Value- Aware Active Learning

Sayin, Burcu; Yang, Jie; Passerini, Andrea; Casati, Fabio

DOI
10.3233/FAIA230085

Publication date
2023

Document Version
Final published version

Published in
HHAI 2023

HHAI 2023: Augmenting Human Intellect
P. Lukowicz et al. (Eds.)

doi:10.3233/FAIA230085

Value-Aware Active Learning

Burcu SAYIN [a,1], Jie YANG [b], Andrea PASSERINI [a] and Fabio CASATI [c]
[a] *University of Trento, Italy*
[b] *Delft University of Technology, Netherlands*
[c] *Servicenow, Santa Clara, USA*

Abