

Xinzhi WANG, Jiahao LI, Ze ZHENG, Yudong CHANG, Min ZHU

Entity and relation extraction with rule-guided dictionary as domain knowledge

© Higher Education Press 2022

Abstract Entity and relation extraction is an indispensable part of domain knowledge graph construction, which can serve relevant knowledge needs in a specific domain, such as providing support for product research, sales, risk control, and domain hotspot analysis. The existing entity and relation extraction methods that depend on pretrained models have shown promising performance on open datasets. However, the performance of these methods degrades when they face domain-specific datasets. Entity extraction models treat characters as basic semantic units while ignoring known character dependency in specific domains. Relation extraction is based on the hypothesis that the relations hidden in sentences are unified, thereby neglecting that relations may be diverse in different entity tuples. To address the problems above, this paper first introduced prior knowledge composed of domain dictionaries to enhance characters' dependence. Second, domain rules were built to eliminate noise in entity relations and promote potential entity relation extraction. Finally, experiments were designed to verify the effectiveness of our proposed methods. Experimental results on two domains, including laser industry and unmanned ship, showed the superiority of our methods. The *F1* value on laser industry

entity, unmanned ship entity, laser industry relation, and unmanned ship relation datasets is improved by +1%, +6%, +2%, and +1%, respectively. In addition, the extraction accuracy of entity relation triplet reaches 83% and 76% on laser industry entity pair and unmanned ship entity pair datasets, respectively.

Keywords entity extraction, relation extraction, prior knowledge, domain rule

1 Introduction

Entity and relation extraction aims to identify triplets from unstructured text. These extracted triplets are beneficial for discovering the disadvantage of domains, thereby providing support for product development and sales, business risk control, and commercial hotspot analysis.

The existing entity and relation extraction studies include three categories (rule based, statistical machine learning, and deep learning based).

Rule-based approaches are built on handcrafted dictionaries or linguistic rules. Syntactic-lexical patterns are usually exploited to construct rules. Rule-based entity extraction first appeared in typical Proteus systems (Grishman, 1995) and Lasie systems (Humphreys et al., 1998). These systems employ heuristic algorithms and rule templates. Wang and Hirst (2009) employed WordNet (Miller, 1995) to extract patterns iteratively, thereby improving the recall rate of extraction results. Hearst (1992) extracted upper and lower semantic relations based on template matching (e.g., “X such as Y”). Rule-based approaches work very well. However, they are transferred difficultly to other domains because of domain-specific rules designed by domain experts.

Machine learning models use statistics to build data representation and then conduct analysis to infer any relationships between variables. Machine learning algorithms are employed to learn patterns hidden in features, thereby

Received November 15, 2021; accepted July 5, 2022

Xinzhi WANG (✉), Jiahao LI, Yudong CHANG
School of Computer Engineering and Science, Shanghai University,
Shanghai 200444, China
E-mail: wxz2017@shu.edu.cn

Ze ZHENG
Baidu (China) Co., Ltd., Beijing 100085, China

Min ZHU
The Sixth Medical Center of PLA General Hospital, Beijing 100048,
China

This work is funded by the Shanghai Sailing Program (Grant No. 20YF1413800), Military Medical Science and Technology Youth Cultivating Program (Grant No. 20QNPY106), High Performance Computing Center of Shanghai University, and Shanghai Engineering Research Center of Intelligent Computing System (Grant No. 19DZ2252600).

recognizing similar patterns from unseen data. Entity extraction solves two problems of entity boundary detection and entity type classification. Isozaki and Kazawa (2002), Yamada et al. (2002), and Lin et al. (2006) converted entity extraction tasks into classification tasks with an accuracy rate of up to 95%. Zhou et al. (2005) utilized convolutional trees to extract the textual semantic information in sentences. Manual feature engineering is critical in machine learning models. However, the selection of features depends on the expert's subjective experience. In addition, the dependency hidden in domain entities is ignored in handcrafted feature engineering.

Deep-learning-based models are a type of artificial intelligence that imitates the way humans gain certain types of knowledge. They have become dominant in recent years. Deep learning consists of multiple processing layers to learn the high-level semantics hidden in data automatically (Thomas and Sangeetha, 2020). Artificial neural networks are typical layers composed of forward and backward passes. Forward pass accumulates the weighted sum of input data. Backward pass computes the gradient of an objective function via the chain rule of derivatives. For instance, some approaches (Li et al., 2020; Shibuya and Hovy, 2020; Waldis and Mazzola, 2021) have successfully handled nested and flat entities simultaneously via a pretrained model. Luo et al. (2020) proposed a hierarchical context-enhanced representation extraction framework, which simultaneously obtains sentence-level and document-level information. Lison et al. (2020) proposed a data self-labeling method based on weakly supervised learning. For relation extraction, some recent methods (Nayak and Ng, 2020; Shen and Han, 2020; Eberts and Ulges, 2021) proposed a joint extraction framework that simultaneously extracts entities and relations. Geng et al. (2020) presented an end-to-end feature extraction network based on a bidirectional tree structure. Nan et al. (2020) introduced a refinement strategy to aggregate incrementally the document information for multihop reasoning. Although deep learning approaches are beneficial for discovering potential patterns, they ignore the known association in domain dictionaries. In addition, when multiple entity tuples exist in sentences, the relation extraction method based on deep learning degrades in terms of accuracy.

Previous methods have successfully extracted entities and relations in open domains. However, entity extraction models regard characters as basic semantic units. Thus, the character dependency hidden in domains is lost. For example, in the Chinese sentence “金运激光与...签约 (Jinyun Laser signed a contract with ...)”, each character is separately fed into a model without considering the dependency hidden in the character of known domain entities, such as “金运 (Jinyun)”. Relation extraction models are based on the premise that the relation of entity tuples identified from sentences is unified without considering that relations may be diverse in different

entity tuples.

The present study first presented a novel domain named entity recognition method, prior bidirectional long short-term memory conditional random field (PBi-LSTM+CRF), to address the issues above. PBi-LSTM+CRF is based on long short-term memory (LSTM) and conditional random field (CRF). PBi-LSTM+CRF integrates neural networks with prior knowledge composed of domain dictionaries to boost characters' dependence. Second, a novel method for relation extraction, multi-entity relation extraction (MRE), was proposed. This method consists of model bidirectional LSTM attention (Bi-LSTM+ATT) and domain extraction rules. MRE aims to extract entity tuples with certain relations when multiple entity tuples exist in a sentence. Domain extraction rules aim to eliminate noise in entity relations and promote potential entity relation extraction. Finally, our two methods were evaluated on two self-built domain data. The experimental results demonstrate that our methods outperform baseline models. For instance, model PBi-LSTM+CRF achieves +1% and +6% *F1* improvement on laser industry entity (LIE) and unmanned ship entity (USE) datasets, respectively. Model Bi-LSTM+ATT achieves +2% and +1% *F1* improvement on laser industry relation (LIR) and unmanned ship relation (USR) datasets, respectively. In addition, the extraction accuracy of MRE reaches 83% and 76% on LIE pair (LIEP) and USE pair (USEP) datasets, respectively.

The main contributions of this study are as follows:

- (1) A novel entity extraction method, PBi-LSTM+CRF, was proposed. PBi-LSTM+CRF integrates prior knowledge and neural networks to solve the problem that existing methods ignore known character dependency in specific domains.
- (2) A relation extraction approach, MRE, which extracts triplets from domain text containing multiple entity tuples, was presented to promote potential entity relation extraction.
- (3) Domain prior knowledge and seed corpus were constructed from web text. In addition, the experimental results on our laser industry and unmanned ship corpus demonstrated the effectiveness of our model.

