Rebalancing Label Distribution while Eliminating Inherent Waiting Time in Multi Label Active Learning applied to Transformers

Maxime Arens[1,2] , Lucile Callebert[2], Jose G. Moreno[1] and Mohand Boughanem[1]
[1] IRIT, Toulouse University, UMR 5505 CNRS, 31400 Toulouse, France
[2] Synapse Développement, 7 Boulevard de la Gare, 31500 Toulouse, France
maxime.arens@gmail.com

Abstract

Data annotation is crucial for machine learning, notably in technical domains, where the quality and quantity of annotated data, significantly affect effectiveness of trained models. Employing humans is costly, especially when annotating for multi-label classification, as instances may bear multiple labels. Active Learning (AL) aims to alleviate annotation costs by intelligently selecting instances for annotation, rather than randomly annotating. Recent attention on transformers has spotlighted the potential of AL in this context. However, in practical settings, implementing AL faces challenges beyond theory. Notably, the gap between AL cycles presents idle time for annotators. To address this issue, we investigate alternative instance selection methods, aiming to maximize annotation efficiency by seamlessly integrating with the AL process. We begin by evaluating two existing methods in our transformer setting, employing respectively random sampling and outdated information. Following this we propose our novel method based on annotating instances to rebalance label distribution. Our approach mitigates biases, enhances model performance (up to 23% improvement on f1score), reduces strategy-dependent disparities (decrease of nearly 50% on standard deviation) and reduces label imbalance (decrease of 30% on Mean Imbalance Ratio).

Keywords: active learning, transformers, wait time, label distribution

1. Introduction

Data annotation is a focal point in both machine and deep learning (Fredriksson et al., 2020) as the accuracy of trained models often hinges on the quantity and quality of the annotated data available. The annotation process, especially in technical domains, is a resource-intensive endeavor, necessitating human involvement, and in some cases, expertise (Wu et al., 2021). This holds even more true with multi-label classification (Zhang, 2022), where each instance may be associated with multiple labels. The primary objective of Active Learning (AL) lies in mitigating the cost of data annotation (Wang et al., 2021) by judiciously selecting and limiting the data to be annotated. Instead of annotating data in a random manner, AL strategies prioritize the selection of optimal data subsets for annotation, aiming to maximize the model's information gain during subsequent training steps. These chosen data subsets are then annotated by human oracles. These two steps iteratively repeat until a designated stopping criterion is met. The effectiveness of an AL strategy is directly linked to the value of interacting with a human oracle.

The notion of AL in the context of multi-label tasks holds particular interest for research (Liu et al., 2021), especially in addressing challenges such as label imbalance (Tarekegn et al., 2021). Indeed, there is often an imbalance among various labels within a dataset, as well as within individual labels themselves (the ratio of positive to negative representation of a label). This imbalance makes it harder to correctly train models. Moreover, AL tends to perpetuate, and sometimes exacerbate, this imbalance when selecting data for annotation (Attenberg and Ertekin, 2013). Furthermore, recent findings (Wertz et al., 2022) underscore that many existing AL strategies for this task do not seamlessly apply on transformers. Indeed, recent advancements in deep architectures (Vaswani et al., 2017) have prompted investigations around applying AL on transformers. Transformers are well-suited for uncertainty-based AL due to the fact that these strategies induced less computational overhead than other state-of-the-art strategies (Schröder et al., 2022).

In practical work settings, when aiming to implement AL into workflows, both theoretical and practical challenges emerge. Research predominantly delves into theoretical concerns like devising the better strategy to evaluate informativeness and select one instance to annotate over another (Settles, 2009) and overlooks human-in-the-loop or strategic aspects of this solution. Nonetheless, intriguing research avenues lie within practical challenges as well. For instance, a major hurdle in implementing AL in practice is the lack of certainty regarding the effectiveness of a given strategy on an unknown dataset as strategy performance is dataset-linked (Wertz et al., 2022). Furthermore, our background in applying AL has revealed that when used in conjunction with transformers, the time between two AL cycles can become relatively high. Depending on hardware capabilities and the size of unlabeled datasets, this waiting time can reduce or even negate any benefits brought by the implementation of AL. We are then faced with a dilemma: *"either increase hardware costs to reduce this waiting time or pay for annotators waiting time"*. In AL, we target resource-constrained scenarios, where limited access to annotators can corre-

LREC-COLING 2024, pages 13621–13632
20-25 May, 2024.

lates with computer resource constraints. The cost of machine learning today lies in both annotation and the computational resources required for model training, making the idea of reducing training/inference time through more powerful computers impractical. Our approach sidesteps this dilemma by implementing an alternative method to annotate parallel to AL to keep annotators engaged while AL scores are calculated, thereby avoiding wait times and the need for more computation power.

Surprisingly, very little research (Zhang et al., 2022) has been conducted to determine which sampling methods can harmoniously coexist with AL to maximize the entirety of annotation time, rather than just the portion dedicated to AL. Consequently, we explore multiple alternative methods along different AL strategies and show that our realistic workflow reduces some of the biases of AL highlighted in previous studies such as sampling redundant instances (Citovsky et al., 2021). We also present a new approach that not only can be utilized in conjunction with AL and effectively eliminates annotator wait times, but also enhances the performance of the trained model. Moreover, it reduces the importance of the strategy choice as model performances are close irrespective of the chosen AL strategy. Finally, it also addresses another challenge inherent to the multi-label classification tasks: the uneven distribution of labels.

Our work contributions are fourfold: **1.** Providing a workflow to eliminate waiting time for the annotator during AL cycles. **2.** Rebalancing label distribution during annotation. **3.** Reducing differences in performance between different AL strategies. **4.** Improving performances over six classical AL strategies, for two models and four datasets.

