Enhancing Active Learning with Weak Supervision and Transfer Learning by Leveraging Information and Knowledge Sources

Lukas Rauch, Denis Huseljic, and Bernhard Sick

University of Kassel, Wilhelmshöher Allee 73, 34121 Kassel, Germany
{lukas.rauch, dhuseljic, bsick}@uni-kassel.de

Abstract. One of the major limitations of deploying a machine learning model is the availability of labeled training data and the resulting expensive annotation process. Although active learning (AL) methods may reduce the annotation cost by actively selecting the most-useful instances, a costly human annotator usually provides the labels. Therefore, even with AL, we still consider the annotation process to be time-consuming and expensive. Besides human annotators, though, companies often have a vast amount of information and knowledge sources available that can generate low-cost labels (e.g., a black-box model) or improve the learning process (e.g., a pre-trained model). We present a novel approach that enhances AL with weak supervision (WS) and transfer learning (TL) to reduce the annotation cost by leveraging these sources. Specifically, we consider a black-box model like a rule-based system as an error-prone and weakly-supervised annotator that inexpensively provides labels. We estimate its performance with an annotator model to decide whether a human annotation is required. Additionally, we utilize unlabeled internal and external data by transferring knowledge from a pre-trained model to the AL cycle. We sequentially investigate the impact of WS and TL on annotation cost and model performance in an AL cycle through a use case. Our evaluation shows that our approach can reduce annotation cost by 51% while achieving nearly identical model performance compared to a traditional AL approach.

Keywords: Active Learning · Weak Supervision · Transfer Learning · Information and Knowledge Sources.

1 Introduction

In recent years, there has been an increasing interest in machine learning applications across all industries [25]. In particular, (deep) neural networks (NNs) have proven beneficial for unstructured data types such as image or text data. However, one of the major real-world bottlenecks in deploying a NN is the need for large labeled training data sets to reach peak performance [25,30]. To reduce annotation cost for the training process, active learning (AL) [4,31] is a part of human-in-the-loop learning [13] where we actively select the most-useful instances. The goal is to reduce annotation cost while maximizing the performance of a model trained on an actively selected subset from an unlabeled data pool [32,12]. However, since a human annotator (HA) usually provides the labels, the annotation process may still be time-consuming and expensive [11]. Besides HAs, companies usually have a wide range of information and knowledge sources [10] available such as an established black-box model (BBM) like a rule-based system [23] or external data and a pre-trained model from the Internet. These sources can provide labels (information source) or contain beneficial knowledge for training NNs (knowledge source). Nevertheless, they are often ignored or not fully utilized in practice. This raises the question of *how to efficiently leverage and extract* information and knowledge from available sources to further reduce the annotation cost in AL.

To address this question, research fields such as weak supervision (WS) [1,5] and transfer learning (TL)[26] provide suitable methods. Specifically, WS methods generate noisy labels at low cost, e.g., with expert-defined rules or labeling heuristics [2,30] and are typically applied after obtaining a high-quality labeled data set. In TL [26], acquired knowledge of a pre-trained model is transferred to a different but related downstream task. Combining AL-WS and AL-TL has already shown promising results to further reduce the annotation cost in AL [7,33]. However, to the best of our knowledge, there has not yet been a combination of all three fields in which multiple available information and knowledge sources are exploited. Therefore, we investigate the following research questions in this work:

Question 1. How can we enhance AL with WS so that we can leverage an available BBM as an information source to reduce the annotation cost with a competitive model performance compared to a traditional AL approach?

Question 2. How far can the inclusion of TL to leverage unlabeled internal and external data as knowledge sources empower the combination of AL-WS and, thus, further reduce the annotation cost and improve the model performance?

To answer those research questions, we conduct experiments in a real-world use case where we thematically classify banking transactions based on text data. We extend an AL cycle with WS, training a classification and annotator model simultaneously. Specifically, we consider an available BBM (a rule-based system in our use case) as an error-prone and weakly-supervised annotator (WSA). The annotator model allows us to decide whether annotations can be performed at low cost by the WSA without a costly HA (a domain expert in our use case). In addition, we further enhance the AL-WS cycle with TL. We fine-tune a pre-trained model (a language model in our use case) from an external source on unlabeled internal data for the downstream task with unsupervised learning. This allows us to use labeled and unlabeled data to train our models in the AL cycle. By doing so, we are the first to provide an approach to combine AL with WS and TL by leveraging multiple available information and knowledge sources. Based on the evaluation of our experiments, we summarize our contributions as follows:

1. Enhancing AL with WS by leveraging a rule-based system as an information source through an annotator model leads to a reduction of the annotation cost by 43% with a nearly identical model performance compared to a traditional AL approach. Our approach applies without any adjustments to a rule-based system and any BBM that provides class labels (e.g., a classification model).
2. With the addition of TL, we leverage unlabeled internal data for the downstream task and unlabeled external data through a pre-trained model as knowledge sources for the learning process. This enables us to reduce the annotation cost by 51% compared to a traditional AL approach and improve the model performance compared to the combination of AL-WS.

The remainder of this article is structured as follows. Section 2 presents related approaches and illustrates the difference in our work. Subsequently, we propose our approach in Section 3 and evaluate it in Section 4 within a use case. Finally, we conclude our work and present future challenges in Section 5.

