Looking Back on the Past: Active Learning with Historical Evaluation Results

Jing Yao, Zhicheng Dou, *Member, IEEE,* Jian-Yun Nie, *Member, IEEE,* Ji-Rong Wen, *Senior Member, IEEE*

Abstract—Active learning is an effective approach for tasks with limited labeled data. It samples a small set of data to annotate actively and is widely applied in various AI tasks. It uses an iterative process, during which we utilize the current trained model to evaluate all unlabeled samples and annotate the best samples based on a specific query strategy to update the underlying model iteratively. Most existing active learning approaches rely on only the evaluation results generated by the current model and ignore the results from previous iterations. In this paper, we propose using more historical evaluation results which can provide additional information to help better select samples. First, we apply two kinds of heuristic features of the historical evaluation results, the weighted sum of historical results and the fluctuation of the historical evaluation sequence, to improve the effectiveness of active learning sampling. Next, to further and more globally use the information contained in the historical results, we design a novel query strategy that learns how to select samples based on the historical sequences automatically. Our proposed idea is general and can be combined with both basic and state-of-the-art query strategies to achieve improvements. We test our approaches on two common NLP tasks including text classification and named entity recognition. Experimental results show that our methods significantly promote existing methods.

Index Terms—active learning, historical evaluation results, named entity recognition, text classification.

1 Introduction

Active learning is a sub-field of machine learning [1], which in the statistics literature is called optimal experimental design. It is distinctive for selecting a few training instances to annotate actively and strategically, instead of annotating all the unlabeled samples. Benefiting from the effectiveness of sampling, models can achieve high performance with fewer annotated training data. Active learning can play a crucial role when the labeled data are insufficient or the annotation cost is extremely high. It has been widely applied in tasks such as text classification [2], [3], sequence labeling and generation [4], [5], [6], image classification [7], [8], and document ranking in information retrieval [9], [10].

The key challenge of active learning is how to evaluate and select unlabeled samples to annotate for the subsequent training process. Many methods have been proposed for this problem [1], [11]. Basically, these methods can be categorized into three groups: **informative approaches**(such as uncertainty-based methods [12], query-by-committee [13], [14] and expected model change [15]), **representative approaches** [16] (such as density-based methods and cluster-based methods [10], [17]) and **diversity-based approaches** [18]. Active learning uses an iterative process. In general, an active learning approach iteratively selects one or a batch of unlabeled samples to be annotated based on a specific query strategy. It trains and updates the underlying model after the current batch of samples have been labeled, then selects the next batch of samples. In each iteration, the active learning algorithm must evaluate all unlabeled samples using the score function of a specific query strategy which calculates a score for each sample based on the underlying model trained in the previous iteration. Thus, a large set of historical evaluation scores is generated during the iterative process. It contains much information about the behaviors of each sample along with the model updating process, which is potentially useful for measuring the usefulness of the samples for training the model, especially for the informative strategies. Unfortunately, most existing active learning approaches use only the scores generated in the current iteration to select samples, ignoring the information contained in the previous evaluation results. A few studies have paid attention to the historical evaluation scores [19], [20]. However, they only regarded the scores generated in different iterations as discrete, but ignored that those evaluation results were actually a sequence with rich variation and information. Thus, none of these existing approaches took full advantage of the existing information.

In this paper, **we conduct detailed analysis of the historical evaluation results which have been generated in the past iterations, and propose several query strategies to make full use of the information contained in the result sequences to improve the effectiveness of active learning.** We mainly pay attention to the evaluation sequences of informative methods, because informative strategies calculate scores for unlabeled samples based on the underlying model so that the evaluation sequences contain the performance variation. But evaluation scores of the other two kinds of strategies are fixed which depend on the similarity between samples. All the evaluation results are collected along with the active learning process and can reflect the performances

- *J. Yao, Z. Dou, and J.-R Wen are with Gaoling School of Artificial Intelligence, Beijing Key Laboratory of Big Data Management and Analysis Methods, DEKE, Remmin University of China, Beijing 100872, P.R. China.*
  E-mail: jing_yao, dou@ruc.edu.cn, jirong.wen@gmail.com.
  (Corresponding author: Zhicheng Dou)
- *J.-Y Nie is with DIRO, University de Montreal, Quebec, C.P.6128, Succ Centre-Ville, Montreal, Quebec H3C 3J7, Canada.*
  E-mail: nie@iro.umontreal.ca.

Manuscript received April 19, 2005; revised August 26, 2015.

of samples in different stages of the underlying model. They should not be regarded as discrete sets of scores but sequences with a specific variation trend, which could be increasing, decreasing, relatively stable, or fluctuating. We believe the usefulness of a sample for training the model should rely on its performance in both the current and historical iterations. Therefore, **historical evaluation results and the information contained in the evaluation sequence can better measure a sample's value for model training**.

Let us use an example to illustrate this intuition: Consider an entropy query strategy and two samples x_m and x_n with historical evaluation sequences $[0.69, 0.68, 0.69, 0.68, 0.69]$ and $[0.33, 0.68, 0.58, 0.52, 0.69]$ respectively. We find that x_m and x_n have the same entropy 0.69 in the current iteration, but they performed far differently in past iterations during which x_m had higher entropy than x_n, In addition, x_m performed stably all along the process, whereas x_n was fluctuate with a totally different trend. Therefore, it is limiting to regard the two samples as having the same uncertainty based on only the last evaluation result. Taking historical evaluation sequences into account can provide more information to compare the two samples and make a better choice.

Based on the historical evaluation results, we propose several general active learning strategies. To begin with, we design two heuristic methods to apply historical evaluation results to help select samples. The first method calculates a weighted sum of the historical evaluation scores to measure the samples' informativeness and it gives more importance to the closer iterations. In the second method, we think that the fluctuation of the evaluation sequence reflects the sample's uncertainty to some extent; samples showing high fluctuation on the sequence are likely to be less certain. Therefore, we use the fluctuation to measure sample uncertainty more globally. These two heuristic approaches set some rules to use the historical evaluation results, and they both focus on one specific aspect or feature. To further explore more effective information contained in the historical evaluation results, we propose an active learning method that learns how to select samples according to the historical evaluation sequence automatically by a learning to rank algorithm. Through this method, fuller use of these historical evaluations can be made. The proposed strategies are general and we also combine them with some state-of-the-art methods to achieve improvements. Note that in existing active learning algorithms, historical evaluation scores have been calculated from the past iterations. We simply reuse these data, and hence there is no significant efficiency increase between our methods and existing methods.

The main contributions of this paper are as follows: (1) We focus on analyzing and utilizing the information contained in historical evaluation results to help better select samples in active learning. (2) We propose several strategies for incorporating historical evaluation information in active learning, especially a learning based approach, and verify the effectiveness. These methods are not task- or model-specific. They can be combined with existing active learning methods and applied to various tasks. (3) We apply historical evaluation results on several basic and state-of-the-art query strategies. Experimental results show that the effectiveness of these approaches is improved after historical evaluation results are considered.

TABLE 1: Notations used in the paper

Notation	Description
t	an iteration mark
U	unlabeled data set
L	labeled data set
x	a sample in U
x_i	the i-th sample in U
M	the model
$\phi^{\mathcal{S}}(\cdot)$	score function of strategy $\mathcal{S}$
$\phi_t^{\mathcal{S}}(x_i)$	score of x_i calculated by $\phi^{\mathcal{S}}(\cdot)$ in t-th iteration

The remaining of this paper is organized as follows. We first clarify our problem in Section 2, and introduce related works in Section 3. Then we describe our proposed methods in Section 4. In Section 5, we report the experimental settings and results. Finally, we conclude our paper in Section 6.

2 Problem Definition

In this section, we formulate the process of active learning and the problem in detail. The notations we use are listed in Table 1. We use the most widely applied pool-based active learning as the basic framework [1].

In pool-based active learning, there is a small set of labeled data L and a lot unlabeled data U applicable to train a model M . The original labeled data is used to train the model initially. Then the samples in U are selected and annotated iteratively for later model training. The selection process is based on the evaluation scores under a given strategy $\mathcal{S}$. Considering the t^{th} iteration, for a sample x from U, the learner calculates a score $\phi_t^{\mathcal{S}}(x)$ for it, based on a specific query strategy $\mathcal{S}$. The scores computed in every time step in the iterative process can be collected as a historical evaluation sequence $H_t^{\mathcal{S}}(x) = [\phi_1^{\mathcal{S}}(x), ..., \phi_j^{\mathcal{S}}(x), ..., \phi_t^{\mathcal{S}}(x)]$ in order. All samples are ranked according to the corresponding scores measured by the function of a specific query strategy and the most "well-behaved" ones (that is, those with the highest scores) are selected.