2. Related Work

Multi-label classification poses a unique set of challenges in the realm of AL due to the inherent imbalance often observed within multi-label datasets (Tarekegn et al., 2021). The imbalance issue becomes twofold: not only do certain labels occur more frequently than others within the dataset, but the distribution of positive and negative instances for each label can also vary significantly (Ben-Baruch et al., 2021). This entails that AL may inadvertently bias the model towards frequently occurring labels or worsen the imbalance problem by selecting predominantly one class of instances (Attenberg and Ertekin, 2013). Inspirational work can be found in the realm of multi-class classification coupled with AL, where, if imbalance thresholds are reached during the standard AL cycle, it is paused, and rebalancing steps are executed (Aggarwal et al., 2020). The same authors have also delved into addressing this imbalance directly throughout the entire AL cycle by only selecting instance scores from minority classes (Aggarwal et al., 2021).

The remarkable success of transformers (Vaswani et al., 2017) in various natural language processing and computer vision tasks has led to a surge in research dedicated in integrating AL principles with these architectures. While some studies (Ein-Dor et al., 2020; Lu and MacNamee, 2020) have demonstrated that AL can effectively mitigate biases during the initial stages of model training, other preliminary results by D'Arcy and Downey (2022) suggested that applying AL on transformers leads to training instability. Drawing inspiration from the insights of Lu and MacNamee (2020), our focus centers on the study of uncertainty-based strategies for AL within the framework of transformers. The most comprehensive study of the use of uncertainty-based AL strategies within the transformer context (Schröder et al., 2022) reveals substantial performance variations between different active strategies. Moreover, AL strategies are highly dependent on the used dataset. Those results are in line with results on the extreme multi-label task (Wertz et al., 2022). Highlighting one of the current main weaknesses of AL : the lack of evidence in achieving a substantial benefit from its implementation (Ren et al., 2022).

Annotation resources can be budgeted on a per-annotation or per-hour basis, and in the latter case, it becomes imperative to ensure a continuous supply of instances for annotation without causing unnecessary waiting time (Monarch et al., 2021). This motivation first led to the development of batch-mode AL (Settles, 2011), which provides instances in batches, reducing the frequency of model updates. Selecting AL instances in batches aligns well with how transformers are trained. However, Citovsky et al. (2021) demonstrated that annotation of redundant instances is one risk of batch-mode AL. Moreover, even in batch-mode AL, there remains a waiting time for the annotator during the model update between annotated batches. Some approaches (Haertel et al., 2010), partially address this issue by introducing a parallel process to AL that offers instances for annotation based on their informativeness from previous AL cycles, at the cost of reduced accuracy. In contemporary scenarios where models and unlabeled datasets are becoming larger, waiting time is no longer primarily induced by model updates but by model inference on the extensive unlabeled set (Zhang et al., 2022). To mitigate this induced waiting time, Tsvigun et al. (2022) focus on subsampling, where only a part of the unlabeled set is inferred at each AL cycle. The work of Ashrafi Asli et al. (2020) explores AL strategies with pre-calculated features for instance pre-clustering. While promising, we aim to further explore these approaches with the goal of not only mitigating but completely eliminating waiting times for annotators.

3. Eliminating AL's waiting time

In AL, we target resource-constrained scenarios, where limited access to annotators often correlates with computer resource constraints. With transformer models,

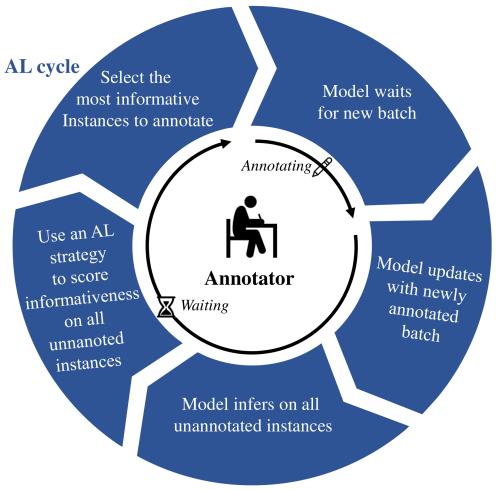

Figure 1: Classic AL cycle without a parallel sampling method to avoid annotator's waiting time

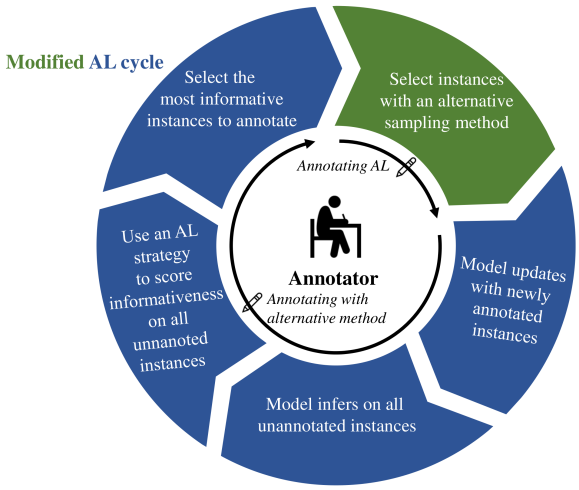

Figure 2: AL cycle with an alternative sampling method to avoid annotator's waiting time

inherent waiting times stem from model predictions on the unannotated dataset, scaling with dataset size. Compared to this, uncertainty-based AL calculations and model updates are relatively swift. Indeed, from our practical experiments conducted using a distilled variant of BERT (Sanh et al., 2019), a dataset of 100,000 unannotated text instances, and an Nvidia V100 (32 GB), we noticed that the average duration between two active learning cycles is roughly 5 minutes. This duration comprises approximately 25 seconds for model updates and about 15 seconds for active learning calculations, which can vary by approximately 10 seconds depending on the chosen strategy.

Although we can reduce waiting times by using augmented computational power, it's not realistic to completely eliminate them that way. So, in the regular AL cycle (shown in Figure 1), we assume that annotators will always have to wait for a noticeable amount of time. To optimize this waiting time and optimize the annotation budget, we suggest using methods to pick which data to annotate that do not necessitate an up-to-date model incorporating the latest annotated data. By employing alternative annotation methods, we thereby obtain a modified AL cycle (shown in Figure 2), where annotators consistently possess data for annotation, eliminating waiting time.

