doi:10.3772/j.issn.1006-6748.2021.04.005

Positive unlabeled named entity recognition with multi-granularity linguistic information①

Ouyang Xiaoye(欧阳小叶)[*,**], Chen Shudong②[*,**], Wang Rong[***]
([*] Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, P. R. China)
([**] University of Chinese Academy of Sciences, Beijing 100049, P. R. China)
([***] Key Laboratory of Space Object Measurement Department, Beijing Institute of Tracking and Telecommunications Technology, Beijing 100094, P. R. China)

Abstract

The research on named entity recognition for label-few domain is becoming increasingly important. In this paper, a novel algorithm, positive unlabeled named entity recognition (PUNER) with multi-granularity language information, is proposed, which combines positive unlabeled (PU) learning and deep learning to obtain the multi-granularity language information from a few labeled instances and many unlabeled instances to recognize named entities. First, PUNER selects reliable negative instances from unlabeled datasets, uses positive instances and a corresponding number of negative instances to train the PU learning classifier, and iterates continuously to label all unlabeled instances. Second, a neural network-based architecture to implement the PU learning classifier is used, and comprehensive text semantics through multi-granular language information are obtained, which helps the classifier correctly recognize named entities. Performance tests of the PUNER are carried out on three multilingual NER datasets, which are CoNLL2003, CoNLL 2002 and SIGHAN Bakeoff 2006. Experimental results demonstrate the effectiveness of the proposed PUNER.

Key words: named entity recognition (NER), deep learning, neural network, positive-unlabeled learning, label-few domain, multi-granularity (PU)

0 Introduction

Named entity recognition (NER) refers to the task of recognizing named entities in text and classifying them into specified types[1]. NER is also a foundation task in natural language processing (NLP), and supports downstream applications such as relation extraction[2], translation[3], question and answer[4]. At present, traditional methods based on supervised learning use a large amount of high-quality labeled data for NER[5]. However, neural NER typically requires a large amount of manually labeled training data, which are not always available in label-few domain, such as biological/medical/ military. Training neural NER with limited labeled data can be very challenging[6-7].

Researchers have investigated a wide variety of methods and resources to boost the performance of label-few domain NER, e.g., annotation learning and reinforcement learning[8], domain-adaptive fine-tuning[9], a fully Bayesian approach to aggregate multiple sequential annotations[10], adversarial transfer network[11], joint sentence and token learning[12], weak supervision to bootstrap NER[13]. Wheras most of the previous studies have injected expert knowledge into the sequence labelling model, which is often critical when data is scarce or non-existent. This work presents a positive unlabeled learning approach, which is positive unlabeled named entity recognition (PUNER), using some positive instances and multi-granularity linguistic information to automatically annotate all unlabeled instances. Positive unlabeled (PU) learning refers to learning a classifier through unlabeled instances and positive instances, classifying unlabeled instances by this classifier, and finally making all unlabeled instances into annotation instances[14-15]. PUNER solves the problem of a large amount of unlabeled data in the label-few domain by PU learning, and effectively parses rich semantic information to identify correct named entities through multi-granular language information.

① Supported by the National Natural Science Foundation of China (No. 61876144) and the Strategy Priority Research Program of Chinese Academy of Sciences (No. XDC02070600).
② To whom correspondence should be addressed. E-mail: chenshudong@ime.ac.cn
Received on Sep. 16, 2020

This paper has the following three contributions. (1) Designed a novel algorithm PUNER, which continuously iterates the unlabeled data through the PU learning method, and combines the neural network-based PU classifier to identify all named entities and their types in the unlabeled data. (2) In PU classifier, there is a multi-granularity language information acquisition module, which integrates multi-granularity embedding of characters, words, and sentences to obtain rich language semantics in the context and helps to understand the meaning of sentences. (3) The experimental results show that PUNER is 1.51% higher than the most advanced AdaPU algorithm on the three multilingual NER data sets, and the performance of PUNER on SIGHAN Bakeoff 2006 is higher than that on CoNLL 2003 and CoNLL 2002 due to the different number of training set.

1 Related work

1.1 Named entity recognition

The NER usually adopts a supervised learning approach that uses a labeled dataset to train the model. In recent years, neural network has become the mainstream of NER[16-17], which achieves most advanced performance. Many works use the long short-term memory (LSTM) and conditional random field (CRF) architecture. Ref. [18] further extended it into bidirectional LSTM-convolutional neural networks (CNNs)-CRF architecture, where the CRF module was added to optimize the output label sequence. Ref. [19] proposed task-aware neural language model (LM) termed LM-LSTM-CRF, where character-aware neural language models were incorporated to extract character-level embedding under a multi-task framework.

1.2 Label-few domain NER

The aim of label-few domain modelling is to reduce the need for hand annotated data in supervised training. A popular method is distant supervision, which relies on external resources such as knowledge bases to automatically label documents with entities that are known to belong to a specific category. Ref. [8] utilized the data generated by distant supervision to perform new type named entity recognition in new domains. The instance selector is based on reinforcement learning and obtains the feedback reward, aiming at choosing positive sentences to reduce the effect of noisy annotation. Ref. [9] proposed domain-adaptive fine-tuning, where contextualized embeddings are first fine-tuned to both the source and target domains with a language modelling loss and subsequently fine-tuned to source domain labelled data. Refs[20,21] generalized this approach with the Snorkel framework which combines various supervision sources using a generative model to estimate the accuracy of each source. Ref. [22] presented a distant supervision approach to NER in the biomedical domain.

1.3 Positive unlabeled learning NER

PU learning is a distant supervision method, which can be regarded as a special classification task, that is, learning how to train a classifier with a small number of positive instances and many unlabeled instances. AdaSampling[23] first randomly selects a part of the unlabeled instances as the negative instances for training, then the process of sampling, modeling, and prediction is repeated for each iteration, and final predicted probability uses the average of the T iterations as the probability of the final prediction. Ref. [24] proposed the unbiased positive-unlabeled learning, and Ref. [25] adopted a bounded non-negative positive-unlabeled learning. Ref. [10] proposed a fully Bayesian approach to the problem of aggregating multiple sequential annotations, using variational expectation-maximization (EM) algorithm to compute posterior distributions over the model parameters. Ref. [13] relied on a broad spectrum of labelling functions to automatically annotate texts from the target domain. These annotations are then merged using a hidden Markov model which captures the varying accuracies and confusions of the labelling functions.

For the label-few domain NER, the PU learning method can solve the problem of only a small amount of labeled data and a large amount of unlabeled data. At the same time, combining the neural network model to realize the PU classifier can obtain multi-granular sentence semantic information and identify named entities. Therefore, a novel PUNER algorithm is adopted, which applies PU learning with multi-granular language information to perform NER in the label-few domain.

2 The proposed PUNER algorithm

2.1 Problem formalization

PU learning can be regarded as a special form of two-class (positive and negative) classification methods, when there are only given a set of positive instances P and a set of unlabeled instances that contains both positive and negative instances. This work uses the binary labeling mechanism for NER tasks, rather than the mainstream B-begin I-inside O-outside (BIO) or B-begin I-inside O-outside E-end S-single (BIOES)

mechanism. This is because the defect of positive instances affects the accuracy of BIO or BIOES mechanism labeling, and the binary labeling mechanism can avoid this effect well. Therefore, the NER task here can be regarded as a binary classification task.

Suppose that there is a unlabeled dataset $U = \{s_1, \cdots, s_N\}$, where $s = \{w_1, \cdots, w_n\}$ represents a sentence with $\{w_i\}_i^n = 1$ as words, and $y \in \{0, 1\}$ denotes a class label, $y = 1$ means w_i is a named entity, otherwise $y = 0$. In positive dataset $P = \{w_1, \cdots, w_m\}$, all the words $\{w_k\}_{k=1}^m$ have the label $y = 1$, which means they are all named entities. The goal is to recognize all the named entities and their types in unlabeled dataset U.

2.2 Algorithm overview

The algorithm of the novel PU learning is shown in Algorithm 1, which is inspired by Ref. [26]. It is a two-step approach, first selecting reliable negative instances from the unlabeled dataset U, then using the positive instances and reliable negative instances to train a classification model for new instances prediction.

