fig. 8. From the input sequence to the predicted tag, the deep learning-based CNER model consists of three basic layers of Character Representation, Context Encoder, and Tag Decoder. The attention layers can be optional between any two layers of basic structure.

The attention mechanism in deep learning is motivated by human's selective visual attention mechanism. The core goal is to select the information that is more critical to the current task goal from a lot of information, and it is first applied in the field of machine vision.

The attention function can be described as mapping a Query and a set of Key-Value pairs to the output, where the Query, Key, Value, and output are all vectors. The output is the weighted sum of the Value, where the weight assigned to each Value is calculated by the compatibility function of the Query and the corresponding Key. In the encoder-decoder framework of general



**Fig. 9.** Schematic diagram of self-attention mechanism in CNER.

tasks, the input Source and output Target content may be different. For example, for English-Chinese machine translation, the Source is an English sentence, and the Target is the corresponding translated Chinese sentence. The attention mechanism is applied between the Query of the Target and all elements of Source. Self-attention is a development of the attention mechanism and focuses on capturing the internal correlation of data or features, it refers not to the attention mechanism between Target and Source but the attention mechanism that applies between the elements within the Source or the Target.

Fig. 9 gives a concept of self-attention mechanism in CNER, the attention layers input the token representations from embedding layer or encoder, then output the attention weighted token representations to encoder or decoder, respectively. The self-attention in CNER is a mechanism that redistributes weights for each token through calculations based on all tokens and assigns more weights to more essential tokens.

In the CNER researches, especially in medical NER tasks, the attention layers are widely used. Some studies like [58,62,66] added attention layers between the character representation and the context encoder, and some [59–61,56,62,65] added between the context encoder and tag decoder. All of these adopted the self-attention mechanism except literature [62]. In the literature [62], a convolutional attention layer was used between the feature representation and the encoder. For character representations containing character embeddings, word segmentation embeddings, and position embeddings, local attention was used to capture the dependence of the central character and surrounding characters in the window range, and the output of the local attention was sent to the CNN. Finally, the local features were obtained by additive pooling and then sent to the context encoder.

Table 10 compares the performance of the attention module in some researches. Their results showed that the use of the attention module could improve the  $F_1$ -score, the improvement varies from 0.58% to 3.74%.

#### 4.5. Adversarial Transfer Learning

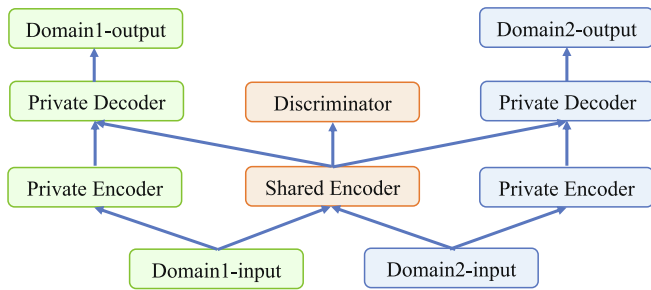
Since the labeled data is usually limited in NER, it is often necessary to improve model performance by using cross-domain or cross-task data. Inspired by the idea of adversarial learning, some works [96–98] fulfilled the knowledge transfer by taking advantage of the adversarial network to learn shared information about the same field or the same task and achieved the purpose of knowledge transfer.

Fig. 10 gives an illustration of cross-domain adversarial transfer learning. After constructing encoder-decoder models on datasets in different domains and adding shared encoders for joint training, the discriminator is used to determine which domain is the current input from. When the discriminator cannot distinguish the source domains, it means that the adversarial training is completed. Then the shared encoder contains the shared information of two



**Table 10**  
Some works using attention modules.

Work	Dataset	Model	F1(%)	Improvement(%)
[58]	CCKS2018	LSTM-CRF	67.32	2.59
		SM-LSTM-CRF	69.91	
		POS + LSTM-CRF	78.94	
[60]	CCKS2017	POS + SM-LSTM-CRF	80.07	1.13
		BILSTM-CRF	88.78	
		Att-BILSTM-CRF	90.11	
		BILSTM-CRF + radical	89.64	1.32
		Att-BILSTM-CRF + radical	90.96	
		BILSTM-CRF + POS	89.06	1.75
		Att-BILSTM-CRF + POS	90.81	
		BILSTM-CRF + radical + POS	90.12	1.23
		Att-BILSTM-CRF + radical + POS	91.35	
[59]	CCKS2018	BILSTM-CRF	86.68	0.58
		Attention-BILSTM-CRF	87.26	
		BILSTM-CRF + dictionary	87.71	0.58
		Attention-BILSTM-CRF + dictionary	88.29	
[56]	CCKS2018	char	86.09	3.17
		char + attention	89.26	
		char + word	90.74	3.74
		char + word + attention	94.48	



**Fig. 10.** Schematic diagram of cross-domain adversarial transfer learning.

domains and does not contain separate information for individual domains.

