CoLAL: Co-learning Active Learning for Text Classification

Linh Le[1], Genghong Zhao[2], Xia Zhang[3], Guido Zuccon[1], Gianluca Demartini[1]

[1]The University of Queensland
[2]Neusoft Research of Intelligent Healthcare Technology, Co. Ltd.
[3]Neusoft Corporation
linh.le,g.zuccon,g.demartini@uq.edu.au, zhaogenghong@neusoft.com, zhangx@neusoft.com

Abstract

In the machine learning field, the challenge of effectively learning with limited data has become increasingly crucial. Active Learning (AL) algorithms play a significant role in this by enhancing model performance. We introduce a novel AL algorithm, termed ***Co-learning*** (CoLAL), designed to select the most diverse and representative samples within a training dataset. This approach utilizes noisy labels and predictions made by the primary model on unlabeled data. By leveraging a probabilistic graphical model, we combine two multi-class classifiers into a binary one. This classifier determines if both the main and the peer models agree on a prediction. If they do, the unlabeled sample is assumed to be easy to classify and is thus not beneficial to increase the target model's performance. We prioritize data that represents the unlabeled set without overlapping decision boundaries. The discrepancies between these boundaries can be estimated by the probability that two models result in the same prediction. Through theoretical analysis and experimental validation, we reveal that the integration of noisy labels into the peer model effectively identifies target model's potential inaccuracies. We evaluated the CoLAL method across seven benchmark datasets: four text datasets (AGNews, DBPedia, PubMed, SST-2) and text-based state-of-the-art (SOTA) baselines, and three image datasets (CIFAR100, MNIST, OpenML-155) and computer vision SOTA baselines. The results show that our CoLAL method significantly outperforms existing SOTA in text-based AL, and is competitive with SOTA image-based AL techniques.

Introduction

Active Learning (AL) is an approach to reduce the amount of labels typically required to train models, thereby improving training efficiency (Settles 2012; Ostapuk, Yang, and Cudre-Mauroux 2019). The core challenge is to identify the most beneficial instances to label at each round of the AL process. Many AL strategies have been proposed to estimate the benefit of each instance on the learning process. The most common AL algorithm family is that of uncertainty-based algorithms, which estimate the "informativeness" of an instance (Settles 2012). These algorithms select instances that the target model is most uncertain about (Siddhant and Lipton 2018; Yoo and Kweon 2019; Linh et al. 2021; Ostapuk, Yang, and Cudre-Mauroux 2019). Another AL family is distribution-based AL. These algorithms estimate the "representativeness" of an instance (Sener and Savarese 2018; Gissin and Shalev-Shwartz 2019; Zhang and Plank 2021; Cui et al. 2022). Despite much progress made in this area, current AL methods still struggle with instances close to the decision boundaries by using the available labeled data. Training on limited labeled data often results in incomplete decision boundaries, causing the model to make numerous erroneous predictions on unlabeled data (Malach and Shalev-Shwartz 2017). This implies that predictions for unlabeled instances made by the target model are potentially incorrect even when the target model predicts them with high confidence scores. This issue happens as the learning patterns representative of unlabeled data emerge with an insufficient probability or do not appear at all in the labeled data distribution.

Thus, in this work we propose a method that can quantify the decision boundaries for unlabeled training data, which are yet unknown to the target model. These unknown decision boundaries are created by defining a peer model, which is a supervised model trained with *noisy labels* (i.e., predictions from the target model for yet unlabeled data). The most beneficial regions identified with these decision boundaries are calculated through a probabilistic graphical model. From this model, two multi-class classifiers (classes from the training data) are fused into a binary classifier indicating whether two models have the same prediction or not. We assume that, if the probability of having the same prediction is high, the unlabeled instance is easy to classify and would not further improve the performance of the target model if labelled. It also means that, unlabeled instances representative of unlabeled training data which do not overlap with the decision boundaries of labeled training data are preferred. We assume that the disparity between decision boundaries can be interpreted through the probability that the two models share the same predictions or the same decision boundaries.

Our contributions are summarized as follows: 1) We analyze the problem, observe it through a probability interpretation, and encapsulate it in the form of the loss function to intuitively explain the region of interests of our AL; 2) We present extensive experimental results aimed at assessing the effectiveness of the proposed method. The experimental results show that our approach outperforms SOTA AL baselines over four text and three image datasets.

Related Work

Uncertainty-based AL

The most common AL algorithm family estimates the classification uncertainty of an instance. This strategy takes the output from the target model for an instance as the input to an estimation function for the "informativeness" of the instance. The outputs of the model may be, for example, the entropy (Dagan and Engelson 1995), the confidence of the prediction (Culotta and McCallum 2005), the margin between the confidence of the two highest predicted classes (Settles 2012), the information benefit from the Bayesian model's parameters (Gal, Islam1, and Ghahramani 2017), the ensemble of multiple variances of uncertainty AL methods for image input (Beluch et al. 2018). Recent uncertainty-based algorithms are based on the loss of the target model (Yoo and Kweon 2019; Linh et al. 2021). However, these approaches often select outliers due to their high uncertainty (Parvaneh et al. 2022).