Algorithm 1 PUNER Algorithm

Data: Positive dataset P and unlabeled dataset U

Result: Predicted classification of all instances $y \in \{0, 1\}$

1. $T_0 \leftarrow P$; //the initial positive training data;
2. $S_0 \subset U$; //treat all unlabeled instances in U as negative instances, get S_0 as initial negative training data;
3. $g_{ner}^1 \leftarrow PULe\_Classifier(P, S_0)$ //PU learning classifier g_{ner}^1 using $\lceil P, y = 1 \rceil \cup \lceil S_0, y = 0 \rceil$ as the initial training dataset;
4. $U_1^L \leftarrow g_{ner}^1(U)$//use g_{ner}^1 to classify unlabeled data U, then get the labeled data U_1^L;
5. $S_1 \leftarrow extract\_Negatives(U_1^L)$ //get negative instances from the labeled data U_1^L;
6. $RN_1 \leftarrow S_0$; //get the initial set of reliable negative instances RN_1;
7. $S_1 \leftarrow S_0$;
8. $T_1 \leftarrow P$;
9. while ($|S_i| \leqslant |S_{i-1}|$ and $|P| \subset |T_i|$) do:
10. $i \leftarrow i + 1$
11. $g_{ner}^i \leftarrow PULe\_Classifier(P, RN_{i-1})$
12. $U_i^L \leftarrow g_{ner}^i(U)$
13. $RN_i \leftarrow extract\_Negatives(U_i^L)$
14. $T_i \leftarrow extract\_Positives(U_i^L)$
15. return (g_{ner}^i)//use g_{ner}^i as the final classifier.

The specific description of the PU learning algorithm is as follows. Lines 1 – 8 of the algorithm are the initialization of reliable negative instances from U. All unlabeled instances in U are treated as negative instances at the beginning. S_0 is the initial negative training data sampling form U, P is the initial positive training data, a binary classifier g_{ner}^1 is trained by P in conjunction with S_0. Then, the unlabeled dataset U can be classified automatically by the classifier g_{ner}^1. The negative instances from the labeled set are chosen as the initial reliable negative instances RN_1. Lines 9 – 15 of the algorithm are to iteratively update the set of reliable negative instances and the classifier until the most accurate classification result is reached. The classifier is used to train the datasets P and RN_1, and classify the RN_1 to get the new positive instances T_i and reliable negative instances RN_i. Line 9 shows the stop condition of the iteration, which means the reliable negative instances have no more changers, and all positive instances are contained in T_i. Under this stop condition, it is shown that the best classifier g_{ner}^i has been generated. All instances in unlabeled dataset U can be unbiased and consistently estimated.

In this paper, the PU learning classifier g_{ner} is a neural-network-based architecture with multi-granularity linguistic information used to recognize named entities and their types, and the specific introduction of g_{ner} is shown in the next section.

2.3 PU learning classifier g_{ner}

In this section, a neural-network-based architecture is adopted to implement PU learning classifier g_{ner}, and this architecture is shared by different entity types, as shown in Fig. 1.

2.3.1 Multi-granularity word processor

In this module, word processor semantically extracts meaningful word representation from different granularities, i.e., the character-granularity representation $e_c(w)$, the word-granularity representation $e_w(w)$, and the sentence-granularity representation $e_s(w)$.

For the word w in the sentence s, the convolution network[27] is used for the char-granularity representation $e_c(w)$ of w, the fine-tuned Stanford's GloVe word embeddings tool[28] for the word-granularity representation $e_w(w)$ of w, and the fine-tuned Bert embedding tool[17] for the sentence-granularity representation $e_s(w)$ of w. The final word presentation is obtained by concatenating these three parts of embeddings:

$$e(w) = [e_c(w) \oplus e_w(w) \oplus e_s(w)] \quad (1)$$

where, $\oplus$ denotes the concatenation operation. Thus, a sequence of word vector $\{v_t\}$ is got.

