An Unsupervised Learning Approach for NER based on Online Encyclopedia

Maolong Li[1], Qiang Yang[4], Fuzhen He[1], Zhixu Li[1,2⋆],
Pengpeng Zhao[1], Lei Zhao[1], and Zhigang Chen[3]

[1]Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, China
[2]IFLYTEK Research, Suzhou, China
[3]State Key Laboratory of Cognitive Intelligence, iFLYTEK, P.R. China
[4]King Abdullah University of Science and Technology, Jeddah, SA
mlli17@stu.suda.edu.cn
{zhixuli, ppzhao, zhaol}@suda.edu.cn, qiangyanghm@hotmail.com,
zgchen@iflytek.com

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 novel way to make full use of the weakly-annotated texts in encyclopedia pages for exactly unsupervised NER learning, which is expected to provide an opportunity to train the NER model with no manually-labeled data at all. Briefly, we roughly divide the sentences of encyclopedia pages into two parts simply according to the density of inner url links contained in each sentence. While a relatively small number of sentences with dense links are used directly for training the NER model initially, the left sentences with sparse links are then smartly selected for gradually promoting the model in several self-training iterations. Given the limited number of sentences with dense links for training, a data augmentation method is proposed, which could generate a lot more training data with the help of the structured data of encyclopedia to greatly augment the training effect. Besides, in the iterative self-training step, we propose to utilize a graph model to help estimate the labeled quality of these sentences with sparse links, among which those with the highest labeled quality would be put into our training set for updating the model in the next iteration. Our empirical study shows that the NER model trained with our unsupervised learning approach could perform even better than several state-of-art models fully trained on newswires data.

Keywords: Named entity recognition · Data agumentation · Enhanced self-training

⋆ The corresponding author

1 Introduction

Named Entity Recognition (NER) aims at recognizing and classifying phrases referring to a set of named entity types [5], such as *PERSON*, *ORGANIZATION*, and *LOCATION* in text. A formal definition of NER is given in Def. 1 below.

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) [26], Support Vector Machines (SVM) [11], Conditional Random Fields (CRF) [8], with many handcrafted 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 [25]. 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 [10, 22], 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 use 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. Someone uses tags and alias in the Wikipedia to build a gazetteer type feature which they use in addition to standard features for CRF model [15], while the others generate NER training data by transforming the inner links to Named Entity (NE) annotations through mapping the Wikipedia pages to entity types [14, 16]. However, all the existing work only consider to use the weakly-annotated texts

with limited number of inner links of entities in online encyclopedias for training, but ignore other possible ways to fully use the data and knowledge contained in online encyclopedias.

In this paper, we propose a novel way to make full use of the weakly-annotated texts in online encyclopedia pages for unsupervised NER learning, which could greatly reduce the amount of manual annotation without hurting the precision of the NER model. The terms or names in encyclopedia pages are linked to the corresponding articles only when they are **first** mentioned in an encyclopedia article, which makes the first paragraph of the encyclopedia pages contain a lot more links than the other paragraphs. Here we roughly divide the sentences of encyclopedia pages into two parts simply according to the density of inner url links contained in each sentence. While a relatively small number of sentences with dense links could be used directly for training the NER model in the first step, the left sentences with sparse links are then smartly selected for gradually promoting the model in several self-training iterations. Given the limited number of sentences with dense links, 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. For those sentences with sparse links, we propose an iterative self-training method to make full use of them, where a graph model is utilized to help estimate the labeled quality of these sentences, among which those with the highest labeled quality would be put into our training set for updating the model in the next iteration.

The contributions of this paper are listed as follows:

1. We propose a novel unsupervised learning approach for NER based on online encyclopedias, which provides an opportunity to train the NER model without using any manually-labeled data.
2. We propose a so-called data augmentation method to greatly augment the training effect with the help of structured knowledge of online encyclopedias.
3. We also propose an iterative self-training method to make full use of these sparse links, where a graph model is utilized to help select sentences for updating the model.