Furthermore, these alternative methods must swiftly select data instances for annotation and deliver them for annotation before annotators complete annotating the set supplied by AL. Within our experimental framework, we assume annotators work at a constant speed and waiting time between AL cycles are constant. Data annotated by alternative methods is incorporated into AL cycles. At each AL update, there is a proportion α of instances composing the batch that come from alternative sampling method and a proportion $1-\alpha$ that has been selected following an AL strategy.

When $\alpha = 0$, it represents the ideal and unrealistic scenario depicted in most AL literature, where waiting time is not taken into account, and all annotated data comes from AL strategy selection. When $\alpha = 1$, it represents a case where we do not perform any AL and thus, deviates from our use case.

Hence, we assess two alternative methods, namely, **Random** and **Stale** (Haertel et al., 2010), alongside introducing our method **Eq_label**, designed to **Eq**uilibrate the **label**s annotated in order to improve performance.

3.1. Random sampling

Random sampling is a frequently used baseline in AL, where instances needing annotation are picked at random from the pool of unlabeled data. In our framework, we can easily apply this selection method, choosing data for annotation randomly while the annotator waits between AL cycles. This approach is worth studying because recent findings have demonstrated that random sampling is a robust baseline when AL is applied to transformer models.

Indeed, some AL strategies have selection biases, resulting in chosen data lacking diversity among them. This may lead to redundancy and even model bias among the selected data, despite AL's inherent focus on selecting the most informative data for annotation. By employing random sampling, we purposefully choose varied and dissimilar data, providing a countermeasure against redundancy and potential bias. Additionally, this method incurs minimal computational overhead.

3.2. Stale uncertainty sampling

An alternate approach for selecting data for annotation involves utilizing the AL scores from the preceding cycle, essentially employing "stale" informativeness scores (Haertel et al., 2010). Consequently, a training batch is composed of two halves: one derived from AL computations based on the model's predictions prior to

updating, and the other derived from AL selection following the model's update. This entails selecting less ranked data w.r.t to their informativeness scores. Furthermore, this alternative method also carries the risk of amplifying the shortcomings of certain AL strategies, possibly exacerbating their associated biases like information redundancy or label imbalance in selected batches.

3.3. Rebalancing label distribution

In this alternative method for selecting data for annotation, we use outdated scores of informativeness, more precisely scores of uncertainty. However, in contrast to the previous method (seen in part 3.2), we do not select data on which the model is most uncertain; instead, we choose data on which the model is most certain. The underlying premise behind this idea arises from the observation that between each model update during AL cycles, the instances on which the model is most uncertain change significantly, whereas the instances on which the model is confident in its predictions remain relatively consistent. In other words, the staleness of uncertainty is far greater than that of certainty.

By weighting uncertainty scores with the probability of the instance being labeled with a rare label (detailed in Section 3.3.1), we achieve a more uniform labeling across different labels. Reducing the label disparity not only enhances overall performance but can also be a desired criterion in projects aiming for equal performance across all labels. To mitigate label imbalance, we rely on label presence probabilities, and inherently, instances on which the model is uncertain are those where these probabilities are least informative. Moreover, it's reasonable to assume that having batches containing both certain and uncertain instances will naturally encourage diversity, as they are likely different, thus enhancing the learning accuracy.

3.3.1. Eq_label method definition

We want to obtain an interest score for an instance to determine which instances are likely to have the most interesting labels. An instance is considered interesting if its labeling helps rebalance the labels we have annotated, or in other words, if this instance is labeled with rare labels.

Notation-wise, we denote our text instances as $x_1, ..., x_n$ and our label space as $l = l^1, ..., l^q$. For a given instance x_i we represent its probability-like predicted label distribution by $y_i = [y_i^1, ..., y_i^q], y_i^j \in [0, 1]$ where the more y_i^j is close to 1, the more the model is confident that x_i is labeled as l^j and where the more y_i^j is close to 0, the more the model is confident that x_i is not labeled as l^j.

Our calculation method consists of the product of two factors. The first factor represents the model's confidence in its label predictions for an instance, and the second factor is a score associated with which labels the model assigns to this instance. The higher the rarity of the labels, the higher this score becomes.

Let the first factor of the final product be the certainty score c_i (inverse of the uncertainty scores calculated by uncertainty-based AL strategy seen in part 4.2) associated with the instance x_i.

The second factor of the final product is calculated through a series of calculation steps. Let $L(t) = [L(t)^1, \ldots, L(t)^q]$ with $L(t)^j$ be the sum of instances annotated positively with label j at time t.

And, let $\omega(t) = [\omega(t)^1, \ldots, \omega(t)^q]$ be the intermediate weighting vector, where $\omega(t)^j = (\max(L(t)) + L(t)^j)/(2 \cdot L(t)^j)$ if $L(t)^j \neq 0$.

The cases where $L(t)^j = 0$ are computed in a subsequent step to equal $2 \cdot \max(\omega_t^j)$. At this stage, the rarer a label is, the higher its associated $\omega(t)^j$.

To stabilize our method and make the comparison more reliable among different instances, we normalize the obtained scores. Let $\omega'(t) = [\omega'(t)^1, \ldots, \omega'(t)^q]$ be the normalized weighting vector, with $\omega'(t) = \textit{softmax}(\omega(t))$.

Finally, we get the second factor of the final product be the label equilibrating score $e(t)_i$ associated with the instance x_i at time t, we have: $e(t)_i = \sum_{j=1}^{q} y_i^j \cdot \omega'(t)^j$.

This equilibrating score contributes to the instance's interest score (higher scores are selected to be annotated), which is given by $score(t)_i = c_i \cdot e(t)_i$.

4. Experiments

Various values of alpha may be explored in future experiments. However, based on our empirical investigations, if one aims for regular model updates and, therefore, more precise AL, only around half of the annotation time can be allocated to AL. In our work, aside from the initial training batch (an initial random model initialization), we set $\alpha = 0.5$ and all data batches used for model training are thus equally composed of data from our alternative method and data from AL.

In our experiments, multi-label AL consists in the following process: first, our models are initialized by training them with 25 randomly selected instances. Following (Schröder et al., 2022), we then perform 50 iterations of AL where each batch is composed of 25 instances, thus a total of 1250 annotated instances is collected. To meet our target of $\alpha = 0.5$, at each iteration, the batch consists of 12 instances selected by an alternative method and 13 instances chosen through an active learning strategy. After each iteration, we further train the model with the newly annotated batch of instances.