2 Model

How syntactic rules and our method, MRE, work are described in this section. The architecture of our proposed method, MRE, is shown in Fig. 1. MRE comprises three components: Bi-LSTM+ATT for relation extraction, PBi-LSTM+CRF for entity extraction, and entity tuple mining. First, our Bi-LSTM+ATT is employed to identify relations from unstructured sentences. Second, our PBi-LSTM+CRF aims to recognize domain entities from sentences. Finally, the mining

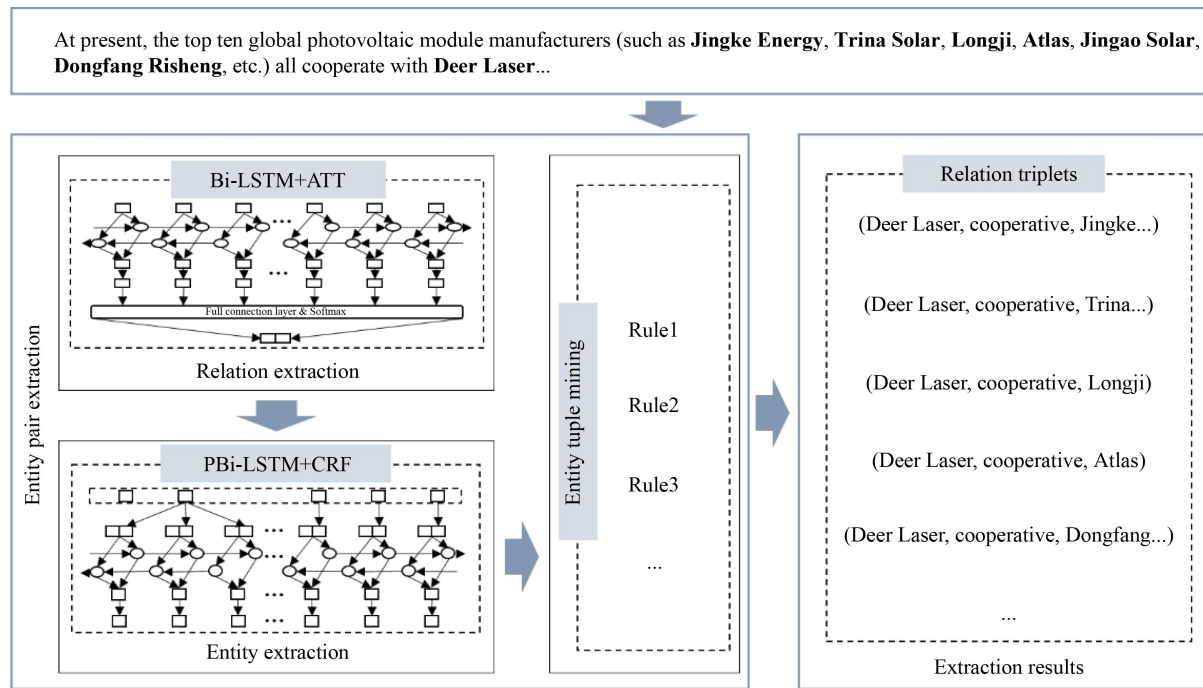


Fig. 1 MRE architecture.

algorithm extracts triplets (head entity, relation, and tail entity).

2.1 Syntactic rules

Rule-based methods are a widely used technique to obtain high-quality seed corpus quickly. Some syntactic rules are introduced to promote the construction of prior knowledge and training datasets by analyzing the corpus of the laser industry and unmanned ships, thereby efficiently extracting triplets. Our handcrafted rules contain regular expression operators (e.g., *, ?, and +) or keywords expressing certain relations.

Rule 1 (same type): The entities matched by pause patterns are identified as the same type. A pause pattern means that the entities are connected by pause and comma signs. For example, in the sentence "... A, B, ...", entities A and B types are equal.

Rule 2 (verb relation): If a sentence has two entities and the sentence satisfies the pattern $\{\wedge.*?(preposition|conjunction).*(verb).*?\$ \}$ ($\{\wedge.*?(介词|连词).*(动词).*?\$ \}$), then the relations between two entities are related to the verb. For example, in the sentence "... A and B sign contracts ...", cooperative relations indicated by the verb "sign" exist in entities A and B.

Rule 3 (coreference resolution): The sentence has entities judged as certain relations. If every character in $entity_1$ is found in $entity_2$, then $entity_1$ and $entity_2$ are equal. For example, in the sentence "... CC... DCCD...", entity CC and DCCD are identical.

Rule 4 (parallel relation): If a sentence S contains

syntactic relations COO (coordinate), ATT (affixed in), or RAD (right appended), then entities $word_i$ and $word_j$ in S are in parallel relations. For example, in the sentence "... A and B ...", entities A and B have parallel relations implied by the coordinative word "and".

Rule 5 (one-to-one relation): For the sentence identified as a certain relation, if the sentence has only two entities, $entity_1$ and $entity_2$, then a relationship exists between $entity_1$ and $entity_2$. For example, in the sentence "... A ... B ...", certain relations exist in entities A and B.

Rule 6 (many-to-many relation): If a sentence is classified as certain relations and conforms to the pattern $\{.*organization.*together[subscribe|sign|cooperate]\}$ ($\{.*方.*共同[签订|签署|合作]\}$), then the entities in this sentence show a mutual relation. For example, in the sentence "... organization A, B, and C together cooperate", cooperative relation exists among entities A, B, and C.

Rule 7 (one-to-many relation): Let E denote the entity set extracted from the sentence with certain relations, and the size of E is greater than 2. The entity $word_i$ and other entities in E do not meet Rule 3, but all entities in E except $word_i$ meet heuristic Rule 3. Then, $word_i$ and other entities in E have a relationship. For example, in the sentence "... A ... B ... C ...", A is not the same (Rule 3) as B and C, but B and C are identical. A relationship exists between A with entities B and C.

2.2 Text feature representation

Text feature representation is a method of vector modeling

text, transforming unstructured text into a highly abstract vector to mine the hidden information in a text. For a computer to understand unstructured text, each character (or word) in a sentence is converted into a character- or word-based vector. These vectors are built on a look-up table representing character (or word) embedding. They are randomly initialized with the size of all possible characters (or words). With model training, character (or word) embedding is updated iteratively to learn language representation.

2.3 Relation extraction

Relation extraction aims to detect relations among entities. It is necessary for the field of information retrieval.

Our target is to classify sentences into predefined relations and recognize all entities in sentences using entity extraction methods. Model Bi-LSTM+ATT is employed to detect relations from sentences. Figure 2 describes the frame of model Bi-LSTM+ATT. First, unstructured text is vectorized in this study with a word embedding model and Bi-LSTM neural network. An attention layer is also employed to enhance the weight of the keywords beneficial for relation extraction and reduce the influence of noisy information (e.g., a, an, and the).