2 Related Work

Since AL is the backbone of our approach, we focus on related work regarding combinations of AL-WS and AL-TL. To the best of our knowledge, there has been no attempt yet to enhance AL with WS and TL.

Active Learning and Weak Supervision. Similar to our approach, [24] and [2] combine AL and WS. However, in their approaches, human experts actively select and annotate instances to improve a generative model that converts one-hot-encoded into probabilistic labels. Moreover, the authors of [3] use this combination to improve the expert rules of a WS model with interactive user feedback. In contrast to our approach, these methods primarily focus on WS and try to improve it with AL techniques. Instead, we focus on an AL cycle and enhance it with WS to reduce the annotation cost. Additionally, these works require labeling functions that are created from scratch. We, on the contrary, can automatically leverage information from any existing BBM that generates class labels without necessarily designing labeling functions. This simplification saves the effort to decompose an existing BBM for a generative model and enables us to treat it as a WSA in an AL cycle.

In comparison, [7] and [28] follow a similar objective as we do since they also aim to enhance a traditional AL cycle with WS techniques to reduce human interaction. The authors of [7] assign a pseudo label for a given instance in a self-training setting if the classifier's predicted probability exceeds a certain threshold. Additionally, they automatically assign the majority class label of similar instances to all unlabeled instances in a cluster. Moreover, instead of annotating single instances, [28] use human labels to annotate a cluster of similar instances to reduce human effort. However, these works do not consider a BBM that generates class labels in a real-world setting. We automatically leverage this existing knowledge source through an annotator model, reducing the annotation cost in an AL cycle.

Active Learning and Transfer Learning. The authors of [27] combine AL and TL but from a different perspective. While we aim to improve AL with TL, they enhance TL by actively selecting the most-suitable instances for the source domain from the target domain. Furthermore, [14] actively fine-tune a pre-trained model based on the contribution of an instance for the feature representation and performance of a classification model on a target task to reduce the annotation cost. In contrast, we do not actively select instances in the TL process but enhance a classification and annotator model within an AL cycle with transferred knowledge. Additionally, [17] investigate how TL mitigates the random initialization cold start and reduces label queries. The authors of [33] also leverage available unlabeled data but through unsupervised feature learning at the beginning of an AL cycle and semi-supervised learning during the cycle. They employ unsupervised pre-training by clustering the features and train a semi-supervised model by generating pseudo-labels for unlabeled instances. This way, they improve the model's performance while requiring less labeled data [33]. In our approach, however, we apply unsupervised learning not only on existing internal data but also propose to utilize external knowledge sources with TL.

3 Proposed Approach

In Section 3.1, we first give a formal definition of our problem setting. Consequently, we describe our proposed approach in Section 3.2 as shown in Figure 1. We design a modular approach so that we can selectively combine AL with WS and TL. This enables us to compare the influence of the individual components on the model performance and annotation cost.

3.1 Problem Setting

Problem. We consider a classification problem where we have a D-dimensional instance that is described by a feature vector $\mathbf{x} \in \mathcal{X}$ where $\mathcal{X} = \mathbb{R}^D$ describes the feature space. An instance $\mathbf{x}$ is drawn independently from the same distribution and belongs to a ground truth class label $y \in \mathcal{Y}$ where the set $\mathcal{Y} = \{1, ..., C\}$ defines the space of all class labels and C is the number of classes. In a pool-based AL scenario, we are given an unlabeled pool data set $\mathcal{U}(t) \subseteq \mathcal{X}$ without class labels. At each cycle iteration $t \in \mathbb{N}$, we aggregate the most-useful instances $\mathbf{x}^*$ in a batch $\mathcal{B}(t) \subset \mathcal{U}(t)$ with the size $b \in \mathbb{N}$. These instances require labels for the next cycle t+1 that annotators provide. Therefore, we define a set of annotators $\mathcal{A} = \{\mathrm{HA}, \mathrm{WSA}\}$, where we treat the HA as omniscient, providing a costly ground truth class label and an available BBM as a WSA, providing an error-prone class label at a low cost. Besides the class labels y to train the classification model, we also add a binary agreement label $z \in \mathcal{Z}$ with the set $\mathcal{Z} = \{0, 1\}$ to every instance in a batch to train the annotator model. We determine z based on the agreement between the labels provided by the HA and the WSA. It represents which instances were correctly classified (1) or misclassified (0) by the WSA. This means that we have to retrieve the WSA label at every selected instance. Thus, we denote the annotated batch as $\mathcal{B}^*(t) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ and the labeled data set as $\mathcal{L}(t) \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$.

