Rapid Performance Gain through Active Model Reuse

Feng Shi Yu-Feng Li
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
{shif, liyf}@lamda.nju.edu.cn

Abstract

Model reuse aims at reducing the need of learning resources for a newly target task. In previous model reuse studies, the target task usually receives labeled data *passively*, which results in a slow performance improvement. However, learning models for target tasks are often required to achieve good enough performance rapidly for practical usage. In this paper, we propose the ACMR (Active Model Reuse) method for the rapid performance improvement problem. Firstly, we construct queries through pre-trained models to facilitate the active learner when labeled examples are insufficient in the target task. Secondly, we consider that pre-trained models are able to filter out not very necessary queries so that AcMR can save considerable queries compared with direct active learning. Theoretical analysis verifies that ACMR requires fewer queries than direct active learning. Experimental results validate the effectiveness of ACMR.

1 Introduction

In traditional machine learning, a learning model is tailored to the target task. Nowadays, however, with the popularity of machine learning, unlike training data which is often hard to share due to the privacy issue [Albrecht, 2016], a great deal of well-trained machine learning models have been available for use. For example, these models have been trained with a large amount of labeled training data [Parkhi *et al.*, 2015]; with smart optimization techniques [Kingma and Ba, 2014]. All these models work well on the specific tasks, but have to be discarded once the target task changes, which causes the user to retrain a new model for the target task.

Model reuse [Zhou, 2016] tries to reduce the learning resources for a new target task with the exploitation of pre-trained models (as demonstrated in Figure 1), which has attracted much attention and has shown promising performance when the labeled examples are limited for the target task. For example, Yang *et al.* [2017b] makes predictions for new instances from a collection of different pre-trained models and shows encouraging results; Ye *et al.* [2018] adapts the pre-trained models to a new environment with different features and shows that the performance will be boosted. However,

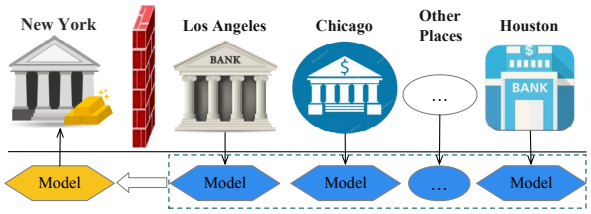

Figure 1: A practical example of model reuse. It is often challenging to share bank data due to the privacy issue. Instead we can reuse pre-trained models of banks to help improve the performance.

previous model reuse studies usually assume that the labeled data for the target task are *passively* collected. This leads to a limited performance improvement for the target task, which may be difficult to meet the demands. Since in many practical applications, it is expected that the performance of the target model can be good enough and quickly improved. It is evident that this scenario is quite different from the classic model reuse studies. We call this kind of model reuse the "rapid performance improvement" problem.

The problem is related to but different to standard active learning [Settles, 2012], which tries to select the most informative instances to be labeled. The informativeness is typically defined as maximal expected improvement in classification accuracy [Huang *et al.*, 2010]. Many sample selection criteria (e.g., uncertainty sampling [Lewis and Gale, 1994], query by committee (QBC) [Seung *et al.*, 1992]) have been devoted. However, they ignore the use of many pre-trained models, which results in a slow performance improvement for target tasks or needing a large amount of query costs.

As illustrated in Figure 2, for a new classification task in 20 Newsgroups, we directly employ active learning method (e.g., QBC [Seung *et al.*, 1992]) or model reuse method (e.g., Safer [Li *et al.*, 2017]). As shown, both of the two approaches yield slow performance gains or require a large number of queries to achieve good enough accuracy.

In this paper, we study the rapid performance improvement problem and propose the ACMR (Active Model Reuse) method. As illustrated in Figure 3, unlike traditional active learning, we consider pre-trained models to help construct queries, facilitating active learner when labeled examples are insufficient on the target task. Moreover, we leverage pre-trained models to filter out not very necessary queries so that considerable queries could be saved compared with direct ac-

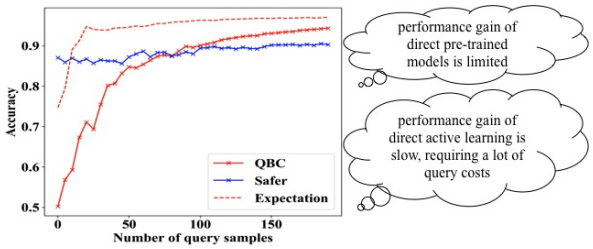

Figure 2: A true example of learning curves of two methods on a classification task. We expect that with the help of pre-trained models, the performance of active learner can get rapid improvement.

tive learning. During the learning process, the relationships between pre-trained models and the target task are continually updated such that they can predict unlabeled samples more accurately. Theoretical analysis verifies that ACMR requires fewer queries than active learning. Experiments show the superior performance of the proposed method.

This paper is structured as follows. We first review related works and then present our proposed method. Next we show the experiments. Finally we conclude this work.

2 Related Work

Reusability has been emphasized as a crucial characteristic of the new concept of learnware [Zhou, 2016]. It would be ideal if models can be reused in scenarios that are very different from their original training ones. This is of course a big challenge, whereas reusing models have already been demonstrated very useful. Li *et al.* [2013] has shown that by starting from a trained model, it is easier to construct a model. FMR [Yang *et al.*, 2017a] integrates the discriminative ability of fixed models into deep network training, and achieves promising performance in various applications.

