World Wide Web (2020) 23:453–470
https://doi.org/10.1007/s11280-019-00736-3

A crowd-efficient learning approach for NER based on online encyclopedia

Maolong Li[1] · Zhixu Li[1] · Qiang Yang[2] · Zhigang Chen[3,4] · Pengpeng Zhao[1] · Lei Zhao[1]

Received: 1 May 2019 / Revised: 17 August 2019 / Accepted: 22 September 2019 /
Published online: 2 December 2019

Abstract

Named Entity Recognition (NER) is a core task of NLP. State-of-art supervised NER models rely heavily on a large amount of high-quality annotated data, which is quite expensive to obtain. Various existing ways have been proposed to reduce the heavy reliance on large training data, but only with limited effect. In this paper, we propose a crowd-efficient learning approach for supervised NER learning by making full use of the online encyclopedia pages. In our approach, we first define three criteria (representativeness, informativeness, diversity) to help select a much smaller set of samples for crowd labeling. We then propose a data augmentation method, which could generate a lot more training data with the help of the structured knowledge of online encyclopedia to greatly augment the training effect. After conducting model training on the augmented sample set, we re-select some new samples for crowd labeling for model refinement. We perform the training and selection procedure iteratively until the model could not be further improved or the performance of the model meets our requirement. Our empirical study conducted on several real data collections shows that our approach could reduce 50% manual annotations with almost the same NER performance as the fully trained model.

Keywords NER · Crowdsourcing · Crowd-efficient · Named entity recognition

1 Introduction

Named Entity Recognition (NER) aims at recognizing and classifying phrases referring to a set of named entity types [9], such as *PERSON*, *ORGANIZATION*, and *LOCATION* in text. A formal definition of NER is given in Definition 1 below.

This article belongs to the Topical Collection: *Special Issue on Trust, Privacy, and Security in Crowdsourcing Computing*
Guest Editors: An Liu, Guanfeng Liu, Mehmet A. Orgun, and Qing Li

✉ Zhixu Li
zhixuli@suda.edu.cn

Extended author information available on the last page of the article.

Definition 1 Let $T = \{t_1, t_2, ..., t_e\}$ denote a set of named entity types. Given a sequence of tokens $S =< w_1, w_2, ..., w_N >$, the task of NER is to output a list of tuples $< I_s, I_e, t >$, where $I_s, I_e \in [1, N]$ are the start and the end indexes of a named entity mention in S, and $t \in T$ is the corresponding entity type of the mention.

Given an example text "United Nations official Ekeus heads for Baghdad", after NER we could have: "*[ORG United Nations] official [PER Ekeus] heads for [LOC Baghdad].*", where three named entities: *Ekeus* is a person, *United Nations* is an organization and *Baghdad* is a location.

As a core task of Natural Language Processing (NLP), NER is crucial to various applications including question answering, co-reference resolution, and entity linking, etc. Correspondingly, plenty of efforts have been made in the past decades to developing different types of NER systems. For instance, many NER systems are based on feature-engineering and machine learning, such as Hidden Markov Models (HMM) [39], Support Vector Machines (SVM) [17], Conditional Random Fields (CRF) [13], with many hand-crafted features (capitalization, numerals and alphabets, containing underscore or not, trigger words and affixes, etc.). However, these features will be useless when adopted to totally different languages like Chinese.

Later work replace these manually constructed features by combining a single convolution neural network with word embeddings [3]. After that, more and more deep neural network (DNN) NER systems are proposed [37]. However, it is well known that DNN models require a large scale annotated corpus for training. The existing open labeled data for NER model training are mostly from the newswire data [16, 34], which have few mistakes on grammar or morphology. As a result, the models trained on these gold-standard data usually have a bad performance on noisy Web texts. Besides, the NER model trained on such a general corpus could not work well on specific domains, because the contexts of entity names in specific domains are quite different from those in newswires. To get a domain-specific NER model, extra domain-specific labeled corpus for training are required, which, however, are also expensive to achieve.

To deal with the challenges above, many recent work tend to weakly supervised learning for help. On one hand, some people use the structured or semi-structured data in online encyclopedias for NER model training. As the largest open-world knowledge base, Wikipedia contains a large quantity of weakly-annotated texts with inner links of entities, as well as a well-structured knowledge base of various domains. For example, Nothman et. al. generate NER training data by transforming the inner links to Named Entity (NE) annotations through mapping the Wikipedia pages to entity types [23]. Nevertheless, this method will be useless when it comes to languages, like Chinese, which do not have so many structured texts in Wikipedia and the generated training data still have mistakes. On the other hand, some people choose crowdsourcing as an alternative way to obtain labeled data at a lower cost in a short time. For instance, Yang et. al. propose a method to improve the quality of the annotations by adversarial learning based on a common Bi-LSTM and a private Bi-LSTM, which represents annotator-generic and -specific information respectively [38]. However, they only pay attention to the crowd quality and just randomly select samples from the unlabeled set instead of selecting important samples.

In this paper, we propose a crowd-efficient learning approach for NER based on online encyclopedia, which could greatly reduce the amount of crowd annotation without hurting the precision of the NER model. Particularly, we develop a strategy to select some important samples from the unlabeled samples set for crowdsourcing instead of selecting samples at random, where we use three criteria to determine whether a sample is important. The first criterion, called **representativeness**, is to select some samples which are nearest to the centroid of clusters as a candidate sample set. The second criterion is to obtain a batch of **informative** samples from the candidate sample set by computing the entropy of the predictions of the DNN model, where we utilize the linked words in the sentences. The third criterion is **diversity** which makes sentences in a batch disparate at both lexical and syntactic levels. After getting a batch of labeled samples for crowdsourcing, a data augmentation method is proposed, which could generate a lot more training data with the help of the structured knowledge of online encyclopedias to greatly augment the training effect. Finally, we train the model on the augmented training set. The training and selection procedure conduct iteratively until the model could not be improved or the performance of the model meets our requirement.

The contributions of this paper are listed as follows:

1. We put forward a crowd-efficient training framework for NER based on online encyclopedias, which only applies light-weight crowd labelling on the weakly-annotated sentences in online encyclopedia for NER training.
2. We propose a non-trivial sample selection approach for selecting a small set of samples for crowd labelling. This selection approach takes three criteria, i.e., representativeness, informativeness and diversity of samples, into consideration in estimate the potential values of different samples for NER training.
3. We also propose a so-called data augmentation method to greatly augment the training effect with the help of structured knowledge of online encyclopedias.

Our empirical study conducted on several large real data collections shows that the NER model trained in our way could reach almost the same performance as the fully trained models, while our approach uses 50% less crowd annotation.

The remainder of this paper is organized as follows: Section 2 reviews the related work. Section 3 provides a framework of our methods. Section 4 explains how to select important samples for crowdsourcing and Section 5 introduces our data augmentation method for NER training data. After reporting our empirical study in Section 6, we conclude our paper in Section 7.

2 Related work

In the following of this section, we first introduce several state-of-art supervised NER models based on DNN and some semi-supervised NER models using encyclopedia data. After that, we also present some existing work using self-training models. Finally, we cover some related work on data augmentation technics.

Supervised DNN-based NER model Collobert et. al. firstly propose to use a single convolution neural network architecture to output a host of language processing predictions including NER, which is an instance of multi-task learning with weight-sharing [2]. The feature vectors of the architecture are constructed from orthographic features (e.g. capitalization of the first character), dictionaries and lexicons. Later work [3] replaces manually constructed feature vectors with word embeddings (a distributed representations of words in n-dimensional space). Studies have shown the importance of such pre-trained word representations for neural network based NER systems [10], combining with pre-trained character embeddings [14] or language representation models [4, 24]. After that, the Bidirectional Long Short-Term Memory(BI-LSTM) or BI-LSTM with Conditional Random Field (BI-LSTM-CRF), now widely used, is applied to NLP benchmark sequence tagging data sets, along with different word representations [11, 14]. Although the DNN models have a great performance, they need plenty of training data annotated by experts, which is expensive and time-consuming to achieve.