Our empirical study shows that the NER model trained with our unsupervised learning approach could perform even better on Wikipedia texts than several state-of-art models fully trained on newswires data.

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 introduces our data augmentation method for NER training data and Section 5 explains how to enhanced the self-training by a graph-based model. After reporting our experical 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 ency-

clopedia 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).

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 [6,9]. 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 [16]. 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 [14].

Self-training Model. Self-training is a semi-supervised learning technique where there is only a small amount of labeled data but a large number of unlabeled data. It is a widely used approach for various NLP tasks, such as NER [7, 13], part-of-speech(POS) [20] and parsing [12]. The basic idea of self-training is to select a number of instances in the testing set according to some selection strategy, to augment the original training set. The most popular strategy is to select those instances whose predictions given by the baseline models own high confidence values. Kozareva et. al. use two classifiers for labeling the entity names and then one external classifier for voting the labeled results [7]. This method only selects the most confidentially predicted instances and the external classifier for voting may not be reliable. A recall-oriented perceptron is used in [13] for NER, along with self-training on Wikipedia, but they select Wikipedia articles at random and still need annotated sentences.

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 [19]. So far, data augmentation has been proved effective in most image related tasks by flipping, rotating, scaling or cropping the images [23]. When it comes to NLP tasks, data augmentation methods need to be designed specifically [21,24]. Xu et. al. propose a method by changing the direction of relations for relation classification task [24]. They split a rela-

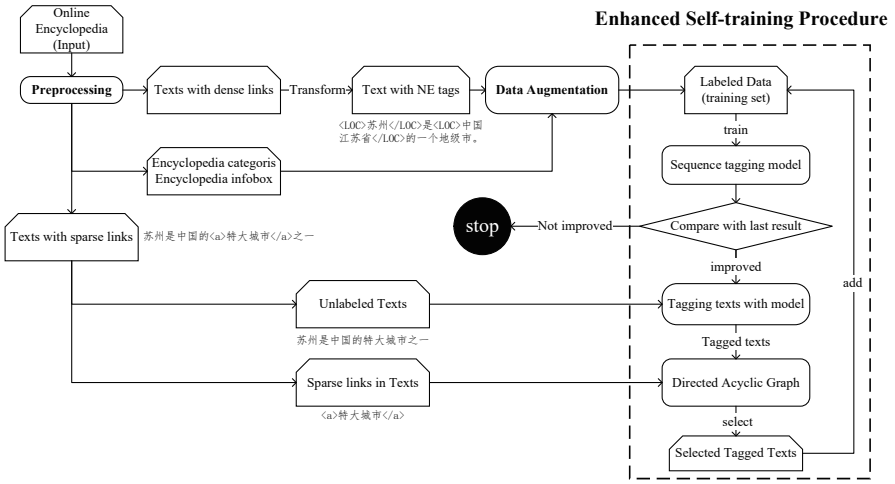


Fig. 1. The framework of the approach

tion 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 [21]. 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 an unsupervised learning approach for NER based on online encyclopedias. Fig. 1 depicts an overview of our unsupervised learning approach. Given a number of encyclopedia texts for a specific purpose, we roughly divide all the sentences into two parts: those with dense links, and those with sparse links. For sentences with dense links, we transform their links into NE tags, and then do data augmentation with the help of the structured knowledge contained in online encyclopedia. After that, we could have a set of labeled data for training a baseline DNN model (BI-LSTM-CRF). Next, we use this DNN model to perform NER tagging on a set of sentences with sparse links. We then utilize a graph model to help estimate the labeled quality of these sentences with sparse links, among which those with the highest labeled quality would be put into our training set for updating the NER model in the next iteration. We repeat this self-training step until the performance of the retrained model could not be improved on the test set. Some details about the framework are given below.