Active learning tries to query the most informative samples and mainly focuses on representative and uncertain information in the unlabeled data [Settles, 2012]. For example, one of the most common strategies is the uncertainty-based selection [Lewis and Gale, 1994], in which the certainties are measured according to the predictions on new unlabeled samples obtained from the initial classifiers. Chakraborty *et al.* [2015] combines the uncertainty and representativeness into a convex framework to perform active learning loops. Huang *et al.* [2010] queries the informative and representative samples based on a min-max framework. Wang and Ye [2015] puts the discrimination and representativeness together via a trade-off parameter to query the i.i.d samples.

Transfer learning is one way of implementing model reuse. However, it typically works on a more general problem setting [Pan and Yang, 2010], such as transferring knowledge through data representation [Pan *et al.*, 2011], model structures [Long *et al.*, 2017] and so on. There are several studies on actively transfer learning [Shi *et al.*, 2008], however, they require additional data rather than strict model reuse setting.

In recent years, robust model reuse based on semi-supervised learning [Li *et al.*, 2017; Li and Liang, 2019; Li *et al.*, 2019] has been proposed, which works on deriving performance-safe prediction when labeled examples in target tasks are limited, where safe prediction means that the performance would not be worse than direct supervised learning using only limited labeled data. Although these studies are able to achieve prudent performance, the performance improvement is often relatively limited since they never consider involving more labeled data.

Table 1: Summary of Notation

Notation	Meaning
N	number of training data
$y_t \in \{+1, -1\}$	ture label of instance $\mathbf{x}_t$
$\mathcal{L}$	labeled data in the target task
$\mathcal{U}$	unlabeled data in the target task
$\{f_1, f_2, \ldots, f_k\}$	k pre-trained models
$f_j^{(t)}$	the prediction of f_j on $\mathbf{x}_t$
$\boldsymbol{\eta} = [\eta_1, \eta_2, \ldots, \eta_k]$	the weight vector of k models
$l_j^{(t)} = (1 - y^{(t)} f_j^{(t)})_+$	the hinge loss of j-th model for $\mathbf{x}_t$
$L_j = \sum_{\mathbf{x}_t \in \mathcal{L}} l_j^{(t)}$	the empirical loss of j-th model
$f_{\mathcal{L}}$	the active learner built on $\mathcal{L}$
$f_{\mathcal{L}}^{(t)}$	the prediction of $f_{\mathcal{L}}$ on $\mathbf{x}_t$

3 The ACMR Method

3.1 Notation and Setting

Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_{n_l}, y_{n_l}), \mathbf{x}_{n_l+1}, \ldots, \mathbf{x}_N\}$ be the training data set of the target task that consists of n_l labeled instances and $n_u = N - n_l$ unlabeled ones. Each instance $\mathbf{x}_t = [x_{t1}, x_{t2}, \ldots, x_{td}]^\top$ is a vector of d dimensions. $\mathcal{L} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_{n_l}, y_{n_l})\}$ is the labeled data set and $\mathcal{U} = \{\mathbf{x}_{n_l+1}, \ldots, \mathbf{x}_N\}$ is the unlabeled one. Table 1 summarizes the notations in the paper.

Formally, suppose that we have obtained k pre-trained models $\{f_1, f_2, \ldots, f_k\}$. We let $f_{\mathcal{L}}$ denote the active learner obtained by training a supervised learner with only labeled data. The goal of model reuse is to derive a better model: $f^+ = g(\{f_1, f_2, \ldots, f_k\}, f_{\mathcal{L}})$, which often outperforms, meanwhile would not be worse than $f_{\mathcal{L}}$. g represents the form of model reuse, specifically, with the help of pre-trained models, f^+ can be build by structural risk minimization:

$$\min_{f^+} \; L(\mathcal{L}_Y, f^+(\mathcal{L}_X)) + \lambda \Omega(f^+) \tag{1}$$

where $\mathcal{L}_X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n_l}\}$, $\mathcal{L}_Y = \{y_1, y_2, \ldots, y_{n_l}\}$, L is a loss function, e.g., mean square loss, hinge loss, etc and Ω is a regularizer. The smaller the value of the loss function, the better the performance. The forms of the loss function, regularizer and f^+ are flexible to design for specific tasks. In this paper, we use the hinge loss and logistic regression [Harrell, 2015] to be the form of loss function and f^+ respectively.

3.2 Deficiencies of Baseline Approaches

Previous works on model reuse typically assume that when labeled examples for the target task are limited, model reuse could guarantee performance improvement [Ye *et al.*, 2018; He *et al.*, 2018]. However, this facilitation is only effective when the number of labeled samples is small and the performance improvement is often relatively limited. To get a good enough model, we still need a lot of queries.

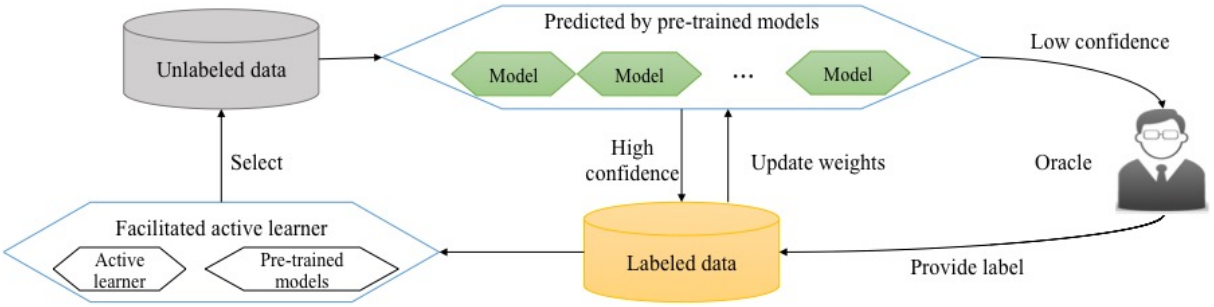

Figure 3: Model architecture of ACMR. In contrast to traditional active learning, we propose to facilitate the active learner with pre-trained models, and we also use pre-trained models to filter out not very necessary queries. During the learning process, the weights of k models are continually updated such that they can predict unlabeled samples more accurately.