Semi-supervised NER using encyclopedia data Online encyclopedias, like Wikipedia, have been widely used to generate weakly labeled NER training data. The main idea is to transform the hyperlinks into NE tags by categorizing the corresponding Wikipedia pages into entity types. Some methods categorize the pages based on manually constructed rules that utilize the category information of Wikipedia [25]. Such rule-based entity type mapping methods have high precision, but low coverage. To achieve a better performance, Nothman et. al. use a classifier trained by the extra manually labeled Wikipedia pages with entity types [23]. Ni et. al. combine decoding constraint and output post-process by utilizing the Wikipedia entity type mappings to their NER system [21].

Crowdsourcing for NER It is costly and non-scalable in time and money to acquire a massive amount of labeled training data annotated by experts. Instead, crowdsourcing is an alternative way to obtain labeled data at a lower cost. Snow et al. demonstrate that non-experts annotations were quite useful for training new systems by collecting labeled results for several NLP tasks from Amazon Mechanical Turk [29].

In recent years, many work have been done on how to use crowdsourcing in NLP tasks such as classification [1, 7] and relation extraction [6]. For crowdsourcing on NER task, Dredze et. al. consider it as a multi-label problem [5], while Rodrigues et. al. take the worker identities into consideration and propose an Expectation-Maximization (EM) algorithm for CRF-based sequence labeling with multiple annotators to jointly learn the CRF model parameters, the reliabilities of the annotators and the estimated ground truth label sequences [26]. To further reduce the noises in crowdsourcing data, Nguyen et. al. consider not only how to best aggregate sequential crowd labels but also how to best predict sequences in unannotated texts by learning a crowd representation [20]. Moreover, Yang et. al. propose a method to improve the quality of the annotations by adversarial learning based on a common Bi-LSTM and a private Bi-LSTM, which represents annotator-generic and -specific information respectively [38]. Unfortunately, all of these works only focus on how to reduce the noise in the crowd annotations but not consider how to better select the samples for workers.

Data augmentation Data augmentation refers to a kind of methods for constructing iterative optimization or sampling algorithms via the introduction of unobserved data or latent variables [32]. So far, data augmentation has been proved effective in most image related tasks by flipping, rotating, scaling or cropping the images [35]. When it comes to NLP tasks, data augmentation methods need to be designed specifically [33, 36]. Xu et. al. propose a method by changing the direction of relations for relation classification task [36]. They split a relation into two sub-paths: *subject-predicate* and *object-predicate*, and then change the order of the two sub-paths to obtain a new data sample with the inverse relationship. Wang et. al. propose a novel approach for automatic categorization of annoying behaviors [33]. They replace the words in tweet with its k-nearest-neighbor (knn) words, found by the cosine similarity of word embeddings. When it comes to Chinese word embeddings, the entity mention often meets Out-of-Vocabulary (OOV) problem.

3 Framework

We propose a crowd-efficient training approach for NER based on online encyclopedia. Figure 1 depicts an overview of our approach. Given a number of encyclopedia texts for a specific purpose, we conduct k-means clustering on all the sentences with links to obtain representative sentences, which are those nearest to the centroid of the clusters. Next, we compute the informativeness of these representative sentences by utilizing the linked words and the confidence of sentence prediction, which is the output of the DNN model. Besides, we would take the diversity in different aspects into account when selecting samples into the batch set. Finally, we augment the newly labeled sentences set with the help of the structured data of online encyclopedia for updating the NER model in the next iteration. The selection procedure and retraining procedure are conducted iteratively until the DNN model meets the convergence point. Some details about the framework are given below.

- **Online Encyclopedia Preprocessing.** We select sentences with inner links of the encyclopedia articles in one domain (e.g. *Geography and places* in Wikipedia) as our source

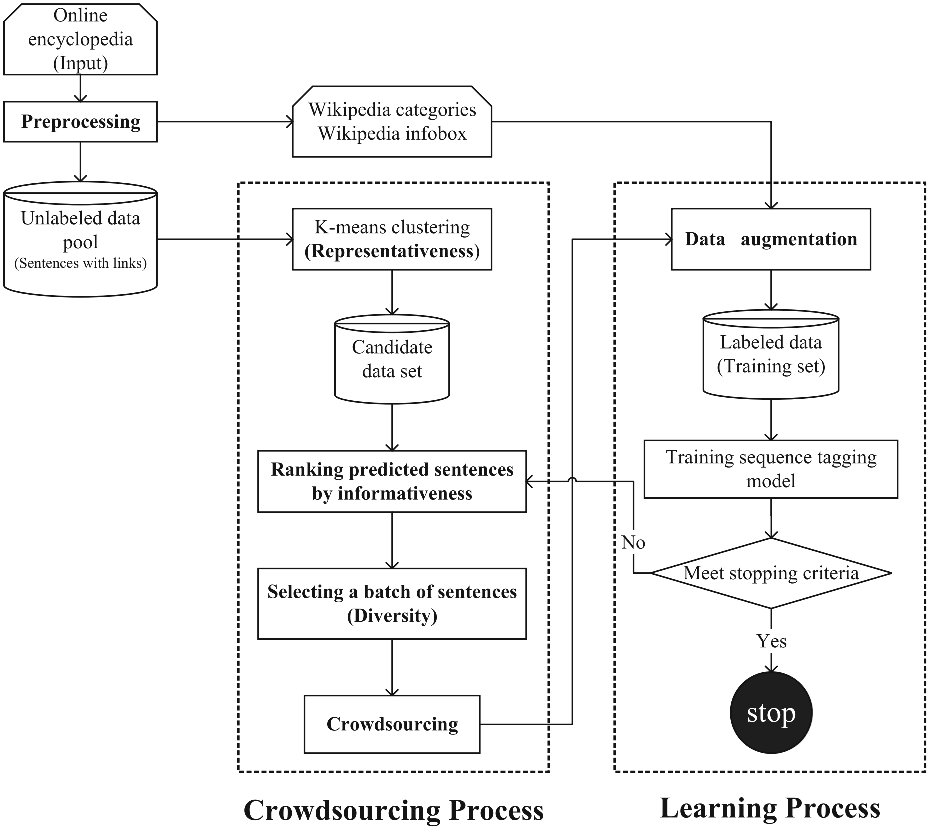


Figure 1 The framework of the approach

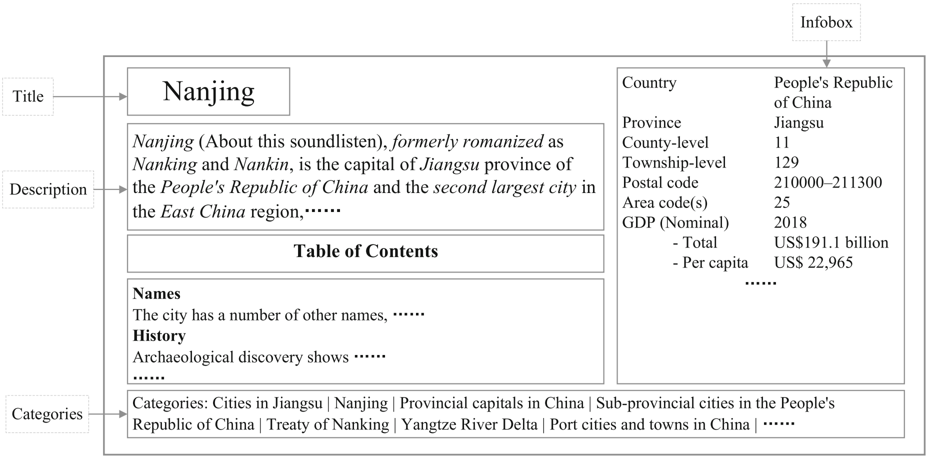
data set. Meanwhile, we extract the **infobox** and **categories** in Wiki pages of all the linked words in the source data set. An example of a Wiki page is shown in Figure 2.

