WeakAL: Combining Active Learning and Weak Supervision

Julius Gonsior[1 [0000−0002−5985−4348]], Maik Thiele[1 [0000−0002−1665−977X]], and Wolfgang Lehner[1 [0000−0001−8107−2775]]

Technische Universität Dresden, Dresden, Germany
{firstname.lastname}@tu-dresden.de

Abstract. Supervised Learning requires a huge amount of labeled data, making efficient labeling one of the most critical components for the success of Machine Learning (ML). One well-known method to gain labeled data efficiently is Active Learning (AL), where the learner interactively asks human experts to label the most informative data point. Nevertheless, even by applying AL in labeling tasks the amount of human effort is still too high and should be minimized further.
In this paper therefore we propose WeakAL, which incorporates Weak Supervision (WS) techniques directly into the AL cycle. This allows us to reduce the number of annotations by human experts while keeping the same level of ML performance. We investigate different WS strategies as well as different parameter combinations for a wide range of real-world datasets. Our evaluation shows that for example in the context of Web table classification, 55% of otherwise manually retrieved labels can be generated by WS techniques with a negligible loss of test accuracy by 0.31% only. To further prove the general applicability of our approach we applied it to six datasets from the AL challenge from Guyon et al., where over 90% of the labels could be computed by the WS techniques, while still achieving competitive competition results.

Keywords: Information Extraction · Active Learning · Semi-supervised · Machine Learning · Weak Supervision · Classification

1 Introduction

Acquiring training data for supervised learning, such as classification, requires substantial human effort, which already led to many research activities with the goal to increase data efficiency and to minimize the need for manual annotation. The first one is *Active Learning* (AL) that deals with the problem of selecting samples from an unlabeled pool for labeling, e.g. by a human annotator, such that the performance of the model to be learned is maximized. The second one is *Weak Supervision* (WS) that uses a labeled ground-truth to compute labels for the unlabeled data, to improve the quality of the classifier.
Traditionally WS is applied after a small high-quality dataset has been obtained, e.g. through AL. In an optimal setting, AL would query only a few representative

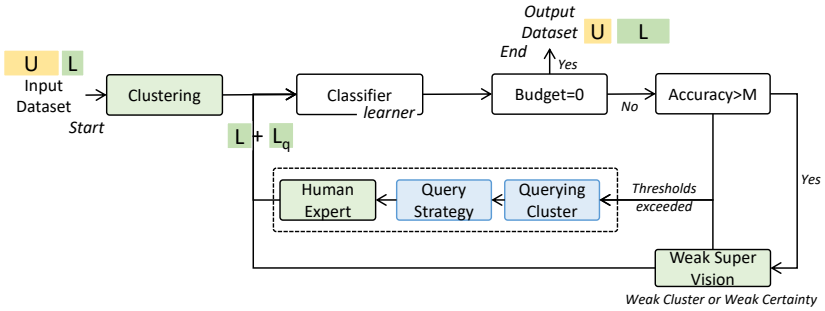


Fig. 1: WeakAL Overview

samples for each class and the other labels would be derived using WS techniques. However, in practice this is often not the case: Either too many redundant labels from the AL cycle were obtained, which could also have been generated by WS, or the obtained labels don't work well in combination with WS and produce a lot of false labels.

Therefore, in this paper, we propose WeakAL, which extends the AL cycle by different WS techniques (see Figure 1). Given a small initial labeled dataset $\mathcal{L}$ and a large unlabeled pool $\mathcal{U}$, we first cluster the combined samples of $\mathcal{L}$ and $\mathcal{U}$. Then the classifier is trained on $\mathcal{L}$, and as long as the human labor budget is not exhausted, WeakAL augments the labeled dataset by additional samples. At this point, in the traditional AL cycle, only human experts would be queried. In WeakAL however, also WS techniques are directly incorporated to obtain labels. If a minimum amount of labeled data is available, which is ensured by an accuracy threshold M, the WS strategies are queried. We propose to use two WS techniques: WeakClust and WeakCert. The first one propagates the majority label in a cluster to the unlabeled samples of the cluster, whereas the second one uses the predicted label by the classifier. Based on the parameters for the respective WS strategies, they either return so-called *weak* labels or nothing, indicating that they are not confident enough. Depending on the present labeled data $\mathcal{L}$ and the parameters, the weak labels add more or less label noise. However, by using well-tweaked parameters this can be kept to a minimum. If the WS strategies are not confident enough human experts are consulted, where first a *cluster query strategy* (CQS) identifies a cluster, from which thereafter the *query selection strategy* (QS) selects the samples for the query. The generated labels are be added to the labeled set $\mathcal{L}$ and the cycle starts again.

