Neural Processing Letters (2022) 54:2433–2454
https://doi.org/10.1007/s11063-021-10737-x

LTP: A New Active Learning Strategy for CRF-Based Named Entity Recognition

Mingyi Liu[1] · Zhiying Tu[1] · Tong Zhang[1] · Tonghua Su[1] · Xiaofei Xu[1] · Zhongjie Wang[1]

Accepted: 30 December 2021 / Published online: 12 January 2022

Abstract
In recent years, deep learning has achieved great success in many natural language processing tasks, including named entity recognition. The shortcoming is that a large quantity of manually annotated data is usually required. Previous studies have demonstrated that active learning can considerably reduce the cost of data annotation, but there is still plenty of room for improvement. In real applications, we found that existing uncertainty-based active learning strategies have two shortcomings. First, these strategies prefer to choose long sequences explicitly or implicitly, which increases the annotation burden of annotators. Second, some strategies need to revise and modify the model to generate additional information for sample selection, which increases the workload of the developer and increases the training/prediction time of the model. In this paper, we first examine traditional active learning strategies in specific cases of Word2Vec-BiLSTM-CRF and Bert-CRF that have been widely used in named entity recognition on several typical datasets. Then, we propose an uncertainty-based active learning strategy called the lowest token probability (LTP), which combines the input and output of conditional random field (CRF) to select informative instances. LTP is a simple and powerful strategy that does not favor long sequences and does not need to revise the model. We test LTP on multiple real-world datasets, the experiment results show that compared with existing state-of-the-art selection strategies, LTP can reduce about 20% annotation tokens while maintaining competitive performance on both sentence-level accuracy and entity-level F1-score. Additionally, LTP significantly outperformed all other strategies in selecting valid samples, which dramatically reduced the invalid annotation times of the labelers.

Keywords Active learning · Learning strategies · Named entity recognition · CRF

1 Introduction

Over the past few years, papers applying deep neural networks (DNNs) to the task of named entity recognition (NER) have achieved noteworthy success [4,14,18]. However, under typical

✉ Zhongjie Wang
rainy@hit.edu.cn

[1] Faculty of Computing, Harbin Institute of Technology, Harbin, China

training procedures, the advantages of deep learning mostly rely on a large quantity of labeled data. When applying these methods on domain-related tasks, their main problem lies in their need for a considerable human-annotated training corpus, which requires tedious and expensive work from domain experts. Thus, to make these methods more widely applicable and easier to adapt to various domains, the key is to reduce the number of manually annotated training samples.

Active learning was designed to reduce the amount of data annotation. Unlike the supervised learning setting, in which samples are selected and annotated at random, the process of active learning employs one or more human annotators by asking them to label new samples that are supposed to be the most informative in the creation of a new classifier. The greatest challenge in active learning is to determine which samples are more informative. The most common approach is uncertainty sampling, in which the model preferentially selects samples whose current prediction is least confident.

Many works have been performed to reduce the amount of data annotation for NER tasks through active learning. However, these state-of-the-art approaches mainly face two problems. One of the problems is that they tend to choose the long sequences explicitly or implicitly, which is an undesirable behavior when someone seeks to maximize performance for minimal cost annotation. Another problem is that they may need to revise and modify the original model, which increases the workload of the developer and the computing cost. In this work, we propose a simple but effective active learning strategy that does not prefer a long sequence and does not need to revise the original model.

When evaluating the effect of NER, most of the works only use the value of the entity-level F_1 score. However, in some cases, this can be misleading, especially for languages that do not have a natural separator, such as Chinese. The NER task is often used to support downstream tasks (such as QA and task-oriented dialog), which prefer that all entities in the sentence are correctly identified. Therefore, in this work, we evaluate not only the entity-level F_1 score but also the sentence-level accuracy.

We first experiment with the traditional uncertainty-based active learning algorithms, and then we proposed our own active learning strategy based on the lowest token probability with the best labeling sequence. Experiments show that compared with traditional selection strategies, our strategy **does not favor long samples and does not need to revise model** while maintaining **competitive** performance on both sentence-level accuracy and entity-level F1-score. Finally, we conduct an empirical analysis with different active selection strategies.

The main contribution of this paper is summarized as follows:

1. We proposeed a novel active learning strategy called LTP, which can handle both global and local information without revising NER model. And compared with existing state-of-the-art strategies, LTP can significantly reduce annotation cost.
2. We constructed a large number of experiments on different language datasets with different models. Especially on Chinese datasets, which fills the gap about active learning strategies comparison on Chinese datasets.
3. We discussed the impact factors in active learning and give suggestions on how to choose active learning strategies in practical applications.

The remainder of this paper is organized as follows. In Sect. 2, we summarize the related works in named entity recognition and active learning. In Sect. 3, we briefly introduce the data representation and CRF. Section 4 describes in detail the active learning strategies we propose. Section 5 describes the experimental setting, the datasets, and results. Section 6 discuss the different strategies and gives suggestions on how to choose an appropriate active learning strategy. The last section is the conclusion.

2 Related Work

In this section we summarized related work. One of the related work in this paper is named entity recognition, as this is the background and application scenario of this paper. Another related work is transfer learning and active learning, as both of them are able to reduce the amount of annotated data required in the named entity recognition with different principles. In this paper, LTP considers the use of active learning in transfer learning based models to further reduce the amount of data annotation required.

2.1 Deep Learning for Named Entity Recognition

The framework of NER using a deep neural network can be regarded as a composition of the encoder and decoder. For encoders, there are many options. Collobert et al. [6] first used a convolutional neural network (CNN) as the encoder. Traditional CNNs cannot solve the problem of long-distance dependency. To solve this problem, RNN [24], BiLSTM [12], dilated CNN [38] and bidirectional transformers [9] are proposed to replace CNN as an encoder. For decoders, some works used RNN for decoding tags [21,24]. However, most competitive approaches relied on CRF as a decoder [14,44]. Since the focus of this paper is not the NER model, we will not take too much effect to explain the details of the NER models. Readers who are interested in deep learning for named entity recognition can refer to the very recent survey by Li et al. [16].

Transfer learning is the migration of trained model parameters to new models to facilitate the new model training. We can share the learned model parameters into the new model in a certain way to accelerate and optimize the learning efficiency of the model, instead of learning from zero. So transfer learning could help to achieve better results on a small dataset. However, it should be noted that transfer learning works well only when the sample distributions of the source and target domain are similar. While significant distribution divergence might cause a negative transfer [32]. There are two methods to apply the pre-trained language model to downstream tasks. Feature-based approach (e.g. Word2Vec [22], ELMO [28]) use pretrained representations as input features for downstream task without modifying the original pretrained models, while fine-tuning approach (e.g. GPT [30], Bert [9]) that train the downstream task model by fine-tuning pretrained model parameters. Feature-based methods usually require fewer computational resources, but the performance will be lower, especially when the source and target domains differ significantly. Fine-tune based methods can achieve higher performance and are more tolerant to differences between source and target domains with the cost of requiring more computational resources. In this work, we experiment with both types of transfer learning to show that 1) active learning can complement transfer learning to further reduce the need for labeled data, and 2) our proposed active learning strategy can perform well under different resource conditions.

Active learning strategies have been well studied [1,8], [36]. These strategies can be grouped into the following categories: *query-by-committee (QBC)* [35,40], *information density* [41], *Fisher information* [34] and *uncertainty sample* [7,13,15,33]. *QBC* generates a committee of classifiers based on the current training set, and then evaluate the informative value of each sample and select a subset of most informative samples. *QBC* is usually not suitable for deep learning scenarios as generating a committee of deep learning based classifiers is extremely expensive. The main idea of *information density* is that informative samples should not only be those with high uncertainty, but also those representative of the input distribution. The main bottleneck of *information density* is that each sample has to be

Table 1 Example of data representation. [PAD] tags are not shown

Sentence		Trump	was	born	in	the	United	States	
Tag	[CLS]	B-PER	O	O	O	B-LOC	I-LOC	I-LOC	[SEP]

compared with all other samples in the unlabeled sample pool, which is impractical when the pool contains a large number of unlabeled samples. *Fisher information* tries to evaluate how uncertain a model is about a sample and which model parameter is most responsible for this uncertainty. The computational difficulties of Fisher information ratio and the difficulties in interpreting the parameters of the deep learning model hinder the application of *Fisher information* in deep learning based named entity recognition. *Uncertainty sample* queries the samples which are least certain how to label. This approach is often straightforward. Many works have compared the performance of different types of selection strategies in NER/sequence labeling tasks with the CRF model [3,5,20,34]. These results show that, in most cases, uncertainty-based methods perform better and require less time. For the existing uncertainty-based active learning strategies we will describe them in detail and compare them with LTP in Sect. 4.