Distribution-based AL

This type of AL methods has been shown to deal with the issue of selecting outliers (Abe, Zadrozny, and Langford 2006; Liu et al. 2019a). These methods estimate the "representativeness/diversity" of an instance looking at the distribution of data instances and their feature representation. Clustering-based methods are commonly used in this AL family (Nguyen and Smeulders 2004; Zhu et al. 2008; Nguyen and Patrick 2014), e.g., the work of Sener et al. (Sener and Savarese 2018), which is widely used for image classification. Recently, Gissin et al. (Gissin and Shalev-Shwartz 2019) and Zhang et al. (Zhang and Plank 2021) proposed self-supervised algorithms to leverage data instances' features using a self-supervised mechanism without the need to label data (see Section where we describe our baselines). However, these methods can only interpolate unlabeled training data through the labeled training data (Zhang and Plank 2021), or when features of an unlabeled instance appear in the unlabeled population (Gissin and Shalev-Shwartz 2019) without indicating whether an unlabeled instance could significantly improve the performance of the target model, if used for training.

Learning Hard-To-Classify

Our method is mainly inspired by methods that select hard-to-classify samples (Gissin and Shalev-Shwartz 2019; Zhang and Plank 2021). The DAL method proposed by Gissin and Shalev-Shwartz (2019) targets data samples that make the labeled instances and unlabeled instances indistinguishable. DAL thus builds another model besides the target model to distinguish which unlabeled instances do not belong to the labeled data distribution. The CAL method proposed by Zhang and Plank (2021) targets data samples with low classification accuracy to identify difficult-to-classify unlabeled instances. CAL thus builds another model as a binary classifier to distinguish between unlabeled instances that are hard or not based on a fixed threshold on correctness. The disadvantage of these methods is being unaware that the unlabeled data distribution may vary across the categories in the dataset. This is critical to learn which instances are poorly represented in the training data. From that point, the method we propose focuses on supporting perspectives that are unknown to the target model by means of another model used as a peer model. This peer model takes as input data with noisy labels. Instead of merely using confidence scores predicted by the peer model to select beneficial unlabeled instances, we employ the disagreement between both the target model and the peer model. This disagreement aims to identify significant differences between the two models for an unlabeled instance.

Learning from Noisy Labels

In the noisy label setup, easy data instances are connected with *clean* samples, which are labeled by annotators and increase the performance of the target model. *Noisy* data may be incorrectly labeled and reduce the performance of the target model. Thus, "learning from noisy labels" research looks at a set of solutions to tackle the presence of noisy labels so that the performance of the target model can still be improved (Blum and Mitchell 1998; Han et al. 2018; Rodrigues and Pereira 2018; Malach and Shalev-Shwartz 2017; Jiang et al. 2018; Yao et al. 2020; Berthon et al. 2021). Some related popular approaches in this line of work include (Han et al. 2018; Malach and Shalev-Shwartz 2017; Jiang et al. 2018) which use a peer model to select clean labels. These methods implicitly show that neural networks themselves can detect the easy and hard regions of data even when the labels are noisy (Han et al. 2018). Different from co-teaching (Han et al. 2018) and co-training (Blum and Mitchell 1998), our co-learning method does not use the weights or the output from the target model to remove noisy labels from the peer model. Rather, it uses the output of the peer model to guide the target model which predictions could be incorrect and need to be annotated. Selected unlabeled instances are then annotated and used in the next training iteration as they might be incorrectly predicted and are thus beneficial for the target model to improve its performance.

Our Method: Co-learning AL

Research Problem

Our AL setup consists of unlabeled data U with size N_U, $U = \{x_j\}_{j=1}^{N_U}$, the currently labeled training data L with size N_L, $L = \{x_i\}_{i=1}^{N_L}$, $c \in \{1, 2, ..., C\}$, where C represents the number of classes within the training dataset, a target model f_ψ as a neural network parameterized by ψ, and a peer model f_ξ as a neural network parameterized by ξ. The target model is trained only with labeled data, which are clean labels while the peer model is trained with unlabeled data for which noisy labels are predicted by the target model. Malach and Shalev-Shwartz (2017) have shown that the results of machine learning models are not reliable when they are trained on a small amount of training data. We thus create a peer model to reveal how reliable the target model predictions for $x_j \in U$ (x_j is the vector representation of the j^{th} unlabeled instance) are. This reliability is expressed through the probability that the target model has the same predictions as the peer model. There are two questions addressed by our AL algorithm:

Algorithm 1: CoLAL algorithm

Input: labeled data L, unlabeled data U, seed data I, pool of samples having different predictions from target model and peer model, acquisition size $B = 100$, target model f_Ψ, AL model f_ξ with input U and noisy labels $(f_\Psi(X_u))$
Initialize data I, T
$L \leftarrow I$
while $U \neq \varnothing$ **do**
    Train f_Ψ with labeled data
    Infer noisy-labels for unlabeled training data U
    Train f_ξ with unlabeled data and their noisy labels.
    Clear or Initialize T
    for $i < N_u$ **do**
        Calculate the disagreement in Equation 5 ▷ Compute the disagreement between 2 models f_Ψ, f_ξ
        if $(f_\Psi(x_i)) \neq (f_\xi(x_i))$ **then**
            $T \leftarrow x_i$
        end if
    end for
    if $T \neq \varnothing$ **then**
        Select top B samples with the highest disagreement from T and remove them from U
    else
        Select top B samples with the highest disagreement from U and remove them from U
    end if
end while

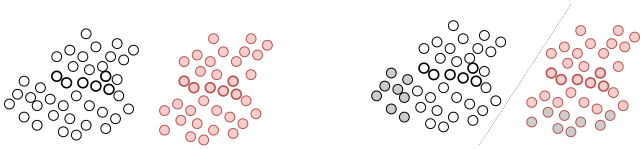
- Can learning from noisy instances detect potentially incorrect predictions made by the target model f_ψ?
- Can incorrect predictions by f_ψ be identified by the disagreement with f_ξ?

