Mind the User! Measures to More Accurately Evaluate the Practical Value of Active Learning Strategies

Julia Romberg
Department of Social Sciences
Heinrich Heine University Düsseldorf, Germany
julia.romberg@hhu.de

Abstract

One solution to limited annotation budgets is *active learning* (AL), a collaborative process of human and machine to strategically select a small but informative set of examples. While current measures optimize AL from a pure machine learning perspective, we argue that for a successful transfer into practice, additional criteria must target the second pillar of AL, the human annotator. In *text classification*, e.g., where practitioners regularly encounter datasets with an increased number of imbalanced classes, measures like F_1 fall short when finding all classes or identifying rare cases is required. We therefore introduce four measures that reflect class-related demands that users place on data acquisition. In a comprehensive comparison of uncertainty-based, diversity-based, and hybrid query strategies on six different datasets, we find that strong F_1 performance is not necessarily associated with full class coverage. Uncertainty sampling outperforms diversity sampling in selecting minority classes and covering classes more efficiently, while diversity sampling excels in selecting less monotonous batches. Our empirical findings emphasize that a holistic view is essential when evaluating AL approaches to ensure their usefulness in practice – the actual, but often overlooked, goal of development. To this end, standard measures for assessing the performance of text classification need to be complemented by such that more appropriately reflect user needs.

1 Introduction

A well-known problem in supervised machine learning (ML) is scenarios where there are limited resources (e.g., budget or time) to annotate data. One approach to solving this problem is *active learning* (AL; Cohn et al. 1996), a collaborative process between human and machine. Through targeted query strategies, AL aims to find a minimal subset of examples whose labels provide the most information for fitting a model.

In *text classification*, many applications have been found to benefit from AL, such as sentiment analysis, intent or topic detection (e.g., Li et al., 2012; Zhang and Zhang, 2019; Tong and Koller, 2001). In addition to these task-specific studies, increased efforts have been made to systematically evaluate the performance of AL strategies across different use cases (e.g., Settles, 2011; Siddhant and Lipton, 2018; Ein-Dor et al., 2020).

Yet many academic studies ignore crucial real-world factors, leading to flawed assessments of practical utility. Literature has pointed out several limitations, including: the difficulty of making a-priori forecasts about the practical value of strategies (Lowell et al., 2019); the fact that actively acquired datasets are often only effective coupled with the respective model (Lowell et al., 2019; Tomanek and Morik, 2011); the need for out-of-distribution generalization (Longpre et al., 2022); taking into account class imbalance that is regularly encountered in real-world text classification (Ein-Dor et al., 2020); and the consideration of extreme multi-label scenarios (Wertz et al., 2022).

While these works seek to optimize AL from a ML perspective, it has been largely neglected that users themselves can present significant challenges that may impact the success of AL. For instance, it has been found that the effectiveness of AL depends on the expertise of the annotators (Baldridge and Palmer, 2009). Furthermore, examples selected by acquisition functions tend to be more ambiguous in terms of class assignment, leading to an increase in annotation uncertainty (Settles, 2011) and annotation time (Hachey et al., 2005). Such details can affect and even challenge the entire AL process.

We therefore argue that a successful transition from research to practice requires a more holistic evaluation that targets both pillars of AL, the

Proceedings of Recent Advances in Natural Language Processing, pages 996–1006
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_107

machine learner and the human annotator. In this work, we focus primarily on the requirements that the human annotator places on a successful AL process. More precisely, we introduce evaluation measures that already take this perspective into account during the development phase of AL approaches, further referred to as "user-centric"[1].

Considering the frequent scenario of multi-class text classification with imbalanced classes (Ein-Dor et al., 2020; Wertz et al., 2022), we contribute through four novel measures that capture class-related demands in AL. We compare different query strategies coupled with BERT across six datasets and analyze the results from both a standard ML and a more user-centric perspective. Our findings indicate that the proposed measures can provide important insights into strengths and weaknesses of AL that complement existing approaches.

2 Related Work

In evaluating the performance of AL, predictive accuracy has generally been the main focus (Kottke et al., 2017). Prior work has relied on task-specific measures, such as accuracy and F_1. Less commonly, AL-specific measures like deficiency (Yanık and Sezgin, 2015) were used. In addition, several measures have addressed desirable characteristics of query strategies, such as uncertainty of the acquired examples (Yuan et al., 2020; Wang et al., 2022), diversity of the acquired examples (Zhdanov, 2019; Yuan et al., 2020), and representativeness w.r.t the full dataset (Zhu et al., 2008; Ein-Dor et al., 2020). The majority of these measures focus on the input or feature space, but representativeness has also been measured in the output label space (Prabhu et al., 2019; Chaudhary et al., 2021). Another focus besides predictive accuracy has been on the computational effort (Schröder et al., 2022).

With a strong emphasis on ML performance, the current measures tend to overlook the human component in the real-world application of AL. Although user studies have proven helpful in uncovering user-centric pitfalls that can get in the way of practicality (Settles, 2011; Peshterliev et al., 2019), they are expensive and time-consuming, which is why they are often avoided in research. To overcome this hurdle, Calma and Sick (2017) suggested to simulate user factors from real-world applications when evaluating AL in an experimental setup (i.e., benchmarking on an already labeled dataset). They addressed error-proneness in AL and presented a theoretical framework for simulating annotation uncertainty of the user.