We also found that existing studies are mainly based on English datasets and do not address Chinese datasets. So that, in this work, four Chinese datasets are selected for experiments to give a relatively comprehensive reference for active learning of Chinese NER. Additionally, traditional uncertain-based strategies always choose long sequences explicitly or implicitly, which significantly increases the burden on annotators. Some strategies [37] revise the model and let the model perform additional tasks for sample selection. Therefore, in this work, we propose a new active learning strategy that does not favor long sequences and does not need to revise the model.

3 CRF-Based NER Model

3.1 Data Representation

We represent each input sentence following the BERT format; each token in the sentence is marked with BIO scheme tags. Special $[CLS]$ and $[SEP]$ tokens are added at the beginning and the end of the tag sequence, respectively. $[PAD]$ tokens are added at the end of sequences to make their lengths uniform. The formatted sentence in length N is denoted as $\mathbf{x} =< x_1, x_2, \ldots, x_N >$, and the corresponding tag sequence is denoted as $\mathbf{y} =< y_1, y_2, \ldots, y_N >$. Table 1 gives an example of data representation.

3.2 CRF Layer

CRF are statistical graphical models that have demonstrated state-of-art accuracy on virtually all of the sequence labeling tasks, including the NER task. The main advantage of CRF is that it can recognize forms, even if they have not been seen in the training corpus. In particular, we use the linear-chain CRF, which is a popular choice for tag decoders and is adopted by most DNNs for NER.

A linear-chain CRF model defines the posterior probability of $\mathbf{y}$ given $\mathbf{x}$ to be:

$$P(\mathbf{y}|\mathbf{x};A) = \frac{\exp\left(P(y_1;\mathbf{x}_1) + \sum_{k=1}^{n-1} P(y_{k+1};\mathbf{x}_{k+1}) + A_{y_k,y_{k+1}}\right)}{Z(\mathbf{x})} \tag{1}$$

where $Z(\mathbf{x})$ is a normalization factor over all possible tags of $\mathbf{x}$, and $P(y_k;\mathbf{x}_k)$ indicates the probability of taking the y_k tag at position k, which is the output of the previous DNN layer, such as bilstm and softmax. A is a parameter called a transfer matrix, which can be set manually or by model learning. In our experiment, we let the model learn the parameter. $A_{y_k,y_{k+1}}$ denotes the probability of a transition from tag states y_k to y_{k+1}. We use $\mathbf{y}^*$ to represent the most likely tag sequence of $\mathbf{x}$:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \tag{2}$$

The parameters A are learned through the maximum log-likelihood estimation, that is, to maximize the log-likelihood function ℓ of training set sequences in the labeled dataset $\mathcal{L}$:

$$\ell(\mathcal{L};A) = \sum_{l=1}^{L} \log P(\mathbf{y}^{(l)}|\mathbf{x}^{(l)};A) \tag{3}$$

where L is the size of the tagged set $\mathcal{L}$.

4 Active Learning Strategies

Algorithm 1 Pool-based active learning framework

Require: Labeled dataset $\mathcal{L}$,
unlabeled data pool $\mathcal{U}$,
selection strategy $\phi(\cdot)$,
query batch size B
while not reach stop condition **do**
// Train the model using labeled set $\mathcal{L}$
$train(\mathcal{L})$;
for $b = 1$ to B **do**
//select the most informative instance
$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{U}} \phi(\mathbf{x})$
$\mathcal{L} = \mathcal{L} \cup < \mathbf{x}^*, label(\mathbf{x}^*) >$
$\mathcal{U} = \mathcal{U} - \mathbf{x}^*$
end for
end while

The greatest challenge in active learning is how to select instances that need to be manually annotated. A good selection strategy $\phi(\mathbf{x})$, which is a function used to evaluate each instance $\mathbf{x}$ in the unlabeled pool $\mathcal{U}$, will select the most informative instance $\mathbf{x}$.

Algorithm 1 illustrates the entire pool-based active learning process. In the remainder of this section, we describe various query strategy formulations of $\phi(\cdot)$ in detail.

4.1 Token-Based (Local) Strategies

The token-based strategy treats the labeling sequence as a set of isolated tokens and evaluates uncertainty by aggregating the information of these tokens.

Minimum token probability (MTP) selects the most informative tokens, regardless of the assignment performed by CRF. This whose highest probability among the labels is lowest:

$$\phi^{MTP}(\mathbf{x}) = 1 - \min_i \max_j P(y_i = j|\mathbf{x}_i; A) \tag{4}$$

where $P(y_i = j)$ is the probability that j is the label at position i in the sequence.

Entropy is a popular measure of informativeness. The entropy of a discrete random variable Y can be represented by $H(Y) = -\sum_i P(y_i) \log P(y_i)$, which means the information needed to "encode" the distribution of outcomes for Y. **Token entropy (TE)** is a method for using the entropy of the model's posteriors over its labeling:

$$\phi^{TE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} P(y_i = j|\mathbf{x}_i; A) \log P(y_i = j|\mathbf{x}_i; A) \tag{5}$$

where N is the length of $\mathbf{x}$ without [PAD] and j ranges over all possible token labels.

Settles [34] argued that querying long sequences should not be explicitly discouraged if they contain more information. They extended **TE** into **maximum token entropy (MTE)**:

$$\phi^{MTE}(\mathbf{x}) = N \times \phi^{TE}(\mathbf{x}) \tag{6}$$

4.2 Sentence-Based (Global) Strategies

Different from token-based strategies, sentence-based strategies treat labeling sequence $\mathbf{y}$ as a whole. Most of these strategies have high complexity or require intrusive models.

Culotta and McCallum [7] employed a simple uncertainty-based strategy for sequence models called least confidence (LC), which sort examples in ascending order according to the probability assigned by the model to the most likely sequence of tags:

$$\phi^{LC}(\mathbf{x}) = 1 - P(\mathbf{y}^*|\mathbf{x}; A) \tag{7}$$

This confidence can be calculated using the posterior probability given by Equation 1. Preliminary analysis revealed that the LC strategy prefers to select longer sentences:

$$P(\mathbf{y}^*|\mathbf{x}; A) \propto \exp\left(P(y_1^*; \mathbf{x}_1) + \sum_{k=1}^{n-1} P(y_{k+1}^*; \mathbf{x}_{k+1}) + A_{y_k^*, y_{k+1}^*}\right) \tag{8}$$

Since Eq. 8 contains summation over tokens, the LC method naturally favors longer sentences. Although the LC method is very simple and has some shortcomings, many works have proven the effectiveness of the method in sequence labeling tasks.

Scheffer et al. [33] proposed a method called **margin**, which queries samples with the smallest margin between the posteriors for its two most likely annotations:

$$\phi^{M}(\mathbf{x}) = -(P(\mathbf{y}_1^*)|\mathbf{x}; A) - P(\mathbf{y}_2^*|\mathbf{x}; A)) \tag{9}$$

where $\mathbf{y}_1^*$ and $\mathbf{y}_2^*$ are the first and second most likely tag sequences of $\mathbf{x}$. **Margin** requires the model to calculate the unnecessary second most likely tag sequence.

Table 2 Qualitative comparison of uncertainty-based active learning strategies

	MTP	LC	TE	TTE	LTP	Margin	SE	BALD
Local(Token) Information	√		√	√	√			
Global(Sentence) Information		√			√	√	√	√
Favor long sequence explicitly		√		√				
Revise model						√	√	
Additional compute						√	√	√

Different from **TE** and **TTE**, **sequence entropy (SE)** considers the entropy of the sequence instead of the entropy of the token:

$$\phi^{SE}(\mathbf{x}) = -\sum_{\hat{\mathbf{y}}} P(\hat{\mathbf{y}}|\mathbf{x};\, A) \log P(\hat{\mathbf{y}}|\mathbf{x};\, A) \tag{10}$$

where $\hat{\mathbf{y}}$ ranges over all possible tag sequences for $\mathbf{x}$. This calculation cost increases exponentially with the length of $\mathbf{x}$ and the number of tag categories.

The most recent uncertainty-based selection strategy is called **Bayesian active learning by disagreement (BALD)** [11,37]. BALD measures the uncertainty of the sample by observing the changes in the forward propagation result of the sample through multiple random dropouts [10]. Let $\tilde{\mathbf{y}}^1, \tilde{\mathbf{y}}^2, \ldots, \tilde{\mathbf{y}}^T$ represent the result from applying T independently sampled dropout masks:

$$\phi^{BALD}(\mathbf{x}) = 1 - \frac{\max_{\tilde{\mathbf{y}}} count(\tilde{\mathbf{y}})}{T} \tag{11}$$

where $count(\tilde{\mathbf{y}})$ means the number of occurrences of $\tilde{\mathbf{y}}$ in $\tilde{\mathbf{y}}^1, \tilde{\mathbf{y}}^2, \ldots, \tilde{\mathbf{y}}^T$. Normally, the value of T is 100. BALD will require considerable time to repeat forward propagation when the data pool is large.

