Multi-AL: Robust Active learning for Multi-label Classifier

M.J.Basting[1] , **T.Younesian**[1] , **A.Ghiassi**[1] , **L.Chen**[1]
[1]TU Delft

Abstract

Multi-label learning is becoming more and more important as real-world data often contains multiple labels. The dataset used for learning such a classifier is of great importance. Acquiring a correctly labelled dataset is however a difficult task. Active learning is a method which can, given a noisy dataset, identify important instances for an expert to label. This greatly reduces the amount of instances needed to train an accurate classifier, and thus reduces the cost of cleaning a noisy dataset. Therefore, this paper aims to present an active learning algorithm, focused on wrongly labeled data, combined with a deep neural network for multi-label image classification. The proposed active learning solution is divided into two measures; a mislabelling likelihood and an informativeness measure together with an option to identify and use highly probable clean instances in the dataset. Experiments performed on the real world dataset, called Microsoft COCO, with 20, 40 and 60% injected label noise show that Multi-AL outperforms the current state-of-the-art multi-label learning algorithm called ASL by 28% while only using 600 labelled instances in total and 250 extracted 'clean' instances. Multi-AL additionally outperforms random sampling by 3% on average for 20 and 40% random label noise when sampling from a wrongly labelled dataset of 23k instances.

1 Introduction

Multi-Label learning, where each instance contains multiple labels, has become more and more important as real world data often contains multiple labels. A key difference and difficulty that arises when dealing with multi-label data is the explosion of possible label combinations [1]. Wrong labels present in the dataset used as ground-truth result in a longer time-to-convergence and an almost surely decrease in robustness [2].

A correctly labeled dataset is of great importance when training a multi-label deep neural network. The acquisition of a dataset which accurately represents real world data and does not contain noise is expensive and time consuming [3]. Acquiring data that does contain noisy labels is often cheap and simple to acquire with crowdsourcing methods such as Amazon Mechanical Turk [4]. Active Learning is a method in machine learning which, given a pool of unlabeled or wrongly labeled instances, can iteratively identify mislabelled instances and select the most informative examples to query an expert for its true label [5]. Therefore, reducing the amount of queries needed to achieve a high accuracy classifier while also limiting the cost of the expert labeller.

Even though multiple active learning solutions exist, there are limited active learning algorithms available for multi-label learning domains and even less that focus on finding mislabelled instances in the training data. The methods discussed in [6], [7] focus on finding the potentially mislabeled instances by calculating the uncertainty of the classifier, additionally [7] calculates an informativeness measure which limits the amount of queries needed even more. However, previous methods all focus on single-labeled data. A multi-label solution is discussed in [8]. This method focuses on the labels of the instances with the most similar features but does not deal with an expert labeller. Multi-label methods discussed in [9]–[12] all discuss potential ways to limit the amount of queries needed yet do not focus on finding the wrongly labeled instances in the dataset.

Therefore, this paper aims to solve the problem of multi-label learning with wrong label noise. The proposed solution is an active learning algorithm called Multi-label Active Learning (Multi-AL). Multi-AL deals with training data that contain wrong labels, that is, the 'true' labels of some images have been switched out with wrong labels. An illustration of this can be seen in Figure 1.

Figure 1: Illustration of data with wrong labels

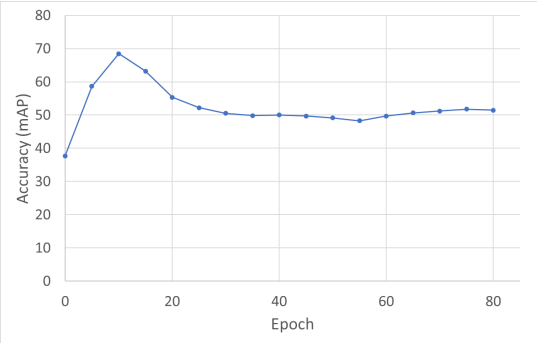


Figure 2: Accuracy ASL with 40% random label noise

Multi-AL is build on a state-of-the-art multi-label classifier that consists of a TresNet [13] backbone with an asymmetric loss function from [14] which we will call ASL from now on. ASL achieves high accuracy when trained on multi-label data that is accurately labelled, however, the performance quickly degrades if the dataset contains random label noise, as can be seen in Figure 2. Multi-AL is divided into two measures where one is used to calculate the mislabelling likelihood of an instance and the other to identify the most informative instances to relabel. The mislabeling likelihood is calculated based on the conflicts between the predicted labels and the labels present in the noisy dataset using the output probability of the deep neural network. This mislabelling likelihood is then used to identify highly probable safe and mislabelled instances in the dataset. Two separate informativeness measures were designed and evaluated, a Conflict-Based Informativeness Measure (CBIM) and an uncertainty-based measure. CBIM, used as a baseline, only calculates a value for the potential mislabelled instances and focuses on the contribution of features towards the predicted label and the potentially wrong label. The uncertainty-based measure focuses on the entire dataset and calculates the uncertainty entropy of every instances. The informativeness measure used in Multi-AL is the uncertainty entropy based on better performance for all levels of noise.