4.1. Dataset

Table 1 showcases the characteristics of the four datasets utilized. Cardinality represents the average number of labels per instance, while density is obtained

Table 1: Features of the benchmark datasets

Name	Labels	Training	Test	Cardinality	Density
Jigsaw_toxic	6	159,571	63,978	0.222	0.037
Go_emotions	27	43,410	5,427	0.848	0.031
EUR_Lex	4,271	55,000	5,000	4.526	0.036
UNFAIR-ToS	8	5,532	1,607	0.124	0.016

by dividing the cardinality by the total number of instances.

We have selected these datasets to conduct our experiments on texts that exhibit variations in the level of language used (ranging from offensive language to legal text, encompassing social media comments) as well as variations in the number of labels associated with each instance (ranging from 6 to 100).

The *Jigsaw toxic comment classification (Jigsaw_Toxic)* dataset originates from a Kaggle competition[1], aiming to detect and classify six distinct types of toxicity prevalent in online content. Instances are extracted from comments on Wikipedia pages. The various labels, corresponding to different forms of toxicity, often exhibit correlations (for instance, all instances of 'severe toxicity' also bear the 'toxicity' label). Nearly 90% of dataset instances display no form of toxicity, hence are unassociated with any label.

Go_Emotions is a dataset consisting of Reddit comments[2] labeled across 27 emotion categories like 'anger' or 'curiosity' (Demszky et al., 2020). Slightly over 30% of instances are labeled as 'neutral', signifying an absence of labels.

EUR_Lex57K (EUR_Lex) stands as a dataset composed of legal texts (Chalkidis et al., 2021) sourced from the corresponding website[3].

UNFAIR - Terms of Services (UNFAIR-ToS) is a dataset comprising texts annotated with eight types of unfair contractual terms (Lippi et al., 2018). These terms potentially violate consumer rights according to European consumer law.

In our research, we utilized the versions of EUR_Lex and UNFAIR-ToS provided within the *Legal General Language Understanding Evaluation* (LexGLUE) (Chalkidis et al., 2022).

Throughout our experiments, 10% of the training dataset is allocated for validation purposes, and the reported performance metrics are obtained from the test dataset.

4.2. Multi-label active learning strategies

The task of multi-label text classification consists of assigning appropriate labels to text instances. Unlike multi-class classification, multiple labels can be assigned to an instance. For each experiment, our label space is predefined and is not extended by the experiment. AL aims to select the best possible instances to annotate in a training process. This selection can be carried out with different *strategies*. All these strategies are based on estimating the *uncertainty* of the model on each instance, that is to say the confidence of the model in predicting the labels associated with an instance. These strategies are based on the hypothesis that by training on hard examples (where the model hesitates), the model will gain in performance.

We follow pool-based AL (Lewis and Gale, 1994) where at each training iteration, the strategies select the instances to annotate from the remaining unlabelled data set. For each unlabelled instance, we compute a score that indicates the uncertainty of the model on its associated predictions.

We use notations from Section 3.3.1. We apply six uncertainty-based multi-label AL strategies to transformers:

Max Loss (ML) selects the instances with the highest loss (Li et al., 2004):

$$\underset{x_i}{\operatorname{argmax}}\Big[\sum_{j=1}^{q}\max\{1-m_j*f_j(x_i),0\}\Big] \quad (1)$$

where $m_j = 1$ if $j = u$, $m_j = -1$ otherwise, u corresponds to the label l^u associated with the greatest probability to a given instance and where $f_j(x_i)$ is defined as:

$$f_j(x_i) = 2 * y_i^j - 1 \quad (2)$$

Mean Max Loss (MML) selects the instances with the highest mean loss (Li et al., 2004):

$$\underset{x_i}{\operatorname{argmax}}\ \frac{1}{q}\Big[\sum_{k=1}^{q}\sum_{j=1}^{q} max\{1-o_{kj}*f_j(x_i),0\}\Big] \quad (3)$$

where $o_{kj} = 1$ if j=k, $o_{kj} = -1$ otherwise and $f_j(x_i)$ is defined in (2).

Minimum Confidence No weighting (CMN) selects the instances where the confidence of the model is the lowest (Esuli and Sebastiani, 2009):

$$\underset{x_i}{\operatorname{argmin}}\Big(\min_{j=1}^{q} f_j(x_i)\Big) \quad (4)$$

with $f_j(x_i)$ being defined in (2).

Max Margin Uncertainty sampling (MMU) selects the instances that maximize the separative margin between the predicted groups of positive and negative labels (Li and Guo, 2013):

$$\underset{x_i}{\operatorname{argmax}}\ \frac{1}{\min pos(x_i) - \max neg(x_i)} \quad (5)$$

where $pos(x_i) = [pos_1(x_i), ..., pos_q(x_i)]$ and $neg(x_i) = [neg_1(x_i), ..., neg_q(x_i)]$, with:

$$pos_j(x_i) = \begin{cases} f_j(x_i) & \text{if } f_j(x_i) > 0 \\ +\infty & \text{otherwise} \end{cases} \quad \text{and} \quad (6)$$

[1] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[2] https://www.reddit.com/

[3] https://eur-lex.europa.eu

$$neg_j(x_i) = \begin{cases} f_j(x_i) & \text{if } f_j(x_i) < 0 \\ -\infty & \text{otherwise} \end{cases} \quad (7)$$

with $f_j(x_i)$ being defined in (2).

Label Cardinality Inconsistency (LCI) selects the instances that maximize the distance between the number of predicted positive labels and the label cardinality of the labelled set (Li and Guo, 2013):

$$\underset{x_i}{\text{argmax}} \sqrt{\left(\sum_{j=1}^{q} y_i^j\right) - L}^{\,2} \quad (8)$$

with L the average number of labels on the already annotated instances.