- **Sample Selection for Crowd Labelling.** Instead of selecting samples for crowdsourcing randomly, we propose to select samples from the unlabeled sample set based on three criteria. The three sample selection criteria will be introduced in details in Section 4.
- **Model Re-training iteratively with Data Augmentation.** After getting a batch of labeled set, we apply a data augmentation method on these labeled data with the help of knowledge in online encyclopedia, and retrain the DNN model again. The crowd labelling and retraining process would be conducted iteratively until the DNN model reaches the convergence point or the performance of it meets our requirement.

4 Sample selection criteria for crowd labelling

For sample selection on NER annotation, we have an assumption that the cost of annotating a sentence is proportional to the number of words in the sentence. In order to save the crowdsourcing cost, we propose a strategy (Algorithm 1) to select more important sentences for crowdsourcing, from which the DNN model can learn more features or knowledge. Since it is time consuming to retrain the DNN model, we use batch-based sample selection and maximize the contribution of a batch on three criteria: *Representativeness, Informativeness, Diversity*. We will introduce the three criteria in the following subsections.

After selecting a batch of sentences, we send these sentences to crowd annotators, which are required to identify the predefined types of entities in the sentences. The annotators are given a guideline document along with 10 annotation examples, as shown in Figure 3. To reduce the noise, each sample from the batch is annotated by one annotator and checked by the other annotator.



Nanjing

Nanjing (About this soundlisten), *formerly romanized* as *Nanking* and *Nankin*, is the capital of *Jiangsu* province of the *People's Republic of China* and the *second largest city* in the *East China* region,······

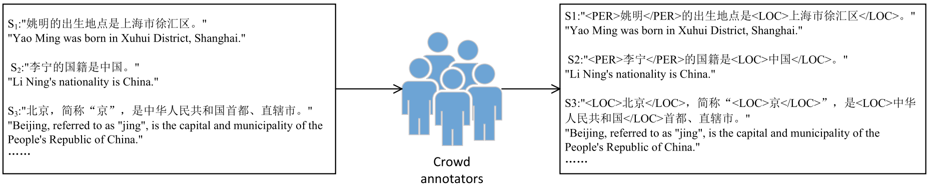
Table of Contents

Names
The city has a number of other names, ······
History
Archaeological discovery shows ······
······

Country	People's Republic of China
Province	Jiangsu
County-level	11
Township-level	129
Postal code	210000–211300
Area code(s)	25
GDP (Nominal)	2018
- Total	US$191.1 billion
- Per capita	US$ 22,965
······	

Categories: Cities in Jiangsu | Nanjing | Provincial capitals in China | Sub-provincial cities in the People's Republic of China | Treaty of Nanking | Yangtze River Delta | Port cities and towns in China | ······

Figure 2 An example Wiki page of Nanjing, where the italic parts have inner-links



S_1:"姚明的出生地点是上海市徐汇区。"
"Yao Ming was born in Xuhui District, Shanghai."

S_2:"李宁的国籍是中国。"
"Li Ning's nationality is China."

S_3:"北京，简称“京”，是中华人民共和国首都、直辖市。"
"Beijing, referred to as "jing", is the capital and municipality of the People's Republic of China."
……

S1:"<PER>姚明</PER>的出生地点是<LOC>上海市徐汇区</LOC>。"
"Yao Ming was born in Xuhui District, Shanghai."

S2:"<PER>李宁</PER>的国籍是<LOC>中国</LOC>。"
"Li Ning's nationality is China."

S3:"<LOC>北京</LOC>，简称“<LOC>京</LOC>”，是<LOC>中华人民共和国</LOC>首都、直辖市。"
"Beijing, referred to as "jing", is the capital and municipality of the People's Republic of China."
……

Figure 3 An example of NE annotations

Algorithm 1 An multi-criteria selection algorithm.

```
Input  : $\mathbb{X}^U$, $M$, $n_b$, $\beta$, where $\mathbb{X}^U$ denotes the unlabeled sentence set, $M$ denotes the
         DNN model, $n_b$ denotes the size of selected-sample batch, $\beta$ is the threshold
         of the similarity.
Output: $\mathbb{X}^{Batch}$, a set of labeled sentence with cardinality of $n_b$.
$i \leftarrow 0, \mathbb{X}^R \leftarrow \emptyset, I \leftarrow \emptyset, \mathbb{X}^{Batch} \leftarrow \emptyset$;
$\mathbb{X}^R \leftarrow Representative(\mathbb{X}^U)$, $\mathbb{X}^R$ is a set of representative samples;
$I \leftarrow Informative(\mathbb{X}^R, M)$, $I$ is a set of informativeness score of all representative
samples;
while $i \leq n_b$ and $\mathbb{X}^R \neq \emptyset$ do
    $s \leftarrow Max(I, \mathbb{X}^R)$, $s$ is the most informative sample;
    $\mathbb{X}^R \leftarrow \mathbb{X}^R - \{s\}$;
    if $Diversity(s, \mathbb{X}^{Batch}) > \beta$ then
        $\mathbb{X}^{Batch} \leftarrow \mathbb{X}^{Batch} \cup \{s\}$;
        $i \leftarrow i + 1$;
    end
end
return $\mathbb{X}^{Batch}$;
```

4.1 Representativeness

We consider representative samples are those with many similar samples in semantics, which means the samples with high representativeness are less likely to be an outlier. Therefore, adding them to the training set will have effects on a large number of unlabeled samples. We get representative samples by selecting the samples which are nearest to the centroid of k-means clusters. The distance between two samples is presented by the cosine similarity of their embeddings. The embedding of any sentence s_i is denoted by $s_i = \sum_{j=9}^{12} h_j$, where $h_9, h_{10}, h_{11}, h_{12}$ are the output of last four hidden layers of the pre-trained BERT model [4], because the last four hidden layers extract semantic features. The input representation of samples is the same as that in BERT paper.

4.2 Informativeness

After getting the representative samples as a candidate set $\mathbb{X}^R$, we take the informativeness into consideration. In our task, we use two measures to evaluate the informativeness:

entropy-based measure for whole sentences and difference-based measure for named entities. We choose informative samples, about which the current model is most uncertain, from representative samples set $\mathbb{X}^R$. First, we introduce the entropy-based measure for sentences. As introduced by Claude Shannon [27], entropy refers to disorder or uncertainty, and the definition of entropy used in information theory is directly analogous to the definition used in statistical thermodynamics. Intuitively, we consider the informativeness of a sample as the entropy degree of the top-k tagging result. This means a sample may be informative for the DNN model if the entropy of top-k tagging sequences is great. Due to benefits of BILSTM-CRF, which uses viterbi decoding [8] to output the top-k most likely tag sequences of the sentences as well as their probabilities, we can compute the entropy of its top-k decoding results. Given a character sequence C of a sample sentence $s \in \mathbb{X}^R$, we get a tag sequence set $T = \{T_1, T_2, \ldots, T_k\}$ and their probability set $P = \{p_1, p_2, \ldots, p_k\}$ from the top-k output of the model, and then the entropy of the tagging result of the sentence is calculated by the following equation:

$$E(s) = H(P) = -\sum_{i=1}^{k} p_i \log p_i \tag{1}$$

Then, we explore the difference-based measure for named entities in sentences by utilizing the linked words in them. Since the linked words in the sentences from Wikipedia are checked by many people, they are more likely to be a meaningful word or entity. Therefore, we assume that if the word segmentation contained entities predicted by the DNN model and that contained linked words differs greatly, the model hasn't learned this sentence thoroughly. We consider the difference as how many different words in word segmentation in two cases. To get two different word segmentations, we use a directed acyclic graph (DAG) to conduct the word segmentation where some certain words would be surely segmented according to some specific conditions. We build a DAG for all possible word segmentations (see an example in Figure 4) where the node denotes one word and the directed path from *start* to *end* denotes a possible word segmentation of the sentence.