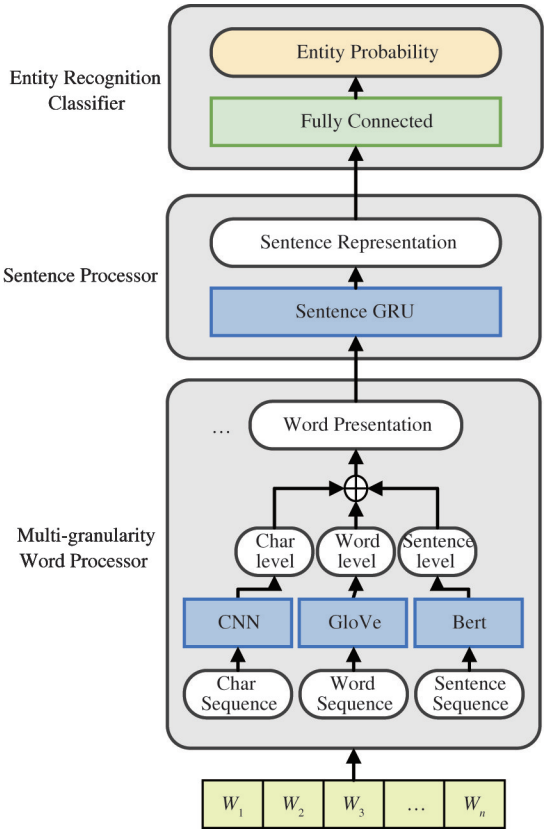

Fig. 1 Architecture of PU learning classifier

The word vector is obtained through the concatenation of multi-granularity linguistic information, that is to obtain multi-granularity features such as char, words, and sentences, and cooperate with the task model. Specifically, this work first uses CNN to generate char-level embedding, GloVe to generate word-level embedding, and Bert to generate sentence-level embedding, and then concatenates the three granular embeddings to obtain a more comprehensive and rich language semantics in the context, which further helps understanding the meaning of the sentence. Finally, it is more effective to cooperate with the upper sentence processor module.

2.3.2 Sentence processor

Based on the word vector $\{v_t\}$, the sentence processor employs a layer of gated recurrent unit (GRU)[29] to learn the contextual information of the sentence, which uses a hidden state vector $\{h_t\}$ to remember important signal. At each step, a new hidden state is computed based on previous hidden state using the same function.

$$\begin{aligned} z_t &= \sigma(W_z v_t + U_z h_{t-1}) \\ r_t &= \sigma(W_r v_t + U_r h_{t-1}) \\ \hat{h}_t &= \tanh(W_h v_t + U_h(r_t \Theta h_{t-1})) \\ h_t &= (1 - z_t) \Theta h_{t-1} + z_t \Theta \hat{h}_t \end{aligned} \tag{2}$$

where, z_t and r_t are an update gate and a reset gate, $\sigma(\cdot)$ is a sigmoid function, W_z, W_r, W_h, U_z, U_r and U_h are parameters. $e(w_k/s)$ is the representation of w_k given s.

2.3.3 Entity recognition classifier

The sentence representation $e(w/s)$ is taken as the entity detection classifier's input, and the probability of the positive class $f(w/s)$ is defined as

$$f(w/s) = \sigma(\boldsymbol{W}_p^{\mathrm{T}} e(w/s) + b) \tag{3}$$

where $\sigma(\cdot)$ is a sigmoid function, $\boldsymbol{W}_p^{\mathrm{T}}$ is a trainable parameter vector and b is the bias parameter. The prediction of the word given label y is modeled by

$$l(f(w/s), y) = |y - f(w/s)| \tag{4}$$

The cross-entropy loss function to learn a better $f(w/s)$ is minimized and defined as

$$-y_i \log(f(w/s)) - (1 - y_i) \log(1 - f(w/s)) \tag{5}$$

After training, PU classifier is used to perform label prediction. However, since a distinct classifier for each entity type is established, the type with the highest prediction probability (evaluated by $f(w/s)$) is chosen. The predictions of other classifiers will be reset to 0. For sentence $s = \{w_1, w_2, w_3, w_4, w_5, w_6\}$, if the label predicted by the classifier of a given type is $L = \{0,1,0,0,1,1\}$, then consider $\{w_2\}$ and $\{w_5, w_6\}$ as two entities of the type.

3 Experiments

In order to demonstrate the performance and adaptability of the algorithm, several methods on three multilingual datasets are compared and details of the implementation and analysis of the experimental results are given.