Category Vector Inconsistency and Ranking of Scores (CVIRS) selects the instances following two measures. The first is based on a rank aggregation of difference margins of classifier predictions. The second is based on the inconsistency of the predicted label sets compared to the label space of the labelled set. This strategy is detailed in Reyes et al. (2018).

Those strategies are called *myopic* strategies, meaning that they evaluate uncertainty instance-wise. As in Reyes et al. (2018), we extend those strategies to batch-mode AL in a simple way: instead of selecting *the* instance for which the model is the most uncertain we select the top instances for which the model is the most uncertain, to fill our batch.

4.3. Study setup

Oracle: The simulation of a human oracle annotating the unlabelled instances selected by the different strategies is carried out using annotated multi-label datasets. In each training iteration, the AL strategy selects the best instances from the training dataset without having access to their corresponding labels. These instances and their corresponding labels constitute the next training batch.

Models: In order to make our experiments more exhaustive, our experiments are done on two different models. As in (Schröder et al., 2022), the two transformers used in this study are based on BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). Considering that (Tsvigun et al., 2022) demonstrate that in AL processes, distilled versions of these models achieve similar performance to the original models while being less computationally intensive, we also utilize the distilled versions of these models (Sanh et al., 2019). On top of both models, we add a dense neuron layer and a sigmoid layer to accomplish multi-label classification.

Implementation Details: DistilBERT consists of 6 layers, hidden units of size 768, and 66 billion parameters. DistilRoBERTa follows a similar structure, with the exception of its parameter count, which is 82 billion. The maximum input token size for both models is set to 128, the number of epochs to 15, and the training batch size to 25. To optimize the model parameters, we chose AdamW with a learning rate of 2e-5. The experiments were conducted on an Nvidia V100 (32 GB). The same hyperparameters were used for both models across the four datasets. For other parameters, we follow the *fine-tuning* process outlined in (Howard and Ruder, 2018). These experiments were conducted using the *small-text* library[4] (Schröder et al., 2023).

4.4. Evaluation

The results presented in this paper, represent an average calculated from five runs of each experiment, as in (Schröder et al., 2022). *Reference Values:* As a point of comparison we will use results from the idealistic setting often used in AL works, where AL update time and model inference time are considered negligible. This scenario (with $\alpha = 0$) referred to as **Classic-AL**, in practice, result in a loss of annotation time (and money) and does not answer our issue. However, it will be compared to results from our three realistic settings where we took into account waiting times by adding different alternative sampling methods.

Performance metric: To evaluate performance, we employ a commonly used metric in multi-label classification (Tsoumakas et al., 2010). We use the notations from section 4.2, with the addition: for a given label l^j we denote the number of true positives (vp^j), false positives (fp^j), and false negatives (fn^j), and define the F1-score as:

$$F1(vp^j, fp^j, fn^j) = \frac{vp^j}{vp^j + \frac{1}{2}(fp^j + fn^j)} \quad (9)$$

We employ a micro-averaged F1-score, meaning that we sum up all true positives, false positives, and false negatives across all labels, then compute the F1-score (higher value indicates better performance):

$$M_{iF1} = F_1(\sum_{j=1}^{q} vp^j, \sum_{j=1}^{q} fp^j, \sum_{j=1}^{q} fn^j) \quad (10)$$

Label imbalance metric: To measure the label imbalance, we employ the Mean Imbalance Ratio (MeanIR) (Tarekegn et al., 2021) (higher value indicates higher imbalance in the label distribution).

$$MeanIR = \frac{1}{q} \sum_{j=1}^{q} IRLbl(l^j) \quad (11)$$

With IRLbl being the Imbalance Ratio per Label, such as:

$$IRLbl(l^j) = \frac{\max_{\lambda \in l} \sum_{i=1}^{n} h(\lambda, y_i)}{\sum_{i=1}^{n} h(l^j, y_i)} \quad (12)$$

, with:

$$h(\lambda, y_i) = \begin{cases} 1 \text{ if } \lambda \in y_i \\ 0 \text{ otherwise} \end{cases} \quad (13)$$

[4] https://small-text.readthedocs.io/en/latest/

5. Results

In our study, we establish a robust point of comparison under idealistic conditions, where the annotator allocates all available time to annotate instances selected through AL.

Table 2 presents the detailed outcomes of our experimental analysis. Primarly, we observe that Random and Eq_label outperforms Stale and the Classic-AL setting in many cases. Indeed, in 48 cases, Random outperforms Classical-AL 32 times, Eq_label outperforms Classical-AL 41 times, and Random outperforms Stale 34 times, while Eq_label outperforms Stale 39 times. Secondly, we observe that the performance trends associated with Stale are not very distinct from those of the Classic-AL setting. This outcome suggests that the shift of a half-batch of annotated data following the Stale method, as compared to Classic-AL setting, does not significantly affect the progression of annotation according to a specific AL strategy. According to conventional statistical criteria, the difference between Stale and the Classic-AL setting is not considered statistically significant (p-value > 0.05 with paired t-test per dataset and model). On the contrary, Random and Eq_label significantly deviate from the Classic-AL setting in an extremely statistically significant manner (p-value < 0.001 with paired t-test per dataset and model).

The overall performance enhancements observed with the Random method likely stem from the diversification of the batch, while for our Eq_label method, these improvements arise not only from text diversification within the batch but also from a more balanced representation of various labels. Furthermore, it is notable in the standard deviation lines that both the Random and Eq_label methods exhibit a reduction in performance differences among different AL strategies. This outcome is expected, as now only half of the annotated data during the experiments originate from these strategies. Nonetheless, this is an interesting result, as one of the significant challenges when implementing AL is selecting an AL strategy beforehand. Following our framework, this choice becomes less critical, simplifying the practical implementation of AL.