Our work follows this lead by incorporating user factors into the laboratory evaluation of AL to provide a simple alternative to costly user studies. However, we focus on the requirements that users place on AL applications in order for them to be considered beneficial in practice. In particular, we address the need for achieving high or full class coverage in a timely manner and covering minority classes. Furthermore, as a solution approach to the annotation uncertainty problem modeled by Calma and Sick (2017), we hypothesize how examples should be acquired to reduce annotation errors and introduce a corresponding measure.

3 Methodology

In this section, we first give a more formal introduction to AL. Then, we motivate and define the four user-centric measures that are central to this work.

3.1 Active Learning

We make use of the pool-based AL scenario (Lewis and Gale, 1994), which assumes that there is a large pool of unlabeled data $\mathcal{U}$ and a small set of labeled data $\mathcal{L}$ at the beginning. We decided to acquire examples in mini-batches, as a practical method.

AL proceeds according to the following scheme: Using some query strategy, a batch $\mathcal{B}$ of examples is selected (and consequently removed) from $\mathcal{U}$. These examples are then labeled by an oracle (e.g., a human annotator) and added to $\mathcal{L}$. Finally, a model is fit to $\mathcal{L}$. This process is repeated until a predefined stop criterion (e.g., a given annotation budget) is met. In the initial run, a default set of labeled examples is used to start the AL process.

3.2 Measures from User-Centric Perspective

In the following, we introduce four measures that reflect demands users may place on AL in practice. The definitions refer to single-label classification.

We draw motivation for the measures from two sources. On the one hand, we refer to the scientific literature, as specified below. On the other hand, we relate directly to the needs of practical users that have been communicated to us in our transdis-

[1] In the following, we will use the terms human annotator and user interchangeably. This terminology is adopted because in certain application scenarios, the human role goes beyond simply annotating data, as AL can simultaneously serve as an analytical tool, e.g., for computational social science.

ciplinary work over several years (among others documented in Romberg and Escher, 2020).

Minority-aware Batch Distribution When "dealing with imbalanced datasets in practice, the rare classes are often the ones that are particularly interesting." as Wertz et al. (2022) state. This is especially true for real-world use cases where AL is used not only for effective dataset creation, but also for efficient dataset analysis (Bonikowski et al., 2022; Yang et al., 2022). In the topic classification of citizens' contributions, e.g., human evaluators are often aware of the common issues in advance (Romberg and Escher, 2022). Thus, from the user's point of view, preference should be given to unexpected classes, which usually corresponds to minority classes. We measure this demand by

$$M(\mathcal{B}) = \frac{1}{n_{\mathcal{B}}} \sum_{c \in C} (1 - \frac{n_{\mathcal{U}_c}}{n_{\mathcal{U}}}) \cdot n_{\mathcal{B}_c} \tag{1}$$

where $n_{\mathcal{B}}$ is the batch size, $n_{\mathcal{U}}$ is the number of examples in $\mathcal{U}$, $n_{\mathcal{U}_c}$ is the number of examples in $\mathcal{U}$ that belong to class c, and $n_{\mathcal{B}_c}$ denotes the number of examples in $\mathcal{B}$ that belong to class c. To give more emphasis to rare classes, we weight all classes by their counter probability of occurring in the initial pool of unlabeled data. $M(\mathcal{B}) \in [0, 1]$, and a higher value indicates more awareness.

Class Coverage It is also of interest to consider how many classes AL can find (Schröder et al., 2021; Wertz et al., 2022). Achieving a high or even full class coverage is desirable for several reasons.

Knowing how query strategies handle the set of classes can be critical to building trust in human-machine collaboration. Indeed, a concern of our practice partners was missing some classes. If there was any potential for incomplete class coverage, this could even be a reason to completely avoid using machine text classification in their use case.

Such needs can relate to task requirements to which the human analyst is also subject. Thus, in these situations, it is not enough to, e.g., simply educate users about the strengths and weaknesses of ML algorithms; ML must meet these requirements.

What is more, with respect to the previously described utilization of AL for data analysis, a timely overview of the collection is an often desired feature, which is given by a fast class coverage.

And overall, having as complete a representation as possible of the classes relevant to the task at hand is generally an important prerequisite for creating reliable datasets.

We measure the class coverage of the examples in $\mathcal{L}$ as

$$K(\mathcal{L}) = \frac{|C_{\mathcal{L}}|}{|C|} \tag{2}$$

where $C_{\mathcal{L}}$ is the set of classes included in $\mathcal{L}$, and C is the total set of classes in the collection.

As a further indicator, we define the full class coverage I_K of an AL experiment as the number of iterations it takes to cover all classes in C.

Variation-aware Batch Distribution The performance of human annotators can be affected by various factors, including declining concentration or fatigue (Calma et al., 2016). One reason for the (more rapid) onset of these factors can be batches that offer little alternation in terms of the classes to be annotated. To reduce error-proneness in annotation caused by monotonous batches, we propose batches to fulfill two conditions: they should represent the available classes (measured by the ratio of acquired to the total number of classes available), and the acquired examples should be uniformly distributed among classes to offer variety (measured via entropy):

$$V(\mathcal{B}) = \frac{|C_{\mathcal{B}}|}{|C_{\mathcal{B}} \cup C_{\mathcal{U}}|} \cdot \sum_{c \in C_{\mathcal{B}}} - \left(\frac{\frac{n_{\mathcal{B}c}}{n_{\mathcal{B}}} \cdot \log_2(\frac{n_{\mathcal{B}c}}{n_{\mathcal{B}}})}{\log_2(|C_{\mathcal{B}}|)} \right) \tag{3}$$

where $C_{\mathcal{B}}$ is the set of classes included in the batch and $C_{\mathcal{U}}$ is the set of classes in the unlabeled pool. $V(\mathcal{B}) \in [0, 1]$, with larger values indicating a more varied set of examples with reference to the classes.