Contribution. In this paper, we introduce WeakAL that extends the AL cycle by different WS techniques. In a comprehensive experimental study, we show that combining AL and WS provides very good results in terms of human effort and classification accuracy for many real-world datasets. Our experiments show that the classification models trained on the data determined by AL and WS can safely reduce the amount of human-retrieved annotations by 50% - 90%

while maintaining the same level of accuracy, and even improving it by a few percentage points.

Outline. The remainder of this paper is organized as follows: In Section 2, we present the typical methods used within an AL cycle, which is extended by WS strategies in Section 3. Section 4 describes the setup of the experiments we conducted to prove our hypothesis. We compare different evaluation metrics and combinations of WS strategies on multiple real-world datasets. The results are shown and discussed in Section 5. Finally, we present related work in Section 6 and conclude in Section 7.

2 Active Learning Foundations

WeakAL makes use of typical AL techniques, such as a *cluster query strategy*, a *query strategy* as well as *batching* of samples. Therefore, in Section 2.1, we give an overview of some popular query strategies, which are used in our experiments and emphasize the importance of the right batch size in Section 2.2.

2.1 Active Learning Query Strategies

In this section, we shortly introduce the different strategies for choosing the most informative queries out of a set of given unlabeled samples. Each strategy approximates the contained informativeness of unlabeled data for a potential classifier.

Random Sampling is a common AL query strategy and found application in [1]. Unlike the other methods, random sampling chooses queries at random and fully independently of their informativeness. However, even with this strategy, a rise in prediction accuracy is possible, since the amount of training data is steadily increased. We use random sampling as a baseline to compare the other strategies.

Uncertainty Sampling chooses queries that are the most uncertain to predict. Hence, learning these queries should result in more certain predictions of the classifier. We compare three uncertainty metrics: least confident, margin sampling, and entropy [2]. Least confidence [3] tries to capture the probability, that the classifier is mislabeling the data using the posterior probability P where $\hat{y}$ is the most likely prediction:

$$QS_{x,LC} = \underset{x}{\operatorname{argmax}}\, 1 - P(\hat{y}|x), x \in \mathcal{U} \tag{1}$$

Information about other classes next to the most probable one is not taken into account by this strategy. Margin sampling [4] in contrast uses the posteriors for the first $\hat{y}_1$ and second most probable classes $\hat{y}_2$ and samples the instances with the smallest margin between those two:

$$QS_{x,SM} = \underset{x}{\operatorname{argmin}}\, P(\hat{y}_1|x) - P(\hat{y}_2|x) \tag{2}$$

Entropy uncertainty [5, 6] uses all possible classes and captures the entropy of a given distribution. It should, therefore, work well on classification problems with many classes:

$$QS_{x,E} = \underset{x}{\operatorname{argmax}} - \sum_i P(y_i|x) \log P(y_i|x) \tag{3}$$

2.2 Batch Sizes

It is common practice in machine learning to train a model on batches of samples instead of single data points. As the retraining of the classifier cannot be done in real-time it is also easier for human experts to label a batch of data points at once. Batches also allow parallelization of the human annotation process. Early experiments suggested that the batch size has no real impact on the efficiency of the AL process. We use therefore a reasonably small batch size of 10, which is small enough to show changes during the AL cycle in detail, but also large enough to keep the experiment runtime under control.

3 Weak Supervision Enhanced Active Learning Cycle

In this section, we propose WeakAL, combining the strengths of AL and WS. We claim, that it is beneficial, to prioritize during AL the retrieval of those unlabeled data points, which do not only directly increase the classifier's performance but also lead to more weakly labeled data. We propose an active learning cycle that incorporates WS, resulting in significantly less human interaction, whereas the accuracy achieved is kept on the same level.
Algorithm 1 shows the overall WeakAL cycle. The AL process starts with two datasets: the unlabeled sample set $\mathcal{U}$ and the already labeled dataset $\mathcal{L}$, where $|\mathcal{L}| << |\mathcal{U}|$. WeakAL requires that both $\mathcal{L}$ and $\mathcal{U}$ consist of clusters, $\mathcal{L}_c$, and $\mathcal{U}_c$. Each cluster is defined as a tuple consisting of the feature vector x and, in case of the labeled set, the corresponding label y:

$$\begin{aligned} \mathcal{L}_c &= \{(x_{l1}, y_{l1}), (x_{l2}, y_{l2}), \ldots\} \\ \mathcal{U}_c &= \{x_{u1}, x_{u2}, \ldots\} \end{aligned} \tag{4}$$