Evaluation performed on the Microsoft COCO dataset [3] for 20, 40 and 60% random label noise indicate that on average Multi-AL outperforms current state-of-the-art multi-label learning algorithm called ASL by 28% while only using 600 labelled instances in total. Multi-AL also shows a 3% increase of accuracy on average for 20 and 40% label noise when compared to random sampling.

The contributions of this paper are as follows:

- A multi-label mislabelling likelihood based on the conflict between predictions of the network and the ground truth.
- An instance based usefulness measure based on the contribution of features towards the predicted and ground truth labels.
- An instance based uncertainty measure that calculates the uncertainty of the classifier.
- An experiment that showcases the effect of using Multi-AL on Microsoft COCO for different levels of random label noise.

This paper is organized as follows. In section 2, we start by discussing the related work. In section 3, an active learning algorithm is proposed which first identifies clean and wrong instances in the dataset using a mislabelling likelihood and then identifies the most important instances to relabel with an informativeness measure. In section 4 an empirical analysis is conducted on the mislabelling likelihood. In section 5 we present the experimental setup and the results of the experiment and discuss the achieved performance of the different methods. In section 6, the ethical implications and reproducibility of the research are discussed. Finally, in section 7, we present the future work and conclusion of the research.

2 Related Work

In this section multiple different types of active learning strategies will be discussed. Ranging from single-label methods to multi-label, instance based to instance-label based methods.

Multi-Class Active Learning

In [6], [7], possible mislabelling likelihood measures are discussed that calculate the uncertainty of the classifier for an instance or instance-label pair. Either the uncertainty is calculated based on the classifier's uncertainty and label uncertainty [6] or based on the contradicting label the classifier outputs and the label in the ground truth [7]. Both these papers however focus on multi-class classification which only deals with single-labeled data.

Active Learning with Label Correlation

A different approach to active learning is to focus on the label correlation [12]. Wu et al. [12] proposes a multi-label active learning algorithm which utilizes label correlations to construct a unified sampling strategy and evaluates the informativeness of each label-pair. A down-fall of this method however is that for each label-pair four classifiers need to be trained.

Feature-based Active Learning

Mikalsen et al. [8] proposes a multi-label dimensionality reduction method which uses the features of instances to construct a neighbourhood graph. The sampling method uses this neighbourhood graph to extract the nearest neighbours and compares its features and labels. This method uses label propagation to label instances instead of a reliable labeller and therefore does not deal with the problem of limited queries.

Impact Based Active Learning

The final active learning algorithms that will be discussed are the strategies that focus on instances that have the most impact on the classifier [9]–[11]. Li, Wang and Sung [10] proposed two selection strategies: Max Loss and Mean Max Loss. The max loss strategy in contrast to the mean max loss strategy only focuses on the most certainly predicted class of the image and ignores the other labels. Both strategies query the instances to relabel which contribute the most to the loss function. Kremer, Sha and Igel [15] proposes to

use noise-aware loss functions to increase the influence of noisy examples. Furthermore, the paper adopts a maximum expected model change strategy. Impact based algorithms are essential in designing a good active learning method.

Previously discussed method mostly focus on identify the most informative instances to label from a selection of unlabeled data. The problem we are trying to answer however has to deal with data with wrong labels. This paper aims to combine a mislabeling likelihood together with an informativeness measure to identify which multi-label instances to relabel. The resulting active learning algorithms should additionally deal with the problem of limiting the amount of queries to the expert while maximizing the performance.

3 Active Learning Multi-label Algorithm

This section introduces formal problem statement, explains the pipeline and architecture of Multi-AL in combination with ASL and explains the designed mislabelling and informativeness measures in more detail.

3.1 Problem Statement

Multi-label classification is similar to multi-class classification, but instead of classifying an instance into a single class the instance can belong to multiple classes (or labels). Assume that $\mathbf{x}$ of size K represents an image with K pixels, Y represents a set of multiple labels of a certain maximum size L. Given a data-set $D = \{(\mathbf{x}_i, Y_i) \mid 1 \leq i \leq N\}$, the goal of multi-label classification is to map images $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}\}$ of K pixels into a set of multiple labels Y of maximum L labels, $\mathbf{x_i} \in \mathbf{X}^{N \times K}$ into $Y_i \subseteq Y = \{y^1, y^2, ..., y^L\}$.

The problem that is discussed here is more specific. The assumption is made that the given data-set D has wrong labels, i.e. a fraction labels of a subset of samples are swapped from its true label. The goal of the classifier is to train an accurate classifier with as little queries to the expert labeller as possible. The amount of queries that are available is called the query budget.

The overall method is as follows. First, identify the mislabeled instances and then choose the most informative instances to relabel by the expert with a limited budget. After training of the multi-label classifier, an evaluation is performed on a test set. In the following section, the calculation for the mislabelling likelihood and different informativeness measures are discussed. But first, the overall method will be discussed in more detail together with the integration into the current state-of-the-art multi-label classifier called ASL [14].

3.2 ASL and Active Learning

Both the mislabelling likelihood and the informativeness measure are combined into one active learning sampling method that needs a classifier prediction. The classifier used is the deep neural network from [14].