– **Online Encyclopedia Preprocessing.** We select the descriptions (paragraphs before *Contents*) of the encyclopedia articles in one domain as source

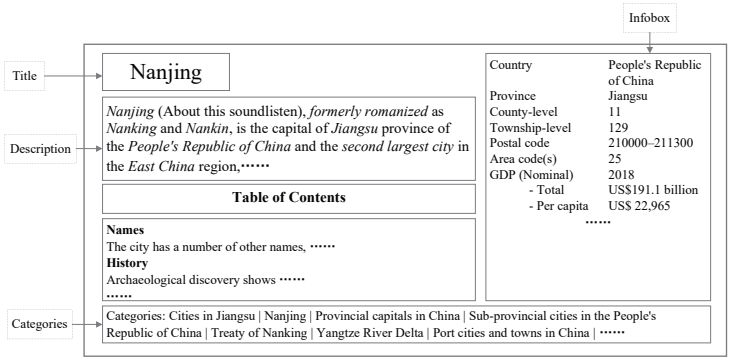


Fig. 2. An example Wiki page of Nanjing, where the italic parts have inner-links.

data (e.g. *Geography and places* in Wikipedia). We classify these encyclopedia texts into sentences with dense links and those with sparse links according to a score equation $S_q(n_l, n) = \frac{n_l}{n}$, where n_l and n are the number of linked words and all words (except stop words) in the sentence respectively. Here we take *topn* sentences ordered by their S_q scores as sentences with dense links, which are then taken as the initial training set D_i, while the left sentences with sparse links as the unlabeled data set D_u. Besides, we extract the **infobox** and **categories** of all the entity Wiki pages in the source data. An example of a Wiki page is shown in Fig 2.

- **Using Sentences with Dense Links.** In order to use sentences with dense links to generate training set, we transform links into NE tags by using the method presented in [16]. Here we call the entity name in texts as *mention* and the canonical name with description, infobox and categories are called *entity*. (e.g: For an entity Wiki page (entity) called *People's Republic of China*, its mention is *China*). We augment the training data by replacing the entity mentions in the texts, using categories and infobox derived from Wikipedia. More details about data augmentation will be given in Section 4. After data augmentation, we train a baseline DNN model (BI-LSTM-CRF) on the augmented training set. The model is based on character level instead of word level, where a character sequence is labeled with a tag sequence, because the word segmentation on web texts still has a few mistakes.
- **Enhanced Self-training Procedure.** The sentences with sparse links can not be added into training data set directly. Therefore, we develop an enhanced self-training to make full use of these links. The main idea of self-training is to augment the initial training set by labeling the unlabeled data with the baseline model, and then use some strategies to select part of these auto-labeled instances to add into the training set. The two steps are conducted for several iterations until the performance of the retrained model could not be further improved on the test set. The performance of the self-training algorithms strongly depends on how to select automatically labeled data in each iteration of the iterative procedure. Therefore, We explore extra metric as an auxiliary measurement to help select auto-labeled instances.

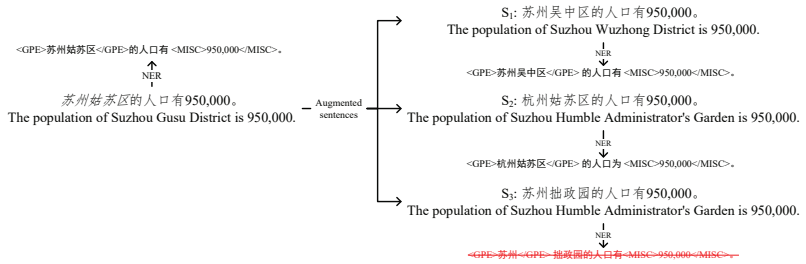


Fig. 3. The NER result of Stanford NER Tagger on augmented multi-entity sentences.

This method, called Enhanced Self-training, will be expressed thoroughly in Section 5.

4 Data Augmentation

The 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 [21,23,24]. The existing data augmentation method used in [21] 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.

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. 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 Fig 3. 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 example, 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 Fig 3.