3.1 Compared methods

Six methods are chosen to compare their performance with the proposed PUNER. The first four are supervised learning methods, which are Stanford NER (MEMM)[30] adds jump features between observation sequences, Stanford NER (CRF)[31] uses global normalization, BiLSTM[32] is combined by forward LSTM and backward LSTM, BiLSTM + CRF[32] uses the BiLSTM as baseline, and learn an optimal path by CRF in the last layer. The last two are applied to the label-few domain, Matching directly uses the constructed named entity positive instances to label the testing set, AdaPU[33] is an adapted PU learning algorithm for NER.

In addition, the partial structure of PUNER is also changed, and the performance of three variations of the proposed MGNER is compared. $PUNER_{ELMO}$ uses ELMo[16] to do sentence embedding instead of Bert;

$PUNER_{biLSTM}$ replaces GRU with BiLSTM neural network; and $PUNER_{att\_}$ implements entity processor module without attention mechanism.

3.2 Data sets

PUNER is evaluated on CoNLL 2003[34], CoNLL 2002[35] and SIGHAN Bakeoff 2006[36]. The corpora statistics of the three datasets are shown in Table 1. These three datasets are labeled with four types, person (PER), location (LOC), organization (ORG), and miscellaneous (MISC), and the training set (TRAIN), development set (DEV) and testing set (TEST) are officially segmented.

CoNLL2003 is an English dataset, collected from Reuters. There are four types of entities in this data set, namely PER, LOC, ORG, and MISC. The official split training set is used for model training, testa is used for development and testb is used for testing in the experiments, which contains 23 407, 5918 and 5620 entities, respectively. Besides, there are about 45.6 k additional unlabeled entities.

CoNLL2002 is a Spanish NER dataset, collected from Spanish EFE News Agency. It is also annotated by PER, LOC, ORG, and MISC types. The esp. train set is used for model training, esp. testa is used for development set and esp. testb is used for testing in the experiments. The TRAIN, DEV and TEST data sets contain 18 752, 4324 and 3551 entities, respectively.

SIGHAN Bakeoff 2006 is a Chinese dataset using multiple data sets provided by different institutions for evaluation. This dataset is also labeled with four types, PER, LOC, ORG, and MISC. It has about 32 317 entities in the training set (ner. train), 3667 entities in development set (ner. dev) and 7403 entities in the testing set (ner. test).

For the qualification of the label-few domain NER, each training set of the dataset is used for training. And it should be noted that the data annotation information during training is not used. The method of building named entity dictionary given in Ref. [33] is used to construct positive instances. For CoNLL2003, most popular and common names of person, location and organizations from Wikipedia are collected to construct the dictionary. For CoNLL2002, Google translator is used to translate the English PER, LOC, ORG, MISC dictionary into Spanish. And for SIGHAN Bakeoff 2006, a dictionary based on Baidu Baike is built.

3.3 Implementation details

If the comparison methods and PUNER method are all in the identical experimental environment, the results of these experiments will be copied directly, otherwise the methods will be reproduced in the context of this paper.

Table 1 Corpora statistics for the CoNLL(en), CoNLL(sp) and Bakeoff (ch) datasets

Datasets		TRAIN	DEV	TEST
CoNLL (en)	sentence	14 987	3466	3684
	entities	23 407	5918	5620
CoNLL(sp)	sentence	8322	1914	1516
	entities	18 752	4324	3551
Bakeoff (ch)	sentence	20 864	2317	4635
	entities	32 317	3667	7403

The proposed algorithm is implemented using Pytorch libraries. A random search[37] is used for superparameter optimization, and the best performance setting is chosen as the final setting. In this experiment, the Adam optimizer with the learning rate decay is applied. The learning rate starts from 0.001 and begins to decrease by 0.9. The batch size is set to 20. The word presentation consists of three parts, pretrained GloVe word embedding, sentence Bert embedding, along with a randomly initialized training CNN encoder for character embeddings. And the dimensionality of word embedding is set as 300. In order to prevent over-fitting, all the GRU layers dropout rates are set to 0.4. Besides, the positive instances in the PU learning algorithm are selected following previous work[33].

3.4 Results

Experiment results on the CoNLL 2003, CoNLL 2002 and SIGHAN Bakeoff 2006 datasets are shown in Table 2. As can be seen from Table 2 (2), among the methods applied in label-few domain, performance of the proposed PUNER is better than others on three different datasets. PUNER achieves excellent results in label-few domain.