The main task of the AL cycle is to iteratively increase the set of labeled data $\mathcal{L}$ by identifying the most promising cells in $\mathcal{U}$. The cycle stops when a predefined *budget* B (line 1 in Algorithm 1) of available user interaction is exhausted. At the beginning of each cycle, the classifier f is retrained on the labeled set $\mathcal{L}$.
If a minimum training accuracy M is reached (line 3), WeakAL utilizes WeakClust and WeakCert (Section 3.2) instead of asking the human experts. The budget remains untouched for WS labels, as these queries come for free without human interaction. Both WS strategies have threshold parameters, α, β, and γ. If the thresholds are not met, human experts are used instead. For that, first, a cluster $\mathcal{U}_c$ of the unlabeled data is selected based on the *cluster query strategy* CQS (line 10, Section 3.1). Then the utilized *query strategy* (line 14, Section 2.1)

selects as much, as per the *batch size* BS defined, unlabeled samples q from the selected cluster $\mathcal{U}_c$. The human experts are then asked for the label, and the budget is reduced accordingly. At the end of each cycle the newly labeled data $\mathcal{L}_q$ is added to $\mathcal{L}$ (line 18) and q removed from $\mathcal{U}$ (line 19), and the process starts again by retraining the classifier on the extended dataset.

Algorithm 1 WeakAL

Input: small clustered labeled start set $\mathcal{L}$, large unlabeled clustered dataset $\mathcal{U}$, query strategy QS, batch size BS, human user interaction budget B, minimum training accuracy before WS M, minimum certainty threshold α, minimum cluster homogeneity β, minimum labeled cluster size γ and a cluster query strategy CQS
Output: labels for $\mathcal{U}$

1: **while** $B > 0$ **do**
2: $f \leftarrow \text{TRAIN}(\mathcal{L})$
3: **if** $\text{ACC}(f, \mathcal{L}) > M$ **then** ▷ Weak Supervision
4: $q, y_q \leftarrow \text{WeakClust}(\mathcal{L}, \mathcal{U}, \beta, \gamma)$
5: **if** $y_q = \emptyset$ **then**
6: $q, y_q \leftarrow \text{WeakCert}(\mathcal{U}, f, \alpha)$
7: **end if**
8: **end if**
9: **if** $y_q = \emptyset$ **then** ▷ Traditional Active Learning
10: $\mathcal{U}_c \leftarrow CQS(\mathcal{U})$
11: $\text{INIT}(q)$
12: **for** $1, \ldots, \text{MIN}(BS, \text{COUNT}(\mathcal{U}_c))$ **do**
13: $B \leftarrow B - 1$
14: $\text{APPEND}(q, \text{QS}(\mathcal{U}_c, f))$
15: **end for**
16: $y_q \leftarrow \text{AskHumanExperts}(q)$
17: **end if**
18: $\text{Merge}(\mathcal{L}, (q, y_q))$
19: $\text{Remove}(\mathcal{U}, q)$
20: **end while**
21: **return** $\mathcal{L}$

3.1 Cluster Query Strategies to support WeakClust

The basic idea of the clustering approach is to save human effort by labeling the entire cluster instead of individual data points. This strategy requires a minimum amount of labels per cluster. We investigate the following three clustering strategies:

Single Cluster Strategy. To compare the approach of limiting the human experts' queries to a single cluster $\mathcal{U}_c$ per AL cycle to the typical approach of using the entire set of unlabeled points $\mathcal{U}$, the *single cluster strategy* puts all unlabeled data into a single cluster, simulating thereby the absence of a cluster strategy.

Random Cluster Strategy. This strategy selects a cluster at random and acts as a second baseline.

Most Uncertain Cluster Strategy. The *most uncertain cluster strategy* can be used in three different flavors, depending on the used uncertainty query strategy: least confidence, smallest margin, and entropy (see Section 2.1). By obtaining

the labels for the most uncertain points per cluster only those remain unlabeled that are more likely part of the class-homogeneous core of the cluster. For each cluster $\mathcal{U}_c \in \mathcal{U}$, the query selection strategy is used first to calculate $QS(x)$ for each sample x. After that, the cluster samples are sorted in descending order based on the value of the query selection strategy. The highest most uncertain data points within the batch size BS are stored in $\overline{\mathcal{U}_c}$. The cluster with the highest sum of query selection certainties is selected accordingly:

$$\mathcal{U}_c = \underset{\mathcal{U}_c}{\operatorname{argmax}} \sum_{x \in \overline{\mathcal{U}_c}} QS(x), \text{ for } \mathcal{U}_c \in \mathcal{U} \tag{5}$$

3.2 Weak Supervision Techniques

We selected two WS techniques, WeakClust and WeakCert, which we believe work best alongside the AL process, and can easily be incorporated into it. WeakClust propagates the labels of a partially labeled cluster to the entire cluster, and WeakCert returns the predicted labels of the trained classifier. Especially the WeakClust technique is optimal for AL, as in an ideal scenario first one sample gets queried per cluster, and then using more most uncertain samples from the cluster, the hypothesis of the first sample gets confirmed or dismissed. Each WS strategy has thresholds that have to be met to confidently add the weak labels to the labeled dataset. A minimum amount of labeled data M needs to be made available first for both WS techniques to justify applying WS. Otherwise, the risk of many false labels from a severely overtrained classifier is increasing. All parameters have to be chosen carefully, as WS automatically computes the annotations and with a suboptimal starting point, many wrong labels can be produced.