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 Fig 3. To reduce this side effect, we conduct the data augmentation differently according to the number of the entities in text. For 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, because the new entities with same relation often have some common attributes with the replaced entities, which keeps the context of entities in augmented sentences correct. More details are shown as follows.

4.1 Augmenting with Single-Entity Sentences

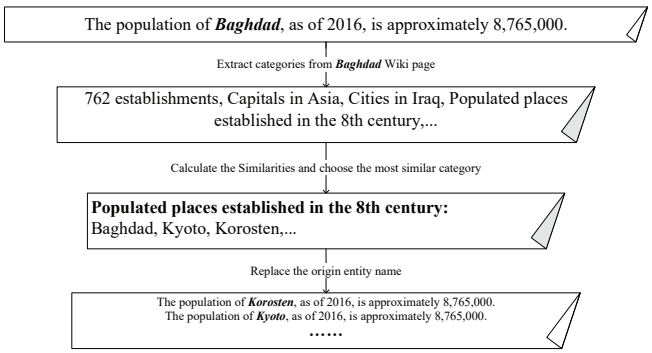


Fig. 4. The example of one **single-entity sentence**: The entity name in the example sentence is ***Baghdad*** and the categories in the corresponding Wiki page are in the second blocks. Then, we find the most similar category is ***Populated places established in the 8th century***. There are 35 entities in this category. Finally, we replace the ***Baghdad*** with the top-n most similar entity names in the category, like ***Kyoto***, ***Korosten***.

Given a single-entity sentence s which contains a mention m_o, we augment the orginal sentence by replacing the original mention m_o in the text with its related entities' mentions. First, we get the most similar category $c_{sim} = \text{argmax}_c CosSim(c_i, s)(c_i \in Cate(e_o))$, where 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 Equation 1 below:

$$E_{rep} = \begin{cases} E_r & \|E_r\| \leq n_{max} \\ TOP(n_{max}, E_r, e_o) & \|E_r\| > n_{max} \end{cases} \tag{1}$$

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 Fig 4.

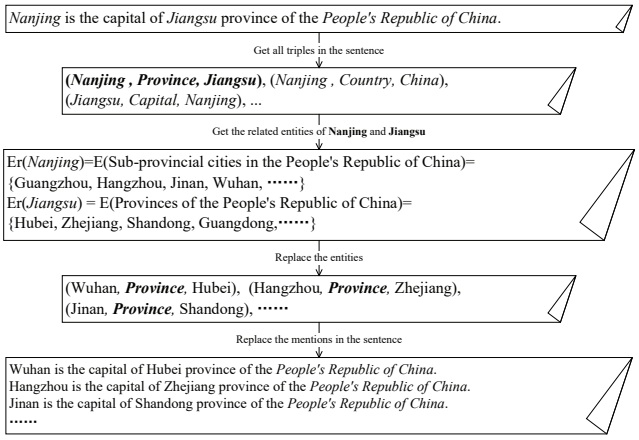


Fig. 5. The example of one **multi-entity sentence**: The triples in the sentence is in the second block. Then, we give the example of getting the related entities of the subject and object in the triple **(Nanjing, Province, Jiangsu)**. After that, we get the new relation triples illustrated in the forth block. Finally, we replace the original mentions in the sentence with the mentions of entities in new triples.

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 sentence S_3 shown in Fig 3. To reduce these mistakes, we should also take the attributes of the entity into consideration, which is in our future work plan.

4.2 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 Equation 2, 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 $T_{new} = T_1 \cup T_2$.

$$\begin{aligned} T_1 &= \{t | t = (s, r, o), r = r_i, (s \in E_{s_i}, o = o_i) \vee (s = s_i, o \in E_{o_i})\} \\ T_2 &= \{t | t = (s, r, o), r = r_i, s \in E_{s_i} \wedge o \in E_{o_i}\} \end{aligned} \tag{2}$$

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 Fig 5.