4.3 Lowest Token Probability (LTP)

Unlike existing strategies, we believe that local information and global information have their advantages, and the two can complement each other. We look for the most likely sequence assignment (global) and hope that each token (local) in the sequence has a high probability. This goal makes Eq. 1 as large as possible and increases the margin between the best sequence and second best sequence.

$$\phi^{LTP}(\mathbf{x}) = 1 - \min_{y_i^* \in \mathbf{y}^*} P(y_i^*|\mathbf{x}_i;\, A) \tag{12}$$

We proposed our select strategy called **lowest token probability (LTP)**, which selects the tokens whose probability under the most likely tag sequence $\mathbf{y}^*$ is lowest. It is not difficult to infer from the formulation that **LTP** utilizes global and local information and implicitly implements **Margin** but does not require additional calculations.

Table 2 compares all the uncertainty-based active learning strategies mentioned in this section. Strategies that do not need to revise the model and do not require additional calculations are selected as the comparison method of our strategies.

5 Experiments

5.1 Datasets

We have experimented and evaluated the active learning strategies mentioned in Sect. 4 on four Chinese datasets and two English datasets:

- *People's Daily*. People's Daily is the most influential and authoritative newspaper in China. The dataset is a collection of newswire articles annotated with 3 balanced entity types, and is one of the common datasets for Chinese natural language processing related tasks.
- *Boson_NER* [23,29]. This dataset is an annotated dataset released by BosonNLP Lab specifically for Chinese named entity recognition. It consists of a set of online news, compared to *People's Daily*, which is more oriented to daily life. *Boson_NER* contains 6 balanced entity types.
- *Weibo_NER* [25,26]. The Weibo NER dataset is a Chinese Named Entity Recognition dataset drawn from the social media website Sina Weibo, which is Chinese Twitter. This dataset contains 8 extremely unbalanced entity types.
- *OntoNotes-5.0* [42]. OntoNotes 5.0 is a large corpus comprising various genres of text, including news, conversational telephone speech, weblogs, broadcast, etc. In this paper, a collection of broadcast and news articles in Chinese are used. This dataset contains 18 unbalanced entity types.
- *CONLL2003* [39]. CONLL2003 is a named entity recognition dataset released as a part of CONLL2003 shared task. The dataset is in English and was taken from Reuter Corpus, which consists of Reuter news between August 1996 and August 1997. This dataset contains 4 balanced entity types.
- *Ritter* [31]. Ritter is a collection of noisy, informal, but informative 140-character messages drawn from Twitter. This dataset contains 10 unbalanced entity types.

All datasets are formatted in the "BIO" sequence representation. To perform batch training, the length of all samples is limited to 64. Those samples that are originally longer than 64 are split according to commas or directly truncated to meet the length requirement. In terms of dataset partitioning, since all the datasets used in this paper are publicly available standard datasets, if the dataset has been partitioned into a training set and a test set, we directly use this partitioning result, and if it has not been partitioned, we partition the training set and test set according to the ratio of 8:2.

Table 3 shows some statistics of the datasets in terms of dimensions, number of entity types, and distribution of the labels. The statistical results show that the 6 selected datasets have distinctive features, covering a wide range of languages, domains, data magnitude and information richness, etc. For example, the gap between the number of sentences(#S) contained in the largest dataset(*People's Daily*) and the smallest dataset(*Ritter*) is nearly 26 times. The number of entity types(#E) in datasets range from 3 to 18. The distribution of entity types on different datasets is quite different, which is shown in Fig. 1. *Boson_NER*, *People's Daily* and *CONLL2003* can be regarded as a dataset with balanced distribution of entity types, while *Weibo_NER*, *OntoNotes-5.0* and *Ritter* can be regarded as unbalanced. We also find that the percentage of tokens with positive label(%PT) in the non-standard text datasets(*Weibo_NER* and *Ritter*) is significantly less than the other datasets. The average entity length(ASE) of English dataset is smaller than that of Chinese dataset. Additionally, we show the distribution of sample lengths on different datasets in Fig. 2. The sample lenght distribution varies widely, especially for *People's Daily*, *CONLL2003* and *Ritter*.

Table 3 Training(Test) data statistics

corpus	#S	#T	#E	ASL	ASE	AEL	%PT	%AC	%DAC
Boson_NER	27350 (6825)	409830 (99616)	6 (6)	14.98 (14.59)	0.67 (0.67)	3.93 (3.87)	17.7% (17.8%)	41.8% (41.8%)	14.7% (14.8%)
Weibo_NER	3664 (591)	85571 (13810)	8 (8)	23.35 (23.36)	0.62 (0.66)	2.60 (2.60)	6.9% (7.3%)	33.6% (36.3%)	14.8% (17.7%)
OntoNotes5.0 (bn-zh)	13798 (1710)	362508 (44790)	18 (18)	26.27 (26.19)	1.91 (1.99)	3.14 (3.07)	22.8% (23.4%)	72.5% (75.4%)	48.0% (51.5%)
People's Daily	50658 (4620)	2169879 (172590)	3 (3)	42.83 (37.35)	1.47 (1.33)	3.23 (3.25)	11.1% (11.6%)	58.3% (54.4%)	35.8% (29.1%)
CONLL2003	13862 (3235)	203442 (51347)	4 (4)	14.67 (15.87)	1.69 (1.83)	1.44 (1.44)	16.7% (16.7%)	79.9% (80.4%)	44.2% (48.8%)
Ritter	1955 (438)	37735 (8733)	10 (10)	19.30 (19.93)	0.62 (0.60)	1.65 (1.62)	5.3% (4.9%)	38.1% (39.2%)	15.3% (15.5%)

#S is the number of total sentences in the dataset, #T is the number of tokens in the dataset, #E is the number of entity types, ASL is the average length of a sentence, ASE is the average number of entities in a sentence, AEL is the average length of a entity, %PT is the percentage of tokens with positive label,%AC is the percentage of a sentences with more than one entity, %DAC is the percentage of sentences that have two or more entities. English datasets are marked in bold