Model training. We express the classification model (e.g., a NN) through its parameters at cycle iteration t as $\boldsymbol{\theta}_t$. This model is trained on the labeled data set $\mathcal{L}(t)$ where either the HA or the WSA provide the class label y. It maps an instance to a vector of class probabilities with $f^{\boldsymbol{\theta}_t} : \mathcal{X} \to \Delta_{C-1}$, where Δ_{C-1} is the $C - 1$ probability simplex spanned by C classes. Given an instance $\mathbf{x} \in \mathcal{X}$, the classification model predicts the probability vector $\hat{\mathbf{p}} = f^{\boldsymbol{\theta}_t}(\mathbf{x})$. This vector corresponds to an estimate of the categorical distribution of the classes made by the model $f^{\boldsymbol{\theta}_t}$. Additionally, we describe the annotator model through its parameters $\boldsymbol{\omega}_t$ which result from training on the binary agreement label z of the labeled data set $\mathcal{L}(t)$. With the function $g^{\boldsymbol{\omega}_t} : \mathcal{X} \to [0, 1]$ the annotator model maps an instance $\mathbf{x} \in \mathcal{X}$ to a probability $\hat{q} = g^{\boldsymbol{\omega}_t}(\mathbf{x})$. Its task is to estimate the probability that the WSA can provide a true class label. Thus, both models receive the same input instances from $\mathcal{L}(t)$ but are trained either on class or binary agreement labels. Moreover, we denote the parameters extracted from a pre-trained model as $\boldsymbol{\phi}$. Since the pre-trained model is only trained once, these parameters are independent of the cycle iterations.

3.2 Proposed Cycle

Our proposed AL cycle is illustrated in Figure 1. In the following paragraphs, we will give a detailed explanation of the steps in our approach.

Step 1 - Initialize Cycle. Before the cycle starts, we fine-tune a pre-trained model on the unlabeled data $\mathcal{U}$ and all additional data that we do not consider for AL with unsupervised learning. This model supplies initial parameters $\boldsymbol{\phi}$ for the classification and annotator model and provides feature representations that are helpful for AL [33]. Thus, we do not randomly initialize the parameters of a model at each cycle iteration. In our case, we utilize a pre-trained language model to extract word embeddings for the downstream task. In the first step, ① at iteration t, the classification and annotator model are initially trained on a small labeled data set $\mathcal{L}(t)$ where the instances $\mathbf{x}$ are drawn randomly from the unlabeled pool data set $\mathcal{U}(t)$. Here, the HA provides the ground truth class labels, and the WSA the error-prone class labels allowing us to compute the binary agreement label, which is utilized for training the annotator model. After the initialization step, we assume to have a trained classification model with the parameters $\boldsymbol{\theta}_t$ and a trained annotator model with the parameters $\boldsymbol{\omega}_t$.

Step 2 - Select Batch. The cycle continues in step ② with the selection algorithm of the **AL** module. We approximate the utility of all instances from the unlabeled pool $\mathcal{U}(t)$ based on the entropy of the predicted probability of the classification model $f^{\boldsymbol{\theta}_t}$. Given a probability vector $\hat{\mathbf{p}}$, the entropy is defined as

$$H(\hat{\mathbf{p}}) = -\sum_{c=1}^{C} \hat{p}_c \ln \hat{p}_c. \tag{1}$$

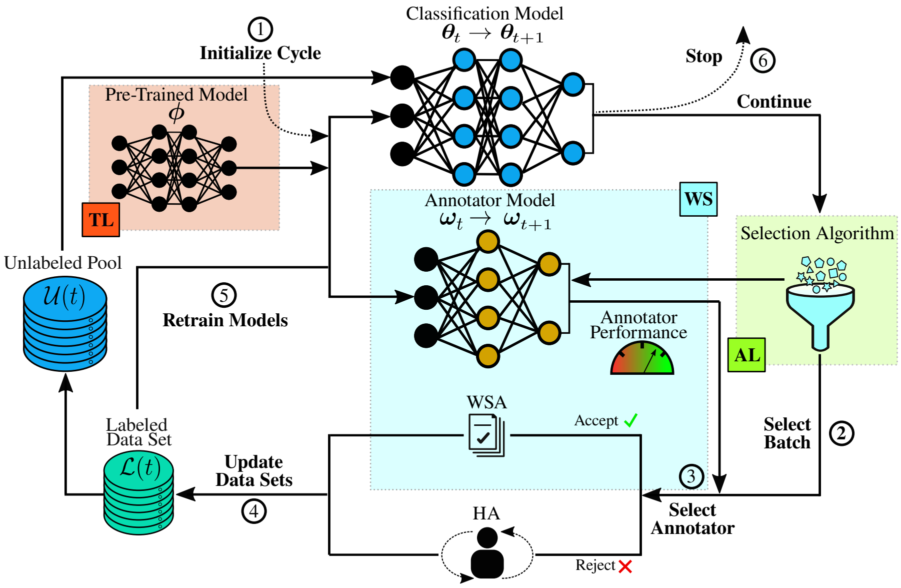

Fig. 1. A schematic illustration of the proposed AL cycle with WS and TL.

At cycle iteration t, we select the instance with maximum entropy according to

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathcal{U}(t)} H\big(f^{\boldsymbol{\theta}_t}(\mathbf{x})\big)\,. \tag{2}$$

To aggregate a batch $\mathcal{B}(t) \subset \mathcal{U}(t)$, we greedily select the most-useful instances $\mathbf{x}^*$ until we reach the desired acquisition batch size $b \in \mathbb{N}$. We refer to this sampling strategy as max-entropy sampling.