From the task formulation above, we can see that the query strategy is a crucial component of active learning. The query strategy measures how valuable a sample is for the later model training and determines which sample should be selected. Generally, these active learning query strategies have their own measurement function $\phi^{\mathcal{S}}(x)$ to evaluate each unlabeled sample x and select the best ones. We have:

$$\begin{aligned} x^* &= \arg\max{}_x \mathcal{F}\left(H_t^{\mathcal{S}}(x)\right) \\ &= \arg\max{}_x \mathcal{F}\left([\phi_1^{\mathcal{S}}(x), ..., \phi_j^{\mathcal{S}}(x), ..., \phi_t^{\mathcal{S}}(x)]\right). \end{aligned} \tag{1}$$

where $\mathcal{F}$ is our measurement function processing the historical evaluation sequence. $\mathcal{F}$ can be a simple linear function or some other complex machine learning algorithms.

Most existing active learning approaches rank samples only according to $\phi_t^{\mathcal{S}}(x)$ which is the evaluation result in the t^{th} iteration. All of them ignore the abundant information delivered by the former elements $\phi_1^{\mathcal{S}}(x)$ to $\phi_{t-1}^{\mathcal{S}}(x)$ that they calculated in the past iterations. So generally for existing methods, we have $\mathcal{F}\left(H_t^{\mathcal{S}}(x)\right) = \phi_t^{\mathcal{S}}(x)$, and Eq. (1) is simplified to:

$$x^* = \arg\max{}_x \phi_t^{\mathcal{S}}(x), \tag{2}$$

where $\phi_t^S(x)$ can be calculated in various ways, as introduced in Section 3. We propose to analyze and explore more information contained in the entire sequence of historical results, i.e., $H_t^S(x)$, to increase the effectiveness of sampling for active learning.

3 Related Works

In this section, we divide the related works into two main parts: (1) general query strategies and (2) task- or model-specific active learning approaches.

3.1 General Query Strategies

Common query strategies [1] can be categorized into three groups: informative models, representative models and diversity-based models. They select samples according to different criteria, and highlight different samples.

3.1.1 *Informative Models*

Informative models select samples that provide the most information for model training. Uncertainty, expected gradient length and disagreement of the model committee are three most popular metrics for the amount of information.

(1) Uncertainty-based methods [21] are the simplest and most commonly used approaches, especially for probabilistic models. They select samples with high uncertainty on label predictions and that are difficult for models to learn. Two typical uncertainty-based methods are *least confidence (LC)* and *entropy*. In the *LC* method, confidence of a sample refers to the model's prediction probability for its label, so the evaluation score is calculated by:

$$\phi_t^{\mathrm{LC}}(x) = 1 - P_\theta(y^*|x). \tag{3}$$

Here $y^* = \max_{y_i} P(y_i|x)$, and y_i ranges over all possible labels. θ are the model parameters. *Entropy* measures the uncertainty of a sample based on the entropy of its output probability distribution:

$$\phi_t^{\mathrm{Entropy}}(x) = -\sum_i P_\theta(y_i|x)\log P_\theta(y_i|x). \tag{4}$$

(2) Expected gradient length (EGL) [22] selects samples that would lead to the greatest changes in the underlying model, which can help the model converge to the best as quickly as possible. Because the true label is unknown, we replace the actual gradient with the expectation which is obtained by marginalizing over the gradients calculated on all possible label assignments:

$$\phi_t^{\mathrm{EGL}}(x) = \sum_i P_\theta(y_i|x)\| \bigtriangledown l_\theta(L \cup \langle x, y_i\rangle)\|. \tag{5}$$

Here, θ is parameters of the current model and $l_\theta(L \cup \langle x, y_i\rangle)$ refers to the loss value when the sample x labeled as y_i is added to the labeled set. Unfortunately, the computational cost of the gradient expectation is extremely high, which limits the application of EGL in practice.

(3) Query-by-Committee (QBC) [13], [14] is a framework aiming to minimize the version space of the underlying model. This approach usually maintains a committee of models all of which are trained on the current labeled set but get various parameters. Then, each committee votes on the label prediction of all unlabeled samples and the samples about which they have most disagreement are selected. A common disagreement measure is the average Kullback-Leibler (KL) divergence [23], calculated as:

$$\phi_t^{\mathrm{KL}}(x) = \frac{1}{C}\sum_{c=1}^{C} D(P_c||P_{avg}), \tag{6}$$

where $D(P_c||P_avg)$ is the KL divergence and $P_{avg}(y_i|x)$ is the average of probabilities.

3.1.2 *Representative Models*

The representative method aims to select a small set of samples that can represent the overall distribution and avoid selecting noisy data [10], [16]. It is often combined with uncertainty query strategies to boost each other. A typical method is a density-based model, which is defined as:

$$\phi_t^{\mathrm{DM}}(x) = \phi_t^S(x) \cdot \frac{1}{|U|}\sum_{x_i \in U} \mathrm{sim}\,(x, x_i)\,. \tag{7}$$

Here, $\mathcal{S}$ is an informative query strategy, $\phi_t^S(x)$ represents the informativeness score, and $\mathrm{sim}(x, x_i)$ is a function to measure the similarity between two samples.

3.1.3 *Diversity-based Sampling*

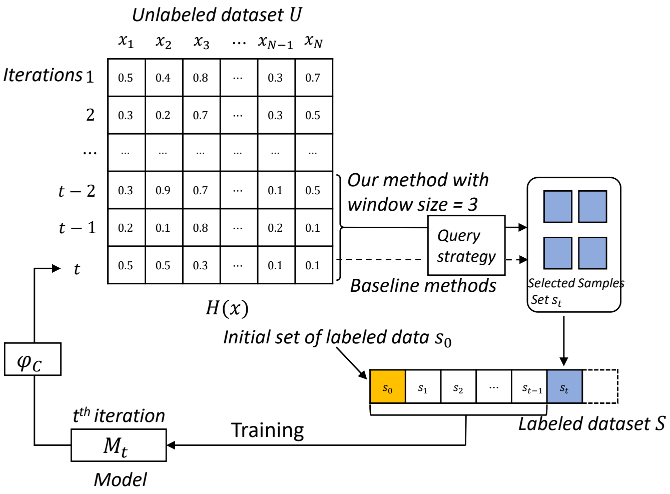
To improve efficiency, we selected a large batch of samples at each iteration sometimes. A diversity criterion is usually applied in such batch-mode active learning frameworks to select various samples. This avoids information redundancy and covers as much information as possible. Generally, we use the dissimilarity between samples to reflect the diversity and combine informativeness to score samples [18]. The most common method follows the maximal marginal relevance (MRR) formula [24]:

$$\phi_t^{\mathrm{MMR}}(x) = \lambda \times \phi_t^S(x) - (1-\lambda) * \max_{x_i \in L} \mathrm{sim}(x, x_i). \tag{8}$$

where λ is a hyper-parameter to balance the influence of the diversity and the informativeness $\phi_t^S(x)$.