For Chinese, the NER tasks and the CWS tasks share much information (mainly boundary information). Wu et al. [33] used a joint learning method for NER task and word segmentation task. However, it did not filter out the unique information of the word segmentation task. Cao et al. [96] constructed models for NER task and CWS task separately and added a shared BILSTM layer for joint training. The shared BILSTM obtained from the adversarial training could learn information that independent of the specific task, thereby filtered out the information unique to the word segmentation task and only incorporated the shared information into the NER task. Wen et al. [97] took the CCKS competition dataset as the source dataset and the online medical consultation text obtained on the Internet as the target dataset, then constructed the NER model of the source data and the target data, respectively. Finally, they learned the parameters of the shared CNN layer through adversarial training to share domain information and realized the purpose of domain knowledge transfer. Hu et al. [98] simultaneously used information from multiple fields (microblogs and news) and multiple tasks (NER and CWS) to build a dual adversarial network. Through joint training on different tasks and different fields, it learned domain-shared information and task-shared information. Finally, the shared information could be used to improve the effect of NER in specific tasks in specific domains.

The above methods using adversarial transfer learning showed that the model performance can be improved after adding shared information from other tasks or domains. However, adversarial transfer learning methods performed inferior to these SOTA models using pre-trained language models, and it might be promising to combine the two of them together.

#### 4.6. Summary and Discussion

Table 11 demonstrates the relevant works on deep learning-based CNER models in recent years and summarizes the performance of these works in the MSRA, WEIBO public datasets, and annual CCKS competition datasets. It is an overall summary of the above tables in this Section 4 and shows what do these works do in the basic architecture with attention layer. Most of CNER works is focused on character representation because of the characteristic of Chinese, while little research on encoder or decoder, because most of them choose LSTM as encoder and CRF as encoder.

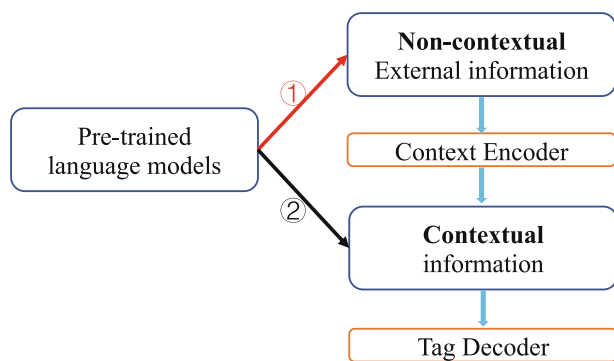
For character representation, it is the focus of CNER's researches. Simple external features such as POS, radical and stroker, contain less information and are effective when the information is insufficient, so these features perform well before BERT is proposed. However, with the development of pre-trained language models like BERT, their representations could catch most of the meanings of Chinese characters, then CNER models can benefit little from simple external features when having used these pre-trained language models. Table 11 shows that these SOTA studies all adopt pre-trained language models for character representation. On the basis of using pre-trained language models, glyph and lexicon are more preferred than other external features for character representation.

For context encoder, it is relied upon by those models adding other external features without contextual information to improve performance. However, pre-trained language models usually contain much context encoding information, and adding a context encoder may reduce the pre-trained encoding information, which can not bring us consistent improvement in our application. Fig. 11 gives an illustration of two ways of concatenating pre-trained language models with external information representations. Compared with concatenating pre-trained language models with non-contextual external information before sent to context encoder (see the red arrow 1), it is promising to concatenate pre-trained language models with the contextual information output from encoders before sent to the tag decoder (see the black arrow 2).