Assume we have obtained confident weights (denoted by $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_k]$) of the pre-trained models about the target task, according to the weighted voting method [Dietterich, 2000], the prediction $\hat{y}^{(t)}$ of the unlabeled instance $\mathbf{x}_t$ is

$$\hat{y}^{(t)} = \underset{c\in\{-1,+1\}}{\arg\max} \sum_{j=1}^{k} \eta_j \cdot \mathbb{I}(f_j^{(t)} = c) \tag{2}$$

where $\mathbb{I}(z) = 1$ if z is true and 0 otherwise. This prediction, however, may be risky since we can not get accurate prior weights especially when there are very few labeled examples in target tasks [Pan and Yang, 2010]. As illustrated in Figure 2, we consider that the pre-trained models are helpful while their performance is not good enough.

Thus, rather than fully trusting pre-trained models, we consider trusting the pre-trained models partially and leverage pre-trained models to filter out not very necessary queries generated by direct active learning to facilitate an effective active learner. Moreover, once the label capacity is enhanced, we continually update the confident weights of pre-trained models to improve their confidence for target task. It is worthy that the above two steps promote each other in the active learning iterations. In the following, we introduce them respectively and present the theoretical justification.

3.3 Actively Reuse Pre-trained Models

When the predictions of pre-trained model are risky, we may have the following observations [Shi *et al.*, 2008]: i) Pre-trained models assign $\mathbf{x}_t$ with a class label that is different from the one given by the facilitated active learner; ii) The posteriors of the pre-trained models are low.

According to the above considerations, we can design a query indicator function $\theta(\mathbf{x}_t)$ to reflect the necessity for unlabeled instance $\mathbf{x}_t$ to query the label:

$$P(\hat{y}^{(t)}|\mathbf{x}_t) = \sum_{j=1}^{k} \eta_j P_j(\hat{y}^{(t)}|\mathbf{x}_t) \cdot \mathbb{I}(f_j^{(t)} = \hat{y}^{(t)}) \tag{3}$$

$$\alpha(\mathbf{x}_t) = \big(1 - \mathbb{I}(\hat{y}^{(t)} \neq f_{\mathcal{L}}^{(t)})\big) P(\hat{y}^{(t)}|\mathbf{x}_t) \tag{4}$$

$$\theta(\mathbf{x}_t) = \big(1 + \alpha(\mathbf{x}_t)\big)^{-1} \tag{5}$$

$P_j(\hat{y}^{(t)}|\mathbf{x}_t)$ is the predicted probability of pre-trained model f_j for instance $\mathbf{x}_t$. As it can be seen, $\alpha(\mathbf{x}_t)$ is related to the posteriori probability $P(\hat{y}^{(t)}|\mathbf{x}_t)$, and $\alpha(\mathbf{x}_t) = 0$ if the supervised classifier $f_{\mathcal{L}}^{(t)}$ and the pre-trained models have assigned different labels to $\mathbf{x}_t$. In general, the larger the value $\alpha(\mathbf{x}_t)$ is, the less we need to query the label.

Moreover, because mislabeling of the few instances can put significant negative effect on accuracy, we further set $\theta(\mathbf{x}_t) = \big(1+\alpha(\mathbf{x}_t)\big)^{-1}$ so as to guarantee the possibility (necessity) to query is greater than 50%. In other words, we trust the label given by the pre-trained models only when its confidence reflected by $\alpha(\mathbf{x}_t)$ is very high. Accordingly, with the value of $\theta(\mathbf{x}_t)$, we randomly generate a real number R within 0 to 1, and then the decision function $\mathbb{F}(\mathbf{x}_t)$ is defined as

$$\mathbb{F}(\mathbf{x}_t) = \begin{cases} 0, & \text{if } R > \theta(\mathbf{x}_t) \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

According to Eq.(6), if $\mathbb{F}(\mathbf{x}_t) = 0$, then it means that the instance $\mathbf{x}_t$ should be labeled by the pre-trained models; otherwise, $\mathbf{x}_t$ should be labeled by the domain experts. In other words, we label the instance $\mathbf{x}_t$ by the pre-trained models with probability $1 - \theta(\mathbf{x}_t)$.

We then derive its sampling error bound to demonstrate its ability to mitigate the deficiencies for direct approaches, which utilizes the predictions only when they are accurate enough. Additionally, we show its querying bound to validate the claim that ACMR can reduce labeling cost by querying fewer examples. In the algorithm ACMR, let ε_p and ε_a denote the expected error of the pre-trained models and the active learner $f_{\mathcal{L}}$ respectively, and let $\delta = \varepsilon_p + \varepsilon_a$. The results of theoretical analysis are shown in Table 2.