The last set of methods shown in Table 2(2) are deformations of the proposed PUNER. By using Bert for embedding instead of ELMo, it increases $F1$ score 1.4% on the CoNLL(en) dataset, 1.3% on the CoNLL(sp) dataset and 0.8% on the Bakeoff (ch) dataset. Choosing GRU to extract semantics instead of BiLSTM, $F1$ score is improved by 0.8% on the CoNLL (en) dataset, 0.3% on the CoNLL(sp) dataset and 0.7% on the Bakeoff (ch) dataset. The attention mechanism improves $F1$ score by 1.2% on the CoNLL (en) dataset, 0.8% on the CoNLL(sp) dataset and 1% on the Bakeoff (ch) dataset. The initial multigranularity linguistic information of word embedding has important effect on subsequent tasks, and at the

same time, the attention mechanism also significantly helps to extract important semantics.

Table 2 $F1$ scores on CoNLL (en), CoNLL (sp) and Bakeoff (ch) testing set for NER

(1) Four supervised learning methods

Method	CoNLL (en)	CoNLL(sp)	Bakeoff (ch)
MEMM	86.13%	81.14%	86.68%
CRF	87.94%	82.63%	89.13%
BiLSTM	88.30%	80.28%	88.77%
BiLSTM + CRF	90.01%	84.74%	91.73%

(2) Methods for label-few domain

Method	CoNLL(en)	CoNLL(sp)	Bakeoff (ch)
Matching	44.90%	42.23%	45.93%
AdaPU	82.94%	75.85%	83.41%
$\text{PUNER}_{\text{ELMo}}$	83.08%	75.94%	83.93%
$\text{PUNER}_{\text{biLSTM}}$	83.81%	76.95%	84.76%
$\text{PUNER}_{\text{att}-}$	83.24%	76.45%	84.07%
PUNER	84.48%	77.24%	85.02%

Analyzing different performance results of these three datasets, the ranking of $F1$ value on the three data sets are Bakeoff (ch), CoNLL (en) and CoNLL (sp). $F1$ score on the Chinese dataset is 0.54% higher than English dataset and 7.78% higher than Spanish dataset. Considering the data set analysis information provided in Table 1, it is believed that the performance difference between different data sets is mainly caused by the difference in the number of sentences and entities. Specifically, the number of Bakeoff (ch) sets is larger than that of CoNLL (en) and CoNLL (sp), and the number of data sets directly affects the effect of model training. From the experimental results, $F1$ score on the CoNLL (sp) is the worst. This may also be caused by the low quality of the positive instances of CoNLL (sp), because the Spanish positive samples are translated from the positive instances of CoNLL (en). The translation process may produce noise data, which affects accuracy.

Moreover, compared with the previous AdaPU, the performance of the proposed method is improved, because the combined use of Bert, GRU neural network and attention mechanism can improve the semantic understanding of context. However, compared with Table 2(1), the performance of PUNER is still worse than that of supervised learning.

Experiments are conducted on three datasets, using different sizes of training sets to train the model, and studying the impact on $F1$ values. On three data sets, 20%, 40%, 50%, 60%, 80%, and 100% training sets are selected for training PUNER, respectively. Fig. 2 describes the results of this study on three datasets. It can be seen from Fig. 2 that as the number of training sets increases, the overall performance of the model also increases, although there are fluctuations. Therefore, the amount of data has an impact on the performance of the model. Meanwhile, the performance of the supervised learning method BiLSTM + CRF in Fig. 2 shows that the gap between supervised learning and unsupervised learning and research on unsupervised learning are also very meaningful.