The first question is mostly answered based on how deep neural networks memorize. This is beneficial for filtering noisy labels while effectively learning the patterns of unlabeled data:

Random predictions can be made when the categorical distribution is not significantly different among classes. In these cases, if few clean samples are included in the training process of f_ξ, f_ξ can learn from clean samples instead of noisy samples. This is shown by Bo Han et al. (Han et al. 2018) who observed that deep networks are capable of identifying and learning clean patterns during the early stages of training, even with the presence of noisy labels (e.g., see Figure 1(b)). Instances that are most representative of a specific region of data are learned first in the early epochs. In our work, f_ξ is trained with only 2 epochs to enforce this behaviour.

From an intuitive perspective, various classifiers can create distinct decision boundaries, leading to different learning capabilities (see Figures 3(a), 3(b), 3(c)). Thus our work proposes a co-learning mechanism using two models: one target model and one peer model. While the peer model f_ξ is trained with a high number of unlabeled data for which we generate noisy labels and a small number of epochs (2 epochs), the target model f_ψ is trained on a small amount of good quality labeled data and a higher number of training epochs (15 epochs).

The second question is addressed in the following sections.



((a)) Instances learned first in early training or with few epochs (e.g., 2 epochs).

((b)) Gray points are noisy instances and are filtered in the early stage of training.

Figure 1: Example of memorization mechanism. The training set has two classes: white with a black border and red with a red border. Samples highlighted with bolder borders are learned first.

Theoretical Foundation of Co-learning

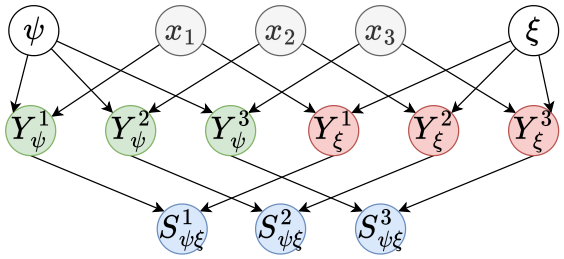
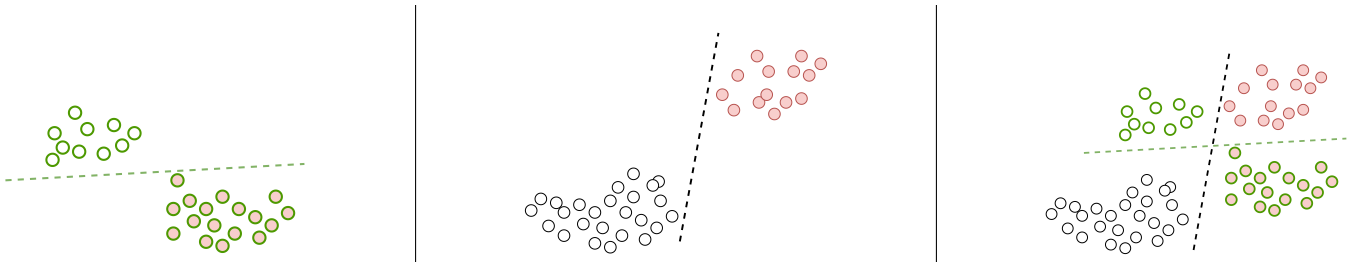


Figure 2: Graphical representation of the combined classification task; x_j represents nodes of U, Y_ψ^j and Y_ξ^j represents class labels from the target model and the peer model, S_{ij} is the similarity between the prediction from the target model and peer model, ψ and ξ represent the neural network parameters of the target model and peer model respectively.

In this work, we propose a novel method that combines the output of two models: the target model trained on the labeled training data L and the peer model trained on the unlabeled training data U. We assume that the combined model can identify the incorrect predictions made by the target model. Let $S_{\psi\xi}^j$ indicate the same prediction from the two models, $\hat{S}_{\psi\xi}^j$ denote the probability that f_ψ has the same prediction as f_ξ. Y_ψ^j, Y_ξ^j are the labels for an instance at index j from two models f_ψ, f_ξ. $P(S_{\psi\xi}^j | Y_\psi^j, Y_\xi^j) = 1$ when $Y_\psi^j = Y_\xi^j$. The output for the unlabeled sample j by the target model parameterized by ψ is $f(x_j; \psi) = P(Y_\psi^j | x_j; \psi)$. The output for the unlabeled sample j by the peer model parameterized by ξ is $f(x_j; \xi) = P(Y_\xi^j | x_j; \xi)$. Now, we set up our research problem through a probabilistic graphical model G (see Figure 2). This describes the graph to observe the similarity S in predictions between two models as follows:

$$P(G) = P(U, Y, S; \psi; \xi) = P(S|Y)P(Y|U; \psi; \xi)P(U) \quad (1)$$



((a)) The model trained on labeled data with a green decision boundary.