#### 5. Future: Trends and Challenges

In the environment of big data, deep learning has brought fiery development of CNER. However, most researches were solidified in





**Fig. 11.** Two ways of concatenating the representations of pre-trained language models and external information. The second way is more recommended.

adjusting classic models or using more external features, and the performance reached a bottleneck. Future research may focus on the following areas:

### 5.1. Improving model performance

Using different character representations according to specific tasks From the recent research, most of recent researches focus on how to introduce more external information in character representations, such as adding character representations or learning cross-domain information. However, introducing external information does not necessarily guarantee the improvement. For example, when the CWS information is introduced, the wrong results of word segmentation will mislead the model to classify entities incorrectly. Especially in professional fields such as medical NER, the word segmentation rules are more complicated and difficult. Therefore, we should carefully select different information according to the specific task. For example, we can introduce rule-based lexicon information and POS information for tasks with regular entities, while information based on the character characteristics, such as glyph information and stroke information, for tasks with many new entities.

Building more effective models Although recent works introduced a lot of external information and achieved good results, the performance of CNER has reached a bottleneck under the current model framework. External information can be better utilized by designing more effective models for CNER. Li et al. [100] proposed a unified MRC framework for NER. Instead of treating NER as a sequence labeling problem, they formulated it as a machine reading comprehension (MRC) task and got competitive results with SOTA models. This method is an effective attempt for better models.

Introducing Cross-lingual information Cross-lingual information is another form of external information which can help CNER. On the one hand, cross-lingual information can be introduced by combining translation tasks with NER tasks, such as translating Chinese into English and then identifying the entities in English and finally re-translating these entities to Chinese. On the other hand, various knowledge like common sense in English can also be used to provide effective information for CNER. For example, Wu et al. [101] proposed a teacher-student learning method, where NER models in the source languages were used as teachers to train a student model on unlabeled data in the target language. This method addressed the limitation that existing cross-lingual NER methods were not applicable if the labeled data in the source languages are unavailable or do not correspond to the unlabeled data in the target language. So that the common knowledge can be transferred from one language to another without the need for pairwise corpora.

### 5.2. More complex applications

Nested NER Nested entity refers to the complete inclusion of one entity in another entity. In the medical NER task of CCKS2020, entities of other types are allowed to be nested in the 'disease' type, which brings challenges to CNER. For example, the 'body' entity '胃 (stomach)' is nested in the 'disease' entity '胃溃疡 (gastric ulcer)'. In English NER, there have been many works [102–108] researching nested entities. Among them, Wang et al. [106] presented Pyramid, a novel layered model for nested NER. Token or text region embeddings were recursively inputted into  $L$  flat NER layers, from bottom to top, stacked in a pyramid shape. The proposed method achieves state-of-the-art F1 scores in nested NER datasets. This method is promising to recognize nested entities in Chinese text.

Fine-grained NER Fine-grained NER refers to the recognition of named entities with thousands of entity types and a hierarchical structure between types, which is expected to provide richer semantic information for downstream NLP applications. However, the publicly available CNER models now support only a small number of non-hierarchical entity types.

Named entity disambiguation Named entity disambiguation is to assigns a unique identity to entities mentioned in the text to determine the real-world entity pointed to by an entity referent. The ambiguity of entities can be summarized into two categories: diversity and ambiguity, which means duplication of names (multiple names with one meaning) and duplication of meanings (multiple meanings with one name). Named entity disambiguation can get the correct semantic relationship between entities, which will help improve the reliability of results.

### 5.3. Solving the problem of data

More and higher quality datasets High-quality datasets are essential for model learning and evaluation. Compared with the full bloom of ENER datasets, CNER datasets are still insufficient in quality and quantity. To carry out research on above-mentioned nested NER, fine-grained NER or named entity disambiguation, the first thing is to solve the problem of lacking high-quality datasets on these tasks. For example, Ding et al. [109] presented FEW-NERD, a large-scale human-annotated few-shot NER dataset with a hierarchy of 8 coarse-grained and 66 fine-grained entity types.

Weakly-supervised and unsupervised learning As the labour cost of corpus labeling is expensive, it is meaningful to develop weakly-supervised or unsupervised algorithms to realize CNER based on fewer or none labeled corpus. Recently, Zeng et al. [110] proposed a weakly supervised method from a causal perspective and provided the interpretability of their method with the structural causal model. When there is no hand-labeled data for the target domain, Lison et al. [111] presented a simple but powerful approach to learn NER models in the absence of labeled data through weak supervision. The approach relied on a broad spectrum of labeling functions to automatically annotate texts from the target domain and a hidden Markov model, which captured the varying accuracies and confusions of these labeling functions. Aly et al. [112] explored the task of zero-shot NER with entity type descriptions to transfer knowledge from observed to unseen classes. Despite the low F1 value, they achieved a breakthrough for unsupervised learning in NER.