4 Evaluation Design

We provide an overview of the study design next by going into detail about the dataset selection, the chosen classification model, the selection of query strategies, and the experimental setup.

4.1 Datasets

We aim at a broad comparison across different datasets to empirically demonstrate the strengths and weaknesses of different query strategies with respect to the introduced user-centric measures. In doing so, we consider six datasets for different multi-class tasks and from diverse domains. An overview is given in Table 1.

DBPedia (Zhang et al., 2015) is a large-scale ontology dataset of Wikipedia articles (title and

Dataset	Task	Domain	$\|C\|$	Train	Val	Test
DBPedia	T	Wikipedia	14	$15,000$	$2,000$	$4,000$
20NG	T	News	20	$2,507$	354	721
ATIS	I	Flight reservations	17	$3,802$	537	$1,093$
TREC-50	Q	Diverse	46	$4,163$	589	$1,196$
BILLS	T	Congressional bills	20	$15,000$	$2,000$	$4,000$
CDB	T	Public participation	29	$1,372$	194	395

Table 1: Details of the six datasets. The task types are topic (T), intent (I), and question (Q) classification. $|C|$ denotes the number of classes.

abstract) and their topics. 20 Newsgroups[2] (20NG) contains messages collected from diverse newsgroups. Airline Travel Information Systems (ATIS; Siddhant and Lipton, 2018) is a dataset of transcribed audio recordings for classifying the intent of costumer utterances. TREC (Li and Roth, 2002) provides answer types for a collection of English-language questions.

These four English-language datasets regularly serve for benchmarking AL. While previous work has mostly relied on TREC-6, which organizes the questions into six main categories, we use the finer answer types of TREC-50 to give more weight to the multi-class setting that motivates this work.

The remaining two datasets come from real-world applications of topic classification in the computational social sciences. The Congressional Bills Corpus (BILLS; Purpura et al., 2008) provides information on bills introduced in the U.S. Congress between 1947 and 2008. One of its purposes is to examine what attention the congress has paid to various issues by thematically analyzing the bill's titles. The Cycling Dialogues Bonn (CDB; Romberg and Escher, 2022) is a German dataset of citizen contributions to a public participation process on cycling infrastructure.

While ATIS, TREC-50, BILLS, and CDB reflect the common class imbalance of real-world data, DBPedia and 20NG have been artificially counterbalanced at creation. To simulate a plausible scenario, we adjust the distribution of the two datasets through sub-sampling. Since we lack knowledge about the original data sources' actual distributions, we assume a distribution according to Zipf's law: the most frequent class should occur about twice as often as the second most frequent class, three times as often as the third most frequent class, and so on.

We follow Ein-Dor et al. (2020) by limiting the size of large datasets to $21K$ (DBPedia and BILLS) and apply a $70\%/10\%/20\%$ split for training, validation and testing. There were predefined splits available for some of the datasets (train/test splits for TREC-50 and 20NG; a train/val/test split for DBPedia), which we rejected for the following reasons: For TREC these are neither consistent in their distribution (Lowell et al., 2019), nor does the test split for TREC-50 contain all of the original 47 classes. For 20NG and DBPedia, we modified the structure of the datasets to a greater extent by adapting them to Zipf's distribution. We therefore decided to define new splits selected according to a stratified random sample. Classes with less than 5 examples were removed.

Detailed insights into the resulting dataset splits and the code for the experiments are available at https://github.com/juliaromberg/ranlp-2023.

4.2 Classification Model

Several studies have shown the potential of AL coupled with pre-trained language models (PTMs) (e.g., Ein-Dor et al. 2020; Yuan et al. 2020; Longpre et al. 2022; Zhang et al. 2022). We adhere to these findings and apply the BERT base model (Devlin et al., 2019), as has been done in much of the related work. For English datasets, we use uncased BERT[3] (pre-trained on English data), and for the German dataset, we rely on cased GBERT[4].

4.3 Query Strategies

We compare a variety of strategies that have stood out in previous work for their strong results and cost-effectiveness when used with PTMs in imbalanced settings. As a baseline, we use *Random Sampling* (Random).

Traditional uncertainty-based acquisition functions select examples according to the confidence of model prediction. They are efficient and have proven to keep up with more advanced AL strategies when used with PTMs (Zhang and Zhang, 2019; Margatina et al., 2021, 2022). We consider *Least Confidence* (LC; Lewis and Gale, 1994), which has proven effective for imbalanced datasets (Ein-Dor et al., 2020; Schröder et al., 2022), and *Breaking Ties* (BT; Luo et al., 2005), which was recommended as a baseline for uncertainty sampling with transformers by Schröder et al. (2022). LC selects those examples for annotation where the model's probability output is lowest for the most likely class, i.e., cases in which the model is least

[2] http://qwone.com/ jason/20Newsgroups/

[3] https://huggingface.co/bert-base-uncased

[4] https://huggingface.co/deepset/gbert-base

confident. BT aims to improve classification confidence by selecting examples where the difference in probability outputs between the two most likely classes is the smallest.