((b)) The peer model trained on unlabeled data with a black decision boundary.

((c)) The model integrates decision boundaries using both labeled and unlabeled data models.

Figure 3: Decision boundaries across training scenarios. In this example, two classes are differentiated by fill colors, with green borders indicating labeled samples.

In Equation 1, S represent the similarities between U and L, Y represents the predictions of U from f_ψ, f_ξ. $P(S|Y) = \prod_j P(S^j_{\psi\xi}|Y^j_\psi, Y^j_\xi)$. To calculate P(G), we need to compute the marginalization through Y as $\sum_Y P(S|Y)P(Y|U;\psi;\xi)$

$$\begin{aligned} P(X,Y,S;\psi;\xi) &\approx \sum_Y P(S|Y)P(Y|U;\psi;\xi) \\ &= \prod_j \Big(\sum_j 1[S^j_{\psi\xi}=1]P(Y^j_\psi|x_j;\psi)P(Y^j_\xi|x_j;\xi) \\ &+ \sum_j 1[S^j_{\psi\xi}=0]P(Y^j_\psi|x_j;\psi)P(Y^j_\xi|x_j;\xi)\Big) \end{aligned} \tag{2}$$

Then we take a negative logarithm on Equation 2 so that we obtain Equation 4 by converting the inner product of two categorical distributions from the two multi-class classifiers into the binary classifier as follows:

$$g(x_j, f(.,\psi), f(.,\xi)) = f(x_j;\psi)^T f(x_j;\xi) = \hat{S}^j_{\psi\xi} \tag{3}$$

$$\begin{aligned} L &= -\sum_j \log\Big(\sum_j 1[S^j_{\psi\xi}=1]P(Y^j_\psi|x_j;\psi)P(Y^j_\xi|x_j;\xi) \\ &+ \sum_j 1[S^j_{\psi\xi}=0]P(Y^j_\psi|x_j;\psi)P(Y^j_\xi|x_j;\xi)\Big) \\ &= -\sum_j S^j_{\psi\xi}\log\big(f(x_j;\psi)^T f(x_j;\xi)\big) \\ &+ (1-S^j_{\psi\xi})\log\big(1-f(x_j;\psi)^T f(x_j;\xi)\big) \\ &= -\sum_j S^j_{\psi\xi}\log\Big(\hat{S}^j_{\psi\xi}\Big) + (1-S^j_{\psi\xi})\log\Big(1-\hat{S}^j_{\psi\xi}\Big) \end{aligned} \tag{4}$$

In this section we show that we can convert the comparison between two multi-class classifier to binary classification by defining the function g indicating the probability of having the same prediction from the two models. In this binary classification, a class of "1" denotes the same prediction between f_ψ and f_ξ, otherwise is "0". To maximize $P(G)$, we minimize Equation 4. The likelihood nature of the loss function in Equation 4 helps us to explore the overlap of decision boundaries between the two models. Specifically, to minimize the loss, the two models must have the same output node when $S^j_{\psi\xi} = 1$. In the case $S^j_{\psi\xi} = 0$, the less overlap between two distributions, the lower the loss. These characteristics illustrate why when we minimize this loss we can learn the similarity between the decision boundaries of two multi-class classification models.

Selection Criterion. In the previous section, we show that the inner product between the categorical distributions from two models can be considered as the probability of having the same prediction from the two models. In our work, we consider the unlabeled samples where their predictions are not the same as the target model, represented by $S^j_{\psi\xi} = 0$. To maximize $P(G)$, we aim to select regions with the least overlap with labeled data or the target model's categorical distributions. We also consider this minimal overlap as the disagreement between the two models. Thus, we formulate our selection criterion (SC) as following:

$$\begin{aligned} &\underset{j=\{1,N_u\}}{\text{argmax}}\ 1 - g(x_j, f(.,\psi), f(.,\xi)) \\ &= \underset{j=\{1,N_u\}}{\text{argmax}}\ 1 - f_\psi(x_j;\psi)^T f_\xi(x_j;\xi) \\ &= \underset{j=\{1,N_u\}}{\text{argmax}}\ 1 - \sum_{c=1}^{C} P(Y^j_\psi = c|x_j;\psi) * P(Y^j_\xi = c|x_j;\xi) \end{aligned} \tag{5}$$

where N_u is the number of unlabeled training data instances. We clarify the mechanism of how this equation is computed through an example. In the first case, we have the categorical distribution of x_i: $f_\psi(x_i) = [0.1, 0.8, 0.1]$ and $f_\xi(x_i) = [0.2, 0.6, 0.2]$, the disagreement between $f_\psi(x_i)$ and $f_\xi(x_i)$=$1 - (0.1 * 0.2 + 0.8 * 0.6 + 0.1 * 0.2) = 0.48$. In the second case, we have the same categorical distribution from the target model $f_\psi(x_i) = [0.1, 0.8, 0.1]$ and $f_\xi(x_i) = [0.8, 0.1, 0.1]$, the disagreement between $f_\psi(x_j)$ and $f_\xi(x_j)$=$1 - (0.1 * 0.8 + 0.8 * 0.1 + 0.1 * 0.1) = 0.83$, which is higher than the first case. It is obvious that the more dissimilar the predictions are, the more disagreement between $f_\psi(x_i)$ and $f_\xi(x_i)$ there would be. Besides, our overlap estimation allows us to halt the selection of non-beneficial

samples when the models achieve a certain level of agreement. This level indicates that further annotation is unlikely to enhance the performance of the target model. It also means that these instances receive the same prediction from the target model and the peer model and thus that these instances do not contain any new features to learn from. Thus, we only consider unlabeled samples if they have different predictions from the two models (see Line 11 in Algorithm 1). To select the most beneficial samples when the two models disagree, we select the top B samples with the highest disagreement (see Lines 16 and 18 in Algorithm 1).