Each iteration of Multi-AL is split up into three phases. The pipeline can be seen in Figure 3.

First, the mislabelling analysis phase, the mislabelling likelihood of all instances in the dataset is calculated and the instances with a mislabelling score lower than t_1 are saved as the 'clean' dataset X_c. Instances with a score higher than a certain threshold are marked as mislabeled.

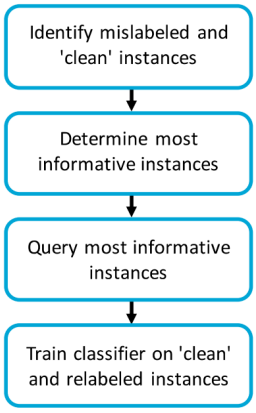


Figure 3: Pipeline of a training round

Second, the selection phase, which starts with evaluating the informativeness for each instance. Subsequently, the k most informative instances are selected for relabeling, and an increasing portion of instances per iteration is selected from X_c.

Finally, the training phase during which both the relabeled instances and the selected 'clean' instances are used to train the qualifier.

3.3 Mislabelling Likelihood

The mislabelling likelihood is used to identify possibly wrongly labelled instances and identify the highly probable clean instances in the dataset. The underlying idea is that the difference in the probability of the predicted labels and observed labels in the training data is high when an instance is mislabeled. The measure used is a slightly adapted version of the measure used in [7].

First, assume Y_t^x are the labels present in the training data for instance $\mathbf{x}$ and Y_p^x are the respective predicted labels where $|Y_t^x| = |Y_p^x| = M$.

$$Y_p^x = \underset{y \in Y', Y' \subset Y, |Y'|=M}{\operatorname{argmax}} \Big(P(y|x) \Big) \tag{1}$$

Thus, Y_p^x denotes the M highest probable labels for instance $\mathbf{x}$. $Y_c^x = Y_p^x \cap Y_t^x$ is used to denote the set of correct labels. Since we only want to calculate the mislabeling likelihood for labels that are wrong, and not for correct labels, we only look at the conflicting labels. This means that we only look at the predicted labels $Y_p^x - Y_c^x$ and compare them to labels present in the training data $Y_t^x - Y_c^x$. From now on, we denote these sets as YC_p^x and YC_t^x respectively. The mislabelling likelihood in [7] is a single-label approach. We define the multi-label mislabelling likelihood as visible in Formula 2. For each possible label pair combination that can be constructed from set YC_p^x and YC_t^x we calculate the mislabelling value and the maximum value encountered is given to instance $\mathbf{x}$. An example of the construction of the label pair combinations can be found in Figure 4.

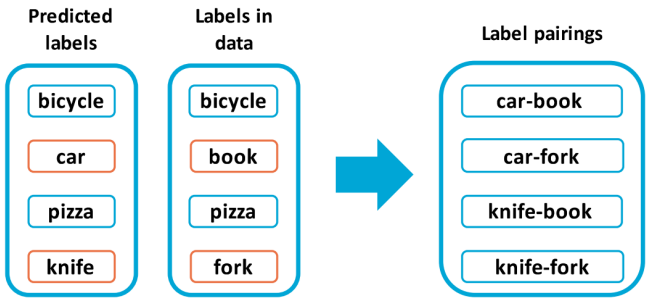


Figure 4: Construction of label pairings

Instances are identified as mislabeled when $D_1(x) > t_1$. The resulting label-pairings that satisfy this threshold are saved for later evaluation together with their score.

$$D_1(x) = \max_{y_p \in YC_p^x, y_t \in YC_t^x} \left(\frac{P(y_p|x) - P(y_t|x)}{\max(P(y_p|x), P(y_t|x))} \right) \quad (2)$$

3.4 Informativeness Measure

In the previous section, the possible mislabeled instances and clean instances are identified. The next step is to select the most informative instances to relabel. The goal of the informativeness measure is to limit the amount of queries to the expert while maximizing the usefulness of the queried instances. Two informativeness measure will be explained; a conflict-based informativeness measure (CBIM) which calculates the effect of each feature present in the neural network on the predicted labels and then uses the informativeness measure from [7], and the uncertainty entropy.

Conflict-Based Informativeness Measure (CBIM)

The Conflict-Based Informativeness Measure (CBIM) is a usefulness measure and the measure is an adaptation from [7]. The measure focuses on the amount of conflicting information between the features.

CBIM is originally designed for single-label data and uses a Support-Vector-Machine as base classifier. When different features can equally attract a certain instance to the predicted label and the label present in the dataset, we say that the instance has conflicting information. According to [7], instances which contain stronger conflicting information are more informative to relabel.

We define the contribution of features, extracted from the DNN, as follows. First, the weights w and output of the second-to-last layer $z^1(x)$ for instance $\mathbf{x}$ are extracted from the network, see Figure 5 for a clearer definition of the weights and outputs. f_i is used to represent a specific feature where $f_i \in F, |F| = \lambda$. Then, a feature contribution matrix FC of all features is calculated using the dot-product between the weight matrix w and output of the second-to-last-layer $z_1(x)$ as can be seen in Formula 3. We define column i as the contribution of feature f_i towards all possible labels. Each row represents how much each feature contributes to classifying an instance in class y_j. Thus column i, row j represents how much feature f_i contributes to classifying instance $\mathbf{x}$ into the label y_j.