3.2 Specific Active Learning Approaches

Basic query strategies introduced in Section 3.1 have been applied to specific AI tasks with some adjustments according to different tasks or models [2], [3], [5], [16], [25], [26]. Davy et al. [20] considered a little historical evaluation results, proposing HUS which directly utilizes the sum of the last several evaluation scores and HKLD using the historical evaluations by composing a committee with models trained in the past k iterations. Recently, many studies [4], [12], [27] combined active learning with neural networks for the task of sequence labeling and generation. Shen et al. [12] used the prediction probability normalized by the sequence length to eliminate the bias of selecting longer sentences, and Zheng et al. [27] considered the uncertainty as the changes between successive epochs. Deng et al. and Sinha et al. [4], [28] used GAN [29] to realize a representative strategy. Chen et al. [30], [31] applied active learning to a specific field, named entity recognition (NER) in clinical text. Besides, Fang et al. [6] proposed policy-based active learning (PAL) to learn a dynamic active learning strategy from data under the framework of reinforcement learning [32]. Similarly, Liu



Iterations	x_1	x_2	x_3	⋯	x_{N-1}	x_N
1	0.5	0.4	0.8	⋯	0.3	0.7
2	0.3	0.2	0.7	⋯	0.3	0.5
⋯	⋯	⋯	⋯	⋯	⋯	⋯
$t-2$	0.3	0.9	0.7	⋯	0.1	0.5
$t-1$	0.2	0.1	0.8	⋯	0.2	0.1
t	0.5	0.5	0.3	⋯	0.1	0.1

Unlabeled dataset U
$H(x)$
Our method with window size = 3
Query strategy
Baseline methods
Selected Samples Set s_t
Initial set of labeled data s_0
s_0 s_1 s_2 ⋯ s_{t-1} s_t
Labeled dataset S
φ_C
t^{th} iteration
M_t
Model
Training

Fig. 1: The general framework of the proposed active learning method. Unlike the existing query strategies, which use only evaluation results in the current iteration t, our strategy uses the historical evaluation results in a fixed-size window (the size is 3 in this example).

et al. [33] tried to learn query strategy with imitation learning [34]. For document ranking, Long et al. [35] adopted active learning to select samples minimizing the expected loss. In text classification, Zhang et al. [3] proposed an AL method (EGL of word embedding) for models with word embeddings by adjusting the standard EGL and achieved strong performance. They assumed word embeddings to be crucial for the text classification model to learn the feature representation of a sentence. Therefore, they highlighted samples having the largest gradient expectation on the word embeddings. Samples selected by this method help the model to learn a great word embedding. More generally, Gal et al. [7] presented an approach suitable for all Bayesian networks and achieved state-of-the-art on image classification, called Bayesian uncertainty (BALD). Bayesian uncertainty is a specific method designed for neural networks to measure a sample's uncertainty and has been proved to be correct and effective both in theory [36] and empirically [37].

All the task- or model- specific approaches described above have the same problem as the general frameworks: they use only the last evaluation score to select samples, except that HUS and HKLD simply apply the historical results. In this paper, we exploit the information of historical evaluation results to extend several state-of-the-art methods, achieving marked improvements.

4 Historical Sequence-based Sampling

4.1 Motivation

Recall that we use pool-based active learning as our basic framework, and the goal is to learn a high-performance model with the fewest human annotation resources. The general process has been described in Section 2. Figure 1 illustrates the general framework of our active learning method. We can see that after t iterations, the scores calculated in each time step construct a sequence $H_t^{\mathcal{S}}(x) = [\phi_1^{\mathcal{S}}(x), \cdots, \phi_j^{\mathcal{S}}(x), \cdots, \phi_t^{\mathcal{S}}(x)]$. We call it **historical sequence** or **historical evaluation results** in this paper. In general, several typical trends can be observed in the historical sequences (shown in Figure 2): (a) relatively stable, (b) increasing, (c) decreasing, and (d) fluctuating. The four trends describe the samples' different performances during the active learning process. Samples with trend (a) or (c) show relatively high evaluation scores for many iterations. These may be more valuable for model training than trend (b), which has a high value in only the last iteration. Besides, a fluctuating case (like trend (d)) shows more uncertainty than a stable case (like trend (a)), which should also be distinguished clearly. Therefore, looking back on past iterations can help us measure a sample more comprehensively.

Let us illustrate the usefulness of historical evaluation results by an example. Assuming that two samples x_m and x_n have the same evaluation result in the t^{th} iteration, in existing methods, they will be regarded equally. However, they have different historical evaluations, say for an entropy-based method with five historical entropy results: $H_5^{\text{Entropy}}(x_m) = [0.69, 0.68, 0.69, 0.68, 0.69]$ and $H_5^{\text{Entropy}}(x_n) = [0.33, 0.42, 0.58, 0.54, 0.69]$. x_m shows great uncertainty along with the model's updating process (like trend (a) in Figure 2), whereas x_n shows such high uncertainty in only the last iteration for one time (like trend (b) in Figure 2). We argue that it is unreasonable to ignore the previous results and consider the two samples equally merely according to their last evaluation results. In this paper, we propose to compare such two samples by their historical performance and make a more accurate choice. Stated in Section 1, we mainly focus on informative strategies. In fact, we have access to a large collection of historical sequences which contain information about the samples' performance variation along with the model's updating steps. We can make a detailed analysis of the historical sequences and use additional information contained in them to improve the performance of sample selecting.

In this paper, we first apply the historical evaluation results to the general query strategies introduced in Section 3.1. Here, we propose several new algorithms, including two heuristic approaches and a learning-based one. Then, we improve several specific state-of-the-art active learning approaches by introducing historical sequences.

4.2 Weighted Sum of Historical Sequence (WSHS)

With regard to the informative model, we calculate the historical evaluation sequences based on a specific informative query strategy $\mathcal{S}$, such as entropy, LC and EGL. We found some typical but different patterns after an analysis of general trends of the sequences, as shown in Figure 2. If only the evaluation result in the last iteration is considered, these kinds of samples would be thought to have almost the same amount of information. However, samples with high informativeness values for many times (like trends (a) and (c) in Figure 2) are substantially more important and valuable for model training than the ones with a high value in only the current iteration (like trend (b) in Figure 2). Consequently, we pay attention to measure the informativeness of a sample based on the evaluation results in both current and former iterations. We design a query strategy that uses

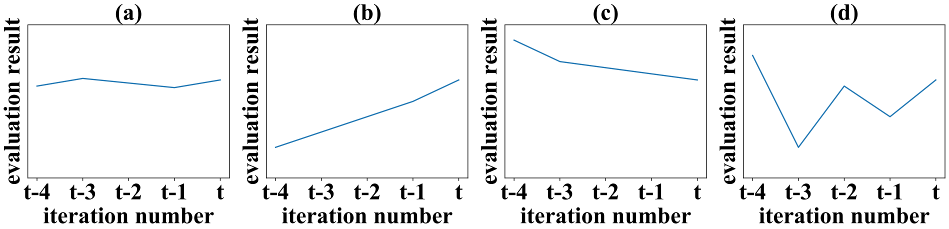


Fig. 2: Four different changing trends. (a) relatively stable, (b) increasing, (c) decreasing and (d) fluctuating.

the weighted sum of values in the historical sequence to select samples. This method highlights samples with high informativeness along the training iterations, formulated as:

$$\mathcal{F}^{\mathrm{WSHS}}\left(H_t^{\mathcal{S}}(x)\right)=\sum_{j=1}^{t} w_j\cdot\phi_j^{\mathcal{S}}(x), \tag{9}$$

where $\phi_j^{\mathcal{S}}(x)$ represents the evaluation result of sample x in the j^{th} iteration, and w_j is the corresponding weight.

We can set w_j in various ways. In this paper, we assume that the model trained in a latter iteration has more influence on the current model training, and earlier models that are far from the current model have less influence. We set the importance of previous results to decrease exponentially, different from directly adding up the historical evaluation results in HUS [20]. This leads to the weighting function:

$$w_j=\begin{cases}0 & 1\le j\le(t-l),\\ 2^{j-t} & (t-l+1)\le j\le t,\end{cases} \tag{10}$$

where l is used to control a window: only the scores generated in the last l iterations are considered. In practice, we can set l according to experimental experience.

Note that our method can be treated as an extension of an existing query strategy $\mathcal{S}$ with the score function $\phi^{\mathcal{S}}(x)$. If the parameter $l=1$, our method degrades to primitive $\phi^{\mathcal{S}}(x)$ only with the current evaluation results.