Algorithmic Complexity Discussion

Our AL approach utilizes a single model trained on unlabeled data, which is comparable to other SOTA methods such as, e.g., Gissin et al. (Gissin and Shalev-Shwartz 2019). A key distinction between our method and these alternatives lies in the labeling of unlabeled data: Our model employs labels derived from the target model, while the aforementioned methods utilize binary labels, with 0 representing unlabeled data and 1 representing labeled data. We introduce an additional simple difference multiplication computation, as detailed in Line 10 of Algorithm 1. The time complexity of our method is equivalent to that of other methods, with an additional $O(N_U)$ factor, where N_U denotes the number of training samples. Consequently, our method is both effective and efficient.

Experimental Setting

Datasets

In this study, we consider four benchmark text classification datasets that were used to evaluate SOTA baselines (Yu et al. 2022; Margatina et al. 2021). The first dataset, AGNews (Zhang, Zhao, and Lecun 2015), is focused on news topic classification and comprises 4 classes, with 119,000 training samples, 1,000 development samples, and 7,600 test samples. The second dataset, DBPedia (Zhang, Zhao, and Lecun 2015), is designed for Wikipedia topic classification and encompasses 14 classes, with 280,000 training samples, 1,000 development samples, and 70,000 test samples. The third dataset, Pubmed (Dernoncourt and Lee 2017), is used for medical abstract classification and includes 5 classes, with 180,000 training samples, 1,000 development samples, and 30,100 test samples. Finally, the fourth dataset, SST-2 (Socher et al. 2021), is used for sentiment analysis and contains 2 classes, with 60,600 training samples, 800 development samples, and 1,800 test samples.

We also consider three benchmark datasets to evaluate image classification SOTA baselines, as in (Parvaneh et al. 2022). These are CIFAR100 (Krizhevsky 2009), MNIST (LeCun et al. 1998) and OpenML [1]. CIFAR100 includes 100 classes of 32x32 coloured images, featuring 50,000 images for training and 10,000 images for testing. MNIST includes 10 classes of 28x28 binary images depicting handwritten single digits, with a training set of 50,000 images and a test set of 10,000 images. Additionally, we have selected the OpenML-155 dataset, which consists of 9 classes of metadata samples, totaling 50,000 training samples and 10,000 test samples, as configured in (Parvaneh et al. 2022).

[1] https://www.openml.org/

Baselines

We compare our approach against seven AL methods: two classic baselines: Random (Settles, Craven, and Ray 2008) and Max-Entropy (Ent.) (Dagan and Engelson 1995); and five recent SOTA baselines: Deep-batch Active Learning (Ash et al. 2020), Contrastive Active Learning (Margatina et al. 2021), Actune Active Learning (Yu et al. 2022), Task-Independent Triplet Loss (Seo et al. 2022), and Feature Mixing for Active Learning (Parvaneh et al. 2022).

Random. Random is a baseline method that selects instances for annotation randomly without any particular strategy or approach. It is often used as a reference point for comparing the performance of other more sophisticated active learning methods.

Max-Entropy (Entropy). Given a classification model's output probability distribution $P(Y_\psi^j = c|x_j, \psi)$ over class c, the entropy H representing the model's uncertainty for an instance x_j is: $H = -\sum_{c=1}^{C} P(Y_\psi^j = c|x_j, \psi) \log(P(Y_\psi^j = c|x_j, \psi))$. Unlabeled samples with higher entropy values (i.e., with greater uncertainty) can be prioritized for labeling.

Deep-batch Active Learning (BADGE). The estimation of informativeness is based on the lower bounds of the gradients of the model's loss function with respect to the model's parameters. The lower bounds are calculated using a variational approximation that involves sampling from the posterior distribution of the model's parameters. This method's estimation is based on multiple lower bounds obtained by sampling from the posterior distribution. It calculates the uncertainty using a variational approximation that involves a probability distribution. The most informative data points for labeling are expected to maximize the reduction of the loss function.

Contrastive Active Learning (CAL). This method leverages unlabeled data by using contrastive samples to select informative instances for labeling. The selection process involves computing the Kullback-Leibler (KL) divergence between labeled and unlabeled data, with the goal of identifying instances that are maximally distinct from the labeled data. The CAL algorithm works by first training a contrastive representation learning model on the available unlabeled data, which generates representations that capture the underlying structure of the data. The KL divergence between the labeled and unlabeled data distributions is then computed using these representations, and the unlabeled instances with the highest KL divergence are selected for labeling.