## 6. Conclusion

This paper intends to act as a practical guide for understanding, using, and developing CNER models. We first give an integral con-

cept of CNER to readers, by introducing some preliminary knowledge of CNER, including datasets, tag schemes, evaluation metrics, difficulties. After introducing the history of CNER traditional methods, we introduce current deep learning-based methods on the basic architecture of character representation, context encoder and tag decoder. Apart from the basic architecture, the intra-architecture attention mechanism and inter-architecture adversarial transfer learning in CNER are introduced. We discuss the characteristics, advantages, and disadvantages of different methods during the elaboration and give our advice based on our experiment experience. Finally, we discuss the possible trends and challenges to CNER and several directions that may be further developed in the future.

### CRedit authorship contribution statement

**Yanming Guo:** Supervision, Writing - review & editing, Validation. **Fenglei Wang:** Resources. **Guohui Li:** Project administration.

### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61806218, 71673293) and the Natural Science Foundation of Hunan Province (No. 2019JJ50722).

### References

- [1] D. Nadeau, S. Sekine, A survey of named entity recognition and classification, *Linguisticae Investigationes* 30 (1) (2007) 3–26.
- [2] A. Mansouri, L.S. Affendey, A. Mamat, Named entity recognition approaches, *International Journal of Computer Science and Network, Security* 8 (2) (2008) 339–344.
- [3] G.K. Palshikar, Techniques for named entity recognition: a survey, in: *Bioinformatics: Concepts, Methodologies, Tools, and Applications*, IGI Global, 2013, pp. 400–426.
- [4] P. Sun, X. Yang, X. Zhao, Z. Wang, An overview of named entity recognition, in: *2018 International Conference on Asian Language Processing (IALP)*, IEEE, 2018, pp. 273–278.
- [5] A. Goyal, V. Gupta, M. Kumar, Recent named entity recognition and classification techniques: a systematic review, *Computer Science Review* 29 (2018) 21–43.
- [6] Y. Wen, C. Fan, G. Chen, X. Chen, M. Chen, A survey on named entity recognition, in: *International Conference in Communications, Signal Processing, and Systems*, Springer, 2019, pp. 1803–1810.
- [7] J. Li, A. Sun, J. Han, C. Li, A survey on deep learning for named entity recognition, *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [8] G. Popovski, B.K. Seljak, T. Eftimov, A survey of named-entity recognition methods for food information extraction, *IEEE Access* 8 (2020) 31586–31594.
- [9] K. Shaalan, H. Raza, Nera: Named entity recognition for arabic, *Journal of the American Society for Information Science and Technology* 60 (8) (2009) 1652–1663.
- [10] K. Shaalan, A survey of arabic named entity recognition and classification, *Computational Linguistics* 40 (2) (2014) 469–510.
- [11] A. Zirikly, M. Diab, Named entity recognition for arabic social media, in: *Proceedings of the 1st workshop on vector space modeling for natural language processing*, 2015, pp. 176–185.
- [12] A. Dandashi, J. Al Jaam, S. Fofouf, Arabic named entity recognition—a survey and analysis, in: *Intelligent Interactive Multimedia Systems and Services 2016*, Springer, 2016, pp. 83–96.
- [13] R.E. Salah, L.Q. binti Zakaria, A comparative review of machine learning for arabic named entity recognition, *International Journal on Advanced Science, Engineering and Information Technology* 7 (2) (2017) 511–518.
- [14] I. El Bazi, N. Laachfoubi, Arabic named entity recognition using deep learning approach, *International Journal of Electrical & Computer Engineering* (2088–8708) 9 (3) (2019).
- [15] L. Liu, J. Shang, J. Han, Arabic named entity recognition: What works and what's next, in: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, 2019, pp. 60–67.
- [16] D. Kaur, V. Gupta, A survey of named entity recognition in english and other indian languages, *International Journal of Computer Science Issues (IJCSI)* 7 (6) (2010) 239.
- [17] B. Sasidhar, P. Yohan, A.V. Babu, A. Govardhan, A survey on named entity recognition in indian languages with particular reference to telugu, *International Journal of Computer Science Issues (IJCSI)* 8 (2) (2011) 438.
- [18] G. Talukdar, P. Protim Borah, A. Baruah, A survey of named entity recognition in assamese and other indian languages, *arXiv e-prints* (2014) arXiv-1407..
- [19] N. Patil, A.S. Patil, B. Pawar, Survey of named entity recognition systems with respect to indian and foreign languages, *International Journal of Computer Applications* 134 (16) (2016).