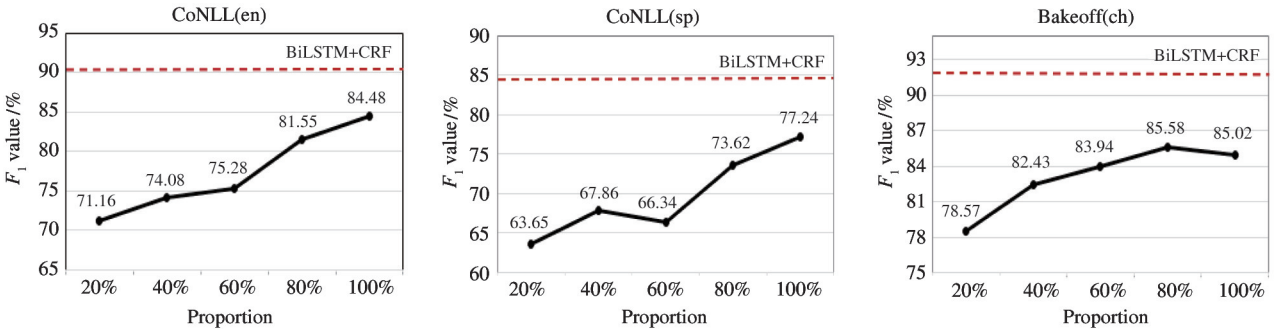

Fig. 2 F1 of PUNER on the testing set of CoNLL (en), CoNLL (sp) and Bakeoff (ch) datasets for training using different segmentation of the training dataset. The dotted line indicates the F_1 value obtained by using the supervised learning method BiLSTM + CRF

4 Conclusion

A novel PUNER algorithm for label-few domain is proposed, which uses PU learning algorithm combined with deep learning method to obtain multi-granularity language information for NER task. In PUNER, PU learning uses the positive instances and many unlabeled instances to effectively solve the labeling problem. Meanwhile, the neural network-based architecture is used to implement the PU learning classifier, which obtains multi-granularity linguistic information and facilitates named entity labeling. Experimental results show that PUNER achieves excellent results in label-

few domain on three multilingual datasets. In future research, graph convolutional network will be considered to model richer sentence semantics.

References

[1] Lin Y, Liu L, Ji H, et al. Reliability-aware dynamic feature composition for name tagging[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019: 165-174

[2] Zheng S, Han X, Lin Y, et al. DIAG-NRE: a neural pattern diagnosis framework for distantly supervised neural relation extraction[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019: 1419-1429

[3] Zhang B, Xiong D, Su J, et al. Neural machine translation with deep attention[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, 42(1): 154-163

[4] Esposito M, Damiano E, Minutolo A, et al. Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering[J]. *Information Sciences*, 2020, 514: 88-105

[5] Liu T, Yao J, Lin C, et al. Towards improving neural named entity recognition with gazetteers[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019: 5301-5307

[6] Wang B R, Chen Y J. NNL: a domain-specific language for neural networks[J]. *High Technology Letters*, 2020, 26(2): 160-167

[7] Lan H Y, Wu L Y, Han D, et al. Assembly language and assembler for deep learning accelerators[J]. *High Technology Letters*, 2019, 25(4): 386-394

[8] Yang Y, Chen W, Li Z, et al. Distantly supervised NER with partial annotation learning and reinforcement learning[C]// Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, USA, 2018: 2159-2169

[9] Han X, Eisenstein J. Unsupervised domain adaptation of contextualized embeddings for sequence labeling[C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 2019: 4237-4247

[10] Simpson E, Gurevych I. A Bayesian approach for sequence tagging with crowds[C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Hong Kong, China, 2019: 1093-1104

[11] Zhou J T, Zhang H, Jin D, et al. Dual adversarial neural transfer for low-resource named entity recognition[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019: 3461-3471

[12] Kruengkrai C, Nguyen T H, Aljunied S M, et al. Improving low-resource named entity recognition using joint sentence and token labeling[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (online), 2020: 5898-5905

[13] Lison P, Hubin A, Barnes J, et al. Named entity recognition without labelled data: a weak supervision approach[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (online), 2020: 1518-1533

[14] Li W, Guo Q, Elkan C, et al. A positive and unlabeled learning algorithm for one-class classification of remote-sensing data[J]. *IEEE Transactions on Geoscience and Remote Sensing*, 2011, 49(2): 717-725

[15] Calvo B, Larranaga P, Lozano J A. Learning Bayesian classifiers from positive and unlabeled examples[J]. *Pattern Recognition Letters*, 2007, 28(16): 2375-2384

[16] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[C]// Proceedings of the North American chapter of the Association for Computational Linguistics, New Orleans, USA, 2018: 2227-2237