Step 3 - Select Annotator. In step ③ with the **WS** module, we estimate the annotator performance of the WSA to decide whether it should provide the class labels for a specific instance. Therefore, we give each instance $\mathbf{x}^*$ of the selected batch $\mathcal{B}(t)$ to the annotator model $g^{\boldsymbol{\omega}_t}$ which estimates the probability $\hat{q}$. Intuitively, we interpret $\hat{q}$ as the probability that the WSA is capable of providing the ground truth class label. This way, the annotator model assesses the performance of the WSA. With the annotator performance estimation we decide whether to reject an error-prone class label of the WSA. In our approach, we investigate a simple reject function[1] that is based on threshold α and the estimated probability $\hat{q}$ as given by

$$r_\alpha(g^{\boldsymbol{\omega}_t}(\mathbf{x}^*)) = \begin{cases} 1, & \text{if } g^{\boldsymbol{\omega}_t}(\mathbf{x}^*) \geq \alpha \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

[1] It should be noted that more complex reject functions are available that could be the focus of future research.

If a class label of the WSA is rejected, the HA has to provide the true class label, enabling us to determine the binary agreement label z. However, suppose we decide that the WSA can provide a ground truth class label. In that case, the binary agreement label is set to 1 as a pseudo-label in the labeled pool. We refer to this as a pseudo-label because no ground truth is available. This technique can be considered semi-supervised learning [33].

Step 4 - Update Data Sets. In ④, we update the unlabeled pool data set $\mathcal{U}(t{+}1) = \mathcal{U}(t) \setminus \mathcal{B}(t)$ with the instances from the aggregated batch. Additionally, we update the labeled training set $\mathcal{L}(t{+}1) = \mathcal{L}(t) \cup \mathcal{B}^{*}(t)$ with the annotated batch including the class and the binary agreement labels.

Step 5 - Retrain Models. In ⑤, the classification and annotator model are retrained from scratch simultaneously. Before training, we initialize the models' parameters with the parameters $\boldsymbol{\phi}$ we obtain from the unsupervised pre-trained model. This leads to an update of the model parameters $\boldsymbol{\theta}_{t+1}$ and $\boldsymbol{\omega}_{t+1}$.

Step 6 - Continue/Stop Cycle. At the end of an iteration, we decide in ⑥ whether to continue or stop the AL cycle with a stopping criterion. AL strategies in literature often use a simple pre-defined stopping criterion such as the desired size of the labeled pool or the maximum number of cycle iterations [20,31]. As this is not in the scope of this work, we choose the maximum number of instances as our stopping criterion.

4 Experimental Evaluation

In Section 4.1, we summarize the experimental setup for our use case. We design our experiments to enhance the AL cycle sequentially with the WS and TL modules to investigate their impact on model performance and annotation cost. The first experiments in Section 4.2 detail our findings where we enhance AL with WS to leverage an available BBM as an information source to reduce the annotation cost. Subsequently, Section 4.3 gives insights on how the addition of TL further improves our approach by utilizing internal and external unlabeled data with a pre-trained model as a knowledge source.

4.1 Experimental Setup

Use Case and Data. The data set in our use case consists of banking transactions. The goal is to predict an appropriate thematic class (e.g., household or insurance) based on short text descriptions of transactions with a NN. We do not have a labeled data set available, but the following information and knowledge sources are at our disposal:

1. **External Data**: Besides internal in-domain data for the downstream task, a vast amount of general-domain text data is available on the Internet [29].

As a pre-trained language model, we employ a fastText model [16] as a knowledge source. This model was trained on a general-domain corpus [8] and is available open-source[2]. We do not employ a deep transformer model in this preliminary investigation to avoid the issues of deep AL.

2. **Internal Data**: We leverage an extensive unlabeled data set with 7.7 million transactions to fine-tune the fastText model in an unsupervised manner with in-domain knowledge. To conduct the experiments efficiently, we randomly sample 9000 instances as the pool data set $\mathcal{U}$ and reserve 2000 instances with ground truth class labels for testing.
3. **Black-Box Model**: A rule-based system that classifies transactions with hand-crafted labeling rules is available. It was developed iteratively over several years by domain experts, and we consider it a BBM since the labeling rules are unavailable. We treat the BBM as the WSA that generates error-prone class labels at low cost. Preliminary studies show that it achieves an accuracy of approximately 86% on the test set.
4. **Human Annotator**: We assume a domain expert as an omniscient annotator that delivers ground truth class labels at a high cost. Specifically, the HA provides the class labels for the actively selected training instances when the label of the WSA is rejected and for the initialization step.

Models. The results are obtained by a classification model in our proposed AL cycle. The classification model is a multi-layer perceptron with an embedding layer to represent the text input with $D = 300$, a hidden layer with a ReLU activation function and an output layer with $C = 36$ neurons for each class. The annotator model is comprised of a similar structure, differing only in the output layer with $C = 1$ neuron as the annotator model solves a binary classification task. In each cycle iteration, we create a new vocabulary from the labeled pool and adapt the input layer of both models. We employ the Adam optimizer [18] to optimize the parameters, and the focal loss [22] as a loss criterion to address class imbalance. Additionally, we add dropout with 20% probability to the hidden neurons. We extract the static word embeddings from the pre-trained fastText model as initial weights of the embedding layers. This process can be considered as sequential TL [29].