Weak Certainty uses the probability of the trained classifier to decide for the unlabeled samples. The pseudocode is given in Algorithm 2. Contrary to the uncertainty AL query strategies, the most certain data points are labeled by this WS strategy. For each unlabeled sample x the predicted label y and the probability σ of the classifier f are calculated (line 4 in Alg. 2). If the probability is higher than the threshold α (line 5), the predicted label gets assigned. All found labels and samples are stored in the lists ys and q (line 6 and 7). WeakCert is therefore basically the application of a single iteration of *self-training* [7].

Algorithm 2 WeakCert

Input: unlabeled data points $\mathcal{U}$, trained classificator f, minimum certainty threshold γ

Output: labels $\boldsymbol{y}$ for a set of unlabeled data points $\boldsymbol{q}$

```
1: Init(q, ys)
2: for U_c ∈ U do
3:     for x ∈ U_c do
4:         y, σ ← ClassWithProb(f, x)
5:         if σ > α then
6:             Append(ys, y)
7:             Append(q, x)
8:         end if
9:     end for
10: end for
11: return q, ys
```

Algorithm 3 WeakClust

Input: labeled data $\mathcal{L}$, unlabeled data $\mathcal{U}$, minimum cluster homogeneity size β, minimum ratio labeled-unlabeled samples γ

Output: labels $\hat{y}$ for the cluster of unlabeled data $\mathcal{U}_c$

```
1: for L_c, U_c ∈ L, U do
2:     if Count(L_c)/Count(U_c) > γ then
3:         ŷ ← MostFrequentLabel(L_c)
4:         if Count(ŷ)/Count(L_c) > β then
5:             return U_c, ŷ
6:         end if
7:     end if
8: end for
9: return ∅, ∅
```

Name	Domain	#Classes	#Features	#Samples	Majority Class
DWTC [9]	Table classification	4	227	5,777	39.84%
HIVA [8]	Chemoinformatics	2	1,617	42,678	96.48%
IBN_SINA [8]	Handwriting recognition	2	92	20,722	62.16%
ORANGE [8]	Marketing	2	230	50,000	98.22%
SYLVA [8]	Ecology	2	216	145,252	93.85%
ZEBRA [8]	Embryology	2	154	61,488	95.42%

Table 1: Datasets used in experiments

Weak Cluster identifies clusters that contain i) a lot of labeled data and ii) almost only samples of with the same label (Algorithm 2). To achieve i) the ratio between labeled and unlabeled samples of the cluster is computed. Only clusters where the ratio is above the threshold γ are considered further (line 2). The second criteria, ensuring ii), is checked by calculating the ratio between the most common class $\hat{y}$ and the size of the cluster (line 4). The first cluster, with a ratio above a threshold β, is returned with $\hat{y}$ as the label for the unlabeled portion. The quality of the underlying clusters has a high impact on the quality of this WS technique. Desirable are many smaller clusters containing only samples of the same class. Note that the propagation of labels from the cluster only applies to unlabeled samples. Possible noise in the clusters should have already been removed by the most uncertainty query strategies (see Section 2.1) before the thresholds for WeakClust are met.

4 Experimental Setup

We first introduce the datasets used in our evaluation in Section 4.1. In Section 4.2, we discuss the parametrization of the clustering approaches. To evaluate the performance of WeakAL we conduct a large hyperparameter search on different real-world datasets, which is described in Section 4.3. Finally, in Section 4.4 we present the evaluation metrics used for our experiments. The code for all experiments is publicly available[1] under the AGPL-3.0 license.

4.1 Datasets

We perform our experiments using six real-world datasets described in Table 1. All datasets are used to train classification models and most of them contain noisy data, have missing values, sparse feature representation, and unbalanced class distributions. Except for DWTC, all datasets come from the *Active Learning Challenge* performed by Guyon et al. in 2010 [8]. In our experiments, 50 % of the data was withheld as a test set.

[1] https://github.com/jgonsior/weakal

Hyperparameter	Search Range
Query Selection	random, uncertainty least confidence, uncertainty max margin, uncertainty entropy
Cluster Selection	dummy, random, most uncertain least confidence, most uncertain max margin, most uncertain entropy
WeakClust?	Yes/No
WeakCert?	Yes/No
M, α, β, γ	$[0.5, 1.0]$

Table 2: Hyperparameter Search Space

4.2 Performed Clustering Strategies

As stated in Section 3, WeakAL expects the input data to be clustered. Since the underlying data characteristics for a dataset to be labeled are often not known, we decided for generally applicable clustering algorithms: For the large datasets, SYLVA and HIVA, we used Mini-batch k-Means [10] and Agglomerative Clustering [11] for the smaller ones. The parameter k, representing the number of clusters, is set to $n\_samples/8$ and the batch size to $min(n\_samples/100, n\_features)$. These parameters ensure an average number of 8 data points per cluster, which proved to work best in our experiments. For our use case, the high number of clusters is not a problem as long as their homogeneity is high. Note, that WeakAL does not depend on a specific cluster strategy, i.e. others can be used as well.