Moreover, the linked words in the sentences and the predicted entities are definitely contained in the DAG. Then, we use dynamic programming to find the most probable word segmentation [30] under two different conditions: 1. one word segmentation must contain linked words, 2. the other word segmentation must contain predicted entities. Thus, the difference of two words segmentations is computed in this way:

$$Diff(W_1, W_2) = \frac{|W_1 \cup W_2 - W_1 \cap W_2|}{|W_1 \cup W_2|} \tag{2}$$

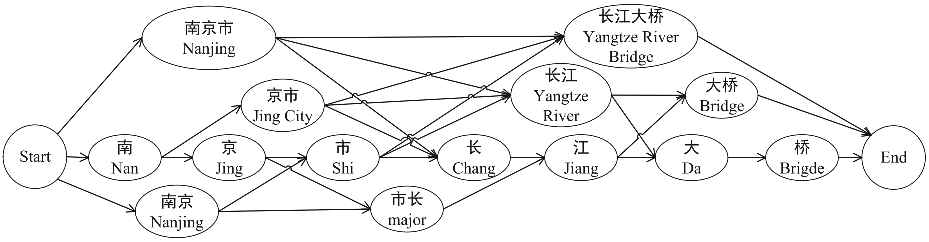


Figure 4 The example of the DAG for sentence:"Yangtze River Bridge of Nanjing". After using dynamic programming on the DAG, we get a word segmentation: "Nanjing", "Yangtze River Bridge"

where W_1 and W_2 denote the word set of two different word segmentations of sentence s respectively.

At last, the informativeness of a sample s is the combination of entropy and difference as follows:

$$Info(s) = E(s) + Diff(W_1, W_2) \tag{3}$$

We prefer selecting a batch of samples with high informativeness score since the sentences with high informativeness score are the ones the DNN model is uncertain about. Therefore, it needs to learn further features from these sentences.

4.3 Diversity

When we select samples into the batch set $\mathbb{X}^{Batch}$, we should take the diversity into consideration, which means to maximize the utility of this batch where the samples have high variance to each other. We compute the diversity from two levels: word-level and sentence-level. For word-level diversity, we compute the cosine similarity of the words in sentences which are represented by Chinese word embeddings [18]. For sentence-level diversity, we compute the cosine similarity of the sentences embedding denoted by $s_i = \sum_{j=5}^{8} hj$, which are the 5-8 hidden layer states of BERT and they extract syntax features. Therefore, the similarity of two sentences is computed by $Sim(s_b, s_u) = Sim(v_b^w, v_u^w) + Sim(v_b^s, v_u^s)$, where s_b and s_u denote sentence from batch set $\mathbb{X}^{Batch}$ and sentence from unlabeled set $\mathbb{X}^U$ respectively, and v^w means the feature vector at word level, while v^s at sentence level. For each sentence s selected from candidate set $\mathbb{X}^R$, we compute the similarity between this sentence and every sentence in the batch set $\mathbb{X}^{Batch}$ as follows:

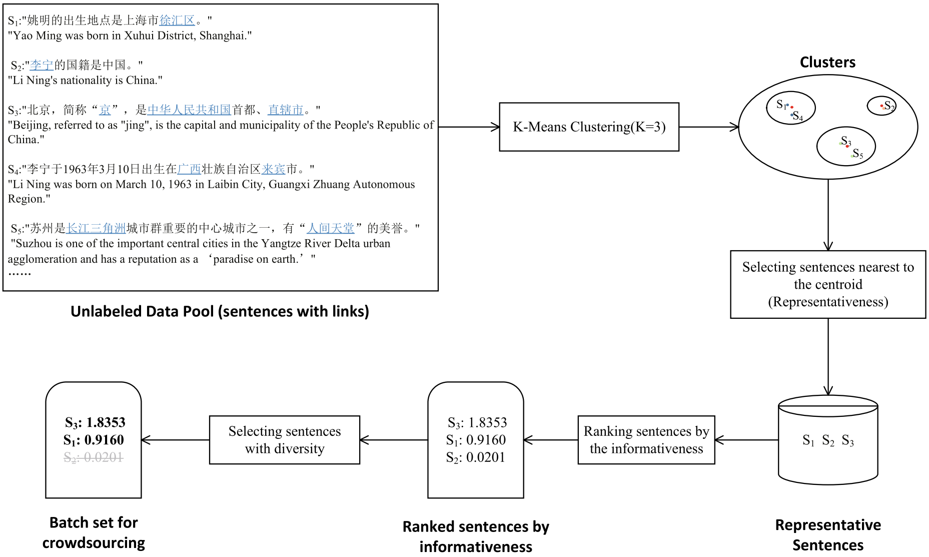
$$Diversity(s, \mathbb{X}^{Batch}) = \frac{\sum_{x \in \mathbb{X}^{Batch}} Sim(s, x)}{|\mathbb{X}^{Batch}|} \tag{4}$$

In this criterion, we avoid selecting too similar samples in a batch by setting a threshold β. Only the samples whose diversity score is larger than β can be added into the batch. We give an example of the multi-criteria selection process in Figure 5.

5 Data augmentation

When we train a deep learning model, what we actually do is tuning its parameters to map particular inputs to some outputs. To get a good performance, we need to show the model a corresponding amount of examples, which leads to complex models in need of large amounts of labeled data. For instance, the Google's Neural Machine Translation(GNMT) model has 380M parameters with 340M words in training data, and the popular BERT-Large model has 340M parameters with 2500M words in English Wikipedia training data set. Both of these models have the state-of-art performance in corresponding task. These facts show that the models get better with the increasing number of data. Therefore, a most effective way to improve the performance of a deep learning model is to add more data to the training set. But in most circumstances, it is difficult or expensive to get plenty of extra annotated data. As an alternative, we could also develop some methods to enhance the data we have already had, which are named as data augmentation [12, 33, 35, 36].

There are many ways to augment existing data and produce more robust models in computer visions like rotation and flipping due to the invariance of convolutional neural networks [15]. Because image pixels are low-level signals, generally continuous and less related to semantics, it is not difficult to do these translations to images. By contrast, natural



S_1:"姚明的出生地点是上海市徐汇区。"
"Yao Ming was born in Xuhui District, Shanghai."

S_2:"李宁的国籍是中国。"
"Li Ning's nationality is China."

S_3:"北京，简称“京”，是中华人民共和国首都、直辖市。"
"Beijing, referred to as "jing", is the capital and municipality of the People's Republic of China."

S_4:"李宁于1963年3月10日出生在广西壮族自治区来宾市。"
"Li Ning was born on March 10, 1963 in Laibin City, Guangxi Zhuang Autonomous Region."