Actune Active Learning (AcTune). This algorithm enhances model performance by selecting annotation-worthy instances via a region-based uncertainty method. This strategy involves segmenting the input space, assessing the uncertainty of each segment, and then giving preference to

highly uncertain regions for annotation to improve prediction accuracy. Additionally, *AcTune* employs a momentum-based memory bank to incorporate representations of prior instances, thereby enriching the training process. The distinctiveness of this method lies in its emphasis on region-aware sampling and the application of weighted k-means clustering. The centroid of every cluster is then computed to identify clusters with high uncertainty levels. Selected data points are the one nearest to the centroid of selected clusters.

Task-Independent Triplet Loss (BATL). This work introduces a task-independent batch acquisition method using triplet loss. This method takes into account both pre-trained linguistic features and task-related features while exploring uncertainty and diversity in the unlabeled dataset. The proposed acquisition function combines sentence representations from a pre-trained language model with task-related features from a classifier's final hidden layer. The triplet loss is based on an anchor, positive, and negative samples where the anchor and positive samples share the same label, while the negative sample has a different label. The selected data points are the ones with the highest triplet loss.

Feature Mixing for Active Learning (AlphaMix). This method identifies novel features in unlabeled data. This involves analyzing inconsistencies in model predictions when the data representations are interpolated. Parvanev et al. created interpolations between labeled and unlabeled data representations and then examined the resulting predicted labels. Their findings indicate that the inconsistencies in these predictions are crucial for identifying features that the model fails to recognize in the unlabeled data. They also proposed an efficient implementation based on a closed-form solution for the optimal interpolation that induces changes in predictions.

Evaluation

We use accuracy to evaluate the effectiveness of the text classification models (Yu et al. 2022; Seo et al. 2022). We compare methods' performance using significance tests (Dror et al. 2018). Specifically, we employ the t-test as done in previous work (Yu et al. 2022).

Training Settings

In our study, we employ RoBERTa-base (Liu et al. 2019b) from the HuggingFace codebase (Wolf et al. 2020) as the backbone for our CoLAL method and for all baselines, except for the Pubmed dataset, where we utilize SciBERT (Beltagy, Lo, and Cohan 2019): a BERT model pre-trained on scientific corpora. To ensure a true low-resource setting and to maintain consistency with previous low-resource AL research, we highlight that we train each model from scratch in every round. This approach helps to avoid overfitting the data collected in earlier rounds, as noted by Hu et al. (Hu et al. 2019). By adhering to these settings, we aim to provide a reliable comparison with the referenced previous work. The configuration for the target model includes training for 15 epochs, using a batch size of 8, a learning rate of 2e-5, and a weight decay of 1e-8. Additionally, we utilize the "Sequence Classification" backbone from HuggingFace for our classification tasks, ensuring compatibility and consistency across experiments. Based on how models memorize, it is essential to have a specific number of epochs for our peer AL model. When incorrect labels exist, models can memorize incorrect labels (Zhang et al. 2017). Our work thus can not run large epoch numbers with F_ξ. "Epoch=1" for F_ξ is empirically selected for BERT-based embeddings through different runs with different epoch settings (i.e., 1, 2, 3, 5). Our experiments are executed with 5 different random seeds on a GPU cluster with 16GB nVidia Tesla V100 GPUs. In our active learning setups, we follow the setup of Yuan et al. (Yuan, Lin, and Boyd-Graber 2020; Yu et al. 2022; Parvaneh et al. 2022) by setting the number of rounds equals to 10, the overall labeling budget for all datasets equals to 1000, and the initial size of the labeled set equals to 100. For CIFAR100, the initial labeled set is 1,000 with a total budget of 10,000. In each round, we sample a batch of 100 samples from the unlabeled set U and query their labels, except for CIFAR100, where 1,000 samples are chosen. Due to the impracticality of large development sets in low-resource settings (Kann, Cho, and Bowman 2019), we limit the size of the development set to 1,000, which is the same as the labeling budget. Additionally, our method does not prescribe a fixed number of initial samples from each class; rather, it starts with a random sample of the available data.

Results and Analysis

Comparison on Text Classification Benchmarks

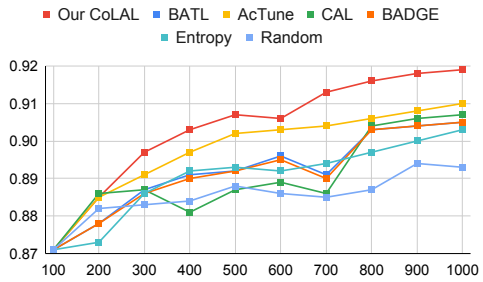
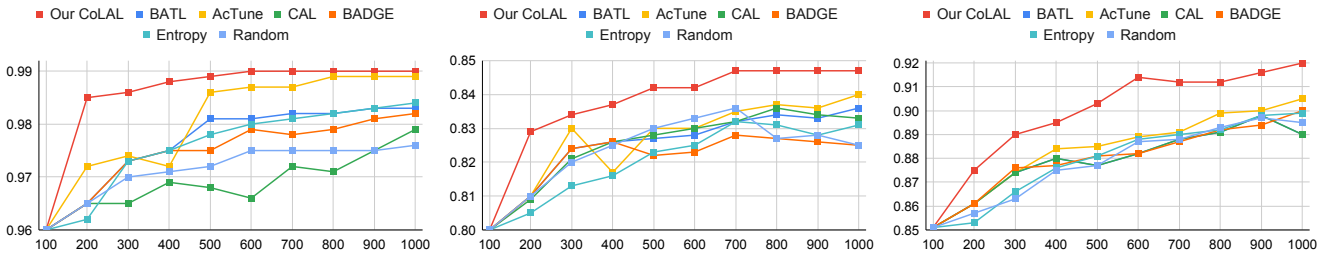


Figure 4: Our method compared to SOTA baselines on AGNews. The horizontal axis represents "Number of used training samples " while the vertical axis represents "Accuracy".