In particular, let $H = \{h_1, h_2, \dots, h_n\}$ be the output of Bi-LSTM, where h_i denotes the semantic vector of the i th word. An attention layer calculates the weight of each word in a sentence, thereby obtaining a rich semantic expression. Unlike Liu and Guo (2019), we operate an attention mechanism after integrating forward and backward LSTM. Each word vector is employed as a query vector to evaluate the relative importance of the query word and other words. After the attention mechanism is

applied, each attention word vector is summed up as a final classification vector. Formally, the formulas for calculating the importance of each word are as follows

$$M = \tanh(H), \alpha = \text{softmax}(w^T M), \\ r = H\alpha^T, h^* = \sum_{i=0}^n \tanh(r_i), \quad (1)$$

where n represents the number of words constituting input sentences. w means model weights. $\alpha \in \mathbf{R}^n$ indicates an attention vector, which is the weight distribution vector of H . r denotes the weighted H . h^* represents the final vector employed for classification (Wang et al., 2021a; 2021b). After the vectorized representation of a sentence, the next step is to identify relations with the softmax function.

$$\tilde{P}(y|S) = \text{softmax}(W^s h^* + b^s), \tilde{y} = \arg \max_{y \in Y(S)} \tilde{P}(y|S), \quad (2)$$

where S denotes input sentences. $Y(S)$ denotes a predefined relation set for S . \tilde{y} means label index with max probability. W^s and b^s are weight vector and bias for the hidden state h^* , respectively. During training, this study leverages cross-entropy to define the loss function

$$J(\theta') = -\frac{1}{m} \sum_{i=1}^m t_i \log(y_i) + \lambda \|\theta'\|_F^2, \quad (3)$$

where $J(\theta')$ is a loss function, and λ is a hyperparameter of L2 regularization. L2 regularization and dropout ensure that neural networks are not overfitting. L2 regularization reduces model overfitting by constraining the weight of different parameters. t_i is a gold tag for a sentence, and y_i means prediction tag by model. m is the number of samples during model training. θ' is the model parameter to be learned during model training.

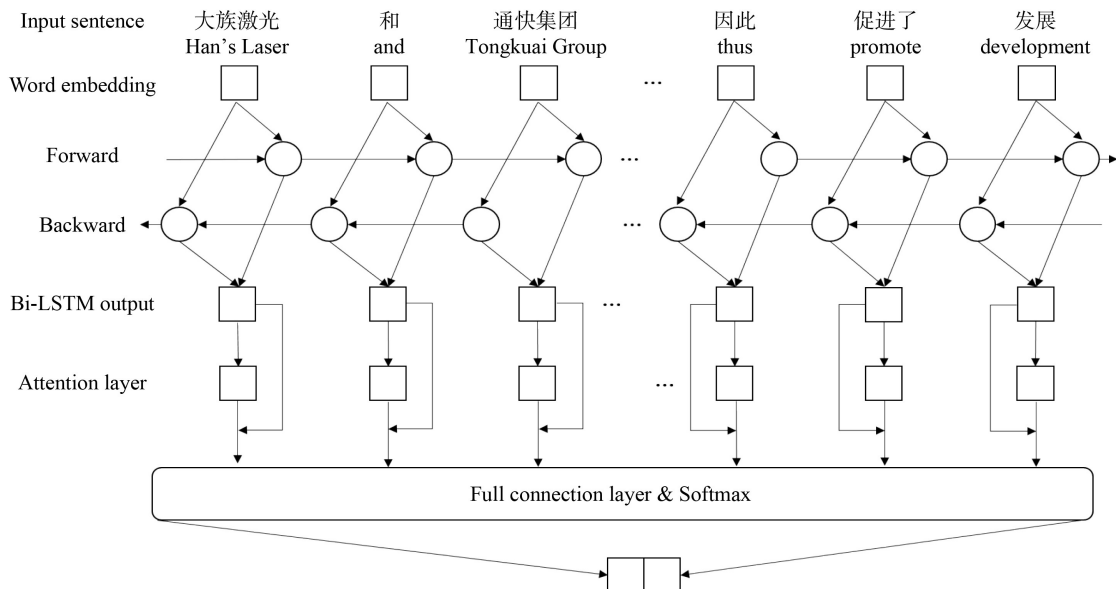


Fig. 2 Bi-LSTM model based on attention mechanism.

2.4 Entity extraction

As with most existing works on entity extraction (also called named entity recognition), the entity extraction in the present study is formulated as a sequence labeling problem. Our target is to classify each character into one of the predefined types, thereby obtaining the boundary and type of each entity.

2.4.1 Prior knowledge hidden in experience

Deep learning models autonomously learn high-level semantic features from input sentences. However, the knowledge that experts have summarized from long-term accumulation, such as domain words and synonyms, is neglected. Models can efficiently learn the pattern hidden in data and improve our model's classification accuracy by using the support of domain knowledge, thereby diminishing the cost of knowledge mining.

The prior knowledge in our work is essentially a textual semantic feature, which comprises a domain dictionary and extended characters. A domain dictionary comes from seed entities. An extended char dictionary is constructed during model training. In particular, k -dimensional prior vectors are randomly initialized for each domain word or character. The following is the procedure for integrating prior knowledge during model training.

1) Based on the entity in the prior knowledge dictionary C , longest inverse matching methods are performed on the input sentence S_i . The entity set e_i , the corresponding position set p_i , and the corresponding entity vector c_i in C are obtained. Then, an empty extended char dictionary L is initialized.

2) Each position word vector in the sentence S_i is matched with p_i . If the match is correct, the current character vector is spliced with the corresponding entity

vector c_i . Otherwise, the current character in S_i is queried whether it is in L . If it does not exist, a k -dimensional vector is randomly initialized and stored in L .

3) The word vector enhanced by prior knowledge is leveraged as the input of the model PBi-LSTM+CRF for training.

2.4.2 Ensemble of prior knowledge and text feature

As shown in Fig. 3, our entity extraction model PBi-LSTM+CRF is proposed to empower character dependency with prior knowledge. The model consists of prior knowledge, character embedding layer, Bi-LSTM layer, and CRF layer. The character embedding layer and Bi-LSTM layer are designed to learn the contextual feature of input sentences. The prior knowledge composed of domain dictionaries is employed to boost characters' dependence between domain entities. A standard CRF model is leveraged to reason best label sequence.

In the model, an input sentence is denoted by $s = \{w_1, w_2, \dots, w_i\}$, where w_i is the character that needs to be predicted. Word2vec algorithm, presented by Mikolov et al. (2013), is employed to learn the distributed character embedding representation $\{w_1, w_2, \dots, w_c\}$ of s , where w_c is the contextual representation of the character w_i . Then, the probability of w_i under the condition of the known context w_c is expressed as

$$P(w_i | w_c) = \frac{\exp(v_{w_i} \cdot h)}{\sum_{i \in V} \exp(v_{w_i} \cdot h)}, \quad (4)$$

where h is expressed as

$$h = \frac{1}{n} \sum_{i=1}^n v_{w_i}, \quad (5)$$

where v_{w_i} is the vector of the character w_i without training, and V is a dictionary library in domain corpus. n represents