Overall Experimental Design. To ensure comparability between our experiments, we define basic AL parameter configurations for all experiments. The configurations are generally based on results from preliminary studies in this use case. Specific settings for the experiments are highlighted in the corresponding sections. The initial labeled data set consists of 250 randomly sampled instances with ground truth labels provided by the HA. In preliminary work, this has proven to be a sufficient initial quantity of instances to enable the models to provide information to select the most-useful instances and suitable annotators. We set the desired size of the labeled data pool to 5370 as a pre-defined stopping criterion and the acquisition batch size b to 32 with 161 cycle iterations t. Our

[2] https://fasttext.cc/docs/en/crawl-vectors.html, accessed 2022-04-20

previous studies have shown that this relatively small number of instances leads to key results while enabling us to conduct experiments efficiently. We employ random sampling as a baseline sampling strategy and compare it to max-entropy sampling (Equation 2) for each experiment. Additionally, we decide between a costly (HA) or low-cost class label (WSA) based on our proposed reject option (Equation 3). Therefore, we define three different annotation scenarios to assess the influence of the WSA and the resulting annotation costs:

1. *full-human*: The HA provides the class labels for all of the selected instances, and we reject the class labels of the WSA. We consider this scenario a conventional AL approach without WS that should achieve the highest performance but generate the greatest baseline annotation cost.
2. *hybrid*: We select the WSA and the HA to provide the class labels based on the assessment of the annotator performance. In preliminary studies, 0.85 has proven to be a simple and promising reject threshold α, ensuring that we only accept labels of the WSA at high annotator performance estimations. At the same time, we ask the HA only for very uncertain instances to minimize annotation cost. Note that we must retrieve the class label of the WSA for every instance to determine the binary agreement label. This scenario reflects our approach combining AL with WS.
3. *full-WSA*: The WSA provides the class labels for all selected instances. This approach is the most inexpensive regarding the annotation cost, but we expect a deterioration of model performance. To ensure comparability, the HA still determines the ground truth class labels for the random initialization step.

As an exemplary cost scheme, we assign a cost of 1 to each annotation by the HA. Since the maintenance of the rule-based system as the BBM and automatically retrieving a class label also generates low cost, we assign 0.1 to an annotation of the WSA. Additionally, each experiment is repeated five times with different random seeds.

4.2 Experiments on AL with WS

This section shows the experimental results to answer research question 1. In these experiments, we utilize the HA and the available rule-based system as information sources with AL and WS.

Question 1. How can we enhance AL with WS so that we can leverage an available BBM as an information source to reduce the annotation cost with a competitive model performance compared to a traditional AL approach?

Findings. In Figure 2, we show the test accuracy and annotation cost for the aforementioned annotation scenarios and sampling strategies for each cycle iteration. Additionally, we report the final results in Table 1 after the AL cycle reaches the stopping criterion. The savings metric represents the cost saved relative to the highest baseline cost with conventional AL. As Figure 2 shows on

Table 1. Mean results (± standard error) of accuracy, annotation cost and savings of the AL cycle with different sampling strategies and annotation scenarios.

Sampling	Scenario	Accuracy(↑)	Cost(↓)	Savings(↑)
random	full-human	0.849±0.001	5370	0
	hybrid	0.842±0.001	1842±221	0.66
	full-wsa	0.823±0.004	762	0.86
max-entropy	full-human	0.873±0.001	5370	0
	hybrid	0.872±0.002	3045±46	0.43
	full-wsa	0.842±0.002	762	0.86

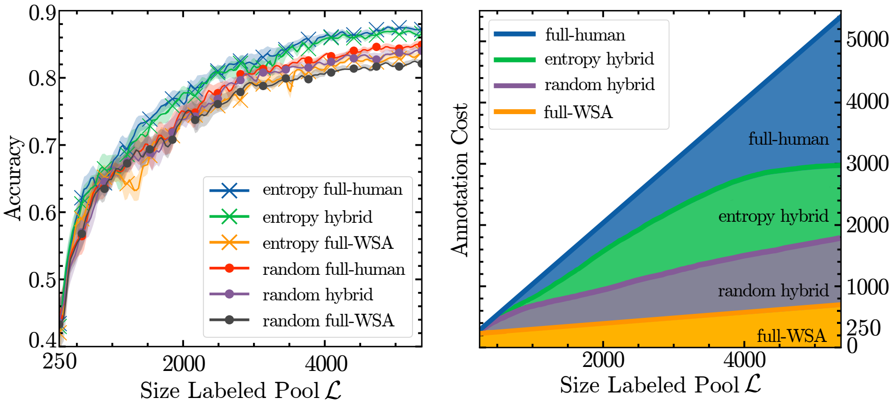

Fig. 2. Test accuracy and annotation cost with increasing size of the labeled data pool in the AL cycle with different sampling strategies and annotation scenarios.