5 Enhanced Self-training Approach

In addition of the sentences with dense links, we also need to find a method to make full use of the sentences with sparse links. Self-training is a useful approach to semi-supervised learning when we only have a small amount of labeled data but a large number of unlabeled data.

5.1 Self-training

The basic self-training algorithm is described in Algorithm 1. The initial training data set D_{init} is generated from Wikipedia pages by transforming the dense links into NE tags, expressed in Section 3. Our baseline model (BI-LSTM-CRF) is trained on the D_{init}. Then, we use the baseline model to predict a batch size n_b of samples selected from unlabeled data set D_u, which consists of the sentences with sparse links. The F_b is the F-measure of baseline model on the test set.

Algorithm 1: An Self-training Algorithm

Input : D_{init}, D_u, M, T, where D_{init} denote the initial training set, D_u denote the unlabeled data set, M denote the baseline model, T denote the max iteration step.

Output: D_l, a set of labeled data.

$D_l \leftarrow D_i, f_{last} \leftarrow F_b, f_{new} \leftarrow -1$;
while $T > 0$ *and* $f_{last} > f_{new}$ **do**
  $D_l \leftarrow D_l \cup Select(M, D_u, n_b)$;
  $M \leftarrow Train(D_l)$;
  $f_{new} \leftarrow Test(M)$;
  $T \leftarrow T - 1$;
end
return D_l;

The *Select* function is the core of the algorithm, which uses some strategies to select high quality labeled data from D_u. However, traditional self-training solutions usually only depend on heuristics such as model confidence for instance selection. Such a selection strategy may not provide a reliable selection [1]. To handle this problem, we develop an enhanced self-training approach.

5.2 Enhanced Self-training

We propose a selection approach by combining a graph model with the confidence-based selection function to help evaluate the quality of the auto-labeled instances. When there are few or even no mentions linked in the sentences of unlabeled data, it is difficult to transform the inner links into NE annotations. For example, the sentence “Beijing is an important world capital and **global**

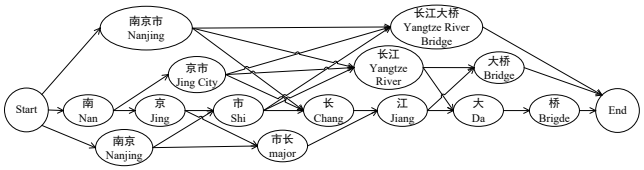


Fig. 6. The example of the DAG for sentence:"Yangtze River Bridge of Nanjing". After using dynamic programming on the DAG, we get a word segmentation: "Nanjing", "Yangtze River Bridge".

power city." has linked words: *global power city*, except *Beijing*. Nevertheless, we can still use the sparse links to help estimate the labeled quality of these sentences due to our model based on character level, where it only tags the characters referred to entity names. Specifically, we consider that the labeled sentences are in high quality when the most probable word segmentation contains the entity names. Therefore, we use a directed acyclic graph (DAG) to conduct the word segmentation, where we can ensure some certain words are segmented. We calculate a score $S_p(y, E, W) = Conf(y) + \frac{|E \cap W|}{|E|}$ of the labeled instances, where $Conf(y)$ denotes the confidence of the result y predicted by model, which are the output of the model, and E denotes entity mentions set in the sentence tagged by model, and W denotes the words of the sentences segmented by DAG. In each self-training iteration, we select labeled sentences with *topn* highest scores to add into the training set.