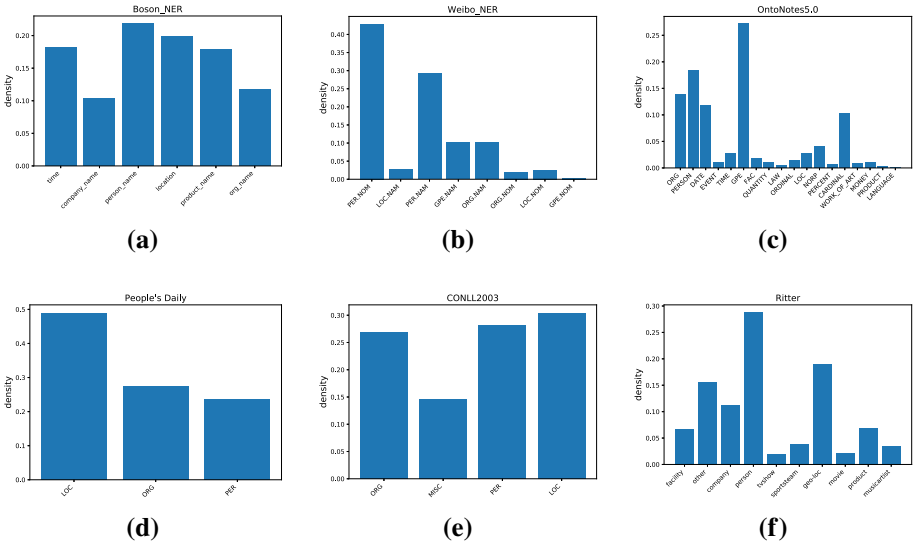


Fig. 1 Distribution of entity types on different datasets

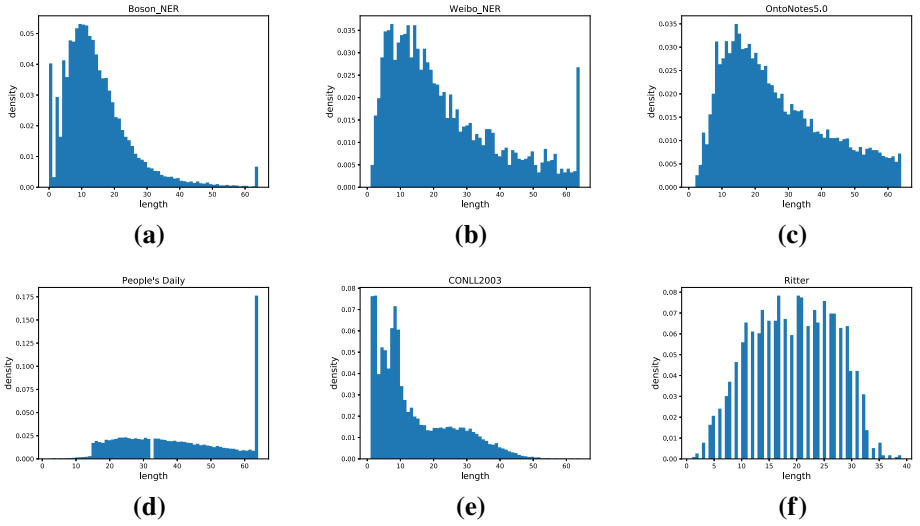


Fig. 2 Distribution of sample length on different datasets

5.2 Experimental Setting

For each dataset, we randomly choose 1% warm-start samples as the initial training set $\mathcal{L}_1$. We train the initial model on these data, then we apply an active learning strategy to choose an additional 2% samples based on the model's uncertainty estimates and train a new model based on these data. In each iteration, we train from scratch to avoid negative effects accumulated from previous training. We train each model to converge in each iteration. We fix the number of active learning iterations at 25 because each algorithm does not improve obviously after the 25 iteration.

We test two different CRF-based NER models on each dataset, namely Word2Vec-BiLSTM-CRF and Bert-CRF. These two models represent different commonly used architectures(BiLSTM encoder and transformer encoder) [16] and different resource restriction scenarios (restricted and sufficient). For Word2Vec-BiLSTM-CRF, we use a 300*d* Glove word embedding pretrained on the Chinese Wikiped corpus [17] for the Chinese datasets a 100*d* GloVe word embedding pretrained on the English Wikipedia corpus [27] for the English datasets. We uniformly set the global learning rate as 0.001 and the training batch size as 64. For Bert-CRF, we use the *bert-base-chinese* and *bert-base-uncased* provided by Transformers [43] as the pre-trained language models for the Chinese and English datasets, respectively. Limited by GPU memory, we set the training batch size as 16. To avoid insufficient learning of the CRF layer, we use different learning rate for different layers. The learning rate of the CRF layer is 0.001, while Bert uses a learning rate of 0.00001 for fine-tuning. Other parameters related to Bert are set to default. The transition matrix A in the CRF is left to let the model learn by itself. It must be noted that the goal of this article is not to obtain SOTA of NER, but to compare the performance of different active learning strategies under the same conditions. Therefore, the NER model and its parameters may not be the best but fair.

We empirically compare the selection strategy proposed in Sect. 4, as well as the uniformly random baseline (**RAND**) and long baseline (**LONG**). We evaluate each selection strategy by constructing learning curves that plot the overall F_1-score (for entities) and $accuracy$ (for sentences). To prevent the contingency of experiments, we performed 5 independent experiments for each selection strategy on each dataset using different random initial training sets $\mathcal{L}_1$. All results reported in this paper are averaged across these experiments.

All the experimental materials and datasets can be found on https://github.com/HIT-ICES/AL-NER.

5.3 Results

In this section, we compare the advantages and disadvantages of different active learning strategies on different NER models in terms of three metrics, **entity-level F_1-scores**, **sentence-level accuracy** and **annotation cost**. **Entity-level F_1-scores** and **sentence-level accuracy** are used to show that LTP can obtain competitive performance compared to existing state-of-the-art strategies. **Anotation cost** is used to show that LTP can significantly reduce the annotation cost.

A sentence is considered to be correctly predicted if all token in the sentence are correctly predicted. So that the sentence-level accuracy can be denoted as:

$$s_{acc} = \frac{\sum_{l=1}^{M} I(\tilde{\mathbf{y}}^{(l)}, \mathbf{y})}{M} \tag{13}$$

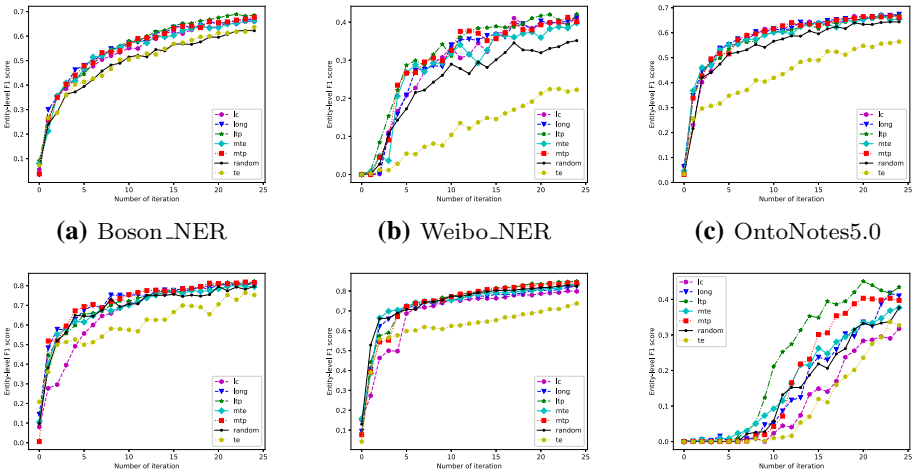
We first present the summarized experimental results in Table 4, where $entity\_F1$ and $sentence\_acc$ are the performance of the models trained by different strategies after 25 iterations of sample selection. We can find that the LTP strategy can usually obtain better performance than other strategies in the case of constrained resources (Word2vec-BiLSTM-CRF) and still achieves competitive performance when computational resources are sufficient (Bert-CRF). Convergency iterations denotes the number of iterations when the model performance is stabilized to 95%(97% and 99%) of the best performance under the given strategy. By comparison, we can find that Bert-CRF usually requires fewer iterations than Word2vec-BiLSTM-CRF, which indicates that transfer learning does reduce the data requirement for

Table 4 Comparison of LTP and other active learning strategies, where the top row in each cell are the results based on Word2vec-BiLSTM-CRF, and those in parentheses are results based on Bert-CRF

Dataset	Strategy	Entity_F1	Sentence_acc	Convergency Iterations			Annotation Cost (relative)		
				95%	97%	99%	95%	97%	99%
Boson_NER	Long	0.432 (0.793)	0.702 (0.868)	21.2 (12.0)	23.6 (22.0)	23.6 (25.0)	1.74 (1.30)	1.74 (1.86)	1.54 (1.50)
	Random	0.372 (0.777)	0.692 (0.8521)	20.6 (18.0)	21.8 (19.0)	24.6 (25.0)	1 (1)	1 (1)	1 (1)
	TE	0.256 (0.608)	0.664 (0.791)	23.6 (22.0)	24.6 (25.0)	24.6 (25.0)	0.49 (0.60)	0.49 (0.70)	0.43 (0.52)
	LC	0.424 (0.800)	0.697 (0.867)	19.2 (12.7)	19.6 (15.3)	24.6 (25.0)	1.63 (1.35)	1.56 (1.45)	1.58 (1.51)
	MTP	0.430 (0.809)	0.704 (0.874)	22.2 (15.0)	23.0 (19.3)	23.0 (20.7)	1.62 (1.34)	1.57 (1.51)	1.39 (1.18)
	MTE	0.407 (0.790)	0.692 (0.861)	20.0 (13.0)	23.0 (16.0)	24.4 (25.0)	1.65 (1.36)	1.70 (1.48)	1.56 (1.49)
	LTP	**0.449 (0.806)**	**0.704 (0.872)**	**22.7 (13.2)**	**22.7 (20.0)**	**22.7 (23.8)**	**1.39 (0.99)**	**1.31 (1.30)**	**1.15 (1.19)**
OntoNotes5.0	Long	0.679 (0.777)	0.523 (0.649)	19.0 (10.0)	23.5 (15.25)	24.3 (18.5)	1.57 (1.48)	1.59 (1.29)	1.49 (1.20)
	Random	0.655 (0.780)	0.504 (0.655)	20.0 (12.8)	22.8 (20.2)	24.8 (25.0)	1 (1)	1 (1)	1 (1)
	TE	0.569 (0.748)	0.441 (0.618)	22.0 (21.0)	24.3 (25.0)	25.0 (25.0)	0.55 (0.77)	0.56 (0.64)	0.54 (0.51)
	LC	0.675 (0.783)	0.522 (0.659)	19.8 (14.0)	20.0 (22.3)	24.4 (25.0)	1.62 (1.99)	1.42 (1.72)	1.48 (1.50)
	MTP	0.675 (0.789)	0.521 (0.664)	19.2 (11.3)	21.2 (16.2)	25.0 (24.4)	1.45 (1.40)	1.37 (1.21)	1.42 (1.36)
	MTE	0.673 (0.773)	0.514 (0.647)	18.3 (11.5)	22.7 (16.5)	24.0 (23.0)	1.47 (1.67)	1.49 (1.39)	1.42 (1.41)
	LTP	**0.675 (0.785)**	**0.517 (0.660)**	**20.0 (13.8)**	**23.4 (18.1)**	**25.0 (22.6)**	**1.26 (1.42)**	**1.28 (1.14)**	**1.25 (1.13)**
Weibo_NER	Long	0.432 (0.702)	0.702 (0.808)	21.2 (18.0)	23.6 (24.0)	23.6 (26.0)	1.74 (2.06)	1.75 (1.90)	1.54 (1.74)
	Random	0.372 (0.652)	0.692 (0.787)	20.6 (16.3)	21.8 (20.7)	24.6 (23.7)	1 (1)	1 (1)	1 (1)
	TE	0.256 (0.540)	0.664 (0.746)	23.6 (23.5)	24.6 (25.0)	24.6 (25.0)	0.49 (0.61)	0.49 (0.53)	0.43 (0.46)
	LC	0.424 (0.686)	0.697 (0.798)	19.2 (18.0)	19.6 (20.8)	24.6 (25.0)	1.63 (2.06)	1.56 (1.74)	1.58 (1.70)
	MTP	0.430 (0.697)	0.706 (0.803)	22.2 (10.0)	23.0 (19.0)	23.0 (23.0)	1.62 (1.06)	1.57 (1.38)	1.39 (1.40)
	MTE	0.407 (0.675)	0.692 (0.799)	20.0 (167)	23.0 (19.7)	24.4 (25.0)	1.65 (1.94)	1.70 (1.68)	1.56 (1.69)
	LTP	**0.449 (0.703)**	**0.707 (0.806)**	**21.2 (15.0)**	**21.2 (22.8)**	**23.2 (25.0)**	**1.31 (1.27)**	**1.23 (1.45)**	**1.18 (1.39)**