S_5:"苏州是长江三角洲城市群重要的中心城市之一，有“人间天堂”的美誉。"
"Suzhou is one of the important central cities in the Yangtze River Delta urban agglomeration and has a reputation as a ‘paradise on earth.’"
……

Unlabeled Data Pool (sentences with links)
K-Means Clustering(K=3)
Clusters
S_1
S_4
S_2
S_3
S_5
Selecting sentences nearest to the centroid (Representativeness)
S_1 S_2 S_3
Representative Sentences
Ranking sentences by the informativeness
S_3: 1.8353
S_1: 0.9160
S_2: 0.0201
Ranked sentences by informativeness
Selecting sentences with diversity
S_3: 1.8353
S_1: 0.9160
~~S_2: 0.0201~~
Batch set for crowdsourcing

Figure 5 First, we have five unlabeled sentences with links as unlabeled data pool. Then, we get three clusters:$\{S_1, S_4\}$, $\{S_2\}$ and $\{S_3, S_5\}$, where red dot means the centroid of each cluster. We select sentences nearest to the centroid as representative sentences set $\{S_1, S_2, S_3\}$. We rank sentences by their informativeness in descending order and greedily choose sentences with higher informativeness. We first choose S_3, then S_1 whose syntax is different from S_3. However, when it comes to S_2 whose syntax is similar with S_1, we would not select S_2 into the batch set

language tokens are discrete: each word well reflects the thought of humans, but neighboring words do not share as much information as pixels in images do [19]. So, there is few data augmentation methods in NLP tasks since its difficulty. The existing data augmentation method used in [33] is not suitable for our model, since our model is based on character level and it is meaningless to replace characters in the Chinese sentence. Considering that Wikipedia contains some weakly labelled data, like inner links, infobox and categories, we decide to utilize it to augment the NER training data set.

We propose a novel data augmentation approach, which could generate a set of sentences from one sentence, by properly replacing each entity mention in the sentence with some other entity mentions of the same entity category. Specifically, for each entity mention in a given sentence, we first identify its most-relevant entity category, and then we augment the sentence by replacing each entity mentions with the other entities in its most related category. For example, we may replace the *Gusu District* in the sentence “The population of Suzhou Gusu District is 950,000.” with *Wuzhong District*, as the example sentence S_1 shown in Figure 6. An important assumption behind our approach is that: sentences in a right presentation form but with incorrect knowledge (which is inconsistent with the real-world case) could also be a sample for training NER. For instance, the mention *Suzhou* in sentence “The population of Suzhou Gusu District is 950,000.” is replaced with *Hangzhou*, as the example sentence S_2 shown in Figure 6.

Our approach, however, may also generate bad sentences if we replace an entity mention with another entity which has a different relation with other entities in the sentence, such as the example sentence S_3 shown in Figure 6. To reduce this side effect, we conduct the data augmentation differently according to the number of the entities in text. For

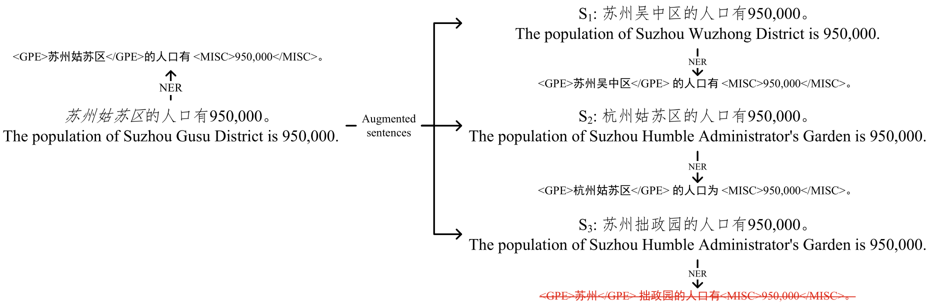

<GPE>苏州姑苏区</GPE>的人口有 <MISC>950,000</MISC>。
NER
苏州姑苏区的人口有950,000。
The population of Suzhou Gusu District is 950,000.
Augmented sentences
S1: 苏州吴中区的人口有950,000。
The population of Suzhou Wuzhong District is 950,000.
NER
<GPE>苏州吴中区</GPE> 的人口有 <MISC>950,000</MISC>。
S2: 杭州姑苏区的人口有950,000。
The population of Suzhou Humble Administrator's Garden is 950,000.
NER
<GPE>杭州姑苏区</GPE> 的人口为 <MISC>950,000</MISC>。
S3: 苏州拙政园的人口有950,000。
The population of Suzhou Humble Administrator's Garden is 950,000.
NER
~~<GPE>苏州</GPE> 拙政园的人口有<MISC>950,000</MISC>。~~

Figure 6 The NER result of Stanford NER Tagger on augmented multi-entity sentences

single-entity sentences, we could replace the entity mention simply. But for multi-entity sentences, we find some entity pairs, having the same relation with the pairs in sentences, and replace mentions with mentions of the new entity pairs. It can keep the context of entities in augmented sentences correct because the new entities with same relation often have some common attributes with the replaced entities. In spite of this, the data agumentation will still bring noises into the labeled data if the entity is linked to the wrong Wikipage. However, we focus on NER task in specific domain, like Geography, where there are few ambiguous words. Therefore, it still has an improvement on performance when we set the max replacing number n_{max} to a proper value. More details are shown as follows.

5.1 Augmenting with single-entity sentences

Given a single-entity sentence s which contains a mention m_o, we augment the original sentence by replacing the original mention m_o in the text with its related entities' mentions. First, we get the most similar category c_{sim} as follows:

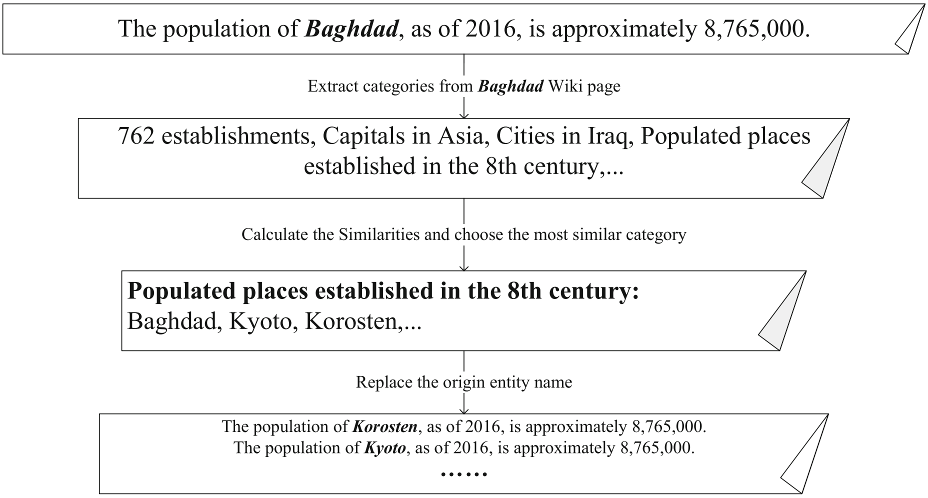
$$c_{sim} = \underset{c}{\operatorname{argmax}}\, CosSim(c_i, s)(c_i \in Cate(e_o)) \tag{5}$$

In above equation, e_o means the corresponding entity of m_o, and $Cate(e)$ means all the categories of entity e, and $CosSim(x, y)$ means the cosine similarity of two texts, which represented by their embeddings, like the BERT hidden states [4]. Then, we get the related entities set $E_r = Ent(c_{sim})$, where $Ent(c)$ denoting all the acquired entities in the category c. However, if none of words in the sentence is contained in the text of category, we set the $E_r = \emptyset$. Note that if we augment the sentence with all the entities in E_r, many similar sentences will be added into the training data set. This will have a bad influence on the model since the augmented sentences bring in a few noises. To deal with this situation, we get the final entities E_{rep} by setting a max-replacing number n_{max} through (6) below:

$$E_{rep} = \begin{cases} E_r & \|E_r\| \leq n_{max} \\ TOP(n_{max}, E_r, e_o) & \|E_r\| > n_{max} \end{cases} \tag{6}$$

where $TOP(n, E, e)$ gets the n entities most similar to e from an entities set E. The similarity of two entities is calculated by the cosine similarity using the first paragraphs of their Wikipedia articles. After that, we replace the original mention with the mentions of entities in E_{rep}. The single-entity sentence example is illustrated in Figure 7.