the right, the annotation costs for the annotation scenarios *full-human* (highest baseline annotation cost and traditional AL) and *full-wsa* (lowest annotation cost without HA) are constant and independent of the sampling strategy. The former cost is identical to the size of the labeled pool since the HA provides labels for each instance. For the latter cost, only the initial labels are provided by the costly HA while the WSA generates the remaining labels at a low cost. With our approach in the *hybrid* scenario, the annotation cost depends on a mix of HA and WSA annotations. More WSA labels are generally rejected when using max-entropy sampling compared to random sampling in our *hybrid* scenario. The savings in Table 1 demonstrate that we can save annotation costs of 43 % with max-entropy sampling and 66 % with random sampling compared to the baseline cost of 5370 in the *full-human* scenario. However, we can see that random sampling degrades test accuracy. We attribute this to the fact that we actively select instances where the classification model is most uncertain in each batch. These also seem to be instances where the annotator model is uncertain and, thus, we more frequently reject the error-prone WSA. Additionally, we can observe a decreasing slope of the green cost curve with max-entropy sampling in our *hybrid* scenario on the left side of Figure 2. This seems intuitive since the high-entropy instances from the unlabeled pool also diminish with cycle iterations. Therefore, a bigger labeled pool as the pre-defined stopping criterion could lead to only a slight increase in annotation cost and more strongly emphasize the benefits of our approach. The slope of the purple cost curve further highlights this assumption as it is monotonously increasing with random sampling, where we draw instances without considering the uncertainty of the classification model.

When looking at the accuracy in Figure 2, we observe that the model performance with max-entropy sampling is consistently superior to random sampling in each annotation scenario. Table 1 supports this observation and shows a performance increase of up to 3 % in accuracy with AL. Accordingly, the classification model's accuracy grows more rapidly in each cycle iteration, and it reaches the highest test accuracy with max-entropy sampling in the *hybrid* and *full-human* annotation scenarios. This demonstrates how AL techniques enable us to obtain a better classification accuracy with the same number of labeled instances compared to random sampling. The worst classification accuracy is obtained by random sampling in the *full-WSA* scenario. Accordingly, the results deteriorate for both selection strategies when only the error-prone WSA provides the class labels. Even though we can obtain savings of 86 % in the *full-WSA* scenario, the accuracy of the BBM (rule-based system) limits the achievable test accuracy of the classification model. This emphasizes the importance of ground-truth class labels from HAs and, thus, strengthens our combined approach in the *hybrid* scenario. As we expect, the classification model provides the best accuracy in the *full-human* scenario with max-entropy sampling as the traditional AL approach. However, our approach in the *hybrid* scenario with max-entropy sampling delivers nearly identical test accuracy while reducing the annotation cost by 43%, as seen by savings in Table 1. Our results show that while costly HAs are important, we can also leverage a BBM as an additional information source. These observations let us conclude that our combination of AL and WS greatly reduces the annotation cost with only a marginal performance loss compared to traditional AL.

4.3 Experiments on AL with WS and TL

In this section, we conduct experiments with our complete proposed approach to tackle the second research question. In addition to WS and AL, we leverage all of the available unlabeled data to train a language model, which serves as a sequential TL approach. We focus on the *hybrid* annotation scenario with and without pre-training. So, we assess the influence of using all available information and knowledge sources on the model performance and annotation cost.

Question 2. How far can the inclusion of TL to leverage unlabeled internal and external data as knowledge sources empower the combination of AL-WS and, thus, further reduce the annotation cost and improve the model performance?

Findings. Figure 3 shows the test accuracy and annotation cost for the aforementioned sampling strategies in the *hybrid* scenario with and without pre-training.

Table 2. Mean results (± standard error) of accuracy, annotation cost, and savings of the AL cycle in different annotation scenarios with and without TL

Sampling	Scenario	Accuracy(↑)	Cost(↓)	Savings(↑)
random	full-human	0.879±0.002	5370	0
	hybrid	0.876±0.002	1819±51	0.66
	full-wsa	0.840±0.003	762	0.86
max-entropy	full-human	0.894±0.003	5370	0
	hybrid	0.893±0.001	2652±40	0.51
	full-wsa	0.847±0.002	762	0.86

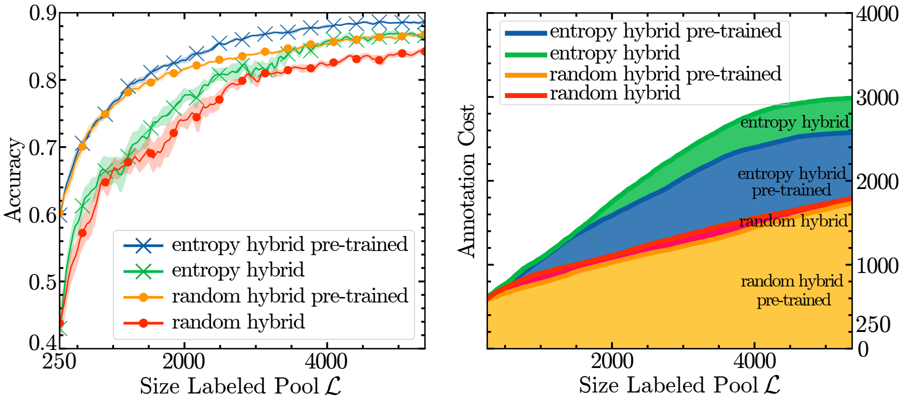

Fig. 3. Test accuracy and annotation cost with increasing size of the labeled data pool in the AL-WS cycle with and without TL.