Table 2: Results of Theoretical Analysis

Evaluation Criterion	Bound of Value
Sampling Error	$\frac{\varepsilon_p^2}{1+(1-\varepsilon_p)}$
Querying Rate	$\delta + \frac{1-\delta}{1+(1-\varepsilon_p)}$

Theorem 1. *In the algorithm* ACMR*, we assume that* $\varepsilon_a \leq \varepsilon_p$*, then the sampling error* ε *for* ACMR *satisfies:*

$$\varepsilon \leq \frac{\varepsilon_p^2}{1 + (1 - \varepsilon_p)} \tag{7}$$

Proof. *According to the analysis of the decision function* $\mathbb{F}(\mathbf{x})$ *described above,* ACMR *makes wrong decision only when both the pre-trained models and the active learner* $f_{\mathcal{L}}$ *agree on the wrong label. In this case,* ACMR *has probability* $1-\theta(\mathbf{x})$ *to trust the classification result given by the pre-trained models, where* $\theta(\mathbf{x})$ *is defined in Eq.*(5). *Thus, the sampling error of* ACMR *can be written as* $\varepsilon = \varepsilon_p\varepsilon_a(1-\theta(\mathbf{x})) \leq \varepsilon_p^2(1-\theta(\mathbf{x}))$. *Moreover, in this situation,* $\theta(\mathbf{x}) = \frac{1}{1+(1-\varepsilon_p)}$. *Thus,*

$$\varepsilon \leq \frac{\varepsilon_p^2 \times (1-\varepsilon_p)}{1+(1-\varepsilon_p)} \leq \frac{\varepsilon_p^2}{1+(1-\varepsilon_p)} \tag{8}$$

Theorem 2. *In the algorithm* ACMR*, for an unlabeled instance, the probability that* ACMR *queries the label from the experts (with cost) satisfies:*

$$P(Q) \leq \delta + \frac{1-\delta}{1+(1-\varepsilon_p)} \tag{9}$$

Proof. *According to the analysis of the decision function* $\mathbb{F}(\mathbf{x})$*,* ACMR *will query the experts to label the instance when the pre-trained models and the active learner hold different predictions on the classification result. And when the two classifiers agree on the result, it still has probability* $\theta(\mathbf{x})$ *to query the experts. Thus:*

$$\begin{aligned} P(Q) &= \varepsilon_a(1-\varepsilon_p) + \big[\varepsilon_p\varepsilon_a + (1-\varepsilon_p)(1-\varepsilon_a)\big]\theta(\mathbf{x}) \\ &\quad + (1-\varepsilon_a)\varepsilon_p \\ &= \theta(\mathbf{x}) + (\varepsilon_p + \varepsilon_a - 2\varepsilon_p\varepsilon_a)\big(1-\theta(\mathbf{x})\big) \\ &\leq \delta + (1-\delta)\theta(\mathbf{x}) \\ &\leq \delta + \frac{1-\delta}{1+(1-\varepsilon_p)} \end{aligned} \tag{10}$$

Remark. *In contrast to Shi* et al. *[2008] which suffers from exponential term* $\exp(-N^{-1})$*, the results in Theorem 1 and Theorem 2 show that* ACMR *avoid this issue and improve the bound evidently. From theoretical analysis, we can find that the sampling error and querying probability of the proposed approach are bounded and related to* ε_p *and* ε_a*. It is worth noticing that the more accurate the pre-trained models and active learner become, the less necessarily will we query the experts to label the instance. In the next section we will talk about how to reduce* ε_p *to improve performance in the whole learning process.*

3.4 Update Weights for Pre-Trained Models

It is well known that ensemble learning often outperforms individual models [Zhou, 2012]. However, simply building a uniformly weighted ensemble of the models is suboptimal, since it does not measure the precise relationships between each pre-trained model and the target task.

Inspired by Murugesan *et al.* [2016], we employ an error-driven update rule in which the weights of pre-trained models are updated only when the prediction of pre-trained models is wrong. Specifically, at time m, the target task receives a training instance $\mathbf{x}_{t(m)}$, the pre-trained models make a prediction and suffer a loss after $y_{t(m)}$ is revealed. Following the error-driven update rule in which the weights are updated only when the pre-trained models make a mistake, we consider the following optimization problem at each time:

$$\boldsymbol{\eta}^{(m+1)} = \arg\min_{\boldsymbol{\eta}\in\Theta} \sum_{j\in[k]} \eta_j l_j^{(t(m))} + \lambda \mathcal{D}_{KL}(\boldsymbol{\eta}||\boldsymbol{\eta}^{(m)}) \tag{11}$$

where $\mathcal{D}_{KL}(\boldsymbol{\eta}||\boldsymbol{\eta}^{(m)})$ denotes the Kullback–Leibler (KL) divergence between current and previous soft-attention distributions, which allows $\boldsymbol{\eta}$ to evolve smoothly over time. The solution for $\boldsymbol{\eta}^{(m+1)}$ is given in a closed form:

$$\eta_j^{(m+1)} = \frac{\eta_j^{(m)} \exp(-l_j^{t(m)}/\lambda)}{\sum_{j'=1}^{k} \eta_{j'}^{(m)} \exp(-l_{j'}^{t(m)}/\lambda)}, \quad j \in [k] \tag{12}$$

Proposition 1. *(Weight Concentration). During the weight update procedure in the whole learning process, the weights will concentrate on those pre-trained models who suffer a small cumulative loss on the target task.*

Proof. *Through the update rule, we know that the weight associated with the* j*-th previous model is equal to* $\eta_j = \frac{\exp(-L_j/\lambda)}{\sum_{j'=1}^{k}\exp(-L_{j'}/\lambda)}, j \in [k]$*. The smaller the loss* L_j *of the pre-trained model, the higher the weight* η_j*.*

We summarize the pseudo-code of ACMR in Algorithm 1.