Table 4 continued

Dataset	Strategy	Entity_F1	Sentence_acc	Convergency Iterations			Annotation Cost (relative)		
				95%	97%	99%	95%	97%	99%
People's Daily	Long	0.818 (0.946)	0.798 (0.927)	16.0 (5.0)	19.5 (9.0)	24.0 (16.0)	1.13 (1)	1.34 (1.11)	1.32 (1.20)
	Random	0.797 (0.944)	0.784 (0.928)	21.0 (7.25)	21.0 (12.25)	25.0 (19.75)	1 (1)	1.0 (1.0)	1.0 (1.0)
	TE	0.764 (0.933)	0.762 (0.915)	22.0 (11.0)	24.0 (15.0)	25.0 (23.0)	0.65 (0.76)	0.74 (0.65)	0.65 (0.73)
	LC	0.816 (0.940)	0.802 (0.923)	17.33 (10.0)	20.33 (12.0)	25.0 (16.0)	1.21 (2.24)	1.38 (1.51)	1.36 (1.20)
	MTP	0.820 (0.950)	0.806 (0.936)	17.5 (4.0)	20.0 (7.5)	23.5 (12.0)	1.05 (0.66)	1.19 (0.76)	1.14 (0.73)
	MTE	0.806 (0.936)	0.789 (0.921)	17.0 (8.0)	21.67 (12.0)	23.0 (18.0)	1.14 (1.74)	1.41 (1.48)	1.23 (1.32)
	LTP	**0.825 (0.946)**	**0.808 (0.930)**	**19.67 (5.57)**	**21.33 (8.57)**	**25.0 (15.43)**	**1.05 (0.82)**	**1.14 (0.78)**	**1.11 (0.91)**
CONLL2003	Long	0.825 (0.858)	0.695 (0.732)	23.2 (16.0)	24.2 (23.0)	24.8 (24.0)	2.26 (2.41)	1.85 (2.08)	1.68 (1.67)
	Random	0.834 (0.874)	0.725 (0.779)	17.83 (13.75)	22.0 (19.25)	24.33 (24.25)	1 (1)	1 (1)	1 (1)
	TE	0.738 (0.779)	0.621 (0.671)	22.6 (22.0)	24.4 (23.0)	25.0 (24.0)	0.50 (0.57)	0.46 (0.44)	0.43 (0.37)
	LC	0.805 (0.850)	0.692 (0.724)	23.6 (22.0)	24.2 (24.0)	25.0 (25.0)	2.27 (2.91)	1.85 (2.13)	1.69 (1.70)
	MTP	0.843 (0.876)	0.732 (0.775)	20.0 (19.0)	21.4 (19.67)	24.8 (23.67)	1.89 (2.39)	1.59 (1.71)	1.56 (1.48)
	MTE	0.835 (0.857)	0.713 (0.743)	22.4 (23.0)	23.8 (24.0)	24.8 (25.0)	2.17 (2.97)	1.80 (2.13)	1.65 (1.70)
	LTP	**0.853 (0.874)**	**0.747 (0.771)**	**20.0 (14.4)**	**22.17 (18.4)**	**24.5 (23.4)**	**1.60 (1.39)**	**1.39 (1.22)**	**1.33 (1.20)**
Ritter	Long	0.424 (0.457)	0.660 (0.691)	24.0 (23.5)	24.8 (25.0)	25.0 (25.0)	1.31 (1.25)	1.33 (1.34)	1.34 (1.34)
	Random	0.381 (0.472)	0.661 (0.704)	24.8 (25.0)	25.0 (25.0)	25.0 (25.0)	1 (1)	1 (1)	1 (1)
	TE	0.342 (0.380)	0.645 (0.677)	23.8 (23.0)	24.0 (23.0)	24.4 (25.0)	0.63 (0.57)	0.63 (0.57)	0.65 (0.65)
	LC	0.338 (0.434)	0.647 (0.688)	23.0 (22.83)	25.0 (24.83)	25.0 (25.0)	1.25 (1.26)	1.33 (1.33)	1.33 (1.34)
	MTP	0.420 (0.521)	0.663 (0.710)	22.2 (25.0)	23.2 (25.0)	25.0 (25.0)	1.12 (1.17)	1.15 (1.17)	1.24 (1.17)
	MTE	0.388 (0.414)	0.650 (0.686)	23.0 (24.0)	25.0 (24.5)	25.0 (24.5)	1.20 (1.29)	1.29 (1.31)	1.29 (1.31)
	LTP	**0.465 (0.502)**	**0.676 (0.703)**	**22.4 (24.75)**	**23.2 (24.75)**	**25.0 (25.0)**	**0.92 (1.00)**	**0.94 (1.00)**	**1.01 (1.01)**



(a) Boson_NER **(b)** Weibo_NER **(c)** OntoNotes5.0

(d) People's Daily (e) CONLL2003 **(f)** Ritter

Fig. 3 Entity-level F_1-score results for Word2vec-BiLSTM-CRF on different datasets

training. And by comparing LTP and Random, we can find that the number of iterations required to converge to stable performance is reduced, which indicates that combining active learning and transfer learning can further reduce the cost of data labeling. The annotation cost indicates how many samples are labeled to achieve 95%(97% and 99%) of the best performance, and for clarity, we use the value relative to Random for this representation. It is clear that LTP has a significantly lower annotation cost compared to other active learning strategies that are competitive in terms of $entity\_F1$ and $sentence\_acc$.

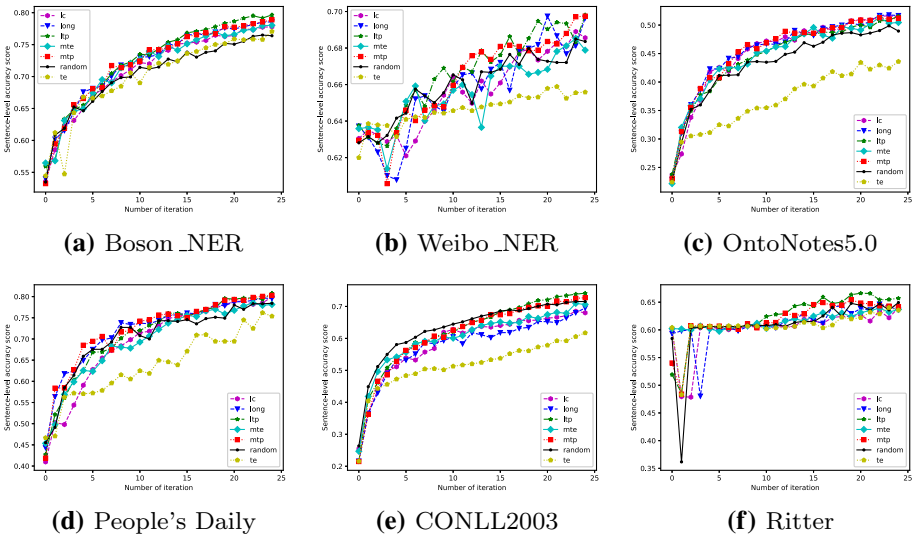
In the remainder of this section we compare the differences between the different active selection strategies at each iteration in more detail.