Diversity-based query strategies aim to select examples that best represent the full dataset. We include *Core-Sets* (Sener and Savarese, 2018), which have been found to select batches of high diversity and representativeness in addition to a promising boost of model performance in imbalanced settings (Ein-Dor et al., 2020). Core-sets are subsets of examples that represent the dataset in a learned feature space (for PTMs: CLS) in the sense that a model trained on a Core-set is competitive to a model trained on the entire dataset. We rely on the lightweight and fast algorithm for building the Core-sets by Bachem et al. (2018).

As a proxy for functions with a hybrid objective, we choose *Contrastive Active Learning* (CAL; Margatina et al., 2021) which has the potential to outperform alternatives such as BADGE (Ash et al., 2020) and ALPS (Yuan et al., 2020) in terms of computational efficiency and accuracy (Margatina et al., 2021). CAL combines the characteristics of uncertainty- and diversity-based strategies by seeking so-called contrastive examples. These are examples that, despite high similarity in the feature space (i.e., among the k nearest neighbors), exhibit maximum mean Kullback-Leibler divergence between their predictive likelihoods.

4.4 Experimental Setup

In each AL iteration, training runs for 30 epochs on a batch size of 12 and the best model, in terms of validation loss, is retained. To avoid overfitting to the data from previous iterations, BERT is fine-tuned from scratch at each iteration (Hu et al., 2019). We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of $2e-5$, beta coefficients of 0.9 and 0.999, and an epsilon of $1e-8$, and set the maximum sequence length to 100 for all datasets.

For each of the six datasets, the unlabeled pool $\mathcal{U}$ is formed by the respective training splits and 50 examples are randomly sampled from the pool to build the set of initially labeled data $\mathcal{L}$. Then, 20 iterations of AL are performed, in each of which a new batch of 50 unlabeled examples is selected from $\mathcal{U}$ according to the respective query strategy. The model performance is evaluated at the end of each iteration using a hold-out test set.

We run the AL simulation five times with different sets of initially labeled data for each combination (datasets $\times$ query strategies). To allow for a fair comparison, these seeds remain the same for each dataset across the different query strategies.

In accordance with our experimental setup, $3,156$ experiments (6 datasets $\times$ (5 query strategies $\times$ 5 initial seeds $\times$ (1 initial model $+$ 20 iterations) $+$ 1 full supervision model)) were conducted. The experiments were run on a single Nvidia Tesla P100-PCIE-16GB GPU and with 2.2 GHz Intel Xeon CPU processor.

We refer the reader to Appendix A for further details on hyperparameter selection, reproducibility of the experiments and computational costs.

5 Results

In this section, we report the experimental results. We start by shedding light on the performance of the different query strategies as is common in the literature via a standard measure for classification tasks, in our case the F_1 score. Using the newly introduced user-centric measures, we then shift our focus to analyzing additional indicators that can help select an appropriate query strategy for practical use.

5.1 F_1 Performance

Figure 1 illustrates how the F_1 score evolves over the iterations of AL in the experiments. It can be seen that full supervision performance can be achieved on all datasets within the chosen annotation budget of 20 iterations, except for BILLS.

Our analysis across all datasets shows a clear pattern of superior performance for uncertainty-based sampling compared to the other strategies. In particular, BT performs consistently strong. While hybrid CAL is in the middle of the rankings, it is evident that the diversity-based strategy mostly underperforms.

Based on these findings, from a ML-perspective that is commonly shared among many studies in the field, it seems an obvious conclusion to recommend BT as the strategy for practical application in imbalanced multi-class settings. In the following, we will examine whether this assumption can be supported from a user-centric perspective.

5.2 User-Centric Measures

Table 2 lists the results of the four user-centric measures for the datasets and query strategies, averaged