[17] Devlin J, Chang M, Lee K, et al. Bert: pre-training of deep bidirectional transformers for language understanding[J]. *Association for Computational Linguistics*, 2019, 1: 4171-4186

[18] Ma X, Hovy E. End-to-end sequence labeling via bi-directional LSTM-CNNS-CRF[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 2016: 1064-1074

[19] Liu L, Shang J, Xu F F, et al. Empower sequence labeling with task-aware neural language model[C]// Proceedings of 32nd AAAI Conference on Artificial Intelligence, New Orleans, USA, 2018: 5253-5260

[20] Ratner A, Bach S H, Ehrenberg H R, et al. Snorkel: rapid training data creation with weak supervision[C]// Proceedings of 43rd International Conference on Very Large Data Bases, München, Deutschland, 2017: 269-282

[21] Ratner A, Bach S H, Ehrenberg H R, et al. Snorkel: rapid training data creation with weak supervision[J]. *The VLDB Journal*, 2020: 709-730

[22] Zhou H, Liu Z, Lang C, et al. Two-perspective biomedical named entity recognition with weakly labeled data correction[C]// Proceedings of 2020 IEEE International Conference on Bioinformatics and Biomedicine, Online Conference, 2020: 941-944

[23] Yang P, Liu W, Yang J Y H. Positive unlabeled learning via wrapper-based adaptive sampling[C]// Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 2017: 3273-3279

[24] Plessis M C D, Niu G, Sugiyama M. Analysis of learning from positive and unlabeled data[J]. *Advances in Neural Information Processing Systems*, 2014(1): 703-711

[25] Kiryo R, Niu G, Du Plessis M C, et al. Positive-unlabeled learning with non-negative risk estimator[C]// Proceedings of Neural Information Processing Systems, Long Beach, USA, 2017: 1675-1685

[26] Fusilier D H, Montesygomez M, Rosso P, et al. Detecting positive and negative deceptive opinions using PU-learning[J]. *Information Processing and Management*, 2015, 51(4): 433-443

[27] Kim Y. Convolutional neural networks for sentence classification[C]// Proceedings of the 2014 Conference on Em-

pirical Methods in Natural Language Processing, Doha, Qatar, 2014: 1746-1751
[28] Pennington J, Socher R, Manning C D. Glove: global vectors for word representation[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 2014: 1532-1543
[29] Cho K, Van M B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 2014:1724-1734
[30] Mccallum A, Freitag D, Pereira F. Maximum entropy Markov models for information extraction and segmentation [C]// International Conference on Machine Learning, Stanford, USA, 2000:591-598
[31] Finkel J R, Grenager T, Manning C D. Incorporating non-local information into information extraction systems by gibbs sampling[C]// Meeting of the Association for Computational Linguistics, Ann Arbor, USA, 2005: 363-370
[32] Rrubaa P, Aravindh A. Bidirectional LSTM-CRF for Named Entity Recognition[C]// Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation, Hong Kong, China, 2018: 531-540
[33] Peng M, Xing X, Zhang Q, et al. Distantly supervised named entity recognition using positive unlabeled learning [C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019: 2409-2419
[34] Sang E F T K, De M F. Introduction to the CONLL-2003 shared task: language independent named entity recognition[C]// Proceedings of North American Chapter of the Association for Computational Linguistics, Stroudsburg, USA, 2003: 142-147
[35] Sang E F T K. Introduction to the CONLL-2002 shared task: language-independent named entity recognition[C] // Proceedings of International Conference on Computational Linguistics, Taibei, China, 2002: 1-4
[36] Levow G. The third international Chinese language processing bakeoff: word segmentation and named entity recognition[C]// Proceedings of Meeting of the Association for Computational Linguistics, Sydney, Australia, 2006: 108-117
[37] Bergstra J, Bengio Y. Random search for hyper-parameter optimization[J]. *Journal of Machine Learning Research*, 2012, 13(1):281-305

Ouyang Xiaoye, born in 1988. She is a Ph. D candidate at Institute of Microelectronics, Chinese Academy of Sciences. She received her M. S. degree from Hefei University of Technology in 2014 and B. S. degree from Hefei University in 2010. Her research interests include the design of algorithms for deep learning, graph computing, distant supervised learning in natural language processing.