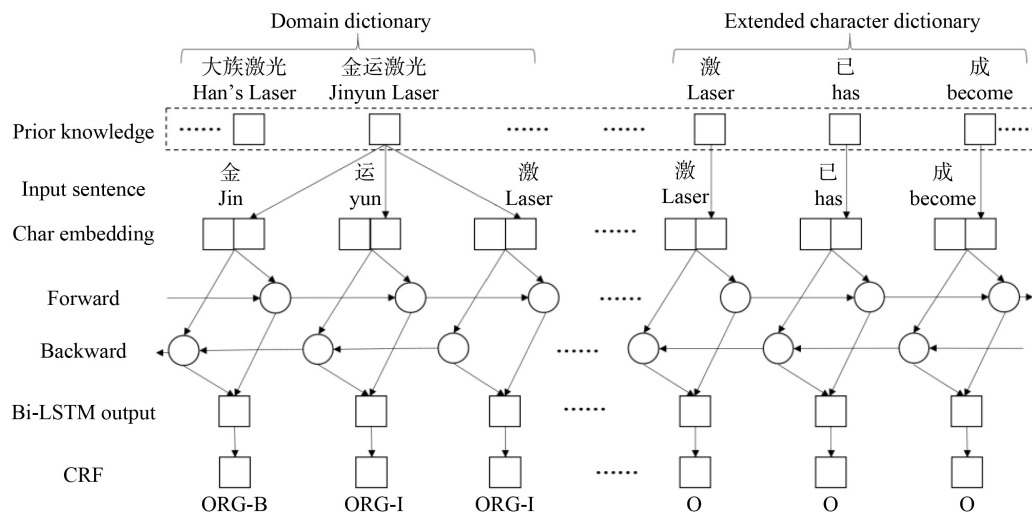


Fig. 3 Domain named entity recognition model incorporating prior knowledge.

the number of context characters of w_i in the current training.

Let $v = (v_1, v_2, \dots, v_n)$ denote the embedding of input sentence that has integrated character embedding and prior knowledge, where $v_i \in \mathbf{R}^{m+k}$, m represents the dimension of the vector obtained from Word2vec, and k represents the dimension of the vector in the prior knowledge dictionary. Bi-LSTM (Lakretz et al., 2019) is employed to learn further the deep semantics of texts (Li et al., 2019). The global contextual features (Park and Kim, 2019) $C = (f_1 \oplus b_1, \dots, f_n \oplus b_n)$, where $f_i \in \mathbf{R}^{2d}$ and $b_i \in \mathbf{R}^{2d}$ separately denote positive-order and reverse-order semantics learned from LSTM. \oplus means concatenating two vectors. Then, a linear transformation is employed to turn $f_i \oplus b_i$ into the d -dimensional vector denoted as $p = (p_1, \dots, p_n)$, where $p_i \in \mathbf{R}^d$.

Afterward, CRF (Lafferty et al., 2001) is employed to obtain the label transition matrix A , where A_{ij} represents the transition score from the i label to the j label. The scoring function is expressed as

$$\text{score}(x, y) = \sum_{i=1}^n p_{i,y_i} + \sum_{i=1}^{n+1} A_{i-1,y_i}. \quad (6)$$

The score between the input sentence x and the golden label y is the sum of the score of each position. The score of each position is composed of two parts: One of which is the p_i , and the other part is the transition matrix A of CRF. The conditional probability of each path y is given by

$$P(y|x, \theta) = \frac{e^{\text{score}(x, y)}}{\sum_{y' \in Y(x)} e^{\text{score}(x, y')}}, \quad (7)$$

where $Y(x)$ denotes all possible paths for x , and θ is the parameter to be learned. During the training period, the model's objective is to maximize the log probability of the golden tag sequence. At inference time, the best tag path y^* is predicted by obtaining the maximum score given by

$$y^* = \arg \max_{y \in Y(x)} P(y|x, \theta). \quad (8)$$

Equation (8) means to find the label sequence with the most remarkable probability score among all possible u^b label sequences. u represents the number of predefined label categories. b is the length of a sentence. Viterbi algorithm is leveraged for inference to reduce reasoning time.

2.5 Relation triplet extraction

Relation triplet extraction means extracting the triplets of the form (head entity, relation label, and tail entity) from a sentence. In our work, the motivations for mining entities and relations separately are as follows: (1) Mining separately can eliminate the information interference between different tasks. Compared with separate entity relation

extraction, joint extraction methods lose entity information when the relations between entities are missing. However, mining separately improves entity and relation extraction completeness. For example, joint extraction approaches ignore corporate entities that do not currently have a partnership. These neglected corporate entities may have cooperative relationships in the future. However, these entities have been abandoned in the knowledge graph construction stage, ultimately affecting the completeness of the knowledge graph. (2) Mining separately improves the applicability of each model. Our entity extraction is a token-level classification task, whereas relation extraction is a sequence-level classification task. Our entity linking study can easily reuse these two model structures in future work.

The detailed steps of knowledge graph construction are as follows: (1) The model Bi-LSTM+ATT identifies whether sentences have cooperative relations. These relations are used as edges in a knowledge graph. (2) Our method MRE employs the entity extraction model PBi-LSTM+CRF to extract the entity words in a sentence with cooperative relations. These entity words are used as nodes in a knowledge graph. (3) Relation triplets are obtained with our syntactic rules that help assign correct relations to each entity tuple. In particular, entity tuples are extracted through Algorithm 1.

Algorithm 1 Entity tuple mining algorithm

Input: A sentence identified as a certain relation

Output: Triplets

1. Extracting entity E from input sentences with the help of model PBi-LSTM+CRF
 2. Using Rule 3 for coreference resolution of the entities in E
 3. If the number of entities in E is equal to 2:
 4. Extracting with Rule 5
 5. Else, if the number of entities in E is greater than 2:
 6. Judging by Rule 6, whether a relationship exists in each entity tuple
 7. If there is a mutual relation between each entity tuple:
 8. Outputting entity tuple
 9. Else:
 10. Using Rule 4 for relation extraction
 11. Using Rule 7 to output entity tuples
-

3 Experiment

3.1 Evaluation metrics

Precision rate (precision, P), recall rate (recall, R), and $F1$ value are employed for model evaluation. The details for evaluation formulas are as follows

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F1 = \frac{2 * P * R}{P + R}. \quad (9)$$

True positive (TP) denotes that predictions match the ground truth. False positive (FP) and false negative (FN) represent that predictions do not match the ground truth.

3.2 Dataset

This section presents the dataset in our experiments. First, 26177 articles about the laser industry and 9069 articles about unmanned ships were collected from web pages. These articles include those from the Baidu Recruitment, Baidu Encyclopedia, industry news, Enterprise Credit System, headlines, and various blog posts. The articles about the laser industry and unmanned ship domain contain 415070 and 318847 sentences, respectively. Second, the unstructured data were manually labeled with auxiliary pattern rules. Finally, our dataset was divided into four parts: Prior knowledge, relation extraction, entity extraction, and entity tuple extraction.

3.2.1 Domain entity dictionary construction

Some extraction patterns were devised to recognize the entities from unstructured texts as seed dictionaries. Table 1 was built on Rule 1 in Section 2.1, which is beneficial for automatically constructing domain entity dictionaries. “NP” in Table 1 denotes words in our seed vocabulary. Pattern No. 0 in Table 1 was taken as an example. In the sentence “A and B, etc.”, if entity A has been in seed vocabulary, then entity B would be inferred as a seed entity.

Table 1 Patterns of extraction

Number	Pattern
0	{NP,*NP}{[and or]NP}?etc. ({NP,*NP}{[和 或]NP}?等)
1	NP[include contain like for example such as]{NP,*NP}{(and or)NP}?etc. (NP[包括 包含 如 例如 像]{NP,*NP}{(和 或)NP}?等)
2	{NP,*NP}{(and or)NP}? ({NP,*NP}{(和 或)NP}?)

According to the construction of Algorithm 2, 29384 and 7652 entities were extracted as prior knowledge from the laser industry and unmanned ship domain, respectively. Algorithm 2 illustrates our seed entity extraction procedure, which iteratively extracts domain entities from our corpus and realizes continuous and incremental extraction.