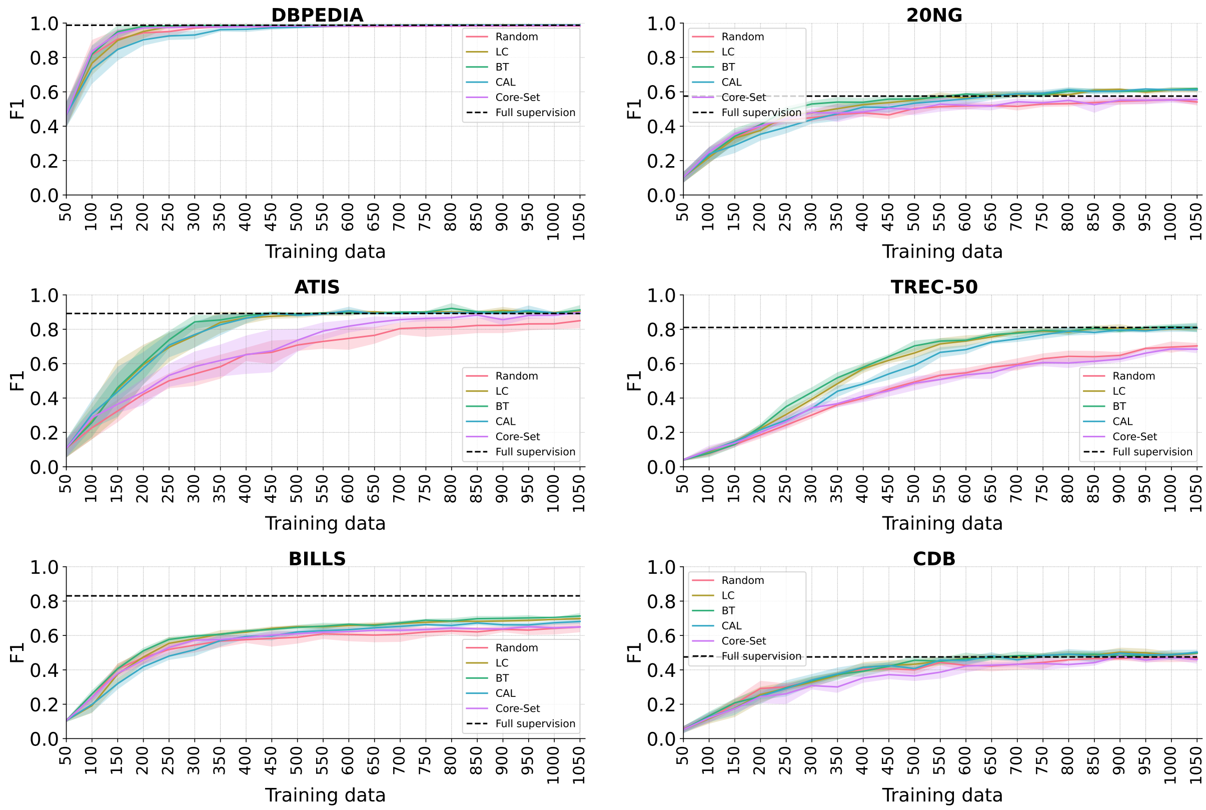

Figure 1: F_1 scores, averaged over the five seeds and with the shaded area illustrating the standard deviation. As a reference for the maximum achievable F_1 score for each dataset, the performance of the BERT models trained on the complete training data is indicated (full supervision).

over the iterations of AL for a better overview.

Which strategies favor minority classes? First, we evaluate whether, among the strategies considered, there are such that promote a higher representation of rare classes in the batches. We apply the minority-aware batch distribution measure $M(\mathcal{B})$ for this purpose.

All advanced strategies are found to consider rare classes more than random sampling. In particular, uncertainty-based strategies promote a higher minority representation on average. A detailed look shows that this trend is consistent among datasets, but there are major differences in how pivotal the choice of query strategy is. For BILLS and CDB, this makes a negligible difference. In contrast, the effect is much more dramatic on ATIS, where the scores range from 0.44 to 0.84.

Which strategies favor class coverage? Next, we examine whether there are any query strategies that prioritize quick and extensive class coverage by applying the class coverage measure $K(\mathcal{L})$.

The results show that uncertainty-based and hybrid query strategies stand out positively. BT achieves the highest average class coverage and turns out to be a good choice for a rapid growth in the coverage curve (as a detailed look at progress between iterations confirms).

Are the strategies capable of finding all classes? As argued in Section 3.2, a realistic requirement of the practice may be that all classes that a dataset comprises are found in the AL process. We measure the full coverage with I_K.

Contrary to our expectation, three strategies failed to find all classes within the budget of 20 annotation cycles on the datasets ATIS and TREC-50. In addition to random sampling and Core-Sets, in TREC-50 this surprisingly also affects the previously excelling strategy BT. The failure is systematic in each case, as we can observe it for several random seeds.

To gain better insight into the extent of the failure, we ran additional experiments beyond the AL budget of 20 iterations until full class coverage was achieved for the affected cases. On TREC-50, Core-Sets and BT both required up to 28 iterations on average. However, the deviations between the different seeds are much more extreme with BT: In