Algorithm 1 The learning algorithm for ACMR

Input: labeled dataset $\mathcal{L}$, unlabeled dataset $\mathcal{U}$, sampling size N, k pre-trained models $\{f_1, f_2, \ldots, f_k\}$ and $\lambda > 0$
Output: the model $f_{\mathcal{L}}$ for the target task

```
Initialize weight vector η = [1/k, ..., 1/k] and f_L.
for m = 1, 2, ··· , N do
    Select an instance x_{t(m)} from U by traditional active learning
    Predict x_{t(m)} by pre-trained models, and calculate the decision function F(x_{t(m)}) via Eq.(6)
    if F(x_{t(m)}) = 0 then
        y_{t(m)} ← ŷ^{(t(m))}
    else
        y_{t(m)} ← query from oracle
        if y_{t(m)} ≠ ŷ^{(t(m))} then
            update the weights via Eq.(12)
        else
            η^{(m+1)} ← η^{(m)}
        end if
    end if
    L = L ∪ (x_{t(m)}, y_{t(m)}), U = U/(x_{t(m)});
    Train the learner f_L with L
    Facilitate active learner f_L via Eq.(1)
end for
return f_L
```

4 Experiment

In this section, we first give the experimental setup and then show the evaluation of our proposal compared to several state-of-the-art algorithms on a number of real-world tasks.

Table 3: Classification accuracy of compared methods with different queries on 6 classification tasks in 20 Newsgroups dataset. The boldfaces denote the best and the second best methods in terms of the accuracy, and Queries mean that the number of samples labeled by experts.

Task	Queries	Safer	QBC	AcTrak-QBC	AcMR-QBC	Random	AcTrak-Ran	AcMR-Ran
Task1	30	$.876 \pm .033$	$.792 \pm .118$	$.817 \pm .107$	$\mathbf{.923 \pm .016}$	$.806 \pm .126$	$.839 \pm .079$	$\mathbf{.926 \pm .014}$
	60	$.880 \pm .029$	$.916 \pm .029$	$.909 \pm .052$	$\mathbf{.932 \pm .013}$	$.911 \pm .030$	$.915 \pm .034$	$\mathbf{.933 \pm .008}$
	90	$.891 \pm .040$	$.930 \pm .022$	$.929 \pm .026$	$\mathbf{.934 \pm .011}$	$.928 \pm .025$	$.934 \pm .019$	$\mathbf{.936 \pm .011}$
Task2	30	$.737 \pm .027$	$.739 \pm .124$	$.711 \pm .143$	$\mathbf{.838 \pm .039}$	$.689 \pm .110$	$.756 \pm .102$	$\mathbf{.839 \pm .049}$
	60	$.738 \pm .036$	$.821 \pm .076$	$.809 \pm .091$	$\mathbf{.870 \pm .044}$	$.833 \pm .059$	$.842 \pm .071$	$\mathbf{.873 \pm .038}$
	90	$.749 \pm .038$	$.856 \pm .065$	$.851 \pm .050$	$\mathbf{.885 \pm .032}$	$.871 \pm .032$	$.861 \pm .056$	$\mathbf{.892 \pm .027}$
Task3	30	$.944 \pm .010$	$.788 \pm .155$	$.803 \pm .128$	$\mathbf{.954 \pm .012}$	$.748 \pm .146$	$.839 \pm .095$	$\mathbf{.952 \pm .012}$
	60	$.945 \pm .011$	$.903 \pm .080$	$.920 \pm .049$	$\mathbf{.956 \pm .011}$	$.868 \pm .115$	$.921 \pm .050$	$\mathbf{.956 \pm .011}$
	90	$.945 \pm .010$	$.937 \pm .040$	$.948 \pm .015$	$\mathbf{.957 \pm .012}$	$.928 \pm .043$	$.951 \pm .020$	$\mathbf{.957 \pm .009}$
Task4	30	$.689 \pm .059$	$.705 \pm .119$	$.664 \pm .133$	$\mathbf{.905 \pm .027}$	$.786 \pm .142$	$.757 \pm .131$	$\mathbf{.910 \pm .027}$
	60	$.680 \pm .049$	$.853 \pm .085$	$.797 \pm .126$	$\mathbf{.928 \pm .012}$	$.875 \pm .063$	$.875 \pm .083$	$\mathbf{.927 \pm .018}$
	90	$.703 \pm .040$	$.907 \pm .044$	$.859 \pm .086$	$\mathbf{.938 \pm .011}$	$.914 \pm .032$	$.922 \pm .023$	$\mathbf{.933 \pm .017}$
Task5	30	$.850 \pm .061$	$.765 \pm .133$	$.753 \pm .142$	$\mathbf{.915 \pm .059}$	$.765 \pm .134$	$.758 \pm .177$	$\mathbf{.941 \pm .015}$
	60	$.869 \pm .053$	$.888 \pm .081$	$.880 \pm .065$	$\mathbf{.941 \pm .029}$	$.864 \pm .085$	$.833 \pm .128$	$\mathbf{.937 \pm .027}$
	90	$.871 \pm .057$	$.925 \pm .030$	$.915 \pm .055$	$\mathbf{.949 \pm .015}$	$.909 \pm .042$	$.886 \pm .087$	$\mathbf{.950 \pm .011}$
Task6	30	$.770 \pm .042$	$.645 \pm .105$	$.646 \pm .093$	$\mathbf{.819 \pm .029}$	$.643 \pm .104$	$.676 \pm .113$	$\mathbf{.821 \pm .040}$
	60	$.763 \pm .042$	$.679 \pm .089$	$.718 \pm .094$	$\mathbf{.838 \pm .032}$	$.743 \pm .103$	$.779 \pm .090$	$\mathbf{.850 \pm .038}$
	90	$.788 \pm .034$	$.747 \pm .106$	$.781 \pm .081$	$\mathbf{.860 \pm .031}$	$.841 \pm .083$	$.823 \pm .075$	$\mathbf{.867 \pm .031}$

4.1 Experimental Setup

In order to better validate our method, the following methods are compared in our experiments. Two sample selection criteria: i) QBC [Seung *et al.*, 1992] selects examples that cause maximum disagreement amongst an ensemble of hypotheses; ii) Random selects examples randomly; one actively transfer learning method AcTraK [Shi *et al.*, 2008] that uses data of other domains to help learn target task; one baseline method Safer [Li *et al.*, 2017] that safely exploits pre-trained models. We employ Naive Bayes [Lewis and Gale, 1994] with default parameters to implement the base classification model.