Algorithm 2 Algorithm for constructing an entity dictionary

Input: A dictionary E of entity seed and a domain corpus C
Output: Extracted entities

1. Splitting C into a sentence set S that meets extraction patterns in Table 1
2. For sentence in S do:
3. Extracting words T , the position of which is identical with NP in patterns, from sentence
4. If T is not in E :
5. Add T to E
6. End for
7. Return E

3.2.2 Relation extraction

Our relation extraction model was evaluated on the LIR

and USR datasets. Then, 4820 and 5071 sentences with a cooperative relation were selected from the laser industry and unmanned ship domain, respectively. LIR and USR datasets were constructed according to the pattern of extracting cooperative relations. Table 2 shows these patterns, which are built on Rule 2. For example, in the sentence “For promoting cooperation with enterprise B, enterprise A provides a series of products”, a cooperative relation was identified because it conforms to pattern No. 1 in Table 2.

Table 2 Patterns of extracting cooperative relation

Number	Pattern
0	{^.*?(with as well as and).*(subscribe sign expand reach).*\$} ({^.*?(与 和).*(签订 签署 展开 达成).*\$})
1	{^.*?(for).*(provide supply offer).*\$} ({^.*?(为).*(提供 供应 供货).*\$})
2	{^.*?(from).*(supply provide reach carry out offer).*\$} ({^.*?(由).*(供应 提供 达成 开展 供货).*\$})

LIR dataset consists of 2800 sentences, which were separated into training set (2000), development set (400), and test set (400). USR dataset consists of 3798 sentences, which were separated into training set (2658), development set (570), and test set (570). The amount of sentences with and without cooperative relation is equal in each set. The overall statistics on datasets is provided in Table 3.

Table 3 Dataset description of relation extraction

Dataset	Class	Training	Development	Test
LIR	Positive	1000	200	200
	Negative	1000	200	200
USR	Positive	1329	285	285
	Negative	1329	285	285

3.2.3 Entity extraction

Our entity extraction model was evaluated on the LIE and USE datasets. Our dataset was labeled with BIEO (Begin, Inside, End, Outside) annotation method (Reimers and Gurevych, 2017). LIE dataset focused on the organization class. USE dataset focused on seven types: Organization, weapon, ship, location, function, measure, and port. Before training, our dataset was divided into three parts: Training, development, and test with ratio of 70:15:15. The statistics of each class are shown in Table 4. The $\#M(\cdot)$ in Table 4 means the number of mentions.

3.2.4 Relation triplet extraction

Our relation triplet extraction method was evaluated on two datasets: LIEP and USEP. LIEP and USEP datasets comprise 1463 and 486 sentences with more than two organization entities, respectively. The two datasets were

Table 4 Dataset statistics of entity extraction

Class	#M(Training)	#M(Development)	#M(Test)
LIE dataset			
Organization	2877	617	618
Total	2877	617	618
USE dataset			
Organization	3375	710	695
Weapon	820	167	156
Ship	3197	641	702
Location	3799	898	855
Function	1073	201	251
Measure	3331	700	735
Port	1776	378	313
Total	17371	3695	3707

derived from the sentences identified as cooperative relations.

3.3 Parameter setting

In our relation extraction experiments, the maximum length of the input sentences for all models was limited to 35. For model Bi-LSTM, Adam (Kingma and Ba, 2014) was employed for optimization with the following parameters: Batch size, 128; epochs, 20; learning rate, 0.001; dropout rate, 0.2; hidden size, 256; embedding size, 300; λ , 0.7. Most of the parameters of model Bi-LSTM+ATT were the same as those of model Bi-LSTM, except that the attention size was 16. For the model logistic regression (LR), L2 regularization improved the model's generalization. The optimization method large BFGS (Broyden-Fletcher-Goldfarb-Shanno) was performed with a maximum number of iterations of 100. For Bayesian models, the Laplace smoothing parameter was 0.2.

In our entity extraction experiments, the maximum length of input sentences for all models was limited to 128. For model Bi-LSTM+CRF and Bi-LSTM, this study employed Adam (Kingma and Ba, 2014) for optimization with the following parameters: Batch size, 64; epochs, 30; learning rate, 0.001; dropout rate, 0.5; hidden size, 128; embedding size, 100. Most of the parameters of model PBi-LSTM+CRF were the same as those of model Bi-LSTM+CRF, except that the priority vector dimension

k was 50. For the hidden Markov model (HMM), the Laplace smoothing parameter was 1E-10.

All neural models in our experiments were implemented with PyTorch and scikit-learn. All the experiments were conducted on NVIDIA RTX 2080 Ti GPUs.

3.4 Result

3.4.1 Result of relation extraction

In this experiment, the metrics in Section 3.1 were adopted for evaluation. The experiments on the dataset of relation extraction were conducted. Different models were evaluated from accuracy, recall, and $F1$ value perspectives.

The experimental results of relation extraction on the test data are shown in Table 5. LR (Bunescu and Mooney, 2005) and Naive Bayes (Kim et al., 2006) perform slightly worse than neural network models. The performance of LR is slightly better than that of Naive Bayes. Given the high-level semantic features and long dependencies of sentences, the model based on Bi-LSTM performs well. Moreover, the performance of model Bi-LSTM+ATT is better than that of model Bi-LSTM. The attention mechanism evaluates the importance of each word vector of the Bi-LSTM output, thereby strengthening the influence of certain words in the domain corpus when identifying cooperative relations. In addition, model Bi-LSTM+ATT obtains the best achievement in all evaluation metrics, outperforming the second-best approach by +2% and +1% $F1$ value on LIR and USR datasets, respectively. All these observations imply that the feature information captured by the combination of Bi-LSTM and attention mechanism is more than that captured by the other baseline models; thus, this combination yields promising relation extraction results.

The $F1$ value change during model Bi-LSTM+ATT learning is reported in Figs. 4(a) and 4(c). $F1$ score synthetically measures model performance. The shift of $F1$ value in the two datasets follows the same trend. According to the two figures, the iterations from 1 to 151 increase the value of $F1$ from 0.5 to 0.95 (Fig. 4(a)) and 0.35 to 0.9 (Fig. 4(c)). The increase is followed by a steady trend after the 151st iteration. These results are consistent with the change of loss. The loss value is

Table 5 Recognition results of cooperative relation

Models	LIR dataset			USR dataset		
	Precision	Recall	$F1$ value	Precision	Recall	$F1$ value
Bi-LSTM+ATT	0.98	0.96	0.97	0.92	0.92	0.92
Bi-LSTM	0.95	0.96	0.95	0.92	0.91	0.91
LR	0.95	0.95	0.95	0.91	0.91	0.91
Naive Bayes	0.94	0.94	0.94	0.86	0.85	0.85