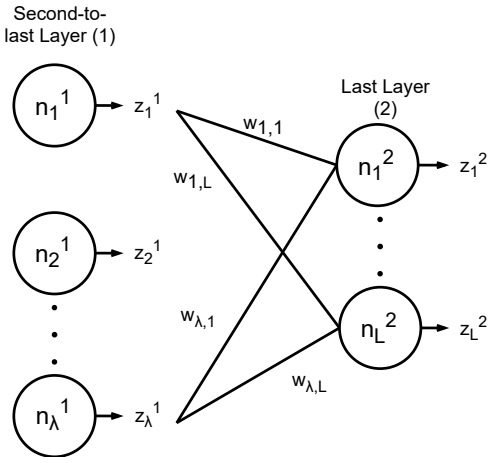


Figure 5: Weights and outputs last two layers of TResNet-m network [13]

$$FC = \begin{bmatrix} w_{1,1} & w_{2,1} & \dots & w_{\lambda,1} \\ w_{1,2} & w_{2,2} & \dots & w_{\lambda,2} \\ \vdots & \vdots & \vdots & \vdots \\ w_{1,L} & w_{2,L} & \dots & w_{\lambda,L} \end{bmatrix} \cdot \begin{bmatrix} z_1{}^1 \\ z_2{}^1 \\ \vdots \\ z_\lambda{}^1 \end{bmatrix}$$
$$FC = \begin{bmatrix} w_{1,1}z_1{}^1 & w_{2,1}z_2{}^1 & \dots & w_{\lambda,1}z_\lambda{}^1 \\ w_{1,2}z_1{}^1 & w_{2,2}z_2{}^1 & \dots & w_{\lambda,2}z_\lambda{}^1 \\ \vdots & \vdots & \vdots & \vdots \\ w_{1,L}z_1{}^1 & w_{2,L}z_2{}^1 & \dots & w_{\lambda,L}z_\lambda{}^1 \end{bmatrix} \quad (3)$$

From this matrix, we divide the features into two sets F_p and F_t to denote the sets of features that contribute more to the predicted label p and training label t respectively, see Formula 4. As can be seen in Formula 5, q_p and q_t are the summed difference in contribution values of features in F_p and F_t respectively.

$$F_p = \{f_i | w_{i,p} z_i{}^1 > w_{i,t} z_i{}^1\}$$
$$F_t = \{f_i | w_{i,p} z_i{}^1 < w_{i,t} z_i{}^1\} \quad (4)$$

$$q_p = \sum_{f_i \in F_p} (w_{i,p} z_i{}^1 - w_{i,t} z_i{}^1)$$
$$q_t = \sum_{f_i \in F_t} (w_{i,t} z_i{}^1 - w_{i,p} z_i{}^1) \quad (5)$$

The usefulness can then be calculated using the original Formula 6 as in [7].

$$I = \sqrt{q_p \times q_t} \quad (6)$$

The focus in this paper however is on multi-label instances. CBIM calculates the informativeness measure I for each saved label pairing, constructed from the sets YC_p^x and YC_t^x as in Figure 4, and outputs the maximum value encountered. Formula 7 shows the final defined formula for CBIM.

$$CBIM(x) = \max_{y_p \in YC_p^x, y_t \in YC_t^x} \sqrt{q_p \times q_t} \quad (7)$$

Uncertainty Entropy

Another possible solution to identify informativeness instances is to query those instances that the classifier is least certain about.

The decimal probabilities for each label can be calculated using a Sigmoid function from the output of the neural network. These probabilities can then be used to calculate the entropy of each individual label $y_i \in Y$ as can be seen in Formula 8.

$$H(x) = -(P(y_i|x) * log_2(P(y_i|x)) \\ +(1 - P(y_i|x)) * log_2(1 - P(y_i|x))) \tag{8}$$

We calculate the total entropy as the individual entropy of each class summed together. This value however will not be between $[0, 1]$ and therefore needs to be divided by the amount of labels present in the data. The full formula can be seen in Formula 9.

$$H(x) = -\sum_{y_i \in Y} (P(y_i|x) * log_2(P(y_i|x)) \\ +(1 - P(y_i|x)) * log_2(1 - P(y_i|x)))/L \tag{9}$$

4 Empirical analysis

In this section, the design choice for the used mislabelling likelihood will be discussed and substantiated.

Two mislabelling measures can be found in [7] called D1 and D2 respectively. D1 calculates a value based on the predicted label of the classifier (y_p) and the label present in the data (y_t), while D2 is calculated from the feature values and the contribution of these features towards the predicted label and the ground-truth label. q_p and q_t reflect the summed difference in contribution values of dominant features for label y_p and y_t. A more detailed mathematical explanation on how these values are calculated can be found in Section 3.4.

$$D_1(x) = \frac{P(y_p|x) - P(y_t|x)}{\max(P(y_p|x), P(y_t|x))} \tag{10}$$

$$D_2(x) = \frac{q_p - q_t}{max(q_p, q_t)} \tag{11}$$

Both D1 and D2 are designed for single-label data but can be slightly modified to work for multi-label data by analysing the D_1 and D_2 for all possible label-pairings for an instance. Both D_1 and D_2 have a range between 0 and 1, where a lower value indicates a lower probability of an instance being mislabeled.