In Table 2, we summarize the final results in all annotation scenarios with pre-training. We can see in Figure 3 that utilizing pre-trained weights gives the classification model a clear head start in performance. After initial training in the *hybrid* scenario, the model already reaches an accuracy of 60% for random (orange curve) and max-entropy (blue curve) sampling. This increase represents a 20% improvement over the green and red curves without pre-training. The advantage generally decreases with more training data but remains fundamentally intact and demonstrates the benefits of adding TL to our WS-AL approach, as also demonstrated in Table 2. We obtain the best results with the fastest accuracy increase in each iteration with pre-training and maximum-entropy sampling (blue curve). However, with the increasing size of the labeled data set, the accuracy of max-entropy sampling without pre-training adjusts to the same level of random sampling with pre-training. This means that max-entropy sampling has the same effect on the final model accuracy as leveraging the knowledge extractable from 7.7 million transactions and shows the general advantage of AL as the backbone of our approach. Table 2 further highlights the increase in accuracy in all annotation scenarios with TL compared to Table 1. Additionally, the curves' trajectories with pre-training are more consistent with much less performance variance across the experiments' seeds.

On the right sight in Figure 3, we can see that the annotation cost with pre-training for max-entropy sampling is lower than without pre-training. Again, random sampling leads to lower annotation costs and poorer accuracy and confirms the benefits of using AL from the results above. Table 2 also highlights the improved savings of 51 % with the addition of TL compared to the baseline cost of 5370 with an 8 % increase relative to AL-WS. Moreover, we assume that the transferred knowledge improves the classification model's and the annotator model's certainty estimations. This means that pre-trained weights enable us to more efficiently select the most-useful instances and the low-cost annotations of the WSA. The results demonstrate the benefits of enhancing our AL-WS approach with TL by also leveraging available unlabeled data as a knowledge source with a pre-trained model. With our AL-WS-TL approach, we can improve the overall test accuracy of the classification model while further reducing the annotation cost.

5 Conclusion and Future Work

This work presented a novel approach to extending AL with WS and TL to reduce the annotation cost by leveraging multiple information and knowledge sources. We treated an established BBM (e.g., a rule-based system) as a weakly-supervised annotator that provides error-prone class labels inexpensively. This assumption made it possible to estimate the performance of this information source with an annotator model to decide whether a costly human annotation in an AL cycle is required. In a use case, we have successfully shown that enhancing AL with WS reduces annotation cost by 43% and leads to an almost identical model performance compared to traditional AL. Moreover, we leveraged unlabeled internal and external data as knowledge sources by fine-tuning a pre-trained language model on all available unlabeled data in an unsupervised manner. We then transferred this knowledge to expand our AL-WS cycle with TL. This enabled us to reduce the annotation cost by 51 % and improve the overall model performance compared to the AL-WS approach.

Since we applied our proposed approach for a shallow NN, we plan to move towards deep AL and the related problems in an application-oriented setting. To provide an accurate probabilistic estimation for the selection of instances, we aim to investigate the uncertainty estimates [15] of our classification and annotator models and calibrate them with methods such as temperature scaling [9] or scaling-binning [21]. Since we greedily acquired a batch of instances without batch-awareness, we intend to use a more complex selection strategy, such as BALD [6,19]. Moreover, we aim to enhance and further investigate the annotator model to measure the label quality of other information sources in the annotation process, such as the HA. Accordingly, we can move towards modern AL settings, where we also consider the HA as error-prone and can determine a more complex cost scheme [11]. This could also be done in a multi-task learning setting by embedding the annotator model directly into the classification model.

Acknowledgments. This work results from the project INFINA, funded by Wirtschafts- und Infrastrukturbank Hessen under the Operational Program for the Promotion of Investments in Growth and Employment in Hessen which is financed by the European Regional Development Fund (ERDF).

References

1. Bach, S.H., Rodriguez, D., Liu, Y., Luo, C., Shao, H., Xia, C., Sen, S., Ratner, A., Hancock, B., Alborzi, H., Kuchhal, R., Ré, C., Malkin, R.: Snorkel DryBell: A Case Study in Deploying Weak Supervision at Industrial Scale. In: Proceedings of the 2019 International Conference on Management of Data. pp. 362–375 (2019). https://doi.org/10.1145/3299869.3314036
2. Biegel, S., El-Khatib, R., Oliveira, L.O.V.B., Baak, M., Aben, N.: Active weasul: Improving weak supervision with active learning. CoRR (2021). https://doi.org/10.48550/arXiv.2104.14847
3. Boecking, B., Neiswanger, W., Xing, E., Dubrawski, A.: Interactive weak supervision: Learning useful heuristics for data labeling. ICLR (2021)
4. Budd, S., Robinson, E.C., Kainz, B.: A Survey on Active Learning and Human-in-the-Loop Deep Learning for Medical Image Analysis. Medical Image Analysis **71**, 102062 (2021). https://doi.org/10.1016/j.media.2021.102062
5. Dunnmon, J.A., Ratner, A.J., Saab, K., Khandwala, N., Markert, M., Sagreiya, H., Goldman, R., Lee-Messer, C., Lungren, M.P., Rubin, D.L., Ré, C.: Cross-Modal Data Programming Enables Rapid Medical Machine Learning. Patterns **1**(2), 100019 (2020). https://doi.org/10.1016/j.patter.2020.100019
6. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70. pp. 1183—1192. ICML (2017)
7. Gonsior, J., Thiele, M., Lehner, W.: WeakAL: Combining Active Learning and Weak Supervision. In: Appice, A., Tsoumakas, G., Manolopoulos, Y., Matwin, S. (eds.) Discovery Science. pp. 34–49. Lecture Notes in Computer Science, Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-61527-7_3
8. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018) (2018)
9. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning. pp. 1321–1330. ICML (2017)
10. Hanika, T., Herde, M., Kuhn, J., Leimeister, J.M., Lukowicz, P., Oeste-Reiß, S., Schmidt, A., Sick, B., Stumme, G., Tomforde, S., Zweig, K.A.: Collaborative Interactive Learning – A clarification of terms and a differentiation from other research fields. CoRR (2019). https://doi.org/10.48550/arXiv.1905.07264
11. Herde, M., Huseljic, D., Sick, B., Calma, A.: A survey on cost types, interaction schemes, and annotator performance models in selection algorithms for active learning in classification. CoRR (2021). https://doi.org/10.48550/arXiv.2109.11301