For AcMR, we choose QBC and Random to implement the base sampling strategy respectively, and we take the pre-trained models as the input to AcMR. For example, in the sentiment analysis task of book, we take the pre-trained models in other domains (DVD, Electronics and Kitchen) as the input to AcMR; in the classification task of rec.autos and sci.crypt, we take the pre-trained models in other domains (e.g. "rec.motorcycles and sci.electronics", "rec.sport.baseball and sci.med", etc) as the input to AcMR. Most simply, we use Logistic Regression [Harrell, 2015] to facilitate the active learner.

Because AcMR need experts to annotate unlabeled samples, we should compare these methods in different number of samples labeled by experts. For each task, we randomly divide the data into two parts: 75% as the unlabeled pool, and the rest 25% as the test set. Experiments are repeated for 30 times, and the average classification accuracy is reported.

4.2 Text Classification Task

The text classification task is collected from 20 Newsgroups[1]. 20 Newsgroups has a two-level hierarchy, for example, the four sub-categories (sci.crypt, sci.electronics, sci.med, sci.space) belong to the top-category sci. Because the difference among top-categories is relatively large and the sub-categories under each top-category are similar, we can design the top-categories classification problem. For example, we generate 6 text classification tasks (comp vs. talk, comp vs. sci, comp vs. rec, rec vs. talk, rec vs. sci, sci vs. talk).

Results are shown in Table 3, which can be seen that AcMR can achieve better performance than other methods. As the number of queries increases, the accuracy of Safer method generally does not change much on different task, and it means that the performance of pre-trained models is limited. It can be seen that AcMR obtains quite promising performance as the number of queries increases, for various sample selection criteria. Actively transfer learning methods also obtain good performance but is not that competitive as AcMR. This suggests that the use of pre-trained models is beneficial to improve performance rapidly.

4.3 Sentiment Analysis Task

The second task is a sentiment analysis problem[1], and the goal is to label the documents with respect to what types of problems they describe. We evaluate our algorithm on product reviews from Amazon on a dataset containing reviews from 4 domains: Book, DVD, Electronics and Kitchen. We consider each domain as a binary classification task: reviews with rating > 3 are labeled positive(+), those with rating < 3 are labeled negative(-). For sentiment analysis dataset we have 4 classification tasks.

Results are shown in Table 4, which can be seen that AcMR can significantly outperform its based active learning methods in most cases. Moreover, AcMR also consistently performs better as the number of labeled examples increases. More importantly, we find that even though the performance improvement of pre-trained models is very limited, our proposal can also use them to improve performance rapidly. It

[1]https://www.cse.ust.hk/TL/

Table 4: Classification accuracy of compared methods with different queries on 4 classification tasks in Sentiment dataset. The boldfaces denote the best and the second best methods in terms of the accuracy, and Queries mean that the number of samples labeled by experts.

Task	Queries	Safer	QBC	AcTrak-QBC	AcMR-QBC	Random	AcTrak-Ran	AcMR-Ran
Task1	30	$\mathbf{.612 \pm .022}$	$.529 \pm .033$	$.578 \pm .033$	$\mathbf{.607 \pm .028}$	$.577 \pm .036$	$.582 \pm .039$	$.598 \pm .032$
	60	$.614 \pm .018$	$.579 \pm .048$	$.609 \pm .035$	$\mathbf{.631 \pm .023}$	$.607 \pm .025$	$.615 \pm .037$	$\mathbf{.631 \pm .026}$
	90	$.614 \pm .018$	$.615 \pm .047$	$.627 \pm .038$	$\mathbf{.645 \pm .023}$	$.625 \pm .024$	$.637 \pm .034$	$\mathbf{.649 \pm .020}$
Task2	30	$\mathbf{.648 \pm .020}$	$.562 \pm .048$	$.582 \pm .041$	$.610 \pm .033$	$.594 \pm .031$	$.594 \pm .036$	$\mathbf{.611 \pm .046}$
	60	$\mathbf{.648 \pm .018}$	$.591 \pm .049$	$.626 \pm .038$	$.631 \pm .030$	$.623 \pm .033$	$.629 \pm .034$	$\mathbf{.643 \pm .032}$
	90	$.648 \pm .020$	$.621 \pm .047$	$.649 \pm .033$	$\mathbf{.660 \pm .031}$	$.650 \pm .026$	$.657 \pm .028$	$\mathbf{.668 \pm .029}$
Task3	30	$\mathbf{.651 \pm .044}$	$.578 \pm .057$	$.629 \pm .033$	$\mathbf{.645 \pm .036}$	$.612 \pm .036$	$.629 \pm .033$	$.642 \pm .037$
	60	$.654 \pm .041$	$.640 \pm .046$	$.665 \pm .024$	$\mathbf{.676 \pm .028}$	$.643 \pm .032$	$.664 \pm .032$	$\mathbf{.676 \pm .026}$
	90	$.665 \pm .033$	$.672 \pm .035$	$.685 \pm .023$	$\mathbf{.694 \pm .025}$	$.671 \pm .027$	$.679 \pm .029$	$\mathbf{.690 \pm .025}$
Task4	30	$.621 \pm .049$	$.628 \pm .032$	$.634 \pm .033$	$\mathbf{.644 \pm .026}$	$.624 \pm .036$	$.628 \pm .045$	$\mathbf{.641 \pm .035}$
	60	$.631 \pm .051$	$.657 \pm .035$	$.666 \pm .024$	$\mathbf{.680 \pm .018}$	$.652 \pm .032$	$.663 \pm .035$	$\mathbf{.684 \pm .029}$
	90	$.643 \pm .054$	$.684 \pm .026$	$.687 \pm .026$	$\mathbf{.702 \pm .022}$	$.678 \pm .026$	$.681 \pm .033$	$\mathbf{.705 \pm .024}$