	Random	LC	BT	CAL	Core-Set
			$M(\mathcal{B})$		
DBPedia	0.852 ± 0.003	$\mathbf{0.918} \pm 0.001$	0.916 ± 0.002	0.916 ± 0.005	0.870 ± 0.002
20NG	0.874 ± 0.002	$\mathbf{0.930} \pm 0.001$	0.928 ± 0.003	0.924 ± 0.001	0.888 ± 0.001
ATIS	0.440 ± 0.006	$\mathbf{0.840} \pm 0.012$	$\mathbf{0.840} \pm 0.007$	0.735 ± 0.009	0.586 ± 0.010
TREC-50	0.925 ± 0.002	$\mathbf{0.947} \pm 0.001$	0.945 ± 0.001	$\mathbf{0.947} \pm 0.001$	0.928 ± 0.001
BILLS	0.918 ± 0.001	$\mathbf{0.931} \pm 0.001$	$\mathbf{0.931} \pm 0.001$	0.928 ± 0.000	0.924 ± 0.001
CDB	0.933 ± 0.001	$\mathbf{0.937} \pm 0.001$	0.936 ± 0.001	0.934 ± 0.000	0.933 ± 0.001
AVG	0.824 ± 0.003	$\mathbf{0.917} \pm 0.003$	0.916 ± 0.002	0.897 ± 0.003	0.855 ± 0.003
			$K(\mathcal{L})$		
DBPedia	0.995 ± 0.023	0.995 ± 0.024	0.995 ± 0.023	0.995 ± 0.026	$\mathbf{0.996} \pm 0.022$
20NG	0.971 ± 0.076	0.979 ± 0.071	$\mathbf{0.982} \pm 0.067$	0.977 ± 0.072	0.977 ± 0.072
ATIS	0.864 ± 0.143	0.915 ± 0.162	$\mathbf{0.926} \pm 0.149$	0.924 ± 0.157	0.867 ± 0.137
TREC-50	0.847 ± 0.138	0.869 ± 0.159	$\mathbf{0.889} \pm 0.151$	0.881 ± 0.161	0.822 ± 0.136
BILLS	0.979 ± 0.051	0.981 ± 0.051	$\mathbf{0.984} \pm 0.048$	0.978 ± 0.056	0.983 ± 0.049
CDB	0.958 ± 0.085	$\mathbf{0.968} \pm 0.077$	0.962 ± 0.080	0.964 ± 0.082	0.962 ± 0.083
AVG	0.936 ± 0.086	0.951 ± 0.091	$\mathbf{0.956} \pm 0.086$	0.953 ± 0.092	0.934 ± 0.083
			I_K		
DBPedia	1.0 ± 1.2	1.2 ± 1.3	1.0 ± 1.2	1.0 ± 1.0	$\mathbf{0.8} \pm 0.8$
20NG	4.2 ± 0.8	2.6 ± 0.9	$\mathbf{2.0} \pm 1.2$	2.6 ± 0.9	2.8 ± 1.3
ATIS	$26.6 \pm 16.4^*$	8.0 ± 2.4	8.8 ± 2.1	$\mathbf{7.6} \pm 1.3$	$22.8 \pm 6.8^*$
TREC-50	$35.2 \pm 8.1^*$	16.2 ± 2.9	$28.0 \pm 23.8^*$	$\mathbf{15.8} \pm 2.7$	$27.8 \pm 5.9^*$
BILLS	4.4 ± 0.9	3.2 ± 0.5	$\mathbf{3.0} \pm 1.2$	3.8 ± 1.1	3.4 ± 2.5
CDB	7.6 ± 2.4	5.8 ± 1.6	6.6 ± 1.1	$\mathbf{5.0} \pm 0.0$	7.0 ± 2.6
AVG	13.2 ± 5.0	6.2 ± 1.6	8.2 ± 5.1	$\mathbf{6.0} \pm 1.2$	10.8 ± 3.3
			$V(\mathcal{B})$		
DBPedia	0.736 ± 0.017	0.516 ± 0.037	0.600 ± 0.018	0.474 ± 0.060	$\mathbf{0.785} \pm 0.007$
20NG	0.636 ± 0.018	0.761 ± 0.008	$\mathbf{0.791} \pm 0.009$	0.737 ± 0.030	0.688 ± 0.014
ATIS	0.216 ± 0.009	0.381 ± 0.020	0.391 ± 0.026	$\mathbf{0.458} \pm 0.007$	0.376 ± 0.010
TREC-50	0.388 ± 0.011	0.393 ± 0.013	$\mathbf{0.426} \pm 0.012$	0.388 ± 0.014	0.400 ± 0.007
BILLS	0.696 ± 0.009	0.676 ± 0.009	0.738 ± 0.019	0.637 ± 0.015	$\mathbf{0.742} \pm 0.016$
CDB	0.606 ± 0.009	0.605 ± 0.016	$\mathbf{0.617} \pm 0.013$	0.581 ± 0.009	0.607 ± 0.006
AVG	0.493 ± 0.012	0.478 ± 0.020	0.512 ± 0.016	0.477 ± 0.021	$\mathbf{0.539} \pm \mathbf{0.008}$

Table 2: Detailed results for $M(\mathcal{B})$, $K(\mathcal{L})$, I_K, and $V(\mathcal{B})$ on the six datasets of evaluation. The scores are averaged over the seeds and iterations of AL, and standard deviation is stated. The best scores are marked in bold. Cases in which a strategy failed to reach full coverage within the given budget are marked with an asterix.

the worst case, BT asked for manual labeling of over three quarters of the pool $\mathcal{U}$, which sums up to 60 iterations of AL.

We further discovered that in case of incomplete class coverage, it was the minority classes that were not found. This is why we repeated the experiments for TREC-50 and ATIS with an increased required minimum class support of 20 to spot check how performance changes. As for Random and Core-Sets, this modification allowed all experiments to achieve full class coverage within the given annotation budget. However, for BT, the undesired effects persisted on TREC-50. Moreover, failure even extended to the other two strategies associated with uncertainty, namely LC and CAL.

Overall, in the average comparison between all strategies, the hybrid CAL stands out, requiring on average only 6 iterations to successfully detect all classes.

How variant are the batches in terms of classes? Last, we apply $V(\mathcal{B})$ in order to account for variance in batches with the goal of reducing monotonous patterns.

Here, it is the diversity-based query strategy Core-Sets that on average produces batches that best fulfill the condition. Individually, though, the results are very mixed for the different acquisition functions and datasets. For example, BT performs best on three of the datasets, rendering this query strategy a strong contender.

6 Discussion

We considered several measures that take into account aspects that may determine the practicality of active learning strategies with respect to specific application scenarios. For the datasets under consideration, it can be seen that the F_1 score, the rapidity of class coverage, and the minority-awareness in the batches advocate for the use of uncertainty-based acquisition functions, in particular BT, in practical scenarios with multiple and imbalanced classes. However, Core-Sets offer the opportunity to add more variety to the monotonous task of annotation by filling batches with rather different classes and in a more balanced way. This may potentially help prevent annotation fatigue and thus human annotation errors that negatively impact AL. In addition, such variation could be a plus in terms of usability.