An experiment was performed to analyse the actual performance of both measures. Firstly, an initial classifier is trained on 100 instances for 20 epochs. Secondly, the mislabelling value for all instances in a noisy dataset is calculated. And lastly, all instances are queried by the expert and the ratio of true clean instances to wrong instances for 0.1 size intervals is calculated and can be seen in Table 1 and Table 11.

Table 2 shows that it can better identify between true clean instances and instances with wrong labels, however when looking at the actual number of clean instances, in Table 3, it can be seen that the true clean instances are more spread out for mislabelling measure D2.

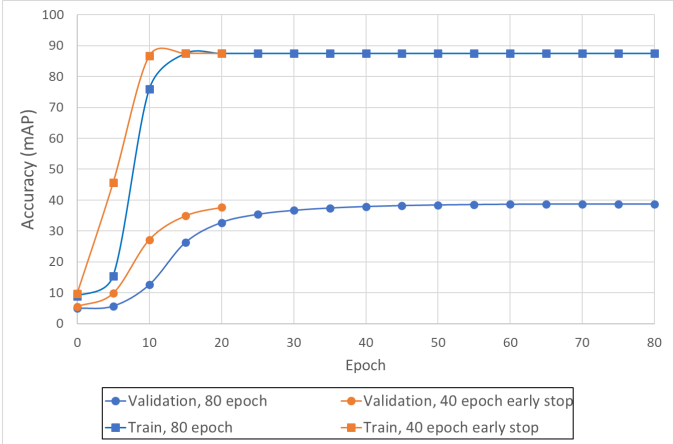


Figure 6: Validation and training accuracy of initial classifier for different training lengths

5 Results

In this section, the experimental setup is discussed in detail and the results of the performed experiment are presented and discussed. .

5.1 Experimental Setup

Experiments are conducted on a portion of the Microsoft COCO [3] dataset. This dataset contains 83k instances and 80 labels. 40% of the samples were used to sample from (23497 instances). The instances with only 1 label are removed, and all images are resized to be 224x224 Pixels. The Mean Average Precision is used to evaluate the accuracy obtained on the validation set.

Initial classifier. A small labelled dataset of 100 instances is used to train an initial classifier with, as active learning relies on a base classifier. The base classifier is a Convolutional Neural Network architecture called TResNet [13], which uses an Asymmetric Loss Function [14]. The specific TResNet architecture is called TresNet-m and a model[1] pretrained on the ImageNet dataset[16] is used to reduce training time needed significantly. The initial classifier is trained for 40 epochs with an early stop at epoch 20. Training for more epochs leads to no large increase of validation and training accuracy as can be seen in Figure 6.

Noise ratios. The sampling set was injected with 20, 40 and 60 % randomized label noise. This means that a certain percentage of labels are swapped with an irrelevant label. The initial training set and the test set are not subject to label noise.

Training procedure. After training the initial classifier, the classifier is trained using the selected active learning method. Each active learning iteration consists of a selection phase, during which 50 instances are relabelled and added to the already labelled dataset. If Safe mode enabled for any of the evaluated sampling methods, 25 instances from the identified set of 'clean' instances X_c are selected for additional

[1] https://miil-public-eu.oss-eu-central-1.aliyuncs.com/model-zoo/ASL/MS_COCO_TRresNet_M_224_81.8.pth

Table 1: Ratio of true clean instances to wrong instances for different intervals using mislabelling measure D1

Noise	Intervals of mislabelling value									
	[0, 0.1)	[0.1, 0.2)	[0.2, 0.3)	[0.3, 0.4)	[0.4, 0.5)	[0.5, 0.6)	[0.6, 0.7)	[0.7, 0.8)	[0.8, 0.9)	[0.9-1.0]
40 %	8.02	2.56	1.85	1.00	0.56	0.29	0.16	0.08	0.05	0.02
60 %	3.05	1.79	0.79	0.51	0.2	0.12	0.05	0.02	0.01	0.02

Table 2: Ratio of true clean instances to wrong instances for different intervals using mislabelling measure D2

Noise	Intervals of mislabelling value									
	[0, 0.1)	[0.1, 0.2)	[0.2, 0.3)	[0.3, 0.4)	[0.4, 0.5)	[0.5, 0.6)	[0.6, 0.7)	[0.7, 0.8)	[0.8, 0.9)	[0.9-1.0]
40 %	8.7	0.00	0.00	0.00	0.00	35.00	1.63	0.40	0.13	0.04
60 %	4.65	0.00	0.00	0.00	0.00	4.0	0.91	0.16	0.03	0.01

Table 3: Number of clean instances for different intervals using mislabelling measure D1 and D2

Noise	Mislabelling measure	Intervals of mislabelling value									
		[0, 0.1)	[0.1, 0.2)	[0.2, 0.3)	[0.3, 0.4)	[0.4, 0.5)	[0.5, 0.6)	[0.6, 0.7)	[0.7, 0.8)	[0.8, 0.9)	[0.9-1.0]
40 %	D1	449	295	494	713	944	972	842	423	116	6
	D2	261	0	0	0	0	35	1327	2358	1189	85
60 %	D1	183	129	175	302	322	375	303	152	42	7
	D2	107	0	0	0	0	16	534	863	435	35

training during the iteration. Every iteration, the model is trained for 20 epochs. Thus, the model will be trained for 200 epochs and in the final iteration, 600 instances labelled by the expert would be used, and 250 instances with the lowest mislabelling value if Safe is enabled. The accuracy of the model is evaluated every 5 epochs on a validation set of 12k instances.