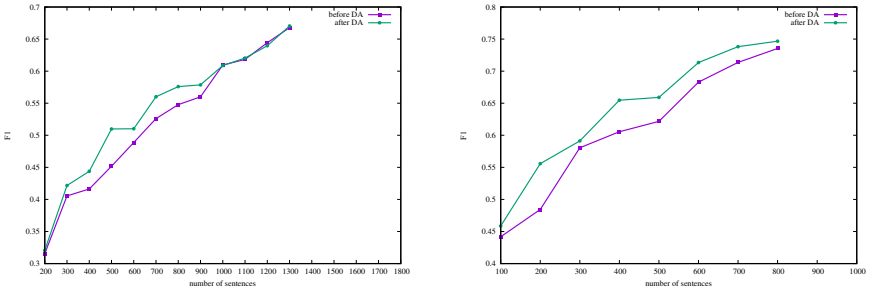
We build a DAG for all possible word segmentations (see an example in Fig. 6) where the node denotes one word and the directed path from *start* to *end* denotes a possible word segmentation of the sentence. In addition, linked words in the sentences with sparse links are definitely contained in the DAG. Each node has a probability calculated by the word frequency. Then, we use dynamic programming to find the most probable word segmentation by calculating the Π of all nodes in the path [17]. Besides, if the linked words in selected sentences are mentions and not tagged by the BI-LSTM-CRF model, we will transform the links into NE annotations and then add them into the training set.

6 Experiments

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

6.1 Datasets and Metrics

Datasets. Since our NER model is unsupervised, all the training data are derived from Wikipedia. We randomly select 8K sentences from the Wiki pages in Geography and extract 1K sentences as initial labeled data set named GEO, 7K sentences as unlabeled data set. We manually label 500 sentences in Wikipedia for testing the model. Besides, we evaluate the data augmentation method on



(a) F1 measure on MSRA (b) F1 measure on GEO

Fig. 7. The improvement of performance of baseline model after DA

the MSRA NER data set, which has 46.4K sentences for training, 4.4k sentences for test [10].

Metrics. Standard precision (P), recall (R) and F1-score (F1) are used as evaluation metrics. We use the conll eval script in CoNLL 2003 [18]. The hyper-parameters of our model are set as follows: $drop = 0.5$, $learningrate = 0.005$, $optimizer = adam$ and $batchsize = 32$.

6.2 Approaches for Comparison

- **BI-LSTM-CRF.** It is the baseline method with sequence tagging model (BI-LSTM-CRF) [6], which is trained on the initial training set transformed by sentences with dense links. We conduct the Chinese NER task on character level without word segmentation.
- **BI-LSTM-CRF+DA.** This method is similar to the first one but it uses the data augmentation (DA for short) on the initial training set and it is trained on the augmented data set.
- **BI-LSTM-CRF+EST.** In this method, we use the enhanced self-training (EST for short) approach on the basis of the baseline model to train the model but without data augmentation.
- **BI-LSTM-CRF+EST+DA.** On the basis of the baseline, we add enhanced self-training method and data augmentation to train our model.

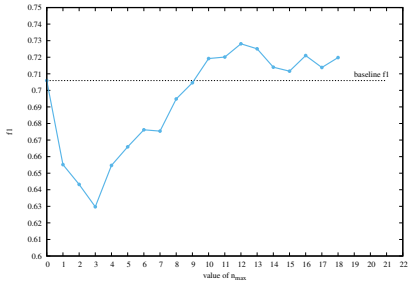
6.3 Experimental Results

Data augmentation. We conduct the experiments on the MSRA NER data set to evaluate the effect of DA. 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.

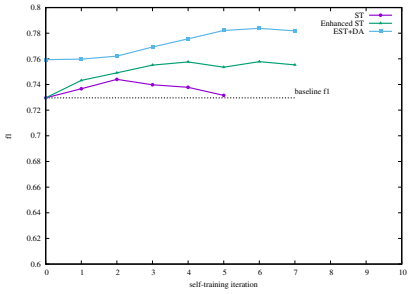
As illustrated in Figure 7, we can see the results of improvements with different size of training data. The improvement increases gradually in the beginning and then progressively decreases, since the diversity of data changes from less to more and becomes stable when applying DA for the small data set. This means

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



(a) The performance of model with different n_{max}



(b) The performance of enhanced self-training

Fig. 8.

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 form 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 Fig 8(a), 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.