Unfortunately, it still brings noises into the training set when we replace the entity mention with the name of a entity, which is not at the same level with the original entity, as the


The population of ***Baghdad***, as of 2016, is approximately 8,765,000.

Extract categories from ***Baghdad*** Wiki page

762 establishments, Capitals in Asia, Cities in Iraq, Populated places established in the 8th century,...

Calculate the Similarities and choose the most similar category

Populated places established in the 8th century:
Baghdad, Kyoto, Korosten,...

Replace the origin entity name

The population of ***Korosten***, as of 2016, is approximately 8,765,000.
The population of ***Kyoto***, as of 2016, is approximately 8,765,000.
......

Figure 7 The example of one **single-entity sentence**: The entity name in the example sentence is ***Baghdad*** and the categories in the corresponding Wiki page are in the second blocks. Then, we find the most similar category is ***Populated places established in the 8th century***. There are 35 entities in this category. Finally, we replace the ***Baghdad*** with the top-n most similar entity names in the category, like ***Kyoto***, ***Korosten***

sentence S_3 shown in Figure 6. To reduce these mistakes, we should also take the attributes of the entity into consideration, which is in our future work plan.

5.2 Augmenting with multi-entity sentences

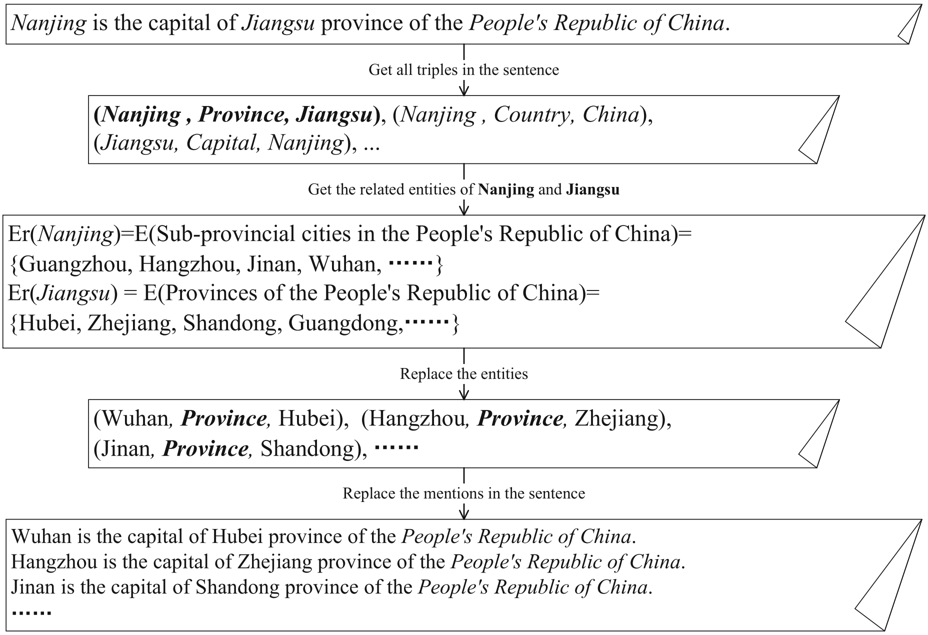
We call a triple denoted by $(subject, relation, object)$ as a Relation Triple, where *subject* and *object* are entities, *relation* denotes the relation between the two entities. Given a multi-entity sentence, we first find all the relation triples T_o in the sentence. For a triple $t_i = (s_i, r_i, o_i) \in T_o$, we get the corresponding replacing relation triple set in two conditions: 1) We get the relation triple set T_1 by only replacing the s or o in one triple. 2) We get the relation triple set T_2 by replacing both s and o of one triple. The two relation triple sets are expressed by (7), where E_{s_i} is the related entities of s_i, so is E_{o_i}. Then, the replacing relation triple set of t_i is $T_{new} = T_1 \cup T_2$.

$$\begin{aligned} T_1 &= \{t | t = (s, r, o), r = r_i, (s \in E_{s_i}, o = o_i) \vee (s = s_i, o \in E_{o_i})\} \\ T_2 &= \{t | t = (s, r, o), r = r_i, s \in E_{s_i} \wedge o \in E_{o_i}\} \end{aligned} \tag{7}$$

After that, we replace the original mentions of s_i, o_i with mentions of s, o in new relation triples T_{new}. The multi-entity sentence example is illustrated in Figure 8.

6 Experiments

We conduct a series of experiments to evaluate our proposed approaches for NER training on real-world data collections.


Nanjing is the capital of *Jiangsu* province of the *People's Republic of China*.

Get all triples in the sentence

(Nanjing , Province, Jiangsu), (*Nanjing , Country, China*), (*Jiangsu, Capital, Nanjing*), ...

Get the related entities of **Nanjing** and **Jiangsu**

Er(*Nanjing*)=E(Sub-provincial cities in the People's Republic of China)= {Guangzhou, Hangzhou, Jinan, Wuhan, ⋯⋯}
Er(*Jiangsu*) = E(Provinces of the People's Republic of China)= {Hubei, Zhejiang, Shandong, Guangdong,⋯⋯}

(Wuhan, ***Province***, Hubei), (Hangzhou, ***Province***, Zhejiang), (Jinan, ***Province***, Shandong), ⋯⋯

Wuhan is the capital of Hubei province of the *People's Republic of China*.
Hangzhou is the capital of Zhejiang province of the *People's Republic of China*.
Jinan is the capital of Shandong province of the *People's Republic of China*.
⋯⋯

Figure 8 The example of one **multi-entity sentence**: The triples in the sentence is in the second block. Then, we give the example of getting the related entities of the subject and object in the triple **(Nanjing, Province, Jiangsu)**. After that, we get the new relation triples illustrated in the forth block. Finally, we replace the original mentions in the sentence with the mentions of entities in new triples

6.1 Datasets and metrics

Datasets The initial unlabeled data are derived from the Wiki pages in Geography. We use multi-criteria strategy to select 10K sentences from unlabeled data set for crowdsourcing, where each sample is assigned to two annotators: one for annotating named entities and the other for checking the annotations. We manually label 500 sentences by ourselves in Wikipedia for testing the model. Besides, we use MSRA NER data set to evaluate data augmentation, which has 46.4K sentences for training, 4.4k sentences for test [16].

Metrics Standard precision (P), recall (R) and F1-score (F1) are used as evaluation metrics. We use the conll eval script in CoNLL 2003 [31]. The hyperparameters of our model are set as follows: $drop = 0.5$, $learningrate = 0.005$, $optimizer = adam$ and $batchsize = 32$. We use the same hyperparameters in all experiments. The effectiveness of different selection strategies is evaluated by the number of samples used when the DNN model reaches the same F1 score.

6.2 Approaches for comparison

We use different selection strategies in NER annotation crowdsourcing.

- **Random Selection (RS).** It is the baseline selection strategy called Random Selection (RS) where we just select the samples from the unlabeled set for crowdsourcing at random.

- **Certain-based Selection (CS).** In the strategy, we select the most informative examples about which the current model are most uncertain, which is a popular strategy used for sample selection [28].
- **Multi-criteria Selection (MCS).** The strategy is to combine the representativeness, informativeness and diversity. First, we consider the representativeness to generate a candidate set for selection. Then, we select samples from the candidate set by the score of their informativeness and diversity.
- **Multi-criteria Selection with Data Augmentation (MCS-DA).** After we getting a batch of samples by using multi-criteria selection, we use a data augmentation to augment the labeled batch set, where the threshold (β) of the Diversity is set to 0.85.