- [20] S. Kale, S. Govilkar, Survey of named entity recognition techniques for various indian regional languages, *International Journal of Computer Applications* 164 (4) (2017) 37–43.
- [21] R. Sharma, S. Morwal, B. Agarwal, Named entity recognition for hindi language: A survey, *Journal of Discrete Mathematical Sciences and Cryptography* 22 (4) (2019) 569–580.
- [22] K. Bhattacharjee, S. Mehta, A. Kumar, R. Mehta, D. Pandya, P. Chaudhari, D. Verma, et al., Named entity recognition: A survey for indian languages, in: *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)*, Vol. 1, IEEE, 2019, pp. 217–220.
- [23] R. Shelke, D.S. Thakore, A survey on various methods used in named entity recognition for hindi language, *Test Engineering and Management* (2020).
- [24] D. Chopra, S. Morwal, Named entity recognition in english using hidden markov model, *International Journal* (2013).
- [25] N. Patil, A.S. Patil, B. Pawar, Issues and challenges in marathi named entity recognition, *International Journal on Natural Language Computing (IJNLC)* 5 (1) (2016) 15–30.
- [26] D. Küçük, N. Arıcı, D. Küçük, Named entity recognition in turkish: Approaches and issues, in: *International Conference on Applications of Natural Language to Information Systems*, Springer, 2017, pp. 176–181.
- [27] L. Akhtyamova, Named entity recognition in spanish biomedical literature: Short review and bert model, in: *2020 26th Conference of Open Innovations Association (FRUCT)*, IEEE, 2020, pp. 1–7.
- [28] R. Weischedel, M. Palmer, M. Marcus, E. Hovy, S. Pradhan, L. Ramshaw, N. Xue, A. Taylor, J. Kaufman, M. Franchini, et al., Ontonotes release 5.0 ldc2013t19, *Linguistic Data Consortium*, Philadelphia, PA 23 (2013).
- [29] G.-A. Levow, The third international chinese language processing bakeoff: Word segmentation and named entity recognition, in: *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 2006, pp. 108–117.
- [30] L. Xu, Q. Dong, C. Yu, Y. Tian, W. Liu, L. Li, X. Zhang, Cluener2020: Fine-grained name entity recognition for chinese, Tech. rep., CLUE Organization (2020).
- [31] P. Zhao, L.-Y. Sun, Y. Wan, N. Ge, Chinese scenic spot named entity recognition based on bert+biLstm+crf(in chinese), *Computer Systems and Applications* 29 (6) (2020) 169–174.
- [32] Y. Gao, L. Gu, Y. Wang, Y. Wang, F. Yang, Constructing a chinese electronic medical record corpus for named entity recognition on resident admit notes, *BMC medical informatics and decision making* 19 (2) (2019) 67–78.
- [33] F. Wu, J. Liu, C. Wu, Y. Huang, X. Xie, Neural chinese named entity recognition via cnn-lstm-crf and joint training with word segmentation, in: *The World Wide Web Conference*, 2019, pp. 3342–3348.
- [34] N. Reimers, I. Gurevych, Optimal hyperparameters for deep lstm-networks for sequence labeling tasks, in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.
- [35] L. Liu, W. Dongbo, A review on named entity recognition(in chinese), *Journal of the China Society for Scientific and Technical, Information* 37 (3) (2018) 329–340.
- [36] M. Collins, Y. Singer, Unsupervised models for named entity classification, in: *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999, pp. 100–110.
- [37] S. Cucerzan, D. Yarowsky, Language independent named entity recognition combining morphological and contextual evidence, in: *1999 joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, 1999, pp. 90–99.
- [38] A. Mikheev, M. Moens, C. Grover, Named entity recognition without gazetteers, in: *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, 1999, pp. 1–8.
- [39] G. Zhou, J. Su, Named entity recognition using an hmm-based chunk tagger, in: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 473–480.
- [40] G. Fu, K.-K. Luke, Chinese named entity recognition using lexicalized hmms, *ACM SIGKDD Explorations Newsletter* 7 (1) (2005) 19–25.
- [41] D.M. Bikel, R. Schwartz, R.M. Weischedel, An algorithm that learns what's in a name, *Machine learning* 34 (1–3) (1999) 211–231.
- [42] A. Borthwick, R. Grishman, A maximum entropy approach to named entity recognition, Ph.D. thesis, Citeseer (1999).
- [43] W. Chen, Y. Zhang, H. Isahara, Chinese named entity recognition with conditional random fields, in: *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 2006, pp. 118–121.
- [44] A. McCallum, W. Li, Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons, *Computer Science Department Faculty Publication Series*. 11 (2003).
- [45] H. Isozaki, H. Kazawa, Efficient support vector classifiers for named entity recognition, in: *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