Enhanced Self-training. We conduct two self-training approaches to train the baseline model without DA, shown in Fig 8(b). The two approaches have approximate performance at the beginning because these selected labeled sentences have high confidence, which means the model has already learned the features of them. After two iterations, the traditional ST model begins to select the left sentences with low confidence, leading to a bad performance. Different from the traditional ST model, our enhanced ST model takes the sparse links into con-

sideration to select and revise the left sentences. Therefore, it still has a positive influence in two more iterations, up to the best on the $4-th$ iteration.

Combination of EST and DA. Fig 8(b) shows that the performance of combining enhanced self-training and data augmentation, is better than the performance of using each of the two. The baseline model has a 3% F1 improvement after using DA and EST. Besides, unlike traditional self-training, the performance of our model will not decrease after several self-training iterations.

7 Conclusions

In this paper, we propose a method to make full use of the weakly-annotated texts in encyclopedia pages for exactly unsupervised NER learning. We propose a so-called data augmentation approach to greatly augment the training effect with the help of structure knowledge of online encyclopedias containing sentences with dense links. In addition, we put forward an iterative self-training method to make fully use of sentences with sparse links by combining the NER model where a graph model is utilized to help select sentences for updating the model. Our empirical study shows that the NER model trained with our unsupervised learning approach could perform even better than several state-of-art models.

In the future, we will improve the data augmentation method by considering the attributes of the entities. Besides, we will find an efficient crowdsourcing approach to reduce the annotation cost as much as possible.

Acknowledgments

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

References

1. Chen, M., Weinberger, K.Q., Blitzer, J.: Co-training for domain adaptation. In: Advances in neural information processing systems. pp. 2456–2464 (2011)
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. Journal of Machine Learning Research **12**(Aug), 2493–2537 (2011)
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. 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)

6. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
7. Kozareva, Z., Bonev, B., Montoyo, A.: Self-training and co-training applied to spanish named entity recognition. In: Mexican International Conference on Artificial Intelligence. pp. 770–779. Springer (2005)
8. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
9. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
10. 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)
11. 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)
12. McClosky, D., Charniak, E., Johnson, M.: Effective self-training for parsing. In: Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics. pp. 152–159. Association for Computational Linguistics (2006)
13. Mohit, B., Schneider, N., Bhowmick, R., Oflazer, K., Smith, N.A.: Recall-oriented learning for named entity recognition in wikipedia. CMU-LTI-11–012, Language Technologies Institute School of Computer Science Carnegie Mellon University 5000 Forbes Ave., Pittsburgh p. 15213 (2011)
14. Nothman, J., Ringland, N., Radford, W., Murphy, T., Curran, J.R.: Learning multilingual named entity recognition from wikipedia. Artificial Intelligence **194**, 151–175 (2013)
15. Radford, W., Carreras, X., Henderson, J.: Named entity recognition with document-specific kb tag gazetteers. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 512–517 (2015)
16. Richman, A.E., Schone, P.: Mining wiki resources for multilingual named entity recognition. Proceedings of ACL-08: HLT pp. 1–9 (2008)
17. Sun, J.: jiebachinese word segmentation tool (2012)
18. Tjong Kim Sang, E.F., 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-Volume 4. pp. 142–147. Association for Computational Linguistics (2003)
19. Van Dyk, D.A., Meng, X.L.: The art of data augmentation. Journal of Computational and Graphical Statistics **10**(1), 1–50 (2001)
20. Wang, W., Huang, Z., Harper, M.: Semi-supervised learning for part-of-speech tagging of mandarin transcribed speech. In: Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on. vol. 4, pp. IV–137. IEEE (2007)
21. 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)
22. 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, Philadelphia, Penn.: Linguistic Data Consortium (2011)
23. Wong, S.C., Gatt, A., Stamatescu, V., McDonnell, M.D.: Understanding data augmentation for classification: when to warp? arXiv preprint arXiv:1609.08764 (2016)

24. 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 preprint arXiv:1601.03651 (2016)
25. 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)
26. Zhou, G., 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)