6.3 Experimental results

Effectiveness of different selection strategies We conduct the sample selection with $batch\_size = 500$ in three strategies: RS, CS, MCS. The result is shown in Figure 9. As we can see from the figure, the CS and MCS strategies use less labeled data than RS strategy when the DNN model achieves the same performance level as the supervised learning. Furthermore, the MCS strategy has a better performance than CS strategy in the later iterations. Impressively, MCS selection strategy achieves 99% performance of the fully-trained deep model using only 50% less of the training data on the crowdsourcing dataset. There are two bottom points on F1 of RS when the number of sentences are 3K and 4.5K , because RS strategy may select some outliers (sentences with semantic errors or syntax errors).

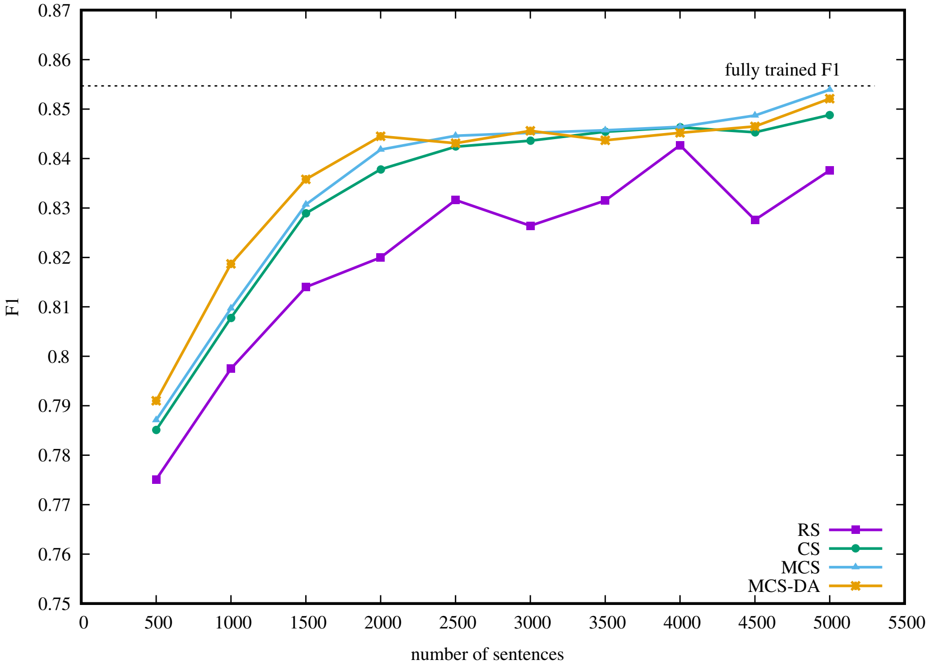


Figure 9 The supervised NER learning on crowd labeled data set with different selection strategies. The dotted line is the F1 of the fully trained model on 10K labeled dataset

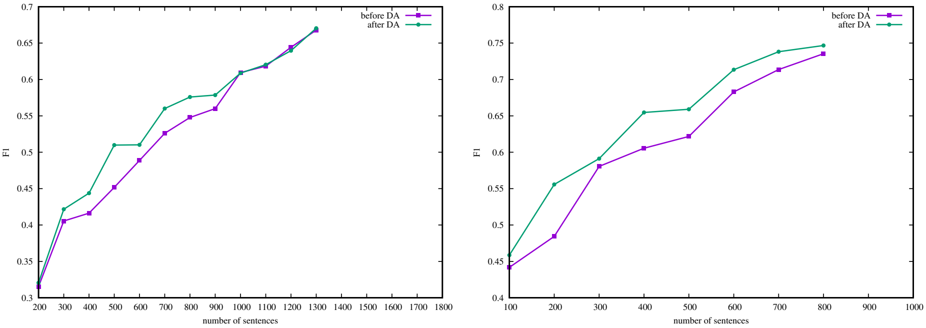
Table 1 The changing of number of sentence, character and entity after DA

Type		Number				
Sentence	before DA	0.1k	0.2k	0.3k	0.4k	0.5k
	after DA	1.4k	2.8k	4.3k	5.4k	7.2k
Character	before DA	4.3k	8.5k	13k	17k	21k
	after DA	79k	162k	253k	314k	422k
Entity	before DA	0.3k	0.6k	1k	1.2k	1.7k
	after DA	8.2k	17k	25k	30k	43k

Data augmentation We conduct the experiments on the MSRA NER data set to evaluate the effect of DA. We use Wikifier to link the entities to the Wikipage [22]. We select different numbers of sentences from the dataset in turn. We set $n_{max} = 10$ for DA experiments, and the change in size of training data is shown in Table 1. We can see that DA effectively augments the training set from the sentence level, character level and entity level.

As illustrated in Figure 10, we can see the results of improvements with different size of training data. The improvement increases gradually in the beginning and then progressively decreases, since the diversity of data changes from less to more and becomes stable when applying DA for the small data set. This means that DA has limited impact on the performance of model when the number of diverse sentences reaches a certain size. Also, we find that MSRA represents the similar results. Apart from that, the DA on Wikipedia data has a better improvement than on MSRA because there are errors in process of linking the mentions to entities on MSRA.

We also conduct experiments on using different values of max-replacing number n_{max}. As shown in Figure 11, the performance shows a declining trend in the beginning since the augmented sentences are inadequate and also brings in noises. After falling to the lowest point, as the number of sentences increases, DA begins to have a positive effect. However, if the n_{max} is set too large, the DA will have little improvement on the performance, even hurt it, because of excess similar augmented sentences along with some noises in the training data. Also, the model will take a longer training time on the immoderately augmented training data. As shown in Figure 9, when we combine the MCS with DA, it would achieve a better performance than other strategies in the former iterations.