Deep learning hyperparameters. We use a weight decay of 10^{-4} and each iteration a scheduler is used to determine the learning rate which lies between 10^{-4} and 10^{-6}. The training phase is done in mini-batches of 32 instances. The Adam optimization algorithm [17] is used as optimizer.

System specifications. The experiments were performed using two google cloud platform virtual machine instances connected to a personal computer using an SSH connection. Both virtual machine instances are equipped with of 8 vCPUs (SkyLake), 30 GB RAM, 60 GB disk space with a Nvidia Tesla T4 GPU and running Ubuntu (20.04).

Alternative baselines. We compare Multi-AL with the following approaches:

- Barebone ASL; all instances are used to train the classifier on and no sampling method is applied.
- Random sampling; randomly select instances each iteration to be relabeled.

Furthermore, random sampling is evaluated with Safe mode enabled, which means that a portion of instances from X_c is used during training and Multi-AL is evaluated with Safe mode disabled.

5.2 Evaluation

In this section, the performance evaluation of the different sampling methods for different levels of noise will be discussed. These experiments were performed using the experimental setup discussed previously.

Figure 7a, Figure 8a, Figure 9a show the validation accuracy during each active learning iteration. Figure 7b, 8b and Figure 9b show the validation accuracy after the final sampling iteration.

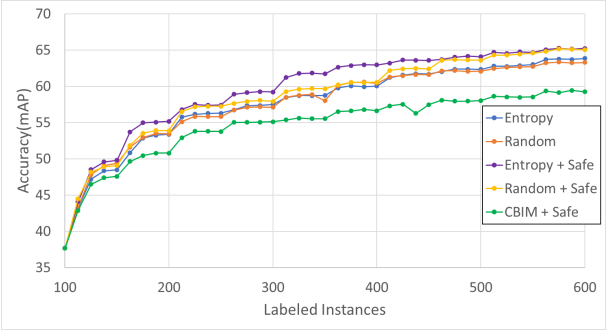
mAP difference using Safe mode. The results for Figure 7a and 8a confirm that Safe mode indeed improves the the achieved validation accuracy. An explanation for this could be that the number of instances used from X_c contain a sufficient number of actual clean instances so the classifier is not negatively influenced. Figure 9a shows that using 'clean' instances for 60 % label noise still improves the accuracy but not as significantly as for lower levels of noise. This further supports this theory.

Difference across approaches. Results from Figure 7, 8 and 9 show only a slight increase of performance when selecting instances based on uncertainty entropy compared to random sampling. Figure 7, 8 and 9 additionally show a significant drop in performance when using the informativeness measure that focuses on calculating the usefulness of mislabeled instances. This achieved result is different from the work it has been based on [7]. A possible explanation might be the difference in setup, as [7] only focuses on single-label data and combines the used informativeness measure with a weighting setup.

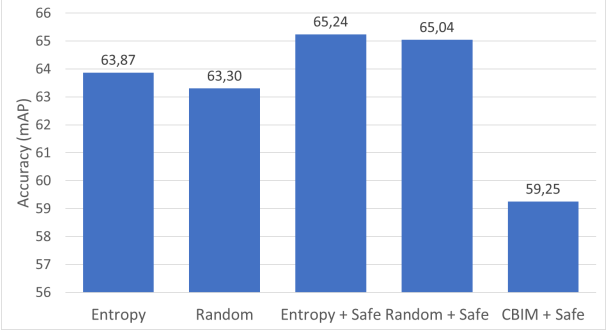
Comparison ASL and Multi-AL In Figure 10 the performance of barebone ASL and Multi-AL are compared for 20, 40 and 60% noise. The results indicate that indeed Multi-AL outperforms ASL significantly for all noise levels even though ASL uses a much larger dataset. 23k instances used vs 100 initial correctly labelled instances, 500 additionally labelled instances and 250 extracted 'clean' instances.

6 Responsible Research

In this section, the ethical implications and reproducibility of the research are discussed. This section focuses on the ethical implications that arise when performing the evaluation and creating a new multi-label deep learning active learning algorithm.