- [46] Y. Hongkui, Z. Huaping, L. Qun, L. Xueqiang, S. Shuicai, Chinese named entity identification using cascaded hidden markov model(in chinese), *Journal on Communications* 27 (2) (2006) 87–94.
- [47] Z. Junsheng, D. Xinyu, Y. Cunyan, C. Jiajun, Automatic recognition of chinese organization name based on cascaded conditional random fields(in chinese), *ACTA ELECTRONICA SINICA* 34 (5) (2006) 804–809.
- [48] H. Wenbo, D. Yuncheng, L. Xueqiang, S. Shuicai, Chinese named entity recognition based on multi-layer conditional random field(in chinese), *Computer Engineering and Applications* 45 (1) (2009) 163–165.
- [49] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *nature* 521 (7553) (2015) 436–444.
- [50] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, Pre-trained models for natural language processing: A survey, *Science China Technological Sciences* (2020) 1–26.
- [51] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, A neural probabilistic language model, *The Journal of Machine Learning Research* 3 (2003) 1137–1155.
- [52] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, *Advances in Neural Information Processing Systems* 26 (2013) 3111–3119.
- [53] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, *Transactions of the Association for Computational Linguistics* 5 (2017) 135–146.
- [54] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.
- [55] C. Dong, J. Zhang, C. Zong, M. Hattori, H. Di, Character-based lstm-crf with radical-level features for chinese named entity recognition, in: *Natural Language Understanding and Intelligent Applications*, Springer, 2016, pp. 239–250.
- [56] Z. Wan, J. Xie, W. Zhang, Z. Huang, Bilstm-crf chinese named entity recognition model with attention mechanism, in: *Journal of Physics: Conference Series*, Vol. 1302, IOP Publishing, 2019, p. 032056.
- [57] Y. Jia, X. Ma, Attention in character-based bilstm-crf for chinese named entity recognition, in: *Proceedings of the 2019 4th International Conference on Mathematics and Artificial Intelligence*, 2019, pp. 1–4.
- [58] X. Cai, S. Dong, J. Hu, A deep learning model incorporating part of speech and self-matching attention for named entity recognition of chinese electronic medical records, *BMC Medical Informatics and Decision Making* 19 (2) (2019) 101–109.
- [59] B. Ji, R. Liu, S. Li, J. Yu, Q. Wu, Y. Tan, J. Wu, A hybrid approach for named entity recognition in chinese electronic medical record, *BMC medical informatics and decision making* 19 (2) (2019) 149–158.
- [60] G. Wu, G. Tang, Z. Wang, Z. Zhang, Z. Wang, An attention-based bilstm-crf model for chinese clinic named entity recognition, *IEEE Access* 7 (2019) 113942–113949.
- [61] L. Li, J. Zhao, L. Hou, Y. Zhai, J. Shi, F. Cui, An attention-based deep learning model for clinical named entity recognition of chinese electronic medical records, *BMC Medical Informatics and Decision Making* 19 (5) (2019) 235.
- [62] Y. Zhu, G. Wang, Can-ner: Convolutional attention network for chinese named entity recognition, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 3384–3393.
- [63] C. Gong, J. Tang, S. Zhou, Z. Hao, J. Wang, Chinese named entity recognition with bert, in: *International Conference on Computer Intelligent Systems and Network Remote Control*, no. cisnrc, 2019, pp. 8–15.
- [64] D. Sui, Y. Chen, K. Liu, J. Zhao, S. Liu, Leverage lexical knowledge for chinese named entity recognition via collaborative graph network, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3821–3831.
- [65] S. Johnson, S. Shen, Y. Liu, Cwpc\_biatt: Character–word–position combined bilstm-attention for chinese named entity recognition, *Information* 11 (1) (2020) 45.
- [66] C. Song, Y. Xiong, W. Huang, L. Ma, Joint self-attention and multi-embeddings for chinese named entity recognition, *Tech. rep., EasyChair* (2020).
- [67] R. Ding, P. Xie, X. Zhang, W. Lu, L. Li, L. Si, A neural multi-digraph model for chinese ner with gazetteers, in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1462–1467.
- [68] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 2227–2237.
- [69] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018).
- [70] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.
- [71] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, H. Wu, Ernie: Enhanced representation through knowledge integration, *arXiv e-prints* (2019) arXiv-1904.
- [72] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, in: *International Conference on Learning Representations*, 2019.
- [73] J. Wei, X. Ren, X. Li, W. Huang, Y. Liao, Y. Wang, J. Lin, X. Jiang, X. Chen, Q. Liu, Nezha: Neural contextualized representation for chinese language understanding, *arXiv e-prints* (2019) arXiv-1909.
- [74] X. Mengge, B. Yu, T. Liu, Y. Zhang, E. Meng, B. Wang, Porous lattice transformer encoder for chinese ner, in: *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 3831–3841.
- [75] R. Ma, M. Peng, Q. Zhang, Z. Wei, X.-J. Huang, Simplify the usage of lexicon in chinese ner, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5951–5960.
- [76] W. Xiao-xue, Z. Qin-hui, Application of pre-training language model in chinese emr named entity recognition(in chinese), *Electronic, Quality* 09 (2020) 61–65.
- [77] C. Xu, F. Wang, J. Han, C. Li, Exploiting multiple embeddings for chinese named entity recognition, in: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2269–2272.
- [78] X. Shi, J. Zhai, X. Yang, Z. Xie, C. Liu, Radical embedding: Delving deeper to chinese radicals, in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2015, pp. 594–598.
- [79] O. Kuru, O.A. Can, D. Yuret, Charner: Character-level named entity recognition, in: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 911–921.
- [80] L. Ling, Y. Zhihao, S. Yawen, L. Nan, L. Hongfei, Chinese clinical named entity recognition based on stroke elmo and multi-task learning(in chinese), *Chinese Journal of Computers* 43 (10) (2020) 1943–1957.
- [81] Y. Meng, W. Wu, F. Wang, X. Li, P. Nie, F. Yin, M. Li, Q. Han, X. Sun, J. Li, Glyce: Glyph-vectors for chinese character representations, in: *Advances in Neural Information Processing Systems*, 2019, pp. 2746–2757.
- [82] Z. Xuan, R. Bao, S. Jiang, Fgn: Fusion glyph network for chinese named entity recognition, *arXiv e-prints* (2020) arXiv-2001.
- [83] A. Sehanobish, C.H. Song, Using chinese glyphs for named entity recognition, *arXiv e-prints* (2019) arXiv-1909.
- [84] H.-Y. Chen, S.-H. Yu, S.-D. Lin, Glyph2vec: Learning chinese out-of-vocabulary word embedding from glyphs, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2865–2871.
- [85] Y. Zhang, J. Yang, Chinese ner using lattice lstm, in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1554–1564.
- [86] W. Liu, T. Xu, Q. Xu, J. Song, Y. Zu, An encoding strategy based word-character lstm for chinese ner, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 2379–2389.
- [87] X. Li, H. Yan, X. Qiu, X.-J. Huang, Flat: Chinese ner using flat-lattice transformer, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6836–6842.
- [88] Q. Wang, Y. Zhou, T. Ruan, D. Gao, Y. Xia, P. He, Incorporating dictionaries into deep neural networks for the chinese clinical named entity recognition, *Journal of biomedical informatics* 92 (2019) 103133.
- [89] Y. Li, B. Yu, X. Mengge, T. Liu, Enhancing pre-trained chinese character representation with word-aligned attention, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 3442–3448.
- [90] H. Duan, Y. Zheng, A study on features of the crfs-based chinese named entity recognition, *International Journal of Advanced Intelligence* 3 (2) (2011) 287–294.
- [91] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, *Deep learning*, Vol. 1, MIT press Cambridge, 2016.
- [92] K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, in: *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, 2014, 2014.
- [93] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural computation* 9 (8) (1997) 1735–1780.
- [94] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [95] J. Lafferty, A. McCallum, F.C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, 2001, pp. 282–289.
- [96] P. Cao, Y. Chen, K. Liu, J. Zhao, S. Liu, Adversarial transfer learning for chinese named entity recognition with self-attention mechanism, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 182–192.
- [97] G. Wen, H. Chen, H. Li, Y. Hu, Y. Li, C. Wang, Cross domains adversarial learning for chinese named entity recognition for online medical consultation, *Journal of Biomedical Informatics* 112 (2020) 103608.