The results of the Random method are generally good, especially when considering its ease of implementation and understanding. Random is thus a strong option to consider when implementing an alternative method of sampling to eliminate wait time. However, this method appears to degrade performance in certain cases, such as the CMN strategy or the Go_emotions dataset. Compared to Random and Eq_label, the results of the Stale method are inferior, making this option the less interesting one to implement in order to eliminate wait times. The results of the Eq_label method are generally the most promising in our experiments, improving on all strategies and datasets, often more than Random. It is worthwhile to focus on another aspect of this method, which is label distribution (see Table 5). Once again it is not surprising to observe no significant difference in average label distribution between Classic-AL setting and Stale. Preliminary experiments have shown that in some cases AL does worsen the label imbalance present in the original dataset. Therefore as expected we observe a significant improvement between Classic-AL/Stale and Random, where the average MeanIR for the latter is notably 10% lower for distilRoBERTa and 20% lower for distilBERT. For Eq_label, an even more egalitarian distribution of annotated labels can be noticed, as in average, MeanIR for Eq_label is 25% lower for distilRoBERTa and 35% lower for distilBERT than for Classic-AL/Stale. The results for each dataset in Table 3 demonstrate that Eq_label outperforms the other methods, achieving superior performance on three out of four datasets and coming close to matching Random on Jigsaw_toxic the fourth one. The trends depicted in Table 4 align with our expectations, indicating that the strategies benefiting the most from the implementation of an alternative annotation method are the ones that initially exhibited poorer performance, such as ML and LCI. Furthermore, it can be observed that our Eq_label method enhances performance similarly or even surpasses Random. Additionally, unlike the other two methods, the inclusion of our approach leads to improvements across all strategies.

6. Discussion and Conclusion

It is noteworthy that our approach, which addresses the practical issue of waiting time, has shown that integrating a parallel annotation method can significantly improve the overall output of the annotation process through AL. Our approach, combining three key concepts - prioritizing certainty over uncertainty regarding staleness, enhancing performance through label distribution rebalancing, and using model prediction certainty for label association - consistently improves results across all datasets, models, and AL strategies. From a practical standpoint, our method successfully eliminates the wait time that can exist during classical AL applications. It also simplifies the decision process at the start of an AL project by making the choice of an AL strategy less crucial for its success. Nonetheless, our findings highlight CMN and MMU as the top-performing strategies. Even when combined with another sampling method, they still exhibit slightly superior performance. While generally weaker, Random is still an option to consider for its ease of implementation. In future work, we would like to explore how our results evolve when varying the proportion of instances originating from AL or an alternative method within the same batch (varying α). We also aim to extend our study to other related tasks such as multi-class and binary classification, where our method can be readily adapted, as well as to more distant tasks like question answering, where an equivalent of our method could involve rebalancing the distribution of documents from which answers are derived.

Table 2: Experimental results (M_{iF1}) of the Classic-AL setting and three parallel sampling methods to AL. With 'dB' being distilBERT, 'dR' being distilRoBERTa and 'Std dev' being standard deviation. Best values for model/strategy are in bold.

	Method →	Classic-AL		Random		Stale		Eq_label	
	Model →	dB	dR	dB	dR	dB	dR	dB	dR
Dataset ↓	**Strategy ↓**								
Jigsaw_toxic	ML	0.412	0.486	**0.57**	**0.577**	0.529	0.522	0.567	0.548
	MML	0.467	0.524	**0.572**	**0.59**	0.51	0.582	0.547	0.581
	CMN	0.617	0.593	0.603	0.6	0.605	**0.619**	**0.631**	0.618
	MMU	0.603	0.614	**0.623**	0.609	0.609	0.613	0.61	**0.615**
	LCI	0.453	0.482	**0.556**	**0.587**	0.496	0.499	0.551	0.546
	CVIRS	0.564	0.575	0.551	0.58	**0.591**	**0.598**	0.537	0.565
	Std dev	0.086	0.056	**0.028**	**0.012**	0.051	0.05	0.038	0.031
Go_emotions	ML	0.404	0.414	0.411	**0.432**	0.386	0.383	**0.412**	0.415
	MML	0.414	0.428	0.404	0.421	0.412	0.427	**0.416**	**0.431**
	CMN	0.438	0.435	0.423	0.416	0.408	0.433	**0.448**	**0.443**
	MMU	0.441	0.435	0.419	0.405	0.418	0.413	**0.455**	**0.441**
	LCI	0.353	0.373	0.383	**0.402**	0.363	0.37	**0.422**	0.389
	CVIRS	0.362	0.4	0.408	**0.441**	**0.41**	0.395	0.405	0.408
	Std dev	0.037	0.024	**0.014**	**0.015**	0.021	0.025	0.02	0.021
EUR_Lex	ML	0.329	0.409	0.495	0.467	0.401	0.42	**0.502**	**0.531**
	MML	0.517	**0.56**	0.52	0.53	0.517	0.513	**0.543**	0.553
	CMN	0.513	0.55	0.464	0.526	0.511	0.522	**0.521**	**0.556**
	MMU	0.53	**0.567**	0.539	0.489	0.518	0.526	**0.545**	0.565
	LCI	0.454	0.488	**0.521**	0.5	0.47	0.478	0.515	**0.525**
	CVIRS	0.479	0.509	**0.565**	0.535	0.562	0.5	0.534	**0.537**
	Std dev	0.075	0.06	0.035	0.027	0.056	0.04	**0.016**	**0.026**
UNFAIR-ToS	ML	0.622	0.718	0.694	0.726	0.682	0.688	**0.703**	**0.757**
	MML	0.647	0.724	0.696	0.71	0.651	0.704	**0.717**	**0.758**
	CMN	0.708	**0.778**	0.756	0.777	0.749	0.689	**0.767**	0.758
	MMU	0.7	0.762	**0.741**	**0.773**	**0.741**	0.763	0.738	0.762
	LCI	0.65	0.744	0.732	**0.753**	0.619	0.713	**0.765**	0.738
	CVIRS	0.708	0.76	0.723	0.746	**0.746**	0.703	0.721	**0.765**
	Std dev	0.037	0.023	0.027	0.026	0.056	0.028	**0.026**	**0.009**

Table 3: Average percentage of M_{iF1} difference from the Classic-AL setting per method, model and dataset (a positive value indicates improvement), 'dB' being distilBERT and 'dR' being distilRoBERTa. Best values are in bold.