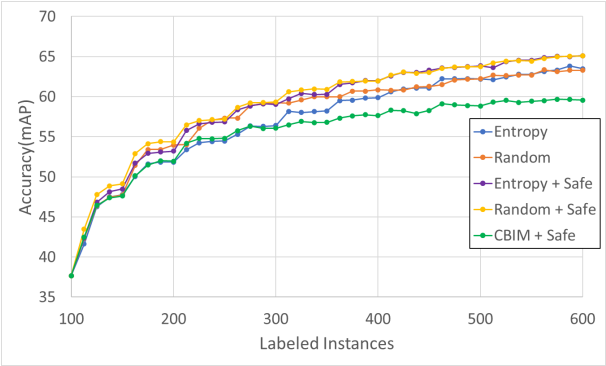


(a) Validation accuracy according to the number of labeled instances

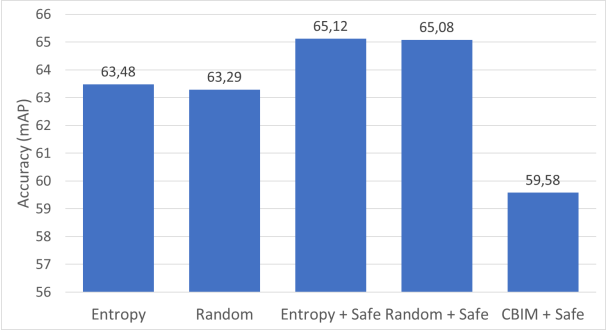


(b) Final Validation Accuracy after final sampling iteration

Figure 7: Reported Accuracy for 20 % label noise according to different relabeling strategies

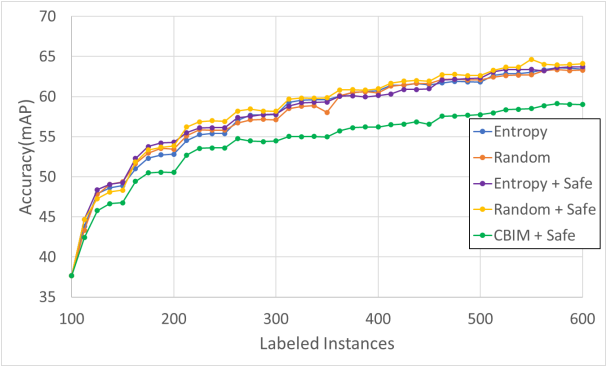


(a) Validation accuracy according to the number of labeled instances

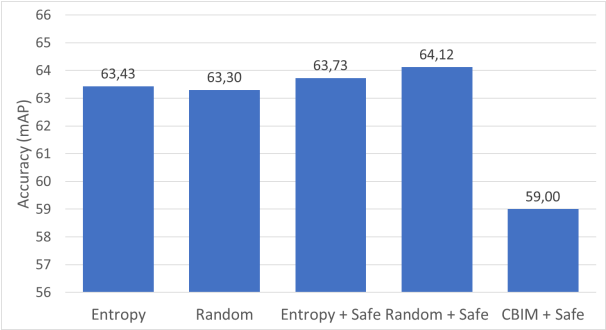


(b) Final Validation Accuracy after final sampling iteration

Figure 8: Reported Accuracy for 40 % label noise according to different relabeling strategies



(a) Validation accuracy according to the number of labeled instances



(b) Final Validation Accuracy after final sampling iteration

Figure 9: Reported Accuracy for 60 % label noise according to different relabeling strategies

6.1 Ethical Implications

When performing an evaluation of any sort it is important to handle result data in a correct way. Leaving results out it is allowed but not without a justifiable explanation. This type of data manipulation is called data trimming and is also a relevant topic in this research. Numerous experiments were performed and not all results obtained proved to display helpful additional information. Thus, the choice has been made to left them out.

One more important ethical implication to keep in mind is data integrity, especially when training a classifier. Data integrity is key to training a high performance classifier, as

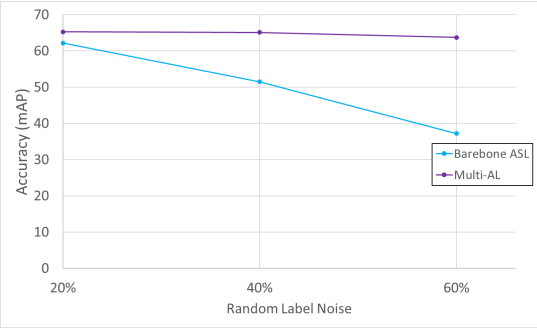


Figure 10: Accuracy barebone ASL compared to Multi-AL for different random label noise levels

a deep neural network will learn from the dataset that it has been given. The main data-set used in this research is the MS COCO dataset. This dataset contains many instances that are not labeled at all or only contain a single label. The choice was made to leave these instances out as the goal is to train a multi-label classifier, furthermore, a single-label instance cannot be given a wrong label without also containing a correct label. The MSCOCO dataset contains around 83k instances, the evaluation system used however could not in reasonable time train a classifier using the entire dataset. That is why only 40 % of the dataset was used.

6.2 Reproducibility

Reproducibility is a key principle to keep in mind when performing research or designing a system of any sort. Especially for research in the deep learning field, reproducibility is a great challenge.

Henderson et al [18] discuss multiple factors that influence the difficulty of reproducing the same results. To minimize the difficulty of reproducing the results, presented in the previous section, the used network and hyper-parameters of the neural network were explained as detailed as possible.