- [98] Y. Hu, C. Zheng, A double adversarial network model for multi-domain and multi-task chinese named entity recognition, *IEICE Transactions on Information and Systems* 103 (7) (2020) 1744–1752.
- [99] S. Wu, X. Song, Z. Feng, MECT: Multi-metadata embedding based cross-transformer for Chinese named entity recognition, in: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, Online, 2021, pp. 1529–1539..
- [100] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, J. Li, A unified mrc framework for named entity recognition, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5849–5859.
- [101] Q. Wu, Z. Lin, B. Karlsson, L. Jian-Guang, B. Huang, Single-/multi-source cross-lingual ner via teacher-student learning on unlabeled data in target language, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6505–6514.
- [102] M. Ju, M. Miwa, S. Ananiadou, A neural layered model for nested named entity recognition, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 1446–1459.
- [103] A. Katiyar, C. Cardie, Nested named entity recognition revisited, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 861–871..
- [104] Y. Luo, H. Zhao, Bipartite flat-graph network for nested named entity recognition, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6408–6418.
- [105] J. Yu, B. Bohnet, M. Poesio, Named entity recognition as dependency parsing, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 6470–6476.
- [106] W. Jue, L. Shou, K. Chen, G. Chen, Pyramid: A layered model for nested named entity recognition, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 5918–5928..
- [107] Y. Wang, H. Shindo, Y. Matsumoto, T. Watanabe, Nested named entity recognition via explicitly excluding the influence of the best path, in: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, Online, 2021, pp. 3547–3557..
- [108] Y. Shen, X. Ma, Z. Tan, S. Zhang, W. Wang, W. Lu, Locate and label: A two-stage identifier for nested named entity recognition, in: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, Online, 2021, pp. 2782–2794..
- [109] N. Ding, G. Xu, Y. Chen, X. Wang, X. Han, P. Xie, H. Zheng, Z. Liu, Few-NERD: A few-shot named entity recognition dataset, in: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, Online, 2021, pp. 3198–3213..
- [110] X. Zeng, Y. Li, Y. Zhai, Y. Zhang, Counterfactual generator: A weakly-supervised method for named entity recognition, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 7270–7280.
- [111] P. Lison, J. Barnes, A. Hubin, S. Touileb, Named entity recognition without labelled data: A weak supervision approach, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 1518–1533.
- [112] R. Aly, A. Vlachos, R. McDonald, Leveraging type descriptions for zero-shot named entity recognition and classification, in: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Association for Computational Linguistics, Online, 2021, pp. 1516–1528..