Our CoLAL approach is statistically significantly better than the baselines on AGNews, DBPedia, PubMed, SST-2 (see Figures 4, 5(a), 5(b), 5(c)). Among the baselines, AcTune yields the best performance except on PubMed dataset, where BADGE slightly outperforms other baselines without significant difference.

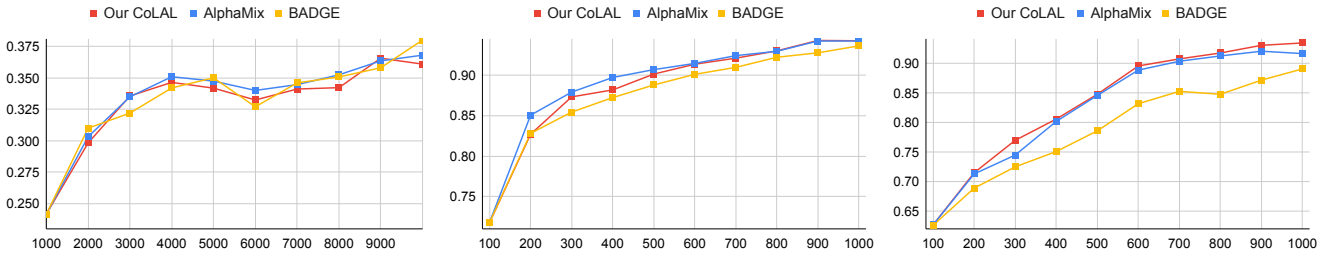
AGNews: Our CoLAL method consistently outperforms other methods across all sample sizes, achieving the highest accuracy at each stage. CoLAL outperforms the best baseline by average gaps ranging from 0.642% to 2.050%, with peak differences between 1.103% and 3.26% at 800 samples.

DBPedia: Our CoLAL exceeds other methods with average accuracy gaps spanning from 0.543% to 1.733%, with its



((a)) Comparison on DBPedia. ((b)) Comparison on PubMed. ((c)) Comparison on SST-2.

Figure 5: Performance comparison on text classification. The horizontal axis represents "Number of used training samples " while the vertical axis represents "Accuracy".



((a)) Comparison on CIFAR100. ((b)) Comparison on MNIST. ((c)) Comparison on OpenML-155.

Figure 6: Performance comparison on image classification. The horizontal axis represents "Number of used training samples " while the vertical axis represents "Accuracy".

highest outperformance between 1.646% and 2.391% at 200 and 400 samples respectively.

PubMed: Our CoLAL method consistently outperforms other methods across all sample sizes, achieving the highest accuracy at each stage. In comparison with other baselines, CoLAL shows average accuracy improvements ranging from 1.026% to 2.046%, with particular distinction in performance against the best baseline BADGE at 1.955%. Interestingly, we observe no significant difference between the baseline methods, while our CoLAL method shows a distinct performance gap. These results serve as evidence for the effectiveness of our CoLAL approach in enhancing the performance of the target model with significant gaps compared to other methods on the Pubmed dataset.

SST-2: Our CoLAL method shows average accuracy gaps over other methods between 1.026% and 2.219%, with maximum improvements observed at 3.628% against BADGE and 2.812% against AcTune at 600 samples.

Comparison on Image Classification Benchmarks

Our CoLAL method exhibits competitive performance compared to the AlphaMix baseline and other baselines for image classification across three datasets: CIFAR100, MNIST, and OpenML-155, as shown in Figures 6(a), 6(b), and 6(c). CoLAL achieves AlphaMix's performance by the 5th iteration on the MNIST dataset and performs comparably to other baselines in the first four iterations on CIFAR100. Furthermore, during iterations 6 to 10 on the OpenML-155 dataset, CoLAL slightly outperforms the baselines.

Conclusions

In this paper, we propose a novel AL algorithm: Co-learning for Active Learning. The aim of this method is to select unlabeled training data points by quantifying the overlap in the categorical distribution with the target model, ensuring the selected data is diverse from the labeled data and representative of the training set. We empirically show that our method is significantly more effective than SOTA AL methods over four benchmark datasets for text classification and competitive with SOTA over three benchmark datasets for image classification. Moreover, our method provides a way for machine learning models to quantify the influence of unlabeled instances through the overlap across different regions of the training data space.

Acknowledgements

This research is supported by the National Key Research and Development Program of China No. 2020AAA0109400 and the Shenyang Science and Technology Plan Fund (No. 21-102-0-09), and by the Swiss National Science Foundation (SNSF) under contract number CRSII5_205975.

References

Abe, N.; Zadrozny, B.; and Langford, J. 2006. Outlier detection by Active Learning. In *SIGKDD*.

Ash, J. T.; Zhang, C.; Krishnamurthy, A.; Langford, J.; and Agarwal, A. 2020. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In *ICLR*.

Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *EMNLP-IJCNLP*.