4.3 Hyperparameter Search

The quality of the used WS technique depends highly on the correct selection of the parameter values. We chose therefore an extensive random hyperparameter search to find optimal values and obtain an understanding of the sensitivity of the WS techniques regarding their parameters. Table 2 lists all the relevant hyperparameters. In total, 37,290 hyperparameters for the DWTC dataset have been tested, which was possible due to its smaller size and 4,922 hyperparameters for all other datasets.
We used a random forest [12] classifier with standard parameters in all experiments since it showed good results for every dataset and is comparatively fast. In addition to that, it has been reported that random forest classifiers are good at dealing with potentially noisy, weak labels [13].

4.4 Evaluation Metrics

To compare the results of an AL run we need to measure its effectiveness in achieving the overall goal of AL, to learn an accurate model with a minimum amount of labeling cost. A desirable metric for WeakAL takes into account a) amount of user-retrieved labels, b) classifier evaluation metrics, such as accuracy,

F1-Score or AUC, and c) an average of the classifier evaluation metrics throughout all AL iterations. The last two options are quite similar but have different objectives. The average is desirable, to not only compare AL runs where only the final iteration resulted in a high-quality run but also those, where no measurable quality drop occurred. As one normally does not know a priori when to optimally stop the AL process, one has to look at the average to not stop before the final "good" queries. As a direct result of this, the final accuracy is needed, as the average loses the information if the quality is good in the end or just in the beginning.
We determine two basic metrics that should be analyzed in conjunction for a meaningful evaluation: the ratio of weakly labeled data *% saved human effort hu* and the final *test accuracy acc_end*. For the saved human effort 0.0 equals zero savings and 1.0 is the optimal case where no human experts were needed for labeling at all. Besides, two compound metrics are calculated: The first one is called *combined score*, which is the harmonic mean of the two basic metrics:

$$combined\_score = \frac{2 * acc\_end * hu}{acc\_end + hu} \tag{6}$$

It captures the tradeoff between a desired low amount of saved human effort and high test accuracy.
To compare ourselves to the results of the Active Learning Challenge described in Section 4.1, we further report the *global score*, which was used in the challenge [8]. Note, that we compute the AUC values for the global score only for human experts' queries, where WS queries are considered "free".

5 Evaluation

In this section, we want to investigate the feasibility of WeakAL and show whether the integration of weak supervision techniques in the AL cycle has the potential to reduce the human labeling effort. We start in Section 5.1 by analyzing the impact of the human experts' query budget. In Section 5.2, we compare the effect of no WS, WeakCert, and WeakClust individually. Further, we combine both strategies, WeakCert and WeakClust, and report the results for the best working parameter combinations for the DWTC dataset. In Section 5.3, we show that the combination of AL and WS even can achieve higher accuracies than AL alone. In Section 5.4 we show on an example how the two WS strategies are applied in practice. The results on the datasets from the AL challenge are given in Section 5.5. In Section 5.6, we provide some rules of thumb for good hyperparameter values.

5.1 Budget size matters

As stated in Section 4.4 the used budget size has a direct impact on the evaluation metrics. Figure 2 plots the best-achieved accuracy for the DWTC dataset for budgets between 0 and 3,000. It can seen that the accuracies for smaller budgets

fluctuate a lot. We focus the following analysis therefore on larger budgets, due to stable and more reproducible results. The best result for the *combined score*, representing the balance of the tradeoff between a low amount of saved human effort and high test accuracy, is achieved for a budget of 260 human experts' queries, with an accuracy of 79.20%.

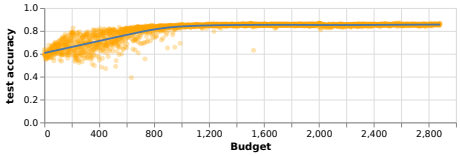


Fig. 2: Comparison of best-achieved test accuracy for different budgets for the DWTC dataset

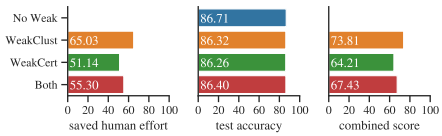


Fig. 3: Comparison of the best result for all possible WS combinations with a budget of 1,500, selected after test accuracy

5.2 Comparison of best-in-class AL + WS

In this analysis, we compare the results for an AL cycle without WS, with each of WeakClust and WeakCert on their own, and with a combination of both. Again we used the DWTC dataset for this experiment. As the saved human effort cannot be calculated, when no WS is being applied, the best results are selected based on the test accuracy, whereas the budget was kept to a fix value of 1,500. Figure 3 shows that the accuracy of WeakAL using a combination of both WS techniques is only 0.31%, WeakCert 0.45%, and WeakClust 1.39% lower than the AL cycle without WS. Hence, it can be concluded that application of WS techniques in WeakAL provides a significant saving of human effort, with a negligible reduction of the test accuracies. WeakCert and WeakClust in combination only achieve a slightly better accuracy than the individual techniques since both often label the same samples. While the savings of human effort are higher for WeakClust compared to WeakCert in this example, this is not true in general but highly depends on the budget.