**Pan Liu** received the B.S. degree in Management Engineering and the M.S. degree in Control Science and Engineering, in 2013 and 2015, respectively, from the National University of Defense Technology, Changsha, China, where he is currently pursuing the Ph.D. degree in the College of Systems Engineering. His current research interests include natural language process, compute vision and deep learning.



**Yanming Guo** is now an associate professor in the College of Systems Engineering, National University of Defense Technology. He received his B.S. and M.S. degrees from the National University of Defense Technology, in 2011 and 2013, respectively, and the Ph.D. degree in Leiden Institute of Advanced Computer Science (LIACS), Leiden University, in 2017. His current interests include computer vision, natural language processing and deep learning. He has served as reviewers of some journals, such as TPAMI, TIP, TNNLS, TMM, PR, Neurocomputing, MTAP.



**Fenglei Wang** received the B.S. and M.S. degrees in Control Science and Engineering, in 2013 and 2015, respectively, from the National University of Defense Technology, Changsha, China, where he is currently pursuing the Ph.D. degree in the in the College of Systems Engineering. His current research interests include image classification, object detection and deep learning.



**Guohui Li** received PhD degree in information system engineering, National University of Defense Technology (NUDT), March 2001. He is the member of a council in Hunan Association for Artificial Intelligence. He has become full professor in NUDT since 1999. He had studied and worked in Communication Lab, Department of Electrical and Computer Engineering, University of Delaware, and Multimedia Communication and Visualization Lab, Department of Computer Science, University of Missouri-Columbia, USA as a research scholar during 1999 to 2000. His research interests include intelligent systems, information retrieval, machine learning and intelligent digital media.