5.3.1 Results of Word2vec-BiLSTM-CRF

Entity-level F_1-scores of Word2vec-BiLSTM-CRF are shown in Fig. 3. It is clear that strategy TE performs the worst on all datasets. The results of strategy LONG illustrate that longer senteces do not necessarily contain richer information. The performance of selection strategies on the Chinese and English datasets differs significantly. First, all active learning strategies (except TE) overall outperform the benchmark strategy RAND on Chinese datasets, while RAND significantly outperforms LC and is comparable to MTE on English datasets. Second, MTE and LC perform similarly on Chinese datasets, but MTE outperforms LC on English datasets. Our strategy LTP performs competitively on all datasets. LTP slightly outperforms other active strategies on English formal text dataset *CONLL2003* and Chinese informal text datasets *Boson_NER* and *Weibo_NER*, and significantly outperforms other strategies on English informal text dataset *Ritter*.

where M is the number of samples in the test set. $\tilde{\mathbf{y}}$ is the model prediction result, and $I(\cdot)$ is an indicator.

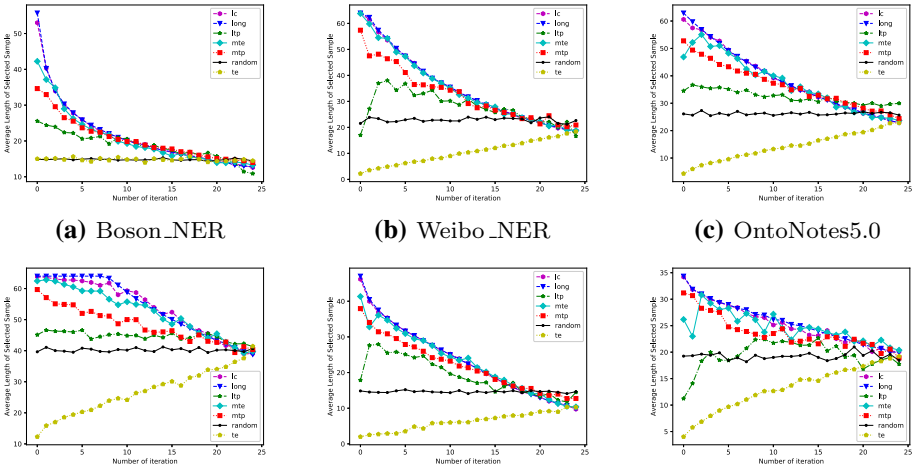
Figure 4 shows the results of **sentence-level accuracy** on six datasets. The results exceeded our expectations and are very interesting. First, the results confirm that the entity-level F_1-score is sometimes misleading as mentioned in Sect. 1. For example, the strategy LONG outperforms LC and MTE, and is competitive with MTP and LTP on *CONLL2003* in terms



(a) Boson _NER **(b)** Weibo _NER **(c)** OntoNotes5.0

(d) People's Daily **(e)** CONLL2003 **(f)** Ritter

Fig. 4 Sentence-level accuracy score results for Word2vec-BiLSTM-CRF on different datasets



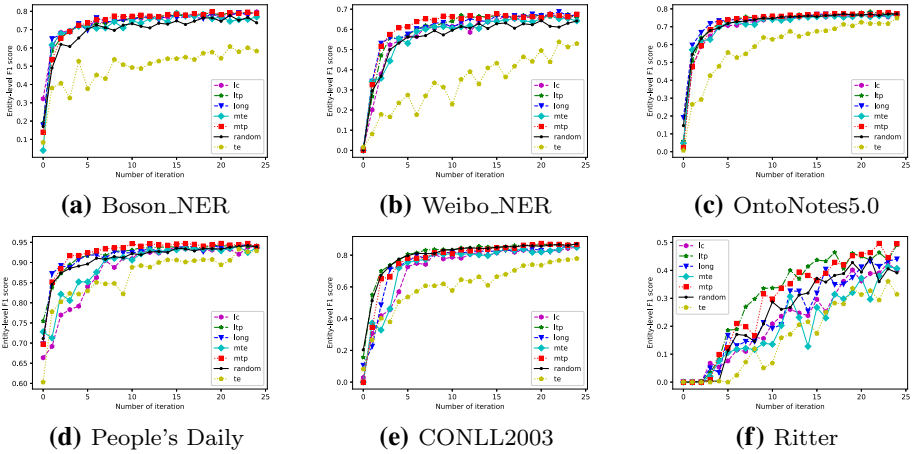
(a) Boson_NER **(b)** Weibo _NER **(c)** OntoNotes5.0

(d) People's Daily **(e)** CONLL2003 **(f)** Ritter

Fig. 5 Average length of the samples selected by active learning strategies. (Word2vec-BiLSTM-CRF)

of entity-level F_1-score, but only outperforms the strategy TE in terms of sentence-level accuracy. On the dataset *Ritter*, the gap between the TE and LONG/MTE is obvious on entity-level F_1-value, but this gap disappears on the sentence-level accuracy. Second, our strategy LTP is better than the rest of the strategies, while it is not obvious on the large datasets of formal text, which is similar to text for pre-trained word embedding.

Figure 5 shows **average length** of the samples selected by different active learning strategies. TE and RAND tend to choose shorter sentences, but their performance (both entity-level F_1-score and sentence-level accuracy) is poor, which can be seen in Figs. 3 and 4. The average length of samples selected by LC, MTP, LONG and MTE decreases gradually with the



(a) Boson_NER **(b)** Weibo_NER **(c)** OntoNotes5.0

(d) People's Daily **(e)** CONLL2003 **(f)** Ritter

Fig. 6 Entity-level F_1-score results for Bert-CRF on different datasets

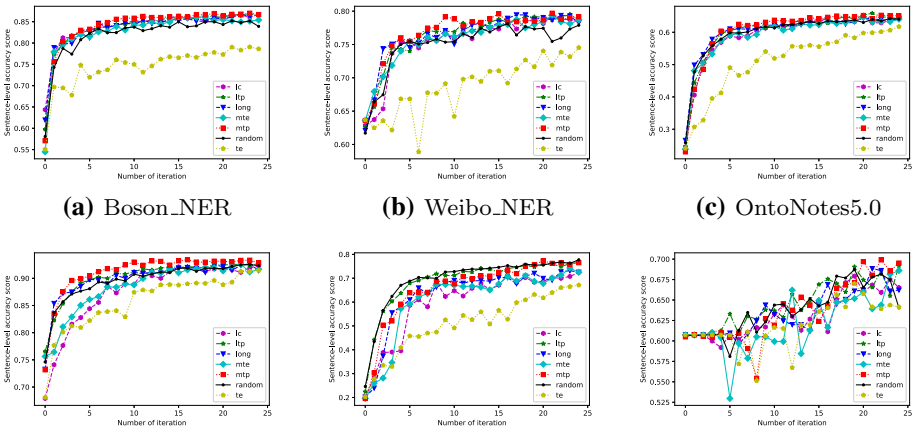
number of iterations. However, the average length of samples selected by LTP is relatively stable. Additionally, it can be seen from the Figs. 3, 4, 5 that the average length of samples selected by LTP before the model performance converges (first 20 iterations) is significantly less than other strategies (except TE and RAND). The preferences of different strategies for selecting the sample length will be discussed in detail in Sect. 6.

5.3.2 Results of Bert-CRF

Entity-level F_1-scores of Bert-CRF are shown in Fig. 6. TE is still the worst-performing strategy, but compared to the performance under Word2Vec-BiLSTM-CRF model, the gap between TE and other strategies has narrowed. All strategies have similar performance on formal text datasets (*OntoNotes5.0*,*People's Daily* and *CONLL2003*). Compared to other strategies, MTP and LTP perform better on the informal text datasets, MTP is slightly better (less than 0.5%) than LTP on *Boson_NER* and *Weibo_NER*, and LTP outperforms MTP on *Ritter*.

Sentence-level accuracy of Bert-CRF are shown in Fig. 7. The performance of strategies on Bert-CRF and Word2Vec-BiLSTM-CRF differs significantly. First, on all Chinese datasets, MTP outperforms the other strategies overall, but it is important to note that our strategy LTP is also competitive. LTP and RAND significantly outperform other strategies on English dataset *CONLL2003*, and all strategies performed similarly on *Ritter*. Second, on the dataset *Weibo_NER*, all strategies perform more consistently on the Bert-CRF model compared to the Word2Vec-BiLSTM-CRF model (Fig. 4).