	Dataset	Jigsaw_toxic		Go_emotions		EUR_Lex		UNFAIR-ToS	
	Model	dB	dR	dB	dR	dB	dR	dB	dR
Method	Random	**+11.521**	**+6.139**	+1.493	-1.489	+9.993	+0.681	+7.608	-0.022
	Stale	+7.189	+4.856	-0.622	-2.575	+5.563	-4.022	+3.792	-5.038
	Eq_label	+10.494	+6.078	**+6.053**	**+1.690**	**+11.977**	**+5.968**	**+9.318**	**+1.159**

Table 4: Average percentage of M_{iF1} difference from the Classic-AL setting per method, model and AL strategy (a positive value indicates improvement). Best values are in bold.

	Method	Random		Stale		Eq_label	
	Model	distilBERT	distilRoBERTa	distilBERT	distilRoBERTa	distilBERT	distilRoBERTa
Strategy	ML	+22.807	+8.633	+13.073	-0.691	**+23.599**	**+11.051**
	MML	+7.188	+0.671	+2.200	-0.447	**+8.704**	**+3.891**
	CMN	-1.318	-1.570	-0.132	-3.947	**+3.998**	**+0.806**
	MMU	+2.111	-4.289	+0.528	-2.649	**+3.254**	**+0.210**
	LCI	+14.764	**+7.427**	+1.990	-1.294	**+17.958**	+5.319
	CVIRS	+6.342	**+2.585**	**+9.276**	-2.139	+3.975	+1.381

Table 5: Average MeanIR per method and model (high value indicates a high label imbalance). 'Full' refers to the MeanIR score calculated for the entire dataset. Best values are in bold.

Dataset	Method / Model	Classic-AL	Random	Stale	Eq_label	Full
Jigsaw_toxic	distilBERT	6.681	6.846	5.776	**4.886**	9.537
	distilRoBERTa	6.782	6.869	6.548	**5.874**	
Go_emotions	distilBERT	12.945	13.035	15.065	**10.017**	12.661
	distilRoBERTa	14.238	12.719	15.111	**10.690**	
EUR_Lex	distilBERT	54.560	37.422	53.409	34.826	**25.918**
	distilRoBERTa	37.745	34.667	38.288	30.048	
UNFAIR-ToS	distilBERT	5.945	5.018	5.781	**2.795**	3.301
	distilRoBERTa	5.960	5.972	7.286	**2.894**	

Acknowledgments

This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD011014063 made by GENCI.
This material is based upon work supported by the ANRT (Association nationale de la recherche et de la technologie) with a CIFRE fellowship granted to Maxime Arens.

Bibliographical References

U. Aggarwal, A. Popescu, and C. Hudelot. 2021. Minority class oriented active learning for imbalanced datasets. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9920–9927, Los Alamitos, CA, USA. IEEE Computer Society.

Umang Aggarwal, Adrian Popescu, and Céline Hudelot. 2020. Active learning for imbalanced datasets. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1417–1426.

Seyed Arad Ashrafi Asli, Behnam Sabeti, Zahra Majdabadi, Preni Golazizian, Reza Fahmi, and Omid Momenzadeh. 2020. Optimizing annotation effort using active learning strategies: A sentiment analysis case study in Persian. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2855–2861, Marseille, France. European Language Resources Association.

Josh Attenberg and Şeyda Ertekin. 2013. Class imbalance and active learning. *Imbalanced Learning: Foundations, Algorithms, and Applications*, pages 101–149.

Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. 2021. Asymmetric loss for multi-label classification.

Jasmin Bogatinovski, Ljupo Todorovski, Sao Deroski, and Dragi Kocev. 2022. Comprehensive comparative study of multi-label classification methods. *Expert Systems with Applications*, 203:117215.

Klaus Brinker. 2006. On active learning in multi-label classification. In *From Data and Information Analysis to Knowledge Engineering*, pages 206–213, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. 2021. MultiEURLEX - A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. *CoRR*, abs/2109.00904.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2022. LexGLUE: A benchmark dataset for legal language understanding in English. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dubln, Ireland.

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S. Dhillon. 2020. Taming pretrained transformers for extreme multi-label text classification. In *KDD 2020*.

Shuyue Chen, Ran Wang, Jian Lu, and Xizhao Wang. 2022. Stable matching-based two-way selection in multi-label active learning with imbalanced data. *Information Sciences*, 610:281–299.

Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. 2021. Batch active learning at scale.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. 1996. Active learning with statistical models. *J. Artif. Int. Res.*, 4(1):129145.

Mike D'Arcy and Doug Downey. 2022. Limitations of active learning with deep transformer language models. URL : https://openreview.net/forum?id=Q8OjAGkxwP5.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim.

2020. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

Andrea Esuli and Fabrizio Sebastiani. 2009. Active learning strategies for multi-label text classification. In *Advances in Information Retrieval*, pages 102–113, Berlin, Heidelberg. Springer Berlin Heidelberg.

Teodor Fredriksson, David Issa Mattos, Jan Bosch, and Helena Holmström Olsson. 2020. Data labeling: An empirical investigation into industrial challenges and mitigation strategies. In *Product-Focused Software Process Improvement*, pages 202–216, Cham. Springer International Publishing.

Daniel Gissin and Shai Shalev-Shwartz. 2019. Discriminative active learning. *CoRR*, abs/1907.06347.

Daniel Gissin and Shai Shalev-Shwartz. 2019. Discriminative active learning. *ArXiv*, abs/1907.06347.

Xiaoqiang Gui, Xudong Lu, and Guoxian Yu. 2021. Cost-effective batch-mode multi-label active learning. *Neurocomputing*, 463:355–367.

Robbie Haertel, Paul Felt, Eric K. Ringger, and Kevin Seppi. 2010. Parallel active learning: Eliminating wait time with minimal staleness. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 33–41, Los Angeles, California. Association for Computational Linguistics.

Alex Holub, Pietro Perona, and Michael C. Burl. 2008. Entropy-based active learning for object recognition. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Sheng-Jun Huang, Songcan Chen, and Zhi-Hua Zhou. 2015. Multi-label active learning: Query type matters. In *IJCAI*.

Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. 2014. Active learning by querying informative and representative examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(10):1936–1949.