Beluch, W. H.; Genewein, T.; Nürnberger, A.; and Köhler, J. M. 2018. The Power of Ensembles for Active Learning in Image Classification. In *CVPR*.

Berthon, A.; Han, B.; Niu, G.; Liu, T.; and Sugiyama, M. 2021. Confidence Scores Make Instance-dependent Label-noise Learning Possible. In *ICML*.

Blum, A.; and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *COLT*.

Cui, L.; Tang, X.; Katariya, S.; Rao, N.; Agrawal, P.; Subbian, K.; and Lee, D. 2022. ALLIE: Active Learning on Large-scale Imbalanced Graphs. In *WWW*.

Culotta, A.; and McCallum, A. 2005. Reducing labeling effort for structured prediction tasks. In *AAAI*.

Dagan, I.; and Engelson, S. P. 1995. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings*.

Dernoncourt, F.; and Lee, J. Y. 2017. PubMed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In *IJCNLP*.

Dror, R.; Baumer, G.; Shlomov, S.; and Reichart, R. 2018. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *ACL*.

Gal, Y.; Islam1, R.; and Ghahramani, Z. 2017. Deep Bayesian ActiveLearning with Image Data. In *ICML*.

Gissin, D.; and Shalev-Shwartz, S. 2019. Discriminative Active Learning. In *ICLR*.

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I. W.-H.; and Sugiyama, M. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*.

Hu, P.; Lipton, Z.; Anandkumar, A.; and Ramanan, D. 2019. Active Learning with Partial Feedback. In *ICLR*.

Jiang, L.; Zhou, Z.; Leung, T.; Li, L.; and Fei-Fei, L. 2018. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In *ICML*.

Kann, K.; Cho, K.; and Bowman, S. R. 2019. Towards Realistic Practices in Low-Resource Natural Language Processing: The Development Set. In *EMNLP-IJCNLP*.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

Linh, L.; Nguyen, M.-T.; Zuccon, G.; and Demartini, G. 2021. Loss-based Active Learning for Named Entity Recognition. In *IJCNN*.

Liu, Y.; Li, Z.; Zhou, C.; Jiang, Y.; Sun, J.; meng Wang; and He, X. 2019a. Generative adversarial active learning for unsupervised outlier detection. In *TKDE*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In *arXiv preprint arXiv:1907.11692*.

Malach, E.; and Shalev-Shwartz, S. 2017. Decoupling "when to update" from "how to update". In *NIPS*.

Margatina, K.; Vernikos, G.; Barrault, L.; and Aletras, N. 2021. Active Learning by Acquiring Contrastive Examples. In *EMNLP*.

Nguyen, D. H. M.; and Patrick, J. D. 2014. Supervised machine learning and active learning in classification of radiology reports. *JAMIA*.

Nguyen, H. T.; and Smeulders, A. 2004. Active learning using pre-clustering. In *ICML*.

Ostapuk, N.; Yang, J.; and Cudre-Mauroux, P. 2019. ActiveLink: Deep Active Learning for Link Prediction in Knowledge Graphs. In *WWW*.

Parvaneh, A.; Abbasnejad, E.; Teney, D.; Haffari, R.; van den Hengel, A.; and Shi, J. Q. 2022. Active Learning by Feature Mixing. In *CVPR*.

Rodrigues, F.; and Pereira, F. 2018. Deep learning from crowds. In *AAAI*.

Sener, O.; and Savarese, S. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *ICLR*.

Seo, S.; Kim, D.; Ahn, Y.; and Lee, K. 2022. Active Learning on Pre-trained Language Model with Task-Independent Triplet Loss. In *AAAI*.

Settles, B. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning.

Settles, B.; Craven, M.; and Ray, S. 2008. Multiple-instance active learning. In *NIPS*.

Siddhant, A.; and Lipton, Z. C. 2018. Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study. In *EMNLP*.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A.; and Potts, C. 2021. Active Learning by Acquiring Contrastive Examples. In *ACL*.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; and Others. 2020. Transformers: State-of-the-Art Natural Language Processing. In *EMNLP*.

Yao, Y.; Liu, T.; Han, B.; Gong, M.; Deng, J.; Niu, G.; and Sugiyama, M. 2020. Dual T: Reducing Estimation Error for Transition Matrix in Label-noise Learning. In *NeurIPS*.

Yoo, D.; and Kweon, I. S. 2019. Learning Loss for Active Learning. In *CVPR*.

Yu, Y.; Kong, L.; Zhang, J.; Zhang, R.; and Zhang, C. 2022. AcTune: Uncertainty-Based Active Self-Training for Active Fine-Tuning of Pretrained Language Models. In *NAACL*.

Yuan, M.; Lin, H.-T.; and Boyd-Graber, J. 2020. Cold-start Active Learning through Self-supervised Language Modeling. In *EMNLP*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*.

Zhang, M.; and Plank, B. 2021. Cartography Active Learning. In *Findings of EMNLP*.

Zhang, X.; Zhao, J. J.; and Lecun, Y. 2015. Character-level convolutional networks for text classfication. In *NIPS*.

Zhu, J.; Wang, H.; Yao, T.; and Tsou, B. K. 2008. Active Learning with Sampling by Uncertainty and Density for Word Sense Disambiguation and Text Classification. In *COLING*.