Table 5: Classification accuracy of compared methods with different queries on 6 classification tasks in Spam dataset. The boldfaces denote the best and the second best methods in terms of the accuracy, and Queries mean that the number of samples labeled by experts.

Task	Queries	Safer	QBC	AcTrak-QBC	AcMR-QBC	Random	AcTrak-Ran	AcMR-Ran
Task1	30	$.925 \pm .028$	$.757 \pm .152$	$.734 \pm .144$	$\mathbf{.956 \pm .016}$	$.673 \pm .148$	$.681 \pm .149$	$\mathbf{.951 \pm .017}$
	60	$.935 \pm .028$	$.778 \pm .142$	$.750 \pm .116$	$\mathbf{.954 \pm .021}$	$.731 \pm .139$	$.755 \pm .113$	$\mathbf{.957 \pm .018}$
	90	$.939 \pm .021$	$.788 \pm .117$	$.761 \pm .092$	$\mathbf{.954 \pm .024}$	$.805 \pm .143$	$.806 \pm .088$	$\mathbf{.957 \pm .023}$
Task2	30	$.906 \pm .037$	$.750 \pm .163$	$.770 \pm .161$	$\mathbf{.952 \pm .032}$	$.707 \pm .170$	$.751 \pm .159$	$\mathbf{.949 \pm .024}$
	60	$.908 \pm .037$	$.824 \pm .118$	$.791 \pm .151$	$\mathbf{.965 \pm .017}$	$.787 \pm .136$	$.849 \pm .110$	$\mathbf{.962 \pm .021}$
	90	$.901 \pm .038$	$.872 \pm .097$	$.838 \pm .144$	$\mathbf{.968 \pm .014}$	$.843 \pm .099$	$.910 \pm .078$	$\mathbf{.960 \pm .019}$
Task3	30	$.897 \pm .051$	$.846 \pm .116$	$.875 \pm .098$	$\mathbf{.970 \pm .024}$	$.860 \pm .064$	$.879 \pm .047$	$\mathbf{.965 \pm .031}$
	60	$.914 \pm .052$	$.895 \pm .052$	$.916 \pm .037$	$\mathbf{.984 \pm .014}$	$.895 \pm .055$	$.895 \pm .044$	$\mathbf{.981 \pm .015}$
	90	$.913 \pm .042$	$.928 \pm .039$	$.908 \pm .042$	$\mathbf{.986 \pm .013}$	$.902 \pm .046$	$.888 \pm .039$	$\mathbf{.983 \pm .014}$
Task4	30	$\mathbf{.962 \pm .023}$	$.928 \pm .072$	$.924 \pm .081$	$.959 \pm .023$	$.944 \pm .030$	$.942 \pm .022$	$\mathbf{.961 \pm .021}$
	60	$.964 \pm .023$	$.964 \pm .023$	$.968 \pm .019$	$\mathbf{.970 \pm .019}$	$.959 \pm .020$	$.967 \pm .015$	$\mathbf{.967 \pm .016}$
	90	$.964 \pm .023$	$.964 \pm .023$	$.968 \pm .019$	$\mathbf{.970 \pm .019}$	$.963 \pm .021$	$.969 \pm .012$	$\mathbf{.973 \pm .013}$
Task5	30	$.653 \pm .137$	$.794 \pm .076$	$.789 \pm .097$	$\mathbf{.896 \pm .045}$	$.762 \pm .093$	$.667 \pm .066$	$\mathbf{.882 \pm .055}$
	60	$.664 \pm .132$	$.802 \pm .070$	$.811 \pm .076$	$\mathbf{.911 \pm .039}$	$.781 \pm .082$	$.702 \pm .081$	$\mathbf{.902 \pm .044}$
	90	$.703 \pm .118$	$.794 \pm .058$	$.822 \pm .068$	$\mathbf{.924 \pm .023}$	$.796 \pm .069$	$.719 \pm .056$	$\mathbf{.906 \pm .031}$
Task6	30	$\mathbf{.782 \pm .061}$	$.669 \pm .097$	$.652 \pm .091$	$\mathbf{.778 \pm .093}$	$.639 \pm .112$	$.647 \pm .107$	$.763 \pm .095$
	60	$.793 \pm .059$	$.768 \pm .102$	$.711 \pm .089$	$\mathbf{.828 \pm .058}$	$.700 \pm .106$	$.682 \pm .103$	$\mathbf{.807 \pm .048}$
	90	$.794 \pm .062$	$.793 \pm .090$	$.793 \pm .076$	$\mathbf{.854 \pm .048}$	$.758 \pm .099$	$.710 \pm .087$	$\mathbf{.820 \pm .044}$

also clearly shows the advantage of our proposed approach which is able to exploit useful pre-trained models.