4.3 Fluctuation of Historical Sequence (FHS)

Uncertainty is one of the most widely explored criteria in active learning. Uncertainty-based query strategies tend to select samples with the least certainty about label prediction. In the t^{th} iteration, each sample has a corresponding historical evaluation sequence $H_t^{\mathcal{S}}(x)$. Two typical trends of sequences related to fluctuation are shown in Figure 2, in which (a) reflects a stable case, and (d) reflects a fluctuating case. In terms of uncertainty, a sample with stable performance and low uncertainty for the updating model tends to be certain. However, great fluctuation in the historical sequence indicates the uncertainty of the instance, which is more likely to be located at the model's decision boundary and to be beneficial for model training. Consequently, the fluctuation of a sample's performance along with the iterations is a crucial measurement for uncertainty. We propose combining the fluctuation of the historical sequence and the evaluation result in the current round as a new score function, highlighting samples that show great fluctuation in the historical sequence and has great informativeness in the current iteration. We have:

$$\mathcal{F}^{\mathrm{FHS}}\left(H_t^{\mathcal{S}}(x)\right)=w_s\cdot\phi_t^{\mathcal{S}}(x)+w_f\cdot V(H_t^{\mathcal{S}}(x)), \tag{11}$$

where $V(H_t^{\mathcal{S}}(x))$ represents the fluctuation of the evaluation sequence. Parameters w_s and w_f are used to balance the two components. The fluctuation of the sequence $H_t^{\mathcal{S}}(x)$ is determined by the variance:

$$V\left(H_t^{\mathcal{S}}(x)\right)=\frac{1}{l}\cdot\sum_{i=1}^{l}\left(\phi_{t-i+1}^{\mathcal{S}}(x)-\frac{1}{l}\cdot\sum_{j=1}^{l}\phi_{t-j+1}^{\mathcal{S}}(x)\right)^2.$$

4.4 Learn from Historical Sequences (LHS)

We use heuristic rules in the above two methods to utilize historical information to help select samples. The weighted sum of historical sequence (WSHS) takes multiple historical evaluation results into consideration. And the second method (FHS) pays attention to the connection between the fluctuation of historical sequences and uncertainty. However, both approaches focus on merely one specific feature and highlight one kind of sample when selecting, ignoring the information of other aspects. As we have found, there may be much effective information contained in the historical sequences and it is difficult to combine these features in a direct way. To take fuller advantage of the historical evaluation results, we further propose a learning based query strategy intended to learn how to select samples with the information contained in the historical sequences automatically. Fang et al. [6] adopted reinforcement learning to train a learning based active learning algorithm and Liu et al. [33] applied imitation learning. In this paper, we select a simple algorithm, the learning to rank (LTR) model, to learn the query strategy effectively. We regard the sampling process of active learning as a ranking problem, where we use a ranker to sort unlabeled samples and select the best ones iteratively. Another advantage of using the LTR model is that it is not necessary to compute the exact value for each unlabeled sample; they can just be compared to get relative relations. Three main components are required to train a ranker: samples to be sorted, samples' features for ranking and their corresponding labels. We will describe these components in detail as follows.

4.4.1 Samples to Be Sorted

Generally, we can directly sort all the samples in the unlabeled set and choose the best ones to annotate. However, usually there are a large amount of unlabeled samples, which causes the ranking space to be too large and will increase the training difficulty and training errors for the ranker. To reduce the sample space without affecting the results, we first select a set of well-performed samples to form a relatively small candidate set, based on the evaluation scores of several traditional query strategies such as the entropy and LC. Then, the ranker is applied to sort the candidates and select the best samples.

4.4.2 Features for Ranking

To make full use of the information contained in the historical evaluation sequences, we extract several features for ranking, elaborated as follows.

• **historical evaluation results**: Referring to our first method WSHS in Section 4.2, we know it is important to consider evaluation results in both the current and past iterations. Therefore, we take the historical evaluation results based on a specific method $\mathcal{S}$ in the last l iterations $[\phi^{\mathcal{S}}_{t-l+1}(x), ...\phi^{\mathcal{S}}_{t}(x)]$ as a main component of the ranking features, where l is a hyper-parameter to control the history window. Compared with the fixed weights in the WSHS, the importance weights of each historical evaluation results can be learned automatically in this method.

• **fluctuation of historical sequence**: Following our FSH method in Section 4.3, we also use the fluctuation of the historical sequences to measure the uncertainty of samples.

• **trend of historical sequence**: The trend of the historical sequence reflects a sample's performance variation on the updating model along the active learning process. For example, considering a historical sequence calculated by an entropy-based strategy, an increasing trend means that the model becomes more and more uncertain about the prediction of this sample. We use MK (Mann-Kendall) Trend [38] to characterize the trend of historical sequences.

• **the predicted next result**: Evaluation results in the historical sequence are collected in a step order so that we can regard the sequence as a time series to some extent. On the basis of the characteristics of time series, we are able to predict the sample's evaluation result in the next iteration based on the historical data, which is of great significance to measure the sample and direct the selection. Various methods have been devised for time series prediction such as the auto-regressive integrated moving average (ARIMA) model [39] and the LSTM neural network [40]. We use the LSTM model to predict the next evaluation score in our study and set the predicted score as a ranking feature. Here, we train the predictor LSTM with the historical evaluation sequences generated on a labeled dataset by a specific query strategy $\mathcal{S}$; the evaluation results in the past k iterations are inputs to predict the current result.

• **output probability of the model**: Many traditional query strategies select samples according to scores calculated on the output probability of the model $P_\theta(y_i|x)$, such as the entropy with Eq. (4) and the LC based on Eq. (3). To generalize existing methods, we directly use the predictive probability distribution as a part of our ranking features.

4.4.3 Labels

The labels used in our LTR framework measures the usefulness of samples for the model training in later iterations. We test the current model M on the testing set and represent its performance as $Eval(M)$. Then, the unlabeled sample x is annotated and added to the labeled set, and we update the model to M', whose performance on the testing set is represented as $Eval(M')$. Consequently, we can use $score(x) = Eval(M') - Eval(M)$ to measure the usefulness of the sample x. The higher the score, the better the sample. However, considering that we add only one labeled sample to update the model and there would be some errors with the score, we convert the absolute score into a level and use the level as the rank label instead of directly using the score. This process also reduces the difficulty of the LTR model training. For example, say there are scores $0.01, 0.015, 0.02, 0.008, 0.025$ for five unlabeled samples. We divide them into three levels with an interval of 0.01: $1[0.008], 2[0.01, 0.015]$ and $3[0.02, 0.025]$, and use the levels $1, 2, 3$ to annotate the samples for training the ranker.

Algorithm 1 Learn an Active Learning Ranker

```
Input:
    A small labeled set $L$, a large unlabeled set $U$, basic
    query strategies $[S_1, S_2, \dots, S_m]$, rank training set $T$=[]
Output:
    An active learning ranker $R$;
repeat
    Use all labeled data in $L$ to train a model $M$;
    Test current model $M$ on testing set, getting $Eval(M)$;
    Use current model $M$ to evaluate all samples in $U$;
    Select batches of samples $[B_1, \dots, B_m]$ with basic
    query strategies (such as entropy, lc) to create candidate set $C$;
    for $i \in \{1, 2, \dots, \|C\|\}$ do
        Add $x_i$ to $L$, update $M$ to $M'$, compute $Eval(M')$;
        Extract features of $x_i$, getting $F_i$;
        Add a rank training sample $(F_i, Eval(M') - Eval(M))$ to $T$
    end for
    Add samples with the highest value of $Eval(M') - Eval(M)$ to L;
until rank training data are enough
Train a ranker with training data $T$;
```

With the three main components defined above, we can train an active learning ranker which prefers samples with the largest value for model training under the framework of learning to rank, and then use the ranker to sort and select unlabeled samples. The training steps are briefly summarized in Algorithm 1.

Looking into this learning-based method, we can find that it requires labeled samples like [6] to train the ranker. Recall that active learning is usually used on datasets with limited labeled samples; therefore, we specifically propose two ways to apply this method. One is to first annotate a portion of the samples to train a ranker and then use the ranker to select the remaining unlabeled samples in the same dataset. And the other approach is to train a ranker on an applicable labeled dataset and apply it on other unlabeled datasets of the same task.

4.5 Improvement for State-of-the-art Methods

The above approaches are general and can be applied to basic query strategies and task- or model-specific active learning algorithms. In this paper, we focus on three state-of-the-art informative active learning methods: EGL of word embedding [3] for text classification, Bayesian Uncertainty [7] for bayesian networks and MNLP for NER.

EGL of word embedding (EGL-word) This method is designed based on Eq. (5). With the motivation that word embedding is crucial for learning vectors of text, this method prefers samples with a maximum gradient expectation on the word embedding layer. A max-over-words approach is used to emphasize the particular word in a sentence. EGL of word embedding is computed as:

$$\phi^{\text{EGL-w}}(x) = \max_{j \in x} \sum_{i} P_{\theta}(y_i|x) \| \bigtriangledown l_{E^{(j)}}(L \cup \langle x, y_i \rangle) \|, \quad (12)$$

where $\bigtriangledown l_{E^{(j)}}$ is the gradient on the embedding of word j.