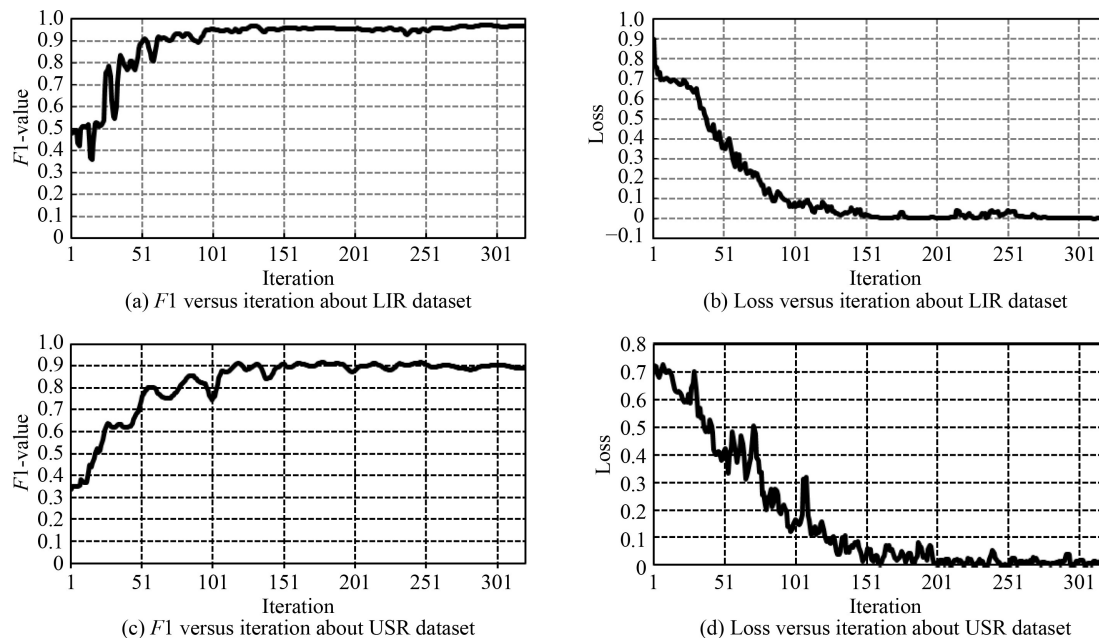


Fig. 4 Model Bi-LSTM+ATT metrics versus iteration during training.

updated each time the model performs backpropagation, thereby affecting the $F1$ value. As the loss value gradually approaches 0, the $F1$ value gradually stabilizes at the maximum value.

The loss change of model Bi-LSTM+ATT is provided in Figs. 4(b) and 4(d). Our loss change further verifies whether model Bi-LSTM+ATT is convergence. Training loss reveals the deviation between model output and the given ground truth. The loss values in the LIR and USR datasets are similar and follow the same trend. At the beginning of training, the loss does not drop until the iteration reaches approximately 30 (Fig. 4(b)) and 15 (Fig. 4(d)) times. The loss value remains steady after iteration 151, implying that a model has learned a group of stable parameters. These observations demonstrate that model parameters have stabilized in a suitable vector space, resulting in a competitive performance on test data.

3.4.2 Result of entity extraction

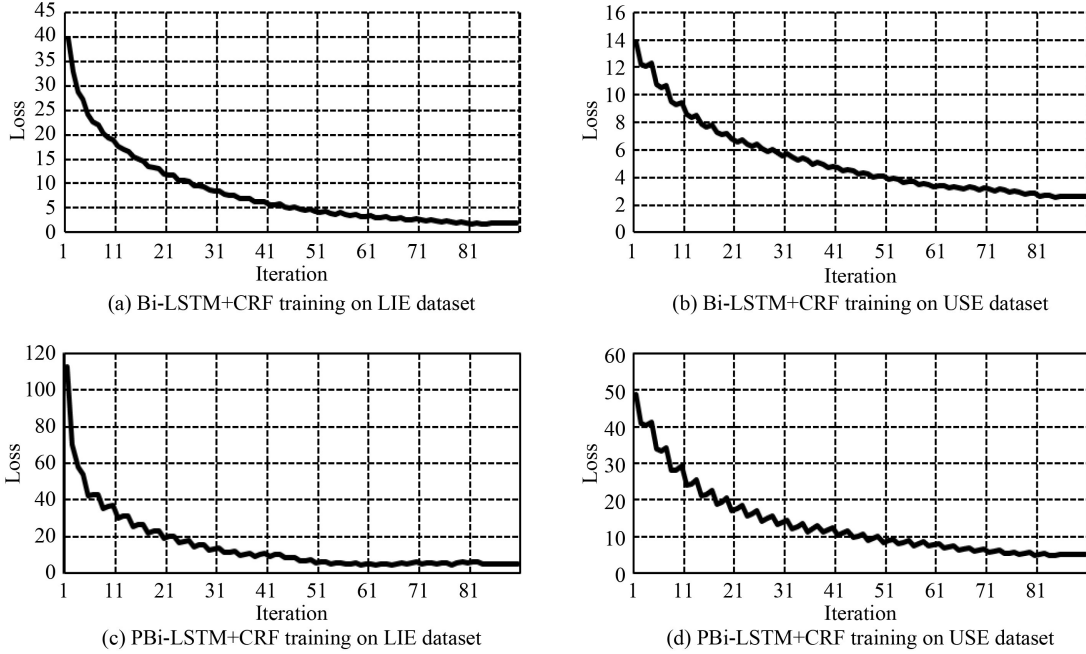
The entity extraction experiment in this study employed four models for comparison experiments. For our proposed model PBi-LSTM+CRF, an early stopping mechanism (Yao et al., 2007) was adopted to verify our model's performance accurately and quickly on a validation set. A gradient descent algorithm was employed to update the neural network parameters in the model. Dropout was utilized in the present study to choose the parameters in neural networks randomly and prevent the model from overfitting. Finally, the training was terminated when the metric $F1$ of the model had no improvement in 10 epochs.

The performance of all systems for entity extraction is shown in Table 6. Precision, recall, and $F1$ entity-level metrics were employed in our experiments. Compared with these baseline models (HMM, Bi-LSTM, and Bi-LSTM+CRF), PBi-LSTM+CRF outperforms other compared methods with a considerable margin on both datasets. The model HMM based on statistics cannot learn abstract semantics from handcrafted features; thus, the model cannot easily obtain high achievement on test data. Moreover, empowering Bi-LSTM with a CRF layer can further boost performance. Compared with Bi-LSTM, the model Bi-LSTM+CRF achieves higher than +10% improvement on two datasets in terms of recall. The model CRF causes this phenomenon. CRF aims to count the probability of dependent conversion between characters and enhance the weight of the correct prediction sequence, thereby improving the proportion of the ground truth identified from the test data. All these observations indicate that the contextualized character representations are helpful for entity extraction tasks on domain texts. Our model PBi-LSTM+CRF achieves the best performance on both datasets, outperforming the second-best approach by +1% and +6% $F1$ value, respectively. These observations demonstrating the usefulness of our prior auxiliary knowledge are in line with our motivation.

The change of loss versus iteration in the period of model learning parameters is plotted in Fig. 5. The loss value initially stands at 40 and 14 for model Bi-LSTM+CRF on LIE and USE datasets, respectively. The loss value is stable at 2 after iteration 81 during the model Bi-LSTM+CRF training. The loss value initially stands at 112 and 48 for model PBi-LSTM+CRF on LIE and USE datasets, respectively. During model PBi-LSTM+CRF

Table 6 Experimental results of entity extraction

Models	LIE dataset			USE dataset		
	<i>Precision</i>	<i>Recall</i>	<i>F1 value</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 value</i>
HMM	0.48	0.50	0.49	0.33	0.48	0.40
Bi-LSTM	0.56	0.49	0.52	0.58	0.50	0.54
Bi-LSTM+CRF	0.61	0.60	0.61	0.65	0.63	0.64
PBi-LSTM+CRF (Ours)	0.64	0.61	0.62	0.69	0.70	0.70

**Fig. 5** Model loss versus iteration during training.

training, the loss value is steady at 5 after iteration 81. Compared with the loss value of model Bi-LSTM+CRF, the loss value of our model is relatively large, and it has apparent fluctuation in all the iterations. This fluctuation is caused by the prior knowledge that gives a high weight to reasonable decoding sequence paths and positively impacts the smooth convergence of models.