4.4 Spam Detection Task

The last task is a spam detection problem, and we use the dataset obtained from ECML PAKDD Discovery challenge[2] to verify whether our method can help improve the performance. We use the task B challenge dataset which consists of labeled training data from the inboxes of 15 users. Each task is a binary classification problem: spam or non-spam. For spam detection task, we test the top 6 classification tasks.

Results from Table 5 show that AcMR achieves highly competitive performance with compared methods. Once again, our proposal works better than state-of-the-art active learning and Safer model prediction algorithms. This shows that in the frame of active learning, taking pre-trained models into account can get a rapid performance improvement.

[2]http://ecmlpkdd2006.org/challenge.html

5 Conclusion

Reusable model design becomes a desire for the rapid expansion of machine learning applications. However, previous model reuse studies assume that the target task receives labeled data *passively*. This leads to a slow performance improvement to the target task. In this paper, we study a kind of new model reuse problem, where the goal is that the model performance for the target task can be quickly improved. We propose the AcMR method which constructs queries through pre-trained models when labeled examples are insufficient for the target task, and leverages pre-trained models to filter out not very necessary queries so that considerable queries could be saved compared with direct active learning. Extending our work into deep learning problem and applying it to more applications are interesting for future study.

Acknowledgments

This research was supported by the National Key R&D Program of China (2018YFB1004300), the National Natural Science Foundation of China (61772262) and the Fundamental Research Funds for the Central Universities (020214380053).

References

[Albrecht, 2016] Jan Philipp Albrecht. How the gdpr will change the world. *European Data Protection Law Review*, 2:287, 2016.

[Chakraborty *et al.*, 2015] Shayok Chakraborty, Vineeth Balasubramanian, Qian Sun, Sethuraman Panchanathan, and Jie-Ping Ye. Active batch selection via convex relaxations with guaranteed solution bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):1945–1958, 2015.

[Dietterich, 2000] Thomas G Dietterich. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, pages 1–15, Günzburg, Germany, 2000.

[Harrell, 2015] Frank E Harrell. Ordinal logistic regression. In *Regression modeling strategies*, pages 311–325. 2015.

[He *et al.*, 2018] Kai-Ming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. *arXiv preprint arXiv:1811.08883*, 2018.

[Huang *et al.*, 2010] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. In *Advances in Neural Information Processing Systems*, pages 892–900, Vancouver, Canada, 2010.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Lewis and Gale, 1994] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

[Li and Liang, 2019] Yu-Feng Li and De-Ming Liang. Safe semi-supervised learning: a brief introduction. *Frontiers of Computer Science*, pages 1–8, 2019.

[Li *et al.*, 2013] Nan Li, Ivor W Tsang, and Zhi-Hua Zhou. Efficient optimization of performance measures by classifier adaptation. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 35(6):1370–1382, 2013.

[Li *et al.*, 2017] Yu-Feng Li, Han-Wen Zha, and Zhi-Hua Zhou. Learning safe prediction for semi-supervised regression. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, volume 2017, pages 2217–2223, San Francisco, CA, 2017.

[Li *et al.*, 2019] Yu-Feng Li, Hai Wang, Tong Wei, and Wei-Wei Tu. Towards automated semi-supervised learning. In *Proceedings of the 33rd AAAI conference on Artificial Intelligence*, Honolulu, HI, 2019.

[Long *et al.*, 2017] Ming-Sheng Long, Han Zhu, Jian-Min Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2208–2217, Sydney, Australia, 2017.

[Murugesan *et al.*, 2016] Keerthiram Murugesan, Han-Xiao Liu, Jaime Carbonell, and Yi-Ming Yang. Adaptive smoothed online multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4296–4304, Barcelona, Spain, 2016.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[Pan *et al.*, 2011] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

[Parkhi *et al.*, 2015] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *Proceedings of the 26th British Machine Vision Conference*, Swansea, UK, 2015.

[Settles, 2012] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[Seung *et al.*, 1992] H Sebastian Seung, Opper Manfred, and Sompolinsky Haim. Query by committee. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 287–294, Pittsburgh, PA, 1992.

[Shi *et al.*, 2008] Xiao-Xiao Shi, Wei Fan, and Jiang-Tao Ren. Actively transfer domain knowledge. In *Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 342–357, Antwerp, Belgium, 2008.

[Wang and Ye, 2015] Zheng Wang and Jie-Ping Ye. Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data*, 9(3):17, 2015.

[Yang *et al.*, 2017a] Yang Yang, De-Chuan Zhan, Ying Fan, Yuan Jiang, and Zhi-Hua Zhou. Deep learning for fixed model reuse. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 2831–2837, San Francisco, CA, 2017.

[Yang *et al.*, 2017b] Yang Yang, De-Chuan Zhan, Xiang-Yu Guo, and Yuan Jiang. Modal consistency based pre-trained multi-model reuse. 2017.

[Ye *et al.*, 2018] Han-Jia Ye, De-Chuan Zhan, Yuan Jiang, and Zhi-Hua Zhou. Rectify heterogeneous models with semantic mapping. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1904–1913, Stockholm, Sweden, 2018.

[Zhou, 2012] Zhi-Hua Zhou. *Ensemble Methods: foundations and algorithms*. 2012.

[Zhou, 2016] Zhi-Hua Zhou. Learnware: on the future of machine learning. *Frontiers of Computer Science*, 10(4):589–590, 2016.