Bayesian Uncertainty (BALD) Bayesian Uncertainty is a way to measure sample uncertainty specially designed for neural networks. We represent its evaluation function as $\phi^{\text{BALD}}(x)$ whose details and theory are in [7].

Maximum Normailized Log Probability (MNLP) In NER task, the log prediction probability is computed as the sum of probabilities over words, so that the LC method naturally tends to select longer sentences. MNLP is proposed to eliminate this bias by normalizing the log probability with sentence length, as:

$$1 - \max_{y_1,\ldots,y_n} \frac{1}{n} \sum_{i=1}^{n} \log P[y_i|y_1, \ldots, y_{i-1}, x_{ij}]. \quad (13)$$

For the three strategies, we can directly apply our methods **WSHS** (in Eq. (9)) and **FHS** (in Eq. (11)) to improve them by introducing historical results, making the sample selection more stable and accurate.

TABLE 2: Comparison of the time and space complexity between basic strategies and our methods.

Complexity	Basic Strategy	WSHS/FHS/LHS
Time	O(T)	O(T+1)
Space	O(N)	O(l*N)

4.6 Discussion about Efficiency

In practical applications, efficiency, including time and space complexity, is an important criterion. Here, we will discuss the efficiencies of our algorithms. According to the problem formulation in Section 2, active learning must evaluate the unlabeled samples in every iteration, and we record the cost time as $O(T)$. Therefore, historical evaluation results can be obtained in each iteration like existing methods without extra computation, and only a small amount of additional time is required to process these historical evaluation results in the current round. This is negligible compared with the model's training and evaluation time, regarded as $O(1)$. Our learning-based method LHS requires some time to train the ranker in advance, but it requires little time to apply the trained ranker to select unlabeled samples. Consequently, **our methods do not have a noticeable increase in time**. As for space complexity, all query strategies need space to store the current evaluation results, as much as $O(N)$. Some extra space is required for storing the historical evaluation results, but this is not a heavy cost currently. Our algorithms need to store only the recent l iterations of results which is usually small, so they do not incur much more space. The comparison of the time and space complexity is shown in table 2. In summary, our methods can not only **achieve the same efficiency as basic methods, but also improve the effect**.

TABLE 3: Statistics of four text classification datasets. #class: number of samples' classes; maxlen: maximum sentence length; N: dataset size; $|V|$: vocabulary size; $Vpre$: number of words with pre-trained embedding.

Dataset	#class	maxlen	N	$\|V\|$	V_{pre}
MR	2	56	10,662	18,765	16,448
SST-2	2	53	9,613	16,185	14,838
Subj	2	23	10,000	21,323	17,913
TREC	6	37	5,952	9,592	9,125

TABLE 4: Statistics of NER datasets. #Docs/ #Sentences/ #Tokens: number of documents/sentences/tokens.

Dataset	Split	#Docs	#Sentences	#Tokens
CoNll-2003 English	Train	946	14,987	203,621
	Dev	216	3,466	51,362
	Test	231	3,684	46,435
CoNll-2002 Spanish	Train	-	8,322	264,715
	Dev	-	1,914	52,923
	Test	-	1,516	51,533
CoNll-2002 Dutch	Train	287	15,806	202,644
	Dev	74	2,895	37,687
	Test	119	5,195	68,875

5 Experiments

In this section, we verify the effectiveness of our methods on two common NLP tasks: text classification and NER. We first decribe the datasets, models and baselines. Then, we report and analyze the experimental results.

5.1 Datasets

5.1.1 *Task1: Text Classification*

We select three widely used benchmarks for the text classification task: two for binary classification and one for multi-class classification. The Subj [41] dataset is used to train the ranker for our LHS method. Statistics of all the datasets are summarized in Table 3.

MR: This dataset [42] contains many movie reviews with one sentence per review. The target is to classify these reviews as positive or negative.

SST-2: This dataset comprises positive and negative reviews from the Stanford Sentiment Treebank SST-1 dataset [43]. The concrete construction process is in [44].

TREC: This is a question dataset [45], involving classifying questions into 6 types.

Subj: This dataset comprises sentences that can be classified as subjective or objective, introduced in [41]. It will be applied in our LHS method to train the ranker.

5.1.2 *Task2: Named Entity Recognition*

For the NER task, we experiment with the most widely used CoNLL-2002/2003 datasets [46] for English, Spanish and Dutch, whose statistics are shown in Table 4. All of them contain four types of named entities: persons (PER), organizations (ORG), locations (LOC) and miscellaneous names (MISC). Following existing work [47], we convert its BIO tagging scheme into the BIOES tagging scheme.

The MR and Subj datasets are randomly split into 10 equal parts for a 10-fold cross validation. For SST-2, TREC and CoNLL-2003, following [43], [45], [46], we use the original split of training/validation/testing sets.

5.2 Models, Parameter Settings, and Metrics

5.2.1 Task1: Text Classification

We select TextCNN as the basic model for text classification; it was proved to achieve strong performance on this task [44], [48]. Its detailed architecture, implementation, and settings for hyperparameters are available in [44]. In this study, we use the word embeddings pretrained by Word2Vec[1]. In terms of active learning, we do 20 rounds of batch sampling for MR and SST-2 with a batch size of 25, while the batch size for TREC is 100 because the multi-class model is harder to train. The first batch of samples is selected at random to initialize the model. Then each time after adding new instances, we fine-tune the model for 10 epochs with the augmented labeled set.

5.2.2 Task2: Named Entity Recognition

We use BiLSTM-CNNs-CRF [47] as the basic model for the NER task. Model details and parameter settings can be found in [47]. We initialize word vectors with public word embedding of corresponding languages. We set the batch size as 100 and develop batch sampling for 20 rounds up to 2,000 annotated samples.

For LHS, we choose LambdaMart [49] as the LTR model and apply the Subj dataset to train the ranker. We use simple LSTM to predict the next evaluation result based the entire historical evaluation sequence.

In each task, for datasets split randomly for cross validation, we repeat the active learning process on each division and calculate the average value as the final result. If the dataset has been split in the original study, we do experiments with its original split for several times.

Metric selection for both tasks follows their original models [44], [47]. For text classification, accuracy was chosen as the evaluation metric, and we use the average F1 on NER. We measure the effect of AL algorithms by the performance of model with the same number of labeled samples or the number of labeled samples required for the model to achieve a certain accuracy.

5.3 Baselines

We compare our methods with several baselines, including basic methods and state-of-the-art specific approaches. In addition, existing methods that have simply made use of historical evaluation results are also considered. Due to our methods focus on informative strategies, the baselines we selected are mainly informative methods.

Random: This method acts as if all unlabeled samples obey i.i.d and randomly samples from the unlabeled set.

TABLE 5: Number of annotated samples required to achieve accuracy of 0.72, 0.73 and 0.735 when various active learning approaches are used for text classification of the MR dataset. Bold indicates the best results, which are achieved by the proposed methods. 500+ means that more than 500 annotated samples are required to achieve the target.

Accuracy	0.72	0.73	0.735
Random	440	500+	500+
Entropy	380	500+	500+
HUS(Entropy)	380	430	500+
WSHS(Entropy)	280	370	500+
FHS(Entropy)	280	405	500+
LHS(Entropy)	**245**	**310**	**340**
LC	300	425	500+
HUS(LC)	370	500	500+
WSHS(LC)	300	420	470
FHS(LC)	260	335	420
LHS(LC)	**245**	**315**	**350**
EGL	300	425	500+
HUS(EGL)	370	500	500+
WSHS(EGL)	300	420	450
FHS(EGL)	270	330	400
LHS(EGL)	**260**	**330**	**370**

Entropy: This method computes entropy for each sample with Eq. (4) and select samples with the largest entropy.

Least confidence (LC): This method calculates scores for samples by Eq. (3), then chooses samples with the least certainty in terms of labeling.

History Uncertainty Sampling (HUS): This method defines uncertainty as the sum of the uncertainty calculated in the last k iterations. In this paper, we use entropy or LC to measure the uncertainty.