(a) F1 measure on MSRA

(b) F1 measure on GEO

Figure 10 The improvement of performance of baseline model after DA

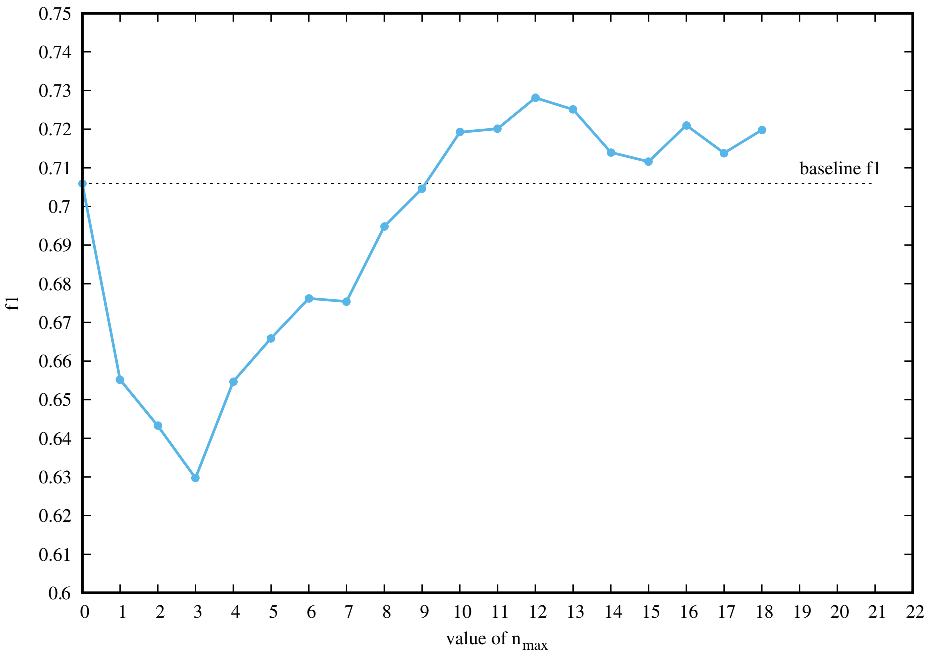


Figure 11 The performance of model with different n_{max}

7 Conclusions

In this paper, we propose a crowd-efficient learning approach for NER based on online encyclopedia, which could greatly reduce the amount of crowd annotation without hurting the precision of the NER model. Particularly, we develop a strategy to select some important samples from the unlabeled samples set for crowdsourcing instead of selecting samples at random, where we use three criteria to determine whether a sample is important. We take representativeness, informativeness and diversity into consideration when we select a batch of samples for crowdsourcing. After getting a batch of labeled samples from crowdsourcing, a data augmentation method is proposed, which could generate a lot more training data with the help of the structured knowledge of online encyclopedias to greatly augment the training effect. Finally, we train the model on the augmented training set. The training and selection procedure conduct iteratively until the improvement of the model are little or the performance of the model meets our requirement.

Acknowledgments This research is partially supported by Natural Science Foundation of Jiangsu Province (No. BK20191420), National Natural Science Foundation of China (Grant No. 61632016, 61572336, 61572335, 61772356), Natural Science Research Project of Jiangsu Higher Education Institution (No. 17KJA520003, 18KJA520010), and the Open Program of Neusoft Corporation (No. SKLSAOP1801).

References

1. Bi, W., Wang, L., Kwok, J.T., Tu, Z.: Learning to predict from crowdsourced data. In: UAI, pp. 82–91 (2014)

2. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine learning, pp. 160–167. ACM (2008)
3. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)
4. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
5. Dredze, M., Talukdar, P.P., Crammer, K.: Sequence learning from data with multiple labels. In: Workshop Co-Chairs, p. 39 (2009)
6. Dumitrache, A., Aroyo, L., Welty, C.: Crowdsourcing ground truth for medical relation extraction. ACM Trans. Interact. Intell. Syst. (TiiS) **8**(2), 12 (2018)
7. Felt, P., Black, K., Ringger, E., Seppi, K., Haertel, R.: Early gains matter: A case for preferring generative over discriminative crowdsourcing models. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 882–891 (2015)
8. Forney, D.G.: The viterbi algorithm. Proc. IEEE **61**(3), 268–278 (1973)
9. Grishman, R., Sundheim, B.: Message understanding conference-6: A brief history. In: COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics, vol. 1 (1996)
10. Habibi, M., Weber, L., Neves, M., Wiegandt, D.L., Leser, U.: Deep learning with word embeddings improves biomedical named entity recognition. Bioinformatics **33**(14), i37–i48 (2017)
11. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. arXiv:1508.01991 (2015)
12. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
13. Lafferty, J., McCallum, A., Pereira, F.CN.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
14. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv:1603.01360 (2016)
15. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. Handbook Brain Theory Neural Netw. **3361**(10), 1995 (1995)
16. Levow, G.-A.: The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, pp. 108–117 (2006)
17. Li, Y., Bontcheva, K., Cunningham, H.: Svm based learning system for information extraction. In: International Workshop on Deterministic and Statistical Methods in Machine Learning, pp. 319–339. Springer (2004)
18. Li, S., Zhao, Z., Hu, R., Li, W., Liu, T., Du, X.: Analogical reasoning on chinese morphological and semantic relations. arXiv:1805.06504 (2018)
19. Mou, L., Meng, Z., Yan, R., Li, G., Xu, Y., Zhang, L., Jin, Z.: How transferable are neural networks in nlp applications? arXiv:1603.06111 (2016)
20. Nguyen, A.T., Wallace, B.C., Li, J.J., Nenkova, A., Lease, M.: Aggregating and predicting sequence labels from crowd annotations. In: Proceedings of the conference. Association for Computational Linguistics. Meeting, vol. 2017, p. 299. NIH Public Access (2017)
21. Ni, J., Florian, R.: Improving multilingual named entity recognition with wikipedia entity type mapping. arXiv:1707.02459 (2017)
22. Noraset, T., Bhagavatula, C., Downey, D.: Websail wikifier at erd 2014. In: Proceedings of the First International Workshop on Entity Recognition & Disambiguation, pp. 119–124. ACM (2014)
23. Nothman, J., Ringland, N., Radford, W., Murphy, T., Curran, J.R.: Learning multilingual named entity recognition from wikipedia. Artif. Intell. **194**, 151–175 (2013)
24. Peters, M.E., Ammar, W., Bhagavatula, C., Power, R.: Semi-supervised sequence tagging with bidirectional language models. arXiv:1705.00108 (2017)
25. Richman, A.E., Schone, P.: Mining wiki resources for multilingual named entity recognition. In: Proceedings of ACL-08: HLT, pp. 1–9 (2008)
26. Rodrigues, F., Pereira, F., Ribeiro, B.: Sequence labeling with multiple annotators. Mach. Learn. **95**(2), 165–181 (2014)
27. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Techn. J. **27**(3), 379–423 (1948)
28. Shen, Y., Yun, H., Lipton, Z.C., Kronrod, Y., Anandkumar, A.: Deep active learning for named entity recognition. arXiv:1707.05928 (2017)

29. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 254–263. Association for Computational Linguistics (2008)
30. Sun, J.: 'jieba'chinese word segmentation tool (2012)
31. Tjong, E.F., Sang, K., De Meulder, F.: Introduction to the conll-2003 shared task: Language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, vol. 4, pp. 142–147. Association for Computational Linguistics (2003)
32. Van Dyk, D.A., Meng, X.-L.: The art of data augmentation. J. Comput. Graph. Stat. **10**(1), 1–50 (2001)
33. Wang, W.Y., Yang, D.: That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2557–2563 (2015)
34. Weischedel, R., Pradhan, S., Ramshaw, L., Palmer, M., Xue, N., Marcus, M., Taylor, A., Greenberg, C., Hovy, E., Belvin, R., et al: Ontonotes release 4.0. LDC2011T03. Linguistic Data Consortium, Philadelphia (2011)
35. Wong, S.C., Gatt, A., Stamatescu, V., McDonnell, M.D.: Understanding data augmentation for classification: when to warp? arXiv:1609.08764 (2016)
36. Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y., Jin, Z.: Improved relation classification by deep recurrent neural networks with data augmentation. arXiv:1601.03651 (2016)
37. Yadav, V., Bethard, S.: A survey on recent advances in named entity recognition from deep learning models. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 2145–2158 (2018)
38. Yang, Y., Zhang, M., Chen, W., Zhang, W., Wang, H., Zhang, M.: Adversarial learning for chinese ner from crowd annotations. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
39. Zhou, G.D., Su, J.: Named entity recognition using an hmm-based chunk tagger. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 473–480. Association for Computational Linguistics (2002)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Maolong Li[1] · Zhixu Li[1] · Qiang Yang[2] · Zhigang Chen[3,4] · Pengpeng Zhao[1] · Lei Zhao[1]

Maolong Li
mlli17@stu.suda.edu.cn

Qiang Yang
qiangyanghm@hotmail.com

Zhigang Chen
zgchen@iflytek.com

Pengpeng Zhao
ppzhao@suda.edu.cn

Lei Zhao
zhaol@suda.edu.cn

[1] Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, Suzhou, China

[2] King Abdullah University of Science and Technology, Jeddah, Saudi Arabia

[3] IFLYTEK Research, Suzhou, China

[4] State Key Laboratory of Cognitive Intelligence, IFLYTEK, Hefei, China