12. Hino, H.: Active learning: Problem settings and recent developments. CoRR (2020). https://doi.org/10.48550/arXiv.2012.04225
13. Holzinger, A., Plass, M., Kickmeier-Rust, M., Holzinger, K., Crişan, G.C., Pintea, C.M., Palade, V.: Interactive machine learning: Experimental evidence for the human in the algorithmic loop: A case study on Ant Colony Optimization. Applied Intelligence **49**(7), 2401–2414 (2019). https://doi.org/10.1007/s10489-018-1361-5
14. Huang, S.J., Zhao, J.W., Liu, Z.Y.: Cost-Effective Training of Deep CNNs with Active Model Adaptation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1580–1588. ACM (2018). https://doi.org/10.1145/3219819.3220026
15. Huseljic, D., Sick, B., Herde, M., Kottke, D.: Separation of aleatoric and epistemic uncertainty in deterministic deep neural networks. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 9172–9179 (2021). https://doi.org/10.1109/ICPR48806.2021.9412616
16. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of Tricks for Efficient Text Classification. CoRR (2016). https://doi.org/10.48550/arXiv.1607.01759
17. Kale, D., Liu, Y.: Accelerating Active Learning with Transfer Learning. In: 2013 IEEE 13th International Conference on Data Mining. pp. 1085–1090 (2013). https://doi.org/10.1109/ICDM.2013.160
18. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. CoRR (2017). https://doi.org/10.48550/arXiv.1412.6980
19. Kirsch, A., van Amersfoort, J., Gal, Y.: BatchBALD: Efficient and diverse batch acquisition for deep bayesian active learning. In: Advances in Neural Information Processing Systems (2019)
20. Kottke, D., Schellinger, J., Huseljic, D., Sick, B.: Limitations of Assessing Active Learning Performance at Runtime. CoRR (2019). https://doi.org/10.48550/arXiv.1901.10338
21. Kumar, A., Liang, P., Ma, T.: Verified uncertainty calibration. In: Advances in Neural Information Processing Systems (NeurIPS) (2019)
22. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal Loss for Dense Object Detection. CoRR (2018). https://doi.org/10.48550/arXiv.1708.02002
23. Liu, H., Gegov, A., Cocea, M.: Rule-based systems: A granular computing perspective. Granular Computing **1**(4), 259–274 (2016). https://doi.org/10.1007/s41066-016-0021-6
24. Nashaat, M., Ghosh, A., Miller, J., Quader, S., Marston, C., Puget, J.F.: Hybridization of Active Learning and Data Programming for Labeling Large Industrial Datasets. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 46–55 (2018). https://doi.org/10.1109/BigData.2018.8622459
25. Paleyes, A., Urma, R.G., Lawrence, N.D.: Challenges in Deploying Machine Learning: A Survey of Case Studies (2021)
26. Pan, S.J., Yang, Q.: A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering **22**(10), 1345–1359 (2010). https://doi.org/10.1109/TKDE.2009.191
27. Peng, Z., Zhang, W., Han, N., Fang, X., Kang, P., Teng, L.: Active Transfer Learning. IEEE Transactions on Circuits and Systems for Video Technology **30**(4), 1022–1036 (2020). https://doi.org/10.1109/TCSVT.2019.2900467
28. Perez, F., Lebret, R., Aberer, K.: Weakly Supervised Active Learning with Cluster Annotation. CoRR (2019). https://doi.org/10.48550/arXiv.1812.11780
29. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X.: Pre-trained Models for Natural Language Processing: A Survey. Science China Technological Sciences **63**(10), 1872–1897 (2020). https://doi.org/10.1007/s11431-020-1647-3

30. Ratner, A., Bach, S.H., Ehrenberg, H., Fries, J., Wu, S., Ré, C.: Snorkel: Rapid training data creation with weak supervision. The VLDB Journal **29**(2), 709–730 (2020). https://doi.org/10.1007/s00778-019-00552-1
31. Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Gupta, B.B., Chen, X., Wang, X.: A survey of deep active learning. ACM Comput. Surv. **54**(9) (2021). https://doi.org/10.1145/3472291
32. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2010)
33. Siméoni, O., Budnik, M., Avrithis, Y., Gravier, G.: Rethinking deep active learning: Using unlabeled data at model training. In: International Conference on Pattern Recognition (ICPR). pp. 1220–1227 (2021). https://doi.org/10.1109/ICPR48806.2021.9412716