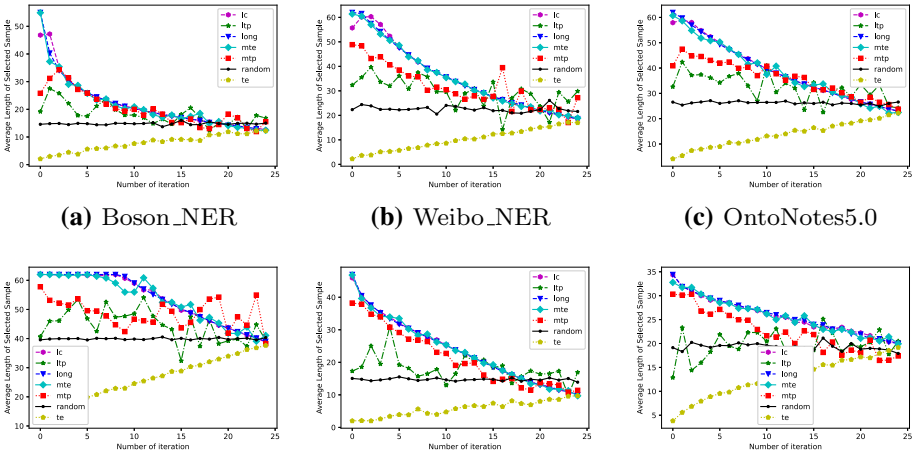
Figure 8 shows **average length** of the samples selected by different active learning strategies. Overall, all strategies perform similarly on Bert-CRF as they do on Word2Vec-BiLSTM-CRF (Fig. 5). The average length of the samples selected by LTP is significantly smaller than that of other strategies (except TE and RAND), and it should be noted that TE and RAND are significantly weaker than LTP in terms of performance (Figs. 6 and 7).



(a) Boson_NER **(b)** Weibo_NER **(c)** OntoNotes5.0

(d) People's Daily **(e)** CONLL2003 **(f)** Ritter

Fig. 7 Sentence-level accuracy score results for Bert-CRF on different datasets



(a) Boson_NER **(b)** Weibo_NER **(c)** OntoNotes5.0

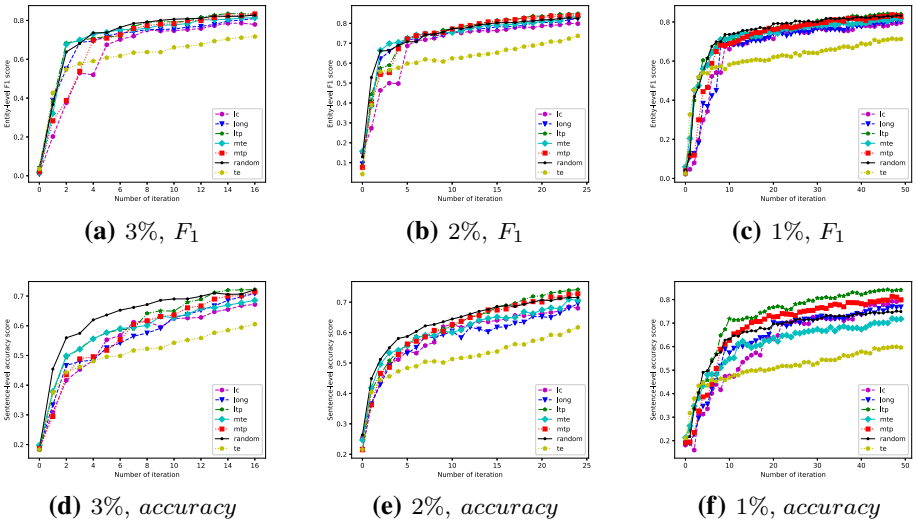
(d) People's Daily **(e)** CONLL2003 **(f)** Ritter

Fig. 8 Average length of the samples selected by active learning strategies. (Bert-CRF)

6 Discussion and Suggestion

6.1 Discussion About Query Batch Size *B*

We know that the most obvious effect of active learning is to select one sample at a time, although this is not realistic due to the cost of retraining. The more samples selected each time, the worse the active learning effect. Therefore, in the case of a large data pool, selecting 2% of the samples in each round cannot clearly reflect the differences between different strategies. To clearly reflect the differences between strategies and understand the effect of query batch size, we constructed an additional experiment on *CONLL2003* with 3% and 1% samples selected for each iteration. This experiment was also repeated 5 times independently and the average results were reported. The results are given in Fig. 9. It is clear that query batch



(a) 3%, F_1 **(b)** 2%, F_1 **(c)** 1%, F_1

(d) 3%, *accuracy* **(e)** 2%, *accuracy* **(f)** 1%, *accuracy*

Fig. 9 Performances of different strategies on CONLL2003 with different percentage samples selected. (Word2Vec-BiLSTM-CRF)

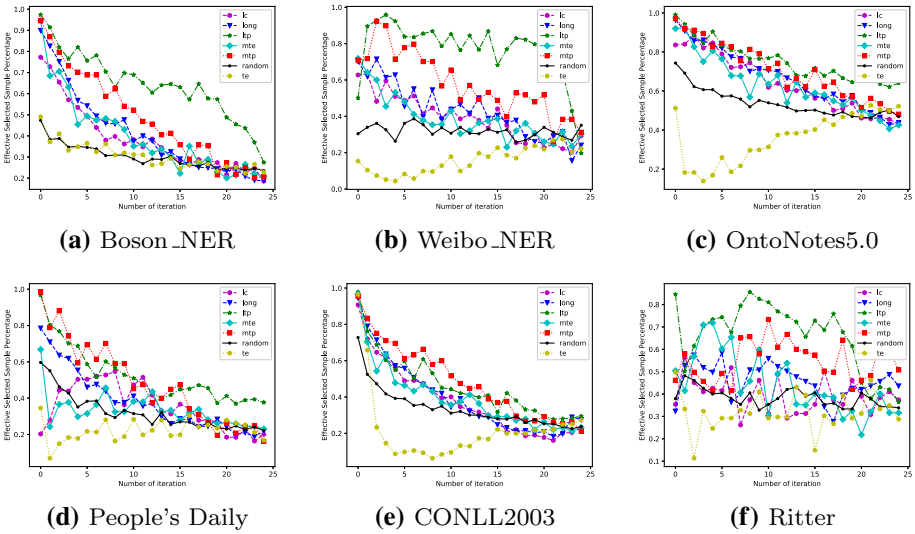
size B has limited effect on entity-level F_1, but a huge effect on sentence-level accuracy. When the value of query batch size is large, the RAND strategy outperforms the other strategies significantly on sentence-level accuracy. The performance gap between LTP and other strategies widens significantly when the value of query batch size B is reduced. This result demonstrates the efficiency of our proposed LTP, especially when the query size is small.

6.2 Discussion About Informative

The core of active learning is to select "informative" samples, but there is no unified standard to measure "informative". One thing is certain, the samples that are not correctly labeled by the model are informative samples for the model. So we need an indicator to evaluate the efficiency of strategies to select informative samples. We formally define effectiveness of each iteration of selection as:

$$effectiveness = 1 - \frac{\sum_{l=1}^{B} I(\tilde{\mathbf{y}}^{(l)}, \mathbf{y})}{B} \tag{14}$$

Figure 10 shows the results. We can observe that the curve of TE always significantly below the curve of other active learning strategies, which is similar to the performance comparison shown in Figs. 3, 4, 6 and 7. We also observe that the effectiveness of selecting samples for all active learning strategies decreases overall with increasing number of iteration rounds on all datasets (expect Ritter, as the NER models do not perform well.). This drop is expected because with the increase of training samples, the performance of the model improves and it becomes more difficult to select samples that do need to be labeled. We can see the LTP achieves better performance through a slower rate of effectiveness decline and the ability to maintain significantly higher values of effectiveness in later iterations than other active learning strategies.



(a) Boson_NER **(b)** Weibo_NER **(c)** OntoNotes5.0

(d) People's Daily **(e)** CONLL2003 **(f)** Ritter

Fig. 10 The results of the effectively selected sample percentage on different datasets (Word2Vec-BiLSTM-CRF)

6.3 Limitation of LTP

Although our proposed LTP enables non-invasive and efficient selection of informative samples without bias to long samples by using complementary local and global information, LTP has some limitations in theory and practice:

- The acquisition of local and global information in LTP relies on the CRF layer in the NER model. LTP cannot be directly applied in NER models that are not based on CRF or in other domain tasks. However, it should be noted that the core idea of LTP to evaluate sample uncertainty using both local and global information has been adapted to other tasks by other researchers, such as question answering [2,19] and role recognition [45].
- LTP is an uncertainty-based method that cannot guarantee the diversity of the selected samples as the information density-based methods. In practice, this results in the samples selected in each round often correcting errors in one aspect of the model. And when the query size B is large it lead to wasted annotation as one aspect of uncertainty the model may only need a number of samples smaller than B to correct. This phenomenon can be solved by selecting the appropriate query size B.