EGL of word embedding(EGL-word): Selects samples resulting in the largest gradient on the word embedding computed as Eq. (12)

Bayesian Uncertainty (BALD): Selects samples with the largest bayesian uncertainty. The score function is in [7].

Maximum Normalized Log Probability (MNLP): Select samples with the largest score calculated by Eq. (13)

5.4 Experimental Results

We conduct various experiments to compare the effectiveness of our methods and baselines. For our first two strategies, WSHS and FHS, we do experiments on both the text classification and NER. For the third strategy LHS, we train the ranker on the binary classification dataset Subj so that we only apply it on the other two 2-class datasets of text classification. Reports and analyses are as follows.

5.4.1 Experimental Results on General Query Strategies

Figure 3 shows the results of existing general query strategies and our general methods WSHS, FHS and LHS on both text classification and NER tasks. Table 5 lists the number of samples to be annotated when various approaches are applied to achieve a certain result. After analyzing Figure 3 and Table 5, we come to four conclusions:

(1) **In most cases, our three general active learning approaches i.e. WSHS, FHS and LHS outperform the corresponding baselines on the two NLP tasks, when combined with the entropy, LC and EGL method. This**

[1] https://code.google.com/archive/p/word2vec/

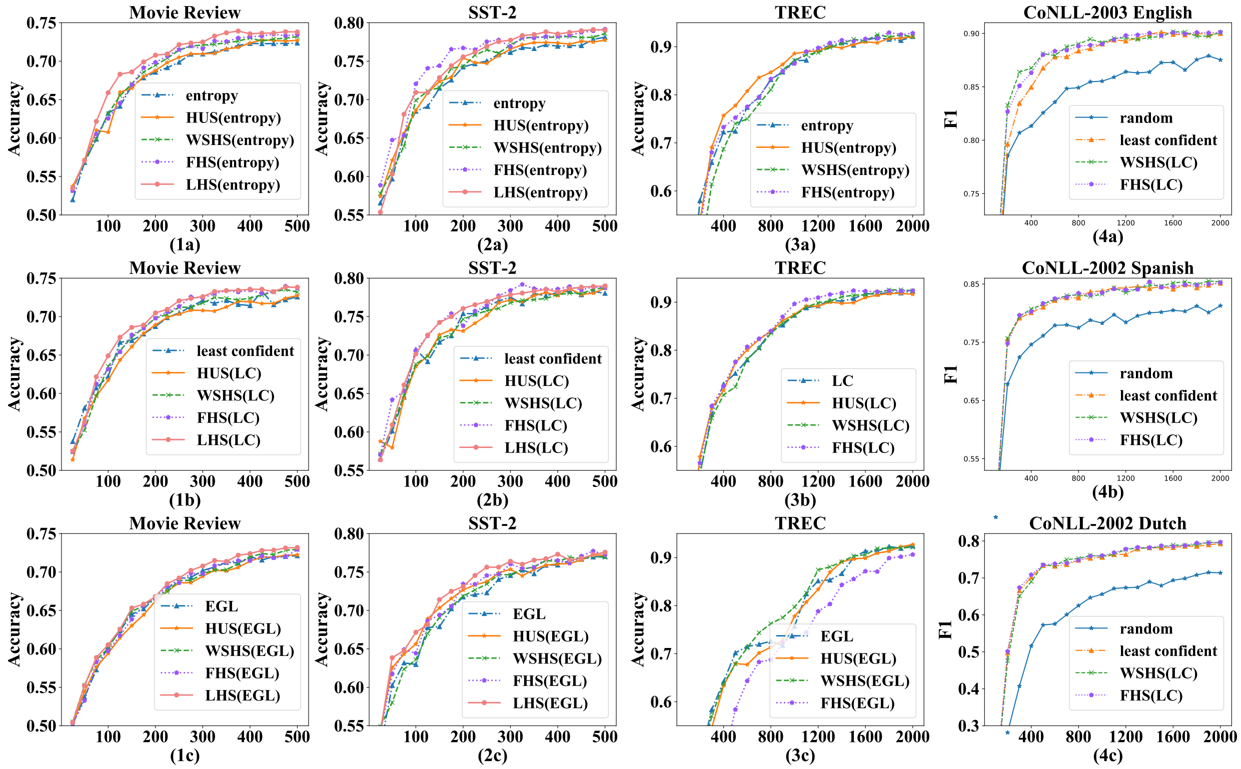


Fig. 3: The performances of general active learning query strategies. The x-axes show the number of labeled samples. The 1^{st}, 2^{nd}, 3^{rd} column: experimental results of text classification, and the 4^{th} column: experimental results of NER.

result fully demonstrates the effectiveness of some information contained in historical evaluation results and our strategies. Here are some examples to show the result more clearly. Focusing on the first column of Figure 3 which shows the experimental results of the text classification on the MR dataset, we find that the curves of our methods lie above the curves of other baselines. In Table 5, the basic entropy query strategy and HUS require 380 labeled samples to achieve an accuracy of 0.72. However, our WSHS and FHS methods combined with the entropy strategy require only 280 samples, saving almost 100 samples' annotation cost. As for NER, as shown in the 4^{th} column, WSHS and FHS can achieve a relatively high F1-score much more quickly than the baselines, showing better performance, especially for the English dataset. When sufficient labeled data are available, our methods are also comparable with other approaches.

(2) **Benefitting from processing and analyzing the historical evaluation results more globally, our three strategies outperform the closest baseline HUS [20].** Pay attention to Figure 3, we find the curves of our three general active learning strategies lie upon the curves of HUS, saving more than 100 samples' annotation costs on the MR dataset. HUS performs similarly to the entropy approach without obvious improvement, the same as the results in [20]. This may be because HUS regards all historical uncertainty scores as having the same importance and uses the direct sum to select samples, which is inconsistent with the intuition and facts. We believe that the latter models should have more influence on the current model and that historical results far from the current iteration may bring some noise.

(3) Comparing our first two heuristic methods, **FHS performs a little better than WHSH in most situations**. Looking at the curves of WSHS and FHS in Figure 3, we find that the curves of FHS lie above those of WSHS in most subfigures. Looking at Table 5, we find that up to 85 samples' annotation costs can be saved by using the approach FHS combined with LC to achieve an accuracy of 0.73 on the MR dataset, compared with WSHS. FHS is designed to incorporate the fluctuation of historical evaluation sequences into sample selection, **and the results prove that the fluctuation of historical sequences is an effective measurement of sample uncertainty**. Therefore, we suggest that higher priority be given to the FHS query strategy.

(4) Focusing on Figure 3, **we observe the learning-based method LHS performs the best on both binary-classification datasets, especially on MR**. The 6^{th} row in Table 5 indicates that LHS combined with entropy requires only 340 annotated samples to achieve an accuracy of 0.735 on the MR dataset, so it incurred the lowest cost of all the general active learning methods. But we find the improvements of LHS over FHS on the SST-2 dataset are not so significant. To make more detailed analysis, we experiment to compare the average WSHS score and FHS score of all selected samples in our three proposed methods. From the results shown in Table 6, we observe that both WSHS and FHS focus on only one aspect of historical evaluation results and select samples with high score on the corresponding feature. However, LHS selects more comprehensive samples with both relatively high WSHS and FHS scores. Furthermore, it also combines some other features extracted from the historical sequences by automatically learning. Thus, we infer the improvements of LHS are not so significant