5.3 General improvement using WS

So far we only compared the results for selected examples of good parameter combinations. In the following, we investigate the overall distribution of *all possible parameter combinations* for a fairly small budget of 200, due too limitations in compute time, for the DWTC dataset. Figure 4 shows three distributions: in blue all parameter combinations without using WS strategies, in orange all parameter combinations showing an accuracy improvement due to the WS-labels, and in green all parameter combinations using WS and showing a performance decline. The improvement was measured by comparing the test accuracy of a classifier trained on the human experts' queries alone, to a classifier trained on the human expert queries and the automatically generated WS-labels. In addition to the

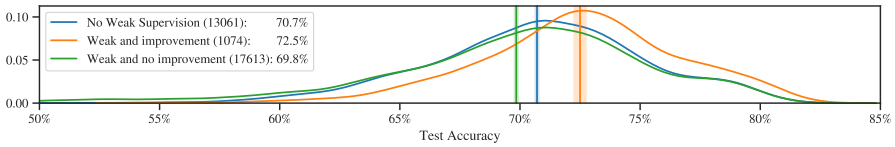
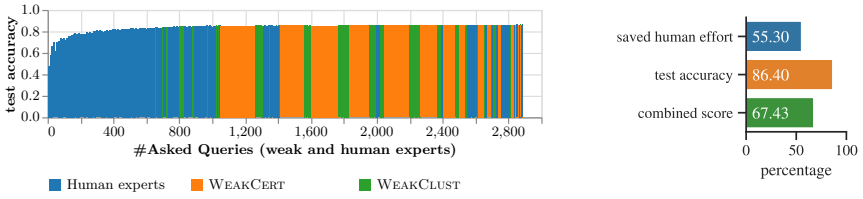


Fig. 4: Kernel density estimation and mean including 95% confidence interval given a budget of 200 samples for the DWTC dataset

kernel density estimations of the distribution, the mean value is shown including the 95% confidence interval. It can been seen that incorporating WS into AL even can improve the average test accuracy by 1.81%. There also exists, a large subset of parameters, which consistently achieve a lower accuracy using WS. Nevertheless, using WS directly within the AL cycle, with the right parameters, has the potential to not only lower the human effort drastically but also to even increase the accuracy.

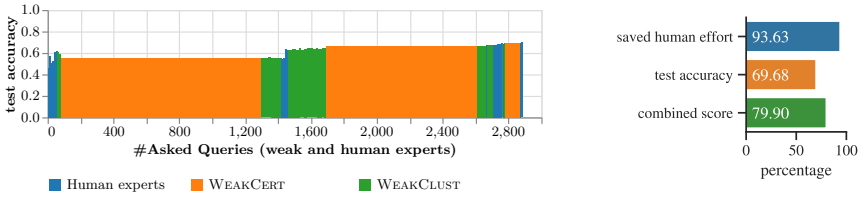
5.4 Detailed results for DWTC dataset

This section gives some deeper insights into how WeakCert and WeakClust work together in detail, shown exemplarily for the DWTC dataset. Figure 5 shows the best-achieved accuracy result for the DWTC dataset with a budget of 1,500 human experts' queries. Plot 5.a made up of colored rectangles, one for each iteration of the AL cycle. The width of a rectangle is the number of retrieved labels during the iteration, the height the achieved test accuracy. In the beginning, a lot of human experts' queries (blue) are requested, until WeakAL is confident enough to apply the WS techniques. From then on, WeakClust (orange), WeakCert (green), and the human expert queries alternate constantly. Most of the labels can be generated automatically by WS, without negatively influencing the accuracy. The alternation between WS and the human experts' queries shows, that it is indeed beneficial to apply WS during the AL cycle, and not after a gold standard is obtained.
In contrast, Figure 6.a displays the test accuracies for a significantly smaller



(a) Test accuracies (b) Metrics

Fig. 5: Highest achieved accuracy result for the DWTC with a budget of 1,500



(a) Test accuracies (b) Metrics

Fig. 6: Best-achieved test accuracy result for DWTC with a budget of 200

budget of only 200 human experts' queries. Here, most of the labels are generated by WeakCert. Interestingly the accuracy is dropping after the first big block of WeakCert labels, but rises quickly again after a few oracle queries. After the second smaller block of human expert queries at around 1,450, the accuracy goes even up purely based on WS labels. So without human interaction, the accuracy of the classifier can be improved, which shows that the effectiveness of WS goes further than just producing redundant labels.
The bar charts in Figure 5.b and Figure 6.b illustrate the results for different evaluation metrics. It is obvious, that smaller budget results in more saved human effort, accepting a loss of the test accuracy. The *combined score* metric shows, that the tradeoffs between the saved human effort and the test accuracy are worse for the bigger budget. This is not surprising, as it always takes much more data to further improve an already good accuracy than a poor one.