What is more, we found weaknesses in reaching full class coverage for all strategies. For random sampling and Core-Sets, we hypothesize that this is caused by extremely rare classes. However, for uncertainty sampling, the problem became even more apparent when excluding those classes. This is of particular interest since full supervision F_1 can be well achieved within the annotation budget (see Figure 1).

Although the F_1 score and some user-centric measures recommend BT as a favorite, the lack of reliability in achieving full class coverage, which we have empirically determined, may become a decisive criterion for practical applicability. Not only can it have a significant impact on human trust in AL. This finding affects AL in general, as the reliability of models strongly depends on the quality of the datasets.

7 Conclusion

With our results, we were able to illustrate that different query strategies stand out in different aspects that might be desirable or even necessary from the user's perspective in the practical application of AL. So what implications can be drawn for AL research beyond this study? The main reason why research on AL exists is its development and improvement for real-world use. In this, AL is a collaberative interaction between human and machine. However, this particular feature of AL seems to have gradually faded from the community's awareness, with the main focus being on optimizing the established performance measure for the particular machine learning task, e.g. classification. It is true that these established measures have important informational value about the methods. But there are additional requirements that arise specifically from the human factor inherent in the nature of AL, which likewise impact the practical value of AL. These should therefore be taken into account.

Therefore, we argue that future studies on AL should report a wider range of measures in their experimental evaluation. With this broader foundation, practitioners will be able to make a more informed decision when selecting an AL strategy based on academic findings in order to comply with their specific needs for a given application. For example, in applications where the annotation step is simultaneously used to analyze the dataset at hand, features such as a quick overview of all classes or, in particular, minority classes can be desired, as we have discussed in more detail in Section 3.2. Surely, the measures we have suggested are by no means exhaustive. Therefore, this work should also serve as a motivation to cover other aspects of the human component of AL in future research.

Ultimately, selecting an appropriate AL strategy for some practical use case is a matter of balancing different needs. The suggested measures make an important contribution to this, as they enable more reflective decisions, especially in combination with common performance measures like the F_1 score.

To sum up, AL has the potential to support ML in scenarios where the annotation budget is limited. We have argued that in order to assist the transfer of such methods from research to practice, both the machine learner and the human annotator must be taken into account. Considering the frequent use case of multi-class text classification with imbalanced classes, we introduced four measures that evaluate the acquired examples w.r.t. class-related requirements from the user's point of view. These measures are based on scientific literature and practical experience. Our results show that as complete a picture as possible should be considered to avoid failures in practical application.

The next step will be to conduct a user study to validate the usefulness of the metrics presented here. In future work, we will also investigate in more detail which influencing factors prevent a fast finding of all classes. This necessitates a study that investigates, among other aspects, the effect of data distribution on the class coverage of the different strategies in order to draw general conclusions.

Acknowledgments

Thanks to Tobias Escher, Gerhard Klassen and Sergej Korlakov for valuable discussions and feedback on earlier forms of this work. This publication is based on research in the project CIMT/Partizipationsnutzen, which is funded by the Federal Ministry of Education and Research of Germany as part of its Social-Ecological Research funding priority, funding no. 01UU1904. Respon-

sibility for the content of this publication lies with the author.

References

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*.

Olivier Bachem, Mario Lucic, and Andreas Krause. 2018. Scalable k-means clustering via lightweight coresets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 1119–1127.

Jason Baldridge and Alexis Palmer. 2009. How well does active learning *actually* work? Time-based evaluation of cost-reduction strategies for language documentation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 296–305.

Bart Bonikowski, Yuchen Luo, and Oscar Stuhler. 2022. Politics as usual? Measuring populism, nationalism, and authoritarianism in U.S. presidential campaigns (1952–2020) with neural language models. *Sociological Methods & Research*, 51(4):1721–1787.

Adrian Calma, Jan Marco Leimeister, Paul Lukowicz, Sarah Oeste-Reiß, Tobias Reitmaier, Albrecht Schmidt, Bernhard Sick, Gerd Stumme, and Katharina Anna Zweig. 2016. From active learning to dedicated collaborative interactive learning. In *29th International Conference on Architecture of Computing Systems*, pages 1–8.

Adrian Calma and Bernhard Sick. 2017. Simulation of annotators for active learning: Uncertain oracles. In *Proceedings of the Workshop and Tutorial on Interactive Adaptive Learning*, pages 49–58.

Aditi Chaudhary, Antonios Anastasopoulos, Zaid Sheikh, and Graham Neubig. 2021. Reducing confusion in active learning for part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 9:1–16.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145.

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.

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*, pages 7949–7962.

Ben Hachey, Beatrice Alex, and Markus Becker. 2005. Investigating the effects of selective sampling on the annotation task. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 144–151.

Peiyun Hu, Zachary C Lipton, Anima Anandkumar, and Deva Ramanan. 2019. Active learning with partial feedback. In *International Conference on Learning Representations*.

Daniel Kottke, Adrian Calma, Denis Huseljic, G. M. Krempl, and Bernhard Sick. 2017. Challenges of reliable, realistic and comparable active learning evaluation. In *Proceedings of the Workshop and Tutorial on Interactive Adaptive Learning*, pages 2–14.

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12.

Shoushan Li, Shengfeng Ju, Guodong Zhou, and Xiaojun Li. 2012. Active learning for imbalanced sentiment classification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 139–148.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 556–562.