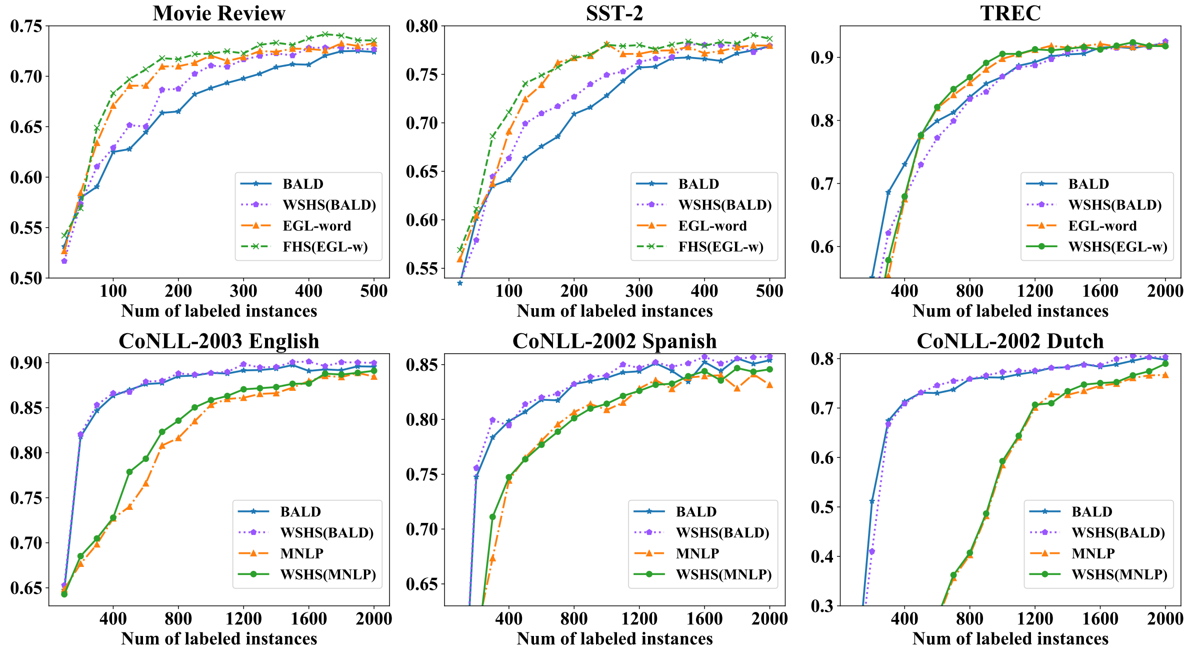


Fig. 4: The performances of state-of-the-art active learning approaches combined with historical evaluation sequences.

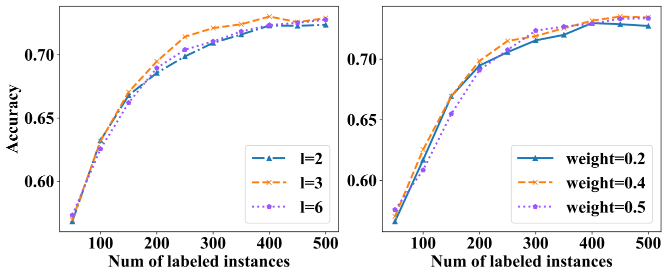


Fig. 5: Analysis of different hyper-parameters.

over FHS on some datasets may due to that LHS does not select better samples based on FHS method but focuses on different samples. But its best performance still proves that **there are many effective features of the historical sequence helpful for selecting samples such as the trend, and the values in the sequence and its fluctuation**.

TABLE 6: Average WSHS/FSH score of selected samples in different methods.

Methods	WSHS score	FHS score
WSHS	**1.1707**	0.000012
FHS	1.0077	**0.005074**
LHS	0.8255	0.001735

5.4.2 *Analysis of Hyper-parameters*

In our first two heuristic strategies WSHS and FHS, there are some critical hyper-parameters, i.e the size of the history window and the weight of the fluctuation. To analyze how these hyper-parameters affect the performance, we conduct experiments on the MR dataset with different sizes of the history window for WSHS, and we fix the size as 3 to experiment with different variance weights for FHS. All the results are illustrated in Figure 5. In the left picture, we find the window size has some influence on the WSHS method, and a moderate size performs the best. We analyze it may be because when the size is small, we can not make full use of the information of the historical evaluation results, but too early results will bring some noise. Thus, we suggest the history window size to be 3-5. As for the FHS, the variance weights close to 0.5 perform better.

TABLE 7: Experimental results of ablation study.

#Samples	100	200	300	400	500
LHS	0.6590	0.7078	0.7250	0.7356	0.7380
-history sequence	0.6544	0.7073	0.7137	0.7263	0.7285
-fluctuation	0.6546	0.7080	0.7191	0.7238	0.7297
-sequence trend	0.6515	0.7131	0.7218	0.7267	0.7361
-next prediction	0.6515	0.7131	0.7218	0.7267	0.7361
-probability	0.6591	0.7114	0.7251	0.7326	0.7356

5.4.3 *Ablation Study of LHS*

In the proposed learning based query strategy LHS which learns how to select samples with the information contained in the historical sequences, we extract a series of features for each unlabeled sample, including historical evaluation results, fluctuation of historical sequences, trend of historical sequences, the predicted next result and output probability of the model. To further analyze the effect of each feature, we carry out an ablation study to turn off all feature one-by-one and compare the results. All the experimental results are illustrated in Table 7. From Table 7, we can see that there are some impacts on the performance after removing any feature, which indicates the various information contained in the historical sequence can help evaluate the effectiveness of unlabeled samples for model training. The two features, historical evaluation results and fluctuation of historical sequences, have the greatest impacts. And they are exactly the two aspects we focus on in the first two heuristic algorithms, demonstrating that WSHS and FHS can mine critical information in the historical sequences.

5.4.4 Improvement on State-of-the-art Methods

In addition to the basic active learning methods, we also combine our proposed methods with state-of-the-art approaches. The comparison results are shown in Figure 4.

Figure 4 indicates that **the EGL-word, BALD and MNLP approaches on both the text classification task and NER are much improved by introducing historical evaluation results with the WSHS or FHS methods**. On the MR dataset, EGL-word with fluctuation of the historical sequence (FHS(EGL-w)) and BALD combined with the WHSH method promote the original approaches greatly. Similarly, results on the SST-2 dataset also support the conclusion. About the NER task, the BALD based methods consistently perform better than those MNLP based methods. But the historical sequences improves the MNLP approach more. In summarize, taking historical evaluation results into consideration also helps these state-of-the-art methods select samples more accurately.

6 Conclusion

In this paper, we argue that historical evaluation results are helpful for comparison when selecting samples in active learning and that full use needs to be made of the information contained in the historical sequences. On the basis of this idea, we proposed several general heuristic methods that incorporate these historical sequences into existing active learning strategies, including the weighted sum and fluctuation of historical evaluation results. We also introduced a learning based query strategy that learns how to select samples automatically based on features extracted from the historical sequences, under the framework of learning to rank (LTR). In addition, we improved some state-of-the-art query strategies by exploiting historical evaluation results. Our methods can be easily implemented in practical applications and would not affect the efficiency. Experimental results on two tasks text classification and NER verified that our approaches could outperform all the basic baselines and state-of-the-art methods. To improve our research in the future, we will further analyze the historical evaluation sequences and explore more effective features.

Acknowledgment

Zhicheng Dou is the corresponding author. This work was supported by National Natural Science Foundation of China No. 61872370 and 61832017, Beijing Outstanding Young Scientist Program No. BJJWZYJH012019100020098 and in part by Shandong Provincial Natural Science Foundation under Grant ZR2019ZD06.

References

[1] B. Settles, “Active learning literature survey,” University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[2] S. Tong and D. Koller, “Support vector machine active learning with applications to text classification,” *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2001.

[3] Y. Zhang, M. Lease, and B. C. Wallace, “Active discriminative text representation learning,” in *AAAI*, 2017, pp. 3386–3392.

[4] Y. Deng, K. Chen, Y. Shen, and H. Jin, “Adversarial active learning for sequences labeling and generation,” in *Proceedings of IJCAI*, 2018, pp. 4012–4018.

[5] A. Ekbal, S. Saha, and D. Singh, “Active machine learning technique for named entity recognition,” in *Proceedings of ICACCI, Chennai, India, August 3-5*, 2012, pp. 180–186.

[6] M. Fang, Y. Li, and T. Cohn, “Learning how to active learn: A deep reinforcement learning approach,” in *Proceedings of EMNLP 2017,the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2017, pp. 595–605.

[7] Y. Gal, R. Islam, and Z. Ghahramani, “Deep bayesian active learning with image data,” in *Proceedings of ICML*, 2017, pp. 1183–1192.

[8] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, “Cost-effective active learning for deep image classification,” *IEEE Trans. Circuits Syst. Video Techn.*, vol. 27, no. 12, pp. 2591–2600, 2017.

[9] R. M. Silva, G. de Castro Mendes Gomes, M. S. Alvim, and M. A. Gonçalves, “Compression-based selective sampling for learning to rank,” in *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, 2016, pp. 247–256.

[10] Z. Li and M. de Rijke, “The impact of linkage methods in hierarchical clustering for active learning to rank,” in *Proceedings of SIGIR*, 2017, pp. 941–944.