5.5 Active Learning Challenge datasets

To compare our results to other common AL strategies we selected the training datasets from the AL challenge [8]. The respective best results are shown in Figure 7. We used a budget of 1,000 and the global score of the AL challenge as the evaluation metric to select the best results. A budget of 500 was too small for most of the datasets and resulted in highly overfitted classifiers with reported test accuracies of under 1%. The figures show, that all AL challenge datasets have high values for all metrics. As the datasets all are highly imbalanced binary

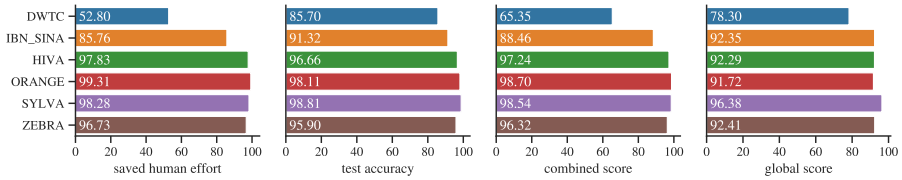
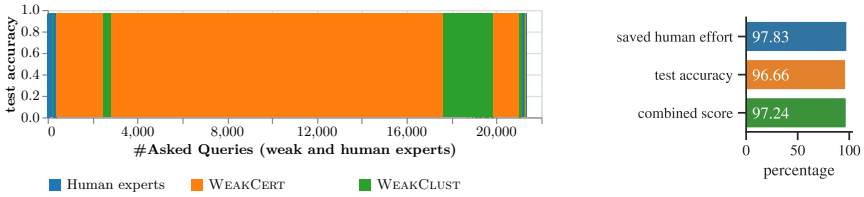


Fig. 7: Comparison of all analyzed datasets with a budget of 1,000 and the combination based on the global score



(a) Test accuracies (b) Metrics

Fig. 8: Best-achieved global score result for IBN_SINA with a budget of 1,000

decision problems, a vast amount of labels can be generated automatically by WS, as most samples are of the same label anyway. Since we have been not able to determine the budgets used in the AL challenge, a comparison of the results is only partly fair. Nevertheless, under the assumption, that a budget of 1,000 is close to the budget used in the competition, our achieved results are competitive to the winners of the AL challenge. Not all datasets are suited for WeakClust as the underlying data could not be clustered well. Clustering worked good for HIVA, IBN_SINA, and ZEBRA.
Figure 8 shows the results for the IBN_SINA dataset in detail. In the beginning, human experts' queries are being collected (blue bars). After that, almost all labels can be generated using the WeakCert (orange) and WeakClust (green). Both WS techniques alternate between each other, with few human experts' queries in between. Again, this is an argument for directly embedding WS into the AL process in WeakAL. The plots for the other datasets from the AL competition looked quite similar. We therefore based our evaluation primarily on the more interesting results for the DWTC dataset.

5.6 Recommended Parameters

Based on the investigation in Section 5.3, we would like to make recommendations which parameter combinations work well in practice: First, the best parameters depend a lot on the desired test accuracy. The higher the test accuracy, the more data is needed, and the higher the thresholds should be set. The minimum training accuracy M should be approximately 10% less than the desired test accuracy. For the query sampling strategies, *uncertainty max margin* performed best, closely followed by *uncertainty least confident*. The selected cluster strategy depends heavily on the quality of the underlying clusters and the amount of available data. For the several datasets, such as ZEBRA, which could be clustered well, and, therefore, WeakClust is applied often, *most uncertain least confident* works best, whereas for those where no meaningful clusters could be found, the dummy cluster strategy is leading. Good values for the threshold α are values between the desired test accuracy up to 1.0. The parameters for WeakClust, β, and γ, should be considered jointly. The lower the cluster homogeneity ratio β, the higher the minimum labeled cluster size γ should be. Good values for

both are between 0.75 and 0.95, keeping in mind the reverse dependency between both.