Sheng-Jun Huang and Zhi-Hua Zhou. 2013. Active query driven by uncertainty and diversity for incremental multi-label learning. In *2013 IEEE 13th International Conference on Data Mining*, pages 1079–1084.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Anders Krogh and Jesper Vedelsby. 1994. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, volume 7. MIT Press.

Punit Kumar and Atul Gupta. 2020. Active learning query strategies for classification, regression, and clustering: A survey. *Journal of Computer Science and Technology*, 35(4):913–945.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *SIGIR '94*, pages 3–12, London. Springer London.

X. Li, L. Wang, and E. Sung. 2004. Multilabel svm active learning for image classification. In *2004 International Conference on Image Processing,2004. ICIP '04.*, volume 4, pages 2207–2210 Vol. 4.

Xin Li and Yuhong Guo. 2013. Active learning with multi-label svm classification. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, page 14791485. AAAI Press.

Marco Lippi, Przemyslaw Palka, Giuseppe Contissa, Francesca Lagioia, Hans-Wolfgang Micklitz, Giovanni Sartor, and Paolo Torroni. 2018. CLAUDETTE: an automated detector of potentially unfair clauses in online terms of service. *CoRR*, abs/1805.01217.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 115124, New York, NY, USA. Association for Computing Machinery.

W. Liu, H. Wang, X. Shen, and I. Tsang. 2021. The emerging trends of multi-label learning. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach.

Jinghui Lu and Brian MacNamee. 2020. Investigating the effectiveness of representations based on pretrained transformer-based language models in active learning for labelling text datasets. *CoRR*, abs/2004.13138.

R. Monarch, R. Munro, and C.D. Manning. 2021. *Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-centered AI*. Manning.

Felipe Kenji Nakano, Ricardo Cerri, and Celine Vens. 2020. Active learning for hierarchical multi-label classification. *Data Mining and Knowledge Discovery*, 34(5):1496–1530.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. Large-scale multi-label text classification - revisiting neural networks. *ArXiv*, abs/1312.5419.

T. RayChaudhuri and L.G.C. Hamey. 1995. Minimisation of data collection by active learning. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 3, pages 1338–1341 vol.3.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM Comput. Surv.*, 54(9).

Simiao Ren, Yang Deng, Willie J Padilla, and Jordan Malof. 2022. Towards robust deep active learning for scientific computing. *arXiv preprint arXiv:2201.12632*.

Oscar Reyes, Carlos Morell, and Sebastián Ventura. 2018. Effective active learning strategy for multi-label learning. *Neurocomputing*, 273:494–508.

Nicholas Roy and Andrew McCallum. 2001. Toward optimal active learning through sampling estimation of error reduction. In *ICML*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Christopher Schröder, Lydia Müller, Andreas Niekler, and Martin Potthast. 2023. Small-text: Active learning for text classification in python. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 84–95, Dubrovnik, Croatia. Association for Computational Linguistics.

Christopher Schröder and Andreas Niekler. 2020. A survey of active learning for text classification using deep neural networks. *CoRR*, abs/2008.07267.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. Revisiting uncertainty-based query strategies for active learning with transformers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland. Association for Computational Linguistics.

Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach.

Burr Settles. 2009. Active learning literature survey. *Technical Report TR-1648. University of Wisconsin-Madison. Department of Computer Sciences*.

Burr Settles. 2011. From theories to queries: Active learning in practice. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, volume 16 of *Proceedings of Machine Learning Research*, pages 1–18, Sardinia, Italy. PMLR.

Burr Settles, Mark Craven, and Soumya Ray. 2007. Multiple-instance active learning. In *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc.

Chuan Shi, Xiangnan Kong, Philip S. Yu, and Bai Wang. 2011. Multi-label ensemble learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 223–239, Berlin, Heidelberg. Springer Berlin Heidelberg.

Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. 2021. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118:107965.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2010. *Mining Multi-label Data*, pages 667–685. Springer US, Boston, MA.

Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, and Ioannis Vlahavas. 2011. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414.

Akim Tsvigun, Artem Shelmanov, Gleb Kuzmin, Leonid Sanochkin, Daniil Larionov, Gleb Gusev, Manvel Avetisian, and Leonid Zhukov. 2022. Towards computationally feasible deep active learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1198–1218, Seattle, United States. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zijie J. Wang, Dongjin Choi, Shenyu Xu, and Diyi Yang. 2021. Putting humans in the natural language processing loop: A survey. In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 47–52, Online. Association for Computational Linguistics.

Lukas Wertz, Katsiaryna Mirylenka, Jonas Kuhn, and Jasmina Bogojeska. 2022. Investigating active learning sampling strategies for extreme multi label text classification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4597–4605, Marseille, France. European Language Resources Association.

Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2021. A survey of human-in-the-loop for machine learning. *ArXiv*, abs/2108.00941.

Bishan Yang, Jian-Tao Sun, Tengjiao Wang, and Zheng Chen. 2009. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, page 917926, New York, NY, USA. Association for Computing Machinery.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. In *NeurIPS*.

Guoxian Yu, Xia Chen, Carlotta Domeniconi, Jun Wang, Zhao Li, Zili Zhang, and Xiangliang Zhang. 2022. Cmal: Cost-effective multi-label active learning by querying subexamples. *IEEE Transactions on Knowledge and Data Engineering*, 34(5):2091–2105.

Hsiang-Fu Yu, Kai Zhong, Inderjit S. Dhillon, Wei-Cheng Wang, and Yiming Yang. 2019. X-bert: extreme multi-label text classification using bidirectional encoder representations from transformers. In *NeurIPS 2019 Workshop on Science Meets Engineering of Deep Learning*.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948, Online. Association for Computational Linguistics.

Jing Zhang. 2022. Knowledge learning with crowdsourcing: A brief review and systematic perspective. *IEEE/CAA Journal of Automatica Sinica*, 9(5):749–762.

Ye Zhang, Matthew Lease, and Byron C. Wallace. 2017. Active discriminative text representation learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 33863392. AAAI Press.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. A survey of active learning for natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6166–6190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.