Shayne Longpre, Julia Reisler, Edward Greg Huang, Yi Lu, Andrew Frank, Nikhil Ramesh, and Chris DuBois. 2022. Active learning over multiple domains in natural language tasks. *arXiv preprint*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

David Lowell, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical obstacles to deploying active learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 21–30.

Tong Luo, Kurt Kramer, Dmitry B. Goldgof, Lawrence O. Hall, Scott Samson, Andrew Remsen, Thomas Hopkins, and David Cohn. 2005. Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research*, 6(4):589–613.

Katerina Margatina, Loic Barrault, and Nikolaos Aletras. 2022. On the importance of effectively adapting pretrained language models for active learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 825–836.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663.

Stanislav Peshterliev, John Kearney, Abhyuday Jagannatha, Imre Kiss, and Spyros Matsoukas. 2019. Active learning for new domains in natural language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 90–96.

Ameya Prabhu, Charles Dognin, and Maneesh Singh. 2019. Sampling bias in deep active classification: An empirical study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4058–4068.

Stephen Purpura, John Wilkerson, and Dustin Hillard. 2008. The U.S. policy agenda legislation corpus volume 1 – a language resource from 1947 - 1998. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation*, pages 403–409.

Julia Romberg and Tobias Escher. 2020. Analyse der Anforderungen an eine Software zur (teil-) automatisierten Unterstützung bei der Auswertung von Beteiligungsverfahren. Working Paper 1, CIMT Research Group, Heinrich Heine University Düsseldorf.

Julia Romberg and Tobias Escher. 2022. Automated topic categorisation of citizens' contributions: Reducing manual labelling efforts through active learning. In *Electronic Government*, pages 369–385.

Christopher Schröder, Kim Bürgl, Yves Annanias, Andreas Niekler, Lydia Müller, Daniel Wiegreffe, Christian Bender, Christoph Mengs, Gerik Scheuermann, and Gerhard Heyer. 2021. Supporting land reuse of former open pit mining sites using text classification and active learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4141–4152.

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.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*.

Burr Settles. 2011. Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1467–1478.

Aditya Siddhant and Zachary C. Lipton. 2018. Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909.

Katrin Tomanek and Katherina Morik. 2011. Inspecting sample reusability for active learning. In *Active Learning and Experimental Design workshop in conjunction with AISTATS 2010*, pages 169–181.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2:45–66.

Yuxia Wang, Daniel Beck, Timothy Baldwin, and Karin Verspoor. 2022. Uncertainty estimation and reduction of pre-trained models for text regression. *Transactions of the Association for Computational Linguistics*, 10:680–696.

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.

Eugene Yang, Sean MacAvaney, David D. Lewis, and Ophir Frieder. 2022. Goldilocks: Just-right tuning of BERT for technology-assisted review. In *Proceedings of the European Conference on Information Retrieval*, page 502–517.

Erelcan Yanık and Tevfik Metin Sezgin. 2015. Active learning for sketch recognition. *Computers & Graphics*, 52:93–105.

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*, pages 7935–7948.

Leihan Zhang and Le Zhang. 2019. An ensemble deep active learning method for intent classification. In *Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence*, pages 107–111.

Shujian Zhang, Chengyue Gong, Xingchao Liu, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. 2022. ALLSH: Active learning guided by local sensitivity and hardness. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1328–1342.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*.

Fedor Zhdanov. 2019. Diverse mini-batch active learning. *arXiv preprint*.

Jingbo Zhu, Huizhen Wang, Tianshun Yao, and Benjamin K. Tsou. 2008. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 1137–1144.

Appendix

A Implementation Details

Hyperparameters The choice of batch size, number of training epochs, and maximum sequence length is a tradeoff between model performance, runtime, and GPU restrictions. We empirically determined that setting the batch size to 12 yielded good results. As for the number of 30 training epochs, we found that model prediction benefits from this increased number especially when there are only a few labeled examples, but also as the AL process progresses. Future work may consider whether the number of epochs can be curtailed as $\mathcal{L}$ grows larger. In consideration with the runtime due to the chosen number of epochs and the total number of experiments, as well as with regard to GPU constraints, we decided on an overall maximum sequence length of 100. For TREC-50 and ATIS, the longest encountered sequence comprises only 41 respectively 52 tokens, so we set the maximum sequence length correspondingly lower in these cases.

Reproducibility Experiments were performed with the same five random seeds, randomly selected from the range $[1, 9999]$, to make them reproducible.

Computational Costs Table 3 provides the average duration of each AL experiment. The decisive factor for the runtime is model fine-tuning.

Full Supervision Models These (c.f. Figure 1 in the main body) were fit on the full training data of the respective dataset with AdamW, $lr = 2e-5$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1e-8$. We trained for five epochs in case of large datasets (DBPedia, BILLS) and for 30 epochs in case of small datasets (20NG, ATIS, TREC-50, CDB), and selected the best model by validation loss. To obtain reliable results, we repeated each experiment five times with different random seeds.

	Random	LC	BT	CAL	Core-Set
DBPEDIA	613	672	670	682	675
20NG	466	474	475	475	473
ATIS	422	442	435	447	436
TREC-50	387	422	405	412	411
BILLS	611	712	710	678	665
CDB	545	561	536	560	547

Table 3: Average runtime (seconds) including model training, inference, batch acquisition, and hold-out test set prediction.