6 Related Work

Semi-supervised learning. There exist various techniques to combine the abundance of unlabeled data with labeled data in a classification setting, and the terminology about that is not always clear according to our experience. The most common term is semi-supervised learning, which uses unlabeled data to verify assumptions based on labeled data [14]. Semi-supervised learning also has been incorporated with AL, e.g. using Expectation-Maximization [15] or using multi-view co-training [16]. Adjacent to semi-supervised learning, weak supervision assumes that high-quality ground truth labels exist, and many noisy labels for the rest of the data. In our case, we produce high-quality data when querying the human experts, and noisy labels when using the WS. Following the terminology introduced in [17], we use the term weak supervision when talking about *inaccurate supervision*. We focus on the aspect of generating labels of weak-supervised learning, intending to reduce the amount of needed ground-truth labels.
Clustering. [18] proposes to query only the cluster centers in different feature spaces, and to use a majority vote afterward for the unlabeled data to determine their labels. In [19] graph-based clustering was directly incorporated into an AL setting. Other techniques, such as *label propagation* [20] iteratively propagate labels based on a small labeled ground truth set using a combination of random walk and clamping. Another approach is to use a small set of ground truth labels and program synthesis techniques to automatically generate labeling functions [21].

7 Conclusions

Annotating training data for supervised learning, such as classification, requires substantial human effort. While utilizing Active Learning during the annotation process already decreases the amount of human labor, we argue that AL should be combined with WS to further reduce the number of annotations made by human experts. Therefore, we proposed WeakAL, a WS extension to a typical AL cycle employing different cluster query strategies to query those samples, which further supports the WS strategies. In a comprehensive study, we selected and compared the proposed strategies as well as multiple parameter combinations. For a Web table classification task the results show that 55.30% of human labeling effort can be saved using automatic WS labels, with only a negligible loss of test accuracy by 0.31%. We showed, that with optimal parameters, a test accuracy improvement by 1.81% can be attributed solely to WS. We further applied WeakAL on datasets from the AL challenge from Guyon et al., where over 90% of the labels could be generated automatically, while still achieving competitive results, thus proving the general applicability of our proposed approach.

Acknowledgements This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) and the European Social Funds (ESF) within the "Innovations for Tomorrow's Production, Services, and Work" Program (funding number 02L18B561) and implemented by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the content of this publication.

References

1. Cohn, D., Atlas, L., Ladner, R.: Improving generalization with active learning. Machine Learning **15**(2), 201–221 (1994)
2. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648 (2010)
3. Lewis, D.D.: A sequential algorithm for training text classifiers: Corrigendum and additional data. SIGIR Forum **29**(2), 13–19 (1995)
4. Scheffer, T., Decomain, C., Wrobel, S.: Active hidden markov models for information extraction. pp. 309–318. IDA (2001)
5. Shannon, C.E.: A mathematical theory of communication. SIGMOBILE Mob. Comput. Commun. Rev. **5**(1), 3–55 (2001)
6. Baram, Y., El-Yaniv, R., Luz, K.: Online choice of active learning algorithms. JMLR **5**, 255–291 (2004)
7. Scudder, H.J.: Probability of error of some adaptive pattern-recognition machines. IEEE Transactions on Information Theory **11**, 363–371 (1965)
8. Guyon, I., Cawley, G., Dror, G., Lemaire, V.: Results of the active learning challenge. JMLR **16**, 19–45 (2011)
9. Eberius, J., Braunschweig, K., Hentsch, M., Thiele, M., Ahmadov, A., Lehner, W.: Building the dresden web table corpus: A classification approach. In: BDC. pp. 41–50. IEEE (2015)
10. Sculley, D.: Web-scale k-means clustering. In: WWW. pp. 1177–1178 (2010)
11. Jr., J.H.W.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association **58**(301), 236–244 (1963)
12. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001)
13. Folleco, A., Khoshgoftaar, T., Van Hulse, J., Napolitano, A.: Identifying learners robust to low quality data. Informatica (Slovenia) **33**, 245–259 (2009)
14. Zhu, X.: Semi-supervised learning literature survey. Comput Sci, University of Wisconsin-Madison **2** (2008)
15. McCallum, A., Nigam, K.: Employing em and pool-based active learning for text classification. p. 350–358. ICML (1998)
16. Muslea, I., Minton, S.N., Knoblock, C.A.: Active + semi-supervised learning = robust multi-view learning. ICML (2002)
17. Zhou, Z.H.: A brief introduction to weakly supervised learning. National Science Review **5**(1), 44–53 (2017)
18. Dara, R., Kremer, S., Stacey, D.: Clustering unlabeled data with soms improves classification of labeled real-world data. vol. 3, pp. 2237 – 2242 (2002)
19. Bodó, Z., Minier, Z., Csató, L.: Active learning with clustering. Active Learning and Experimental Design workshop@AISTATS, vol. 16, pp. 127–139 (2011)
20. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Tech. rep. (2002)
21. Varma, P., Ré, C.: Snuba: Automating weak supervision to label training data. VLDB **12**(3), 223–236 (2018)