3.4.3 Result of relation triplet extraction

The evaluation method of this experiment is the ratio of the number of extracted sentences and the actual number of sentences. The calculation formula is as follows

$$quality = \frac{num_{extracted}}{num_{sentence}}, \quad (10)$$

where *quality* represents the quality of extraction results, $num_{extracted}$ denotes the number of examples extracted, and $num_{sentence}$ is the total number of sentences in our dataset.

All extraction results are shown in Table 7. The sentence containing only two organization entities (one-to-one) corresponding to Rule 5 is the most common in

Table 7 Extraction results

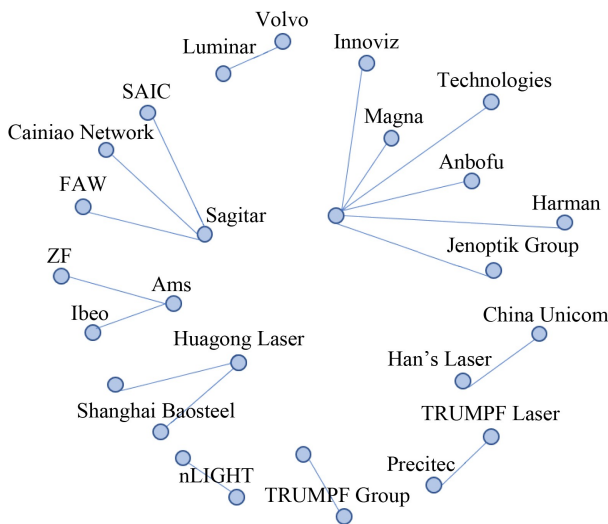
Class	LIEP dataset	USEP dataset
One-to-one	873	249
One-to-many	237	34
Many-to-many	65	53
One-to-one after coreference resolution	44	33
Total	1219	369
Quality	0.83	0.76

LIEP and USEP datasets, with 873 and 249 sentences, respectively. The least number of sentences is coreference resolution (Rule 3). Coreference resolution means that the number of entities in a sentence is two after entity disambiguation. The amount of one-to-many, which corresponds to Rule 7, is the second largest in the LIEP dataset. The proportion of many-to-many, which corresponds to Rule 6, ranks second in the USEP dataset.

The extraction quality of our approach in LIEP and USEP datasets is 0.83 and 0.76, respectively. In particular, the examples of extraction results of entity tuples in the LIEP dataset are shown in Table 8. A visualization

Table 8 Partial extraction results with only two organization entities in sentences

Sentences	Results
At the signing ceremony, Precitec was the sole seller of Joy Laser in the world.	Precitec -- Joy Laser
At the same time, Anbofu also made a small investment in LeddarTech .	Anbofu -- LeddarTech
In September this year, Sagitar Juchuang signed an agreement with FAW to deepen cooperation, and the two sides will conduct research and development cooperation on smart solid-state lidar vehicle plane-level mass production.	FAW -- Sagitar Juchuang
Like Sagitar Juchuang , it also received investment from BAIC New Energy in October last year.	BAIC New Energy -- Sagitar Juchuang
Xiaopeng Motors officially announced a strategic cooperation with China Unicom Guangzhou Branch .	Xiaopeng Motors -- China Unicom Guangzhou Branch
TRUMPF China and Shanghai Jiao Tong University signed a memorandum of cooperation for the future.	Shanghai Jiao Tong University -- TRUMPF China

**Fig. 6** Visualization of partial extraction results.

result is plotted in Fig. 6 to observe our extracted triplets further.

3.5 Analysis

This section focuses on analyzing the how and why of model improvement. Our approaches include MRE and entity extraction model PBi-LSTM+CRF.

Entity extraction: The model PBi-LSTM+CRF aims to solve the phenomenon that the existing pretrained models do not consider the character correlation hidden in the domain text. Compared with baseline models, our model shows a successful performance (Table 6), with 62% and 70% *F1* values on LIE and USE datasets, respectively. These achievements are attributed to the following: (1) Our prior knowledge instantiated by a domain dictionary enhances the association of characters hidden in domain text and boosts character weight, thereby improving the score of correct prediction paths (Figs. 5(c) and 5(d)). (2) Bi-LSTM based on LSTM incorporates forward and backward context semantics to represent the semantics of the current time. (3) Compared with the output of Bi-LSTM, CRF further rectifies the transfer dependency of

prediction labels. CRF is powerful for filtering inaccurate prediction sequences.

The way our entity extraction model outperforms other baseline models is also summarized: (1) **Text feature representation.** The baseline model HMM takes character frequency as a primary feature expression. However, our model maps each character into a 100-dimensional abstract numeric embedding vector, which is randomly initialized and updated during model training. This vectorized operation provides a potential for models to learn abstract semantic representations. (2) **Contextual semantic learning.** The baseline model HMM learns parameters depending on the statistical frequency of each character. However, our model uses a Bi-LSTM model on a per-character vector basis to mine further each character's contextual high-order abstract semantics. Bi-LSTM, comprising a forward pass and backward pass LSTM, fully explores the future and historical temporal information in the whole text sequence. (3) **Feature decoding.** The baseline model Bi-LSTM utilizes a softmax layer to project each character into a predefined class. However, our model employs a CRF layer to identify the label of each character. The model CRF operates a label transition matrix to constrain the dependencies between labels and reduce unreasonable label sequences' scores. The label transition matrix is randomly initialized and optimized during model training. (4) **Integration of domain knowledge.** The baseline model Bi-LSTM+CRF has no known domain information. However, our model incorporates domain knowledge instantiated by entity dictionaries. Each entity word in our domain entity dictionary is randomly assigned with a unique numeric vector. Each character integrates domain knowledge by adding a priori vector. After imparting domain knowledge, the characters in domain entities obtain a higher numerical weight than others in a sentence. The enhanced text vector is fed into Bi-LSTM to learn each character's contextual semantics further. (5) **Model parameter optimization.** The baseline model based on Bi-LSTM has no prior parameters to reduce search space. However, our model introduces domain entity vectors, which assign high weight scores to reasonable label sequences, resulting

in high initial loss values. Domain entity vectors also provide an available domain parameter space for our model, reducing the time for a model to search for optimal parameters.

Relation extraction: The frame MRE aims to eliminate relation noise generated by multiple entity tuples. The frame obtains a promising achievement (Table 7), with 83% and 76% accuracy on LIEP and USEP datasets, respectively. These attainments stem from the following reasons: (1) The model Bi-LSTM+ATT shows excellent prospects for relation extraction (Table 5). It is employed to recognize sentences with predefined relations. The attention mechanism used in our model distributes a higher weight to keywords than the baseline model Bi-LSTM. (2) The model PBi-LSTM+CRF, showing competitive performance in domain entity extraction, is adopted. (3) Domain rules are employed to reduce irrational triplets, thereby improving extraction accuracy.