6.4 Suggesions

Combining the experimental results and discussions, we give some suggestions for when to choose LTP as active learning strategy.

- LTP would be a good choice when faced with constrained resources that prevent the use of large-scale pre-trained language models, such as Bert.
- When the number of annotators is limited and only a small amount of data can be annotated in each iteration, LTP can reduce the annotation cost more than other strategies.
- When the text of the dataset(target domain) and the pre-trained model (source domain) differ significantly, then LTP would be a good choice. However, when the target domain

is similar to source domain and the pre-trained model is large enough. Then there is no need to use active learning strategies, and it is sufficient to randomly select samples for annotation.

7 Conclusion

In this paper, we proposed a new active learning strategy for CRF-based named entity recognition called LTP. The qualitative comparison of existing SOTA uncertainty-based active learning strategies shows that our proposed LTP can leverage both local and global information to find informative samples and does not introduce additional computing. We have constructed a large number of experiments on different language datasets with different models. The experiment shows that compared with the traditional active selection strategies, our strategy does not favor long samples and does not need to revise model while maintaining competitive performance on both sentence-level accuracy and entity-level F1-score. Finally, we detailed discuss the performance of active learning strategies under different conditions and give some usage suggestions of LTP in practical applications.

Furthermore, we believe that the idea behind LTP, using both local and global information to select informative samples can be adapt into other domains. In our further work, we will try to adopt LTP to other domains, such as image classfication.

Acknowledgements Research in this paper is partially supported by the National Key Research and Development Program of China (No 2018YFB1402500), the National Science Foundation of China (61832004, 61772155, 61802089, 61832014).

References

1. Awasthi P, Balcan MF, Long PM (2014) The power of localization for efficiently learning linear separators with noise. In: Proceedings of the forty-sixth annual ACM symposium on Theory of computing, pp 449–458. ACM
2. Boreshban Y, Mirbostani SM, Ghassem-Sani G, Mirroshandel SA, Amiriparian S (2021) Improving question answering performance using knowledge distillation and active learning. arXiv preprint arXiv:2109.12662
3. Chen Y, Lasko TA, Mei Q, Denny JC, Xu H (2015) A study of active learning methods for named entity recognition in clinical text. J Biomed Inform 58:11–18
4. Chiu JP, Nichols E (2016) Named entity recognition with bidirectional lstm-cnns. Trans Assoc Comput Linguist 4:357–370
5. Claveau V, Kijak E (2018) Strategies to select examples for active learning with conditional random fields. In: Gelbukh A (ed) Computational linguistics and intelligent text processing. Springer International Publishing, Cham, pp 30–43
6. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12:2493–2537
7. Culotta A, McCallum A (2005) Reducing labeling effort for structured prediction tasks. In: AAAI, vol 5, pp 746–751
8. Dasgupta S, Kalai AT, Monteleoni C (2005) Analysis of perceptron-based active learning. In: International conference on computational learning theory, pp 249–263. Springer
9. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
10. Gal Y, Ghahramani Z (2016) A theoretically grounded application of dropout in recurrent neural networks. In: Advances in neural information processing systems, pp 1019–1027
11. Gal Y, Islam R, Ghahramani Z (2017) Deep bayesian active learning with image data. In: International conference on machine Learning, pp 1183–1192

12. Huang Z, Xu W, Yu K (2015) Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991
13. Kim S, Song Y, Kim K, Cha JW, Lee GG (2006) Mmr-based active machine learning for bio named entity recognition. In: Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers, pp 69–72. Association for Computational Linguistics
14. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. In: Proceedings of NAACL-HLT, pp 260–270
15. Lewis DD, Catlett J (1994) Heterogeneous uncertainty sampling for supervised learning. In: Machine learning proceedings 1994, pp 148–156. Elsevier
16. Li J, Sun A, Han J, Li C (2020) A survey on deep learning for named entity recognition. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2020.2981314
17. Li S, Zhao Z, Hu R, Li W, Liu T, Du X (2018) Analogical reasoning on chinese morphological and semantic relations. In: Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 2: Short Papers), pp 138–143
18. Limsopatham N, Collier NH (2016) Bidirectional lstm for named entity recognition in twitter messages
19. Lyu Z, Duolikun D, Dai B, Yao Y, Minervini P, Xiao TZ, Gal Y (2020) You need only uncertain answers: Data efficient multilingual question answering. In: TWorkshop on Uncertainty and Ro-Bustness in Deep Learning
20. Marcheggiani D, Artières T (2014) An experimental comparison of active learning strategies for partially labeled sequences. In: EMNLP
21. Mesnil G, He X, Deng L, Bengio Y (2013) Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. In: Interspeech, pp 3771–3775
22. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111–3119
23. Min K, Ma C, Zhao T, Li H (2015) Bosonnlp: An ensemble approach for word segmentation and pos tagging. In: Natural language processing and chinese computing, pp 520–526. Springer
24. Nguyen TH, Sil A, Dinu G, Florian R (2016) Toward mention detection robustness with recurrent neural networks. arXiv preprint arXiv:1602.07749
25. Peng N, Dredze M (2015) Named entity recognition for chinese social media with jointly trained embeddings. In: Processings of the conference on empirical methods in natural language processing (EMNLP), pp 548—554
26. Peng N, Dredze M (2016) Improving named entity recognition for chinese social media with word segmentation representation learning. In: Proceedings of the 54th annual meeting of the association for computational linguistics (ACL), vol 2, pp 149–155
27. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
28. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. arXiv preprint arXiv:1802.05365
29. Qiu X, Qian P, Yin L, Wu S, Huang X (2015) Overview of the nlpcc 2015 shared task: Chinese word segmentation and pos tagging for micro-blog texts. In: Natural language processing and chinese computing, pp 541–549. Springer
30. Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training. https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper.pdf
31. Ritter A, Clark S, Mausam Etzioni O (2011) Named entity recognition in tweets: An experimental study. In: EMNLP
32. Rosenstein MT, Marx Z, Kaelbling LP, Dietterich TG (2005) To transfer or not to transfer. In: NIPS 2005 workshop on transfer learning, vol 898, pp 1–4
33. Scheffer T, Decomain C, Wrobel S (2001) Active hidden markov models for information extraction. In: International symposium on intelligent data analysis, pp 309–318. Springer
34. Settles B, Craven M (2008) An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the conference on empirical methods in natural language processing, pp 1070–1079. Association for Computational Linguistics
35. Seung HS, Opper M, Sompolinsky H (1992) Query by committee. In: Proceedings of the fifth annual workshop on Computational learning theory, pp 287–294. ACM
36. Shen Y, Yun H, Lipton ZC, Kronrod Y, Anandkumar A (2017) Deep active learning for named entity recognition. arXiv preprint arXiv:1707.05928

37. Siddhant A, Lipton ZC (2018) Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 2904–2909
38. Strubell E, Verga P, Belanger D, McCallum A (2017) Fast and accurate entity recognition with iterated dilated convolutions. arXiv preprint arXiv:1702.02098
39. Tjong Kim Sang EF, De Meulder F (2003) 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, pp 142–147. https://aclanthology.org/W03-0419
40. Vandoni J, Aldea E, Le Hégarat-Mascle S (2019) Evidential query-by-committee active learning for pedestrian detection in high-density crowds. Int J Approx Reason 104:166–184
41. Wei K, Iyer R, Bilmes J (2015) Submodularity in data subset selection and active learning. In: International conference on machine learning, pp 1954–1963
42. Weischedel R, Pradhan S, Ramshaw L, Kaufman J, Franchini M, El-Bachouti M, Xue N, Palmer M, Hwang JD, Bonial C, et al (2012) Ontonotes release 5.0
43. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J, Shleifer S, von Platen P, Ma C, Jernite Y, Plu J, Xu C, Scao TL, Gugger S, Drame M, Lhoest Q, Rush AM (2020) Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, pp 38–45. Association for Computational Linguistics, Online. https://www.aclweb.org/anthology/2020.emnlp-demos.6
44. Yang Z, Salakhutdinov R, Cohen W (2016) Multi-task cross-lingual sequence tagging from scratch. arXiv preprint arXiv:1603.06270
45. Zhang Y, Lan M (2021) A unified information extraction system based on role recognition and combination. In: CCF international conference on natural language processing and chinese computing, pp 447–459. Springer