[11] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, and P. S. Yu, “Active learning: A survey,” in *Data Classification: Algorithms and Applications*, 2014, pp. 571–606.

[12] Y. Shen, H. Yun, Z. C. Lipton, Y. Kronrod, and A. Anandkumar, “Deep active learning for named entity recognition,” in *6th International Conference on Learning Representations, ICLR*, 2018.

[13] I. Muslea, S. Minton, and C. A. Knoblock, “Selective sampling with redundant views,” in *Proceedings of AAAI, the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on on Innovative Applications of Artificial Intelligence, July 30 - August 3, 2000, Austin, Texas, USA.*, 2000, pp. 621–626.

[14] H. S. Seung, M. Opper, and H. Sompolinsky, “Query by committee,” in *Proceedings of the Fifth Annual ACM, COLT*, 1992, pp. 287–294.

[15] B. Settles, M. Craven, and S. Ray, “Multiple-instance active learning,” in *Proceedings of NIPS,Vancouver, British Columbia, Canada, December 3-6*, 2007, pp. 1289–1296.

[16] S. Huang, R. Jin, and Z. Zhou, “Active learning by querying informative and representative examples,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 1936–1949, 2014.

[17] H. T. Nguyen and A. W. M. Smeulders, “Active learning using pre-clustering,” in *Proceedings of ICML, Banff, Alberta, Canada, July 4-8*, 2004.

[18] S. Kim, Y. Song, K. Kim, J. Cha, and G. G. Lee, “Mmr-based active machine learning for bio named entity recognition,” in *Proceedings of NAACL-HLT*, 2006.

[19] J. Zhu and M. Y. Ma, “Uncertainty-based active learning with instability estimation for text classification,” *TSLP*, vol. 8, no. 4, pp. 5:1–5:21, 2012.

[20] M. Davy and S. Luz, “Active learning with history-based query selection for text categorisation,” in *Advances in Information Retrieval, 29th European Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007, Proceedings*, 2007, pp. 695–698.

[21] B. Settles and M. Craven, “An analysis of active learning strategies for sequence labeling tasks,” in *Proceedings of EMNLP*, 2008, pp. 1070–1079.

[22] J. Goodman, “Exponential priors for maximum entropy models,” in *Proceedings of NAACL-HLT, Boston, Massachusetts, USA, May 2-7*, 2004, pp. 305–312.

[23] S. Kullback and R. A. Leibler, “On information and sufficiency,” *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[24] J. G. Carbonell and J. Goldstein, “The use of mmr, diversity-based reranking for reordering documents and producing summaries,” in *Proceedings of SIGIR'98*, 1998, pp. 335–336.

[25] B. Du, Z. Wang, L. Zhang, L. Zhang, W. Liu, J. Shen, and D. Tao, “Exploring representativeness and informativeness for

active learning," *IEEE Trans. Cybernetics*, vol. 47, no. 1, pp. 14–26, 2017.

[26] D. Shen, J. Zhang, J. Su, G. Zhou, and C. L. Tan, "Multi-criteria-based active learning for named entity recognition," in *Proceedings of ACL, 21-26 July, Barcelona, Spain.*, 2004, pp. 589–596.

[27] G. Zheng, S. Mukherjee, X. L. Dong, and F. Li, "Opentag: Open attribute value extraction from product profiles," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, 2018, pp. 1049–1058.

[28] S. Sinha, S. Ebrahimi, and T. Darrell, "Variational adversarial active learning," in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 2019, pp. 5971–5980.

[29] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 2672–2680.

[30] Y. Chen, T. A. Lasko, Q. Mei, J. C. Denny, and H. Xu, "A study of active learning methods for named entity recognition in clinical text," *Journal of Biomedical Informatics*, vol. 58, pp. 11–18, 2015.

[31] Y. Chen, T. A. Lasko, Q. Mei, Q. Chen, S. Moon, J. Wang, K. Nguyen, T. Dawodu, T. Cohen, J. C. Denny, and H. Xu, "An active learning-enabled annotation system for clinical named entity recognition," *BMC Med. Inf. & Decision Making*, vol. 17, no. 2, pp. 35–44, 2017.

[32] R. S. Sutton and A. G. Barto, *Reinforcement learning - an introduction*, ser. Adaptive computation and machine learning. MIT Press, 1998.

[33] M. Liu, W. L. Buntine, and G. Haffari, "Learning how to actively learn: A deep imitation learning approach," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 2018, pp. 1874–1883.

[34] S. Ross, G. J. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, 2011, pp. 627–635.

[35] B. Long, J. Bian, O. Chapelle, Y. Zhang, Y. Inagaki, and Y. Chang, "Active learning for ranking through expected loss optimization," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1180–1191, 2015.

[36] N. Houlsby, F. Huszar, Z. Ghahramani, and M. Lengyel, "Bayesian active learning for classification and preference learning," *CoRR*, vol. abs/1112.5745, 2011.

[37] A. Siddhant and Z. C. Lipton, "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, Brussels, Belgium, October 31 - November 4, 2018*, 2018, pp. 2904–2909.

[38] K. H. Hamed and A. R. Rao, "A modified mann-kendall trend test for autocorrelated data," *Journal of Hydrology*, vol. 204, no. 1-4, pp. 182–196, 1998.

[39] D. Ömer Faruk, "A hybrid neural network and arima model for water quality time series prediction," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 4, pp. 586–594, 2010.

[40] F. A. Gers, D. Eck, and J. Schmidhuber, "Applying LSTM to time series predictable through time-window approaches," in *Artificial Neural Networks - ICANN 2001, International Conference Vienna, Austria, August 21-25, 2001 Proceedings*, 2001, pp. 669–676.

[41] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of ACL*, 2004, pp. 271–278.

[42] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *ACL 2005*, 2005, pp. 115–124.

[43] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceddings of EMNLP*, 2013, pp. 1631–1642.

[44] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of EMNLP, October 25-29, Doha, Qatar*, 2014, pp. 1746–1751.

[45] X. Li and D. Roth, "Learning question classifiers," in *19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002*, 2002.

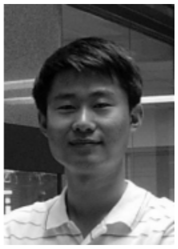
[46] E. F. T. K. Sang and F. D. Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *Proceddings of CoNLL*, 2003, pp. 142–147.

[47] X. Ma and E. H. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," in *Proceedings of ACL,August 7-12, Berlin, Germany, Volume1: Long Papers*, 2016.

[48] Y. Zhang and B. C. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," in *Proceedings of IJCNLP, Taipei, Taiwan, November 27 -December 1*, 2017, pp. 253–263.

[49] Q. Wu, C. J. C. Burges, K. M. Svore, and J. Gao, "Adapting boosting for information retrieval measures," *Inf. Retr.*, vol. 13, no. 3, pp. 254–270, 2010.

Jing Yao is working toward an MS in the School of Information, Renmin University of China. Her research interests include information retrieval, natural language processing and machine learning.



Zhicheng Dou is currently a professor at Renmin University of China. He received his Ph.D. and B.S. degrees in computer science and technology from the Nankai University in 2008 and 2003, respectively. He worked at Microsoft Research Asia from July 2008 to September 2014. His current research interests are Information Retrieval, Natural Language Processing , and Big Data Analysis. He received the Best Paper Runner-Up Award from SIGIR 2013, and the Best Paper Award from AIRS 2012. He served as the program co-chair of the short paper track for SIGIR 2019. His homepage is http://playbigdata.ruc.edu.cn/dou. He is a member of the IEEE.

Jian-Yun Nie Jian-Yun Nie is a professor with the Universit de Montral, Canada. He has published more than 150 papers in information retrieval and natural language processing in journals and conferences. He served as a general cochair of the ACMSIGIR Conference in 2011. He is currently on the editorial board of seven international journals. He has been an invited professor and researcher at several universities and companies. He is a member of the IEEE.

Ji-Rong Wen Ji-Rong Wen received the BS and MS degrees from the Renmin University of China, and the PhD degree from the Chinese Academy of Science, in 1999. He is a professor at the Renmin University of China. He was a senior researcher and research manager with Microsoft Research from 2000 to 2014. His main research interests include web data management, information retrieval (especially web IR), and data mining. He is a senior member of the IEEE.