In addition, the neural network and asymmetric loss function ASL [14] are not changed in the process and are open-source. Further reducement of the randomness that occurs when training a deep neural network was achieved by performing each evaluation two times and taking the average of the results.

7 Conclusions and Future Work

The aim of this report was to present an solution to the problem of multi-label learning with wrong label noise. To achieve this, an active learning algorithm was designed which consists of two measures; a mislabelling measure which can accurately identify clean and mislabelled instances, and, two informativeness measures called the Conflict-Based Informativeness measure and the uncertainty sampling. Additionally, a Safe mode was designed which samples instances from the identified clean instances to use during training. The uncertainty entropy outperformed CBIM for all experiments thus forms the basis of the Multi-AL algorithm. Additionally, a portion of the identified clean instances was used throughout training.

The results of the conducted experiments show that Multi-AL outperforms barebone ASL significantly for 20, 40 and 60% label noise while only using 600 labelled instances in total. Random sampling was outperformed on average by 2% and even more for 20% and 40% random label noise.

The informativeness measures were evaluated and compared, however, the informativeness measure that focused on the conflicting information between the feature, called the usefulness measure, did not performed as expected. Further work is needed to discover the reason behind this decrease in performance. A first step would be to look at the single-label case again or evaluate the method using additional weights. Multiple experiments were performed to analyze the performance but only one dataset was used to experiment on. Further work is necessary to find out the different environments in which Multi-Label AL might not perform well. In the performed experiments 250 'clean' instances were used in the final iteration of Multi-Label AL for all levels of noise. A more thorough analysis needs to be performed to determine if a different number of 'clean' instances leads to significantly better performance for different noise levels.

References

[1] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5172–5181. DOI: 10.1109/CVPR.2019.00532.

[2] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, and S. Lacoste-Julien, *A closer look at memorization in deep networks*, 2017. arXiv: 1706.05394 [stat.ML].

[3] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, *Microsoft coco: Common objects in context*, 2015. arXiv: 1405.0312 [cs.CV].

[4] A. Sorokin and D. Forsyth, "Utility data annotation with amazon mechanical turk," vol. 51, Jul. 2008, pp. 1–8, ISBN: 978-1-4244-2339-2. DOI: 10.1109/CVPRW.2008.4562953.

[5] D. Cohn, "Active learning," in *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010, pp. 10–14, ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_6. [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_6.

[6] C. H. Lin, Mausam, and D. S. Weld, "Re-active learning: Active learning with relabeling," in *AAAI*, 2016, pp. 1845–1852. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12500.

[7] M.-R. Bouguelia, Y. Belaïd, and A. Belaïd, "Stream-based active learning in the presence of label noise," in *Proceedings of the International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM,*, INSTICC, SciTePress, 2015, pp. 25–34, ISBN: 978-989-758-076-5. DOI: 10.5220/0005178900250034.

[8] K. Ø. Mikalsen, C. Soguero-Ruiz, F. M. Bianchi, and R. Jenssen, "Noisy multi-label semi-supervised dimensionality reduction," *Pattern Recognition*, vol. 90, pp. 257–270, 2019, ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2019.01.033. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320319300615.

[9] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," *CoRR*, vol. abs/1703.02910, 2017. arXiv: 1703.02910. [Online]. Available: http://arxiv.org/abs/1703.02910.

[10] X. Li, L. Wang, and E. Sung, "Multilabel svm active learning for image classification," in *2004 International Conference on Image Processing, 2004. ICIP '04.*, vol. 4, 2004, 2207–2210 Vol. 4. DOI: 10.1109/ICIP.2004.1421535.

[11] B. Yang, J.-T. Sun, T. Wang, and Z. Chen, "Effective multi-label active learning for text classification," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09, Paris, France: Association for Computing Machinery, 2009, pp. 917–926, ISBN: 9781605584959. DOI: 10.1145/1557019.1557119. [Online]. Available: https://doi-org.tudelft.idm.oclc.org/10.1145/1557019.1557119.

[12] J. Wu, S. Zhao, V. S. Sheng, J. Zhang, C. Ye, P. Zhao, and Z. Cui, "Weak-labeled active learning with conditional label dependence for multilabel image classification," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1156–1169, 2017. DOI: 10.1109/TMM.2017.2652065.

[13] T. Ridnik, H. Lawen, A. Noy, and I. Friedman, "Tresnet: High performance gpu-dedicated architecture," *CoRR*, vol. abs/2003.13630, 2020. arXiv: 2003.13630. [Online]. Available: https://arxiv.org/abs/2003.13630.

[14] E. Ben-Baruch, T. Ridnik, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, *Asymmetric loss for multi-label classification*, 2020. arXiv: 2009.14119 [cs.CV].

[15] J. Kremer, F. Sha, and C. Igel, "Robust active label correction," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, A. Storkey and F. Perez-Cruz, Eds., ser. Proceedings of Machine Learning Research, vol. 84, PMLR, Apr. 2018, pp. 308–316. [Online]. Available: http://proceedings.mlr.press/v84/kremer18a.html.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[17] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].

[18] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," Sep. 2017.