The way our relation extraction model outperforms other baseline models is summarized as follows: (1) **Text feature representation.** The baseline model LR and Naive Bayes use only a bag-of-words model to characterize text. However, our method uses word embedding to map each word into a 300-dimensional abstract numerical vector and learn high-level semantics. Word embedding represents abstract information about each word rather than counting word frequencies. (2) **Contextual semantic learning.** The baseline model LR employs a layer of perceptrons plus activation function Sigmoid to learn text semantics. The baseline model Naive Bayes exploits feature frequency to model the semantics of each text. However, our model utilizes a Bi-LSTM network to model each word's future and historical contextual semantics simultaneously. (3) **Reinforcement of critical semantic information.** The baseline model Bi-LSTM treats each feature from the same perspective. However, our model stacks an attention mechanism on Bi-LSTM. Each word is treated as a query vector to evaluate the similarity to other words. The attention layer output semantically enhanced word vectors, and each word vector in a sentence is accumulated for vector classification.

The manual examination of the test set of our two datasets indicates that the extracted triplets are unreasonable in approximately 19% of sentences for two reasons. First, the inaccurate organization entity caused by the entity extraction model promotes low performance in entity tuple extraction. Second, the diversity of language expression negatively influences our domain rules.

4 Conclusions

In this study, model PBi-LSTM+CRF and MRE are proposed to perform entity and relation extraction, respectively. Our PBi-LSTM+CRF incorporates prior

knowledge instantiated by domain dictionaries and promotes character association. Our MRE integrates domain pattern rules and diminishes noisy information from unrelated entity tuples. Two domain corpora, including six datasets, are labeled from 26177 laser industry articles and 9069 unmanned ship articles. The experimental results on the laser industry and unmanned ship corpus reveal that our methods outperform baseline models with high *F1* value performance. For example, the model PBi-LSTM+CRF achieves 62% (+1%) and 70% (+6%) *F1* value on LIE and USE datasets, respectively. The model Bi-LSTM+ATT reaches 97% (+2%) and 92% (+1%) *F1* value on LIR and USR datasets, respectively. In addition, the extraction accuracy of MRE reaches 83% and 76% on LIEP and USEP datasets, respectively. Our future work will focus on (1) combining prior knowledge with other models and (2) integrating domain knowledge and rules into entity disambiguation tasks.

References

- Bunescu R C, Mooney R J (2005). A shortest path dependency kernel for relation extraction. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. Vancouver: Association for Computational Linguistics, 724–731
- Eberts M, Ulges A (2021). An end-to-end model for entity-level relation extraction using multi-instance learning. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 3650–3660
- Geng Z, Chen G, Han Y, Lu G, Li F (2020). Semantic relation extraction using sequential and tree-structured LSTM with attention. Information Sciences, 509: 183–192
- Grishman R (1995). The NYU system for MUC-6 or where's the syntax? In: Proceedings of the 6th Conference on Message Understanding. Columbia, MD: Association for Computational Linguistics, 167–175
- Hearst M A (1992). Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th Conference on Computational Linguistics. Nantes: Association for Computational Linguistics, 539–545
- Humphreys K, Gaizauskas R, Azzam S, Huyck C, Mitchell B, Cunningham H, Wilks Y (1998). University of Sheffield: Description of the LaSIE-II system as used for MUC-7. In: Proceedings of the 7th Message Understanding Conference. Fairfax, VA, M98-1007
- Isozaki H, Kazawa H (2002). Efficient support vector classifiers for named entity recognition. In: Proceedings of the 19th International Conference on Computational Linguistics. Taipei: Association for Computational Linguistics, 1–7
- Kim S B, Han K S, Rim H C, Myaeng S H (2006). Some effective techniques for Naive Bayes text classification. IEEE Transactions on Knowledge and Data Engineering, 18(11): 1457–1466
- Kingma D P, Ba J (2014). Adam: A method for stochastic optimization.

- arXiv preprint. arXiv:1412.6980
- Lafferty J D, McCallum A, Pereira F C N (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the 18th International Conference on Machine Learning*. San Francisco, CA: Morgan Kaufmann Publishers Inc., 282–289
- Lakretz Y, Kruszewski G, Desbordes T, Hupkes D, Dehaene S, Baroni M (2019). The emergence of number and syntax units in LSTM language models. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis, MN: Association for Computational Linguistics, 11–20
- Li X, Feng J, Meng Y, Han Q, Wu F, Li J (2020). A unified MRC framework for named entity recognition. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 5849–5859
- Li Z, Yang F, Luo Y (2019). Context embedding based on Bi-LSTM in semi-supervised biomedical word sense disambiguation. *IEEE Access*, 7: 72928–72935
- Lin X D, Peng H, Liu B (2006). Chinese named entity recognition using support vector machines. In: *International Conference on Machine Learning and Cybernetics*. Dalian: IEEE, 4216–4220
- Lison P, Barnes J, Hubin A, Touileb S (2020). Named entity recognition without labelled data: A weak supervision approach. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 1518–1533
- Liu G, Guo J (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337: 325–338
- Luo Y, Xiao F, Zhao H (2020). Hierarchical contextualized representation for named entity recognition. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. New York, NY: AAAI Press, 8441–8448
- Mikolov T, Sutskever I, Chen K, Corrado G S, Dean J (2013). Distributed representations of words and phrases and their compositionality. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems*. Lake Tahoe, NV: Curran Associates Inc., 3111–3119
- Miller G A (1995). WordNet: A lexical database for English. *Communications of the ACM*, 38(11): 39–41
- Nan G, Guo Z, Sekulić I, Lu W (2020). Reasoning with latent structure refinement for document-level relation extraction. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 1546–1557
- Nayak T, Ng H T (2020). Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. New York, NY: AAAI Press, 8528–8535
- Park S, Kim Y (2019). A method for sharing cell state for LSTM-based language model. In: *International Conference on Intelligence Science: Computer and Information Science*. Beijing: Springer, 81–94
- Reimers N, Gurevych I (2017). Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Copenhagen: Association for Computational Linguistics, 338–348
- Shen Y, Han J (2020). Joint extraction of entity and relation with information redundancy elimination. arXiv preprint. arXiv:2011.13565
- Shibuya T, Hovy E (2020). Nested named entity recognition via second-best sequence learning and decoding. *Transactions of the Association for Computational Linguistics*, 8: 605–620
- Thomas A, Sangeetha S (2020). Deep learning architectures for named entity recognition: A survey. In: *Advanced Computing and Intelligent Engineering*. Singapore: Springer, 215–225
- Waldis A, Mazzola L (2021). Nested and balanced entity recognition using multi-task learning. arXiv preprint. arXiv:2106.06216
- Wang T, Hirst G (2009). Extracting synonyms from dictionary definitions. In: *Proceedings of the International Conference RANLP*. Borovets: Association for Computational Linguistics, 471–477
- Wang X, Chang Y, Sugumaran V, Luo X, Wang P, Zhang H (2021a). Implicit emotion relationship mining based on optimal and majority synthesis from multimodal data prediction. *IEEE MultiMedia*, 28(2): 96–105
- Wang X, Kou L, Sugumaran V, Luo X, Zhang H (2021b). Emotion correlation mining through deep learning models on natural language text. *IEEE Transactions on Cybernetics*, 51(9): 4400–4413
- Yamada H, Kudo T, Matsumoto Y (2002). Japanese named entity extraction using support vector machine. *Transactions of Information Processing Society of Japan*, 43(1): 44–53
- Yao Y, Rosasco L, Caponnetto A (2007). On early stopping in gradient descent learning. *Constructive Approximation*, 26(2): 289–315
- Zhou G, Su J, Zhang J, Zhang M (2005). Exploring various knowledge in relation extraction. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Ann Arbor, MI: Association for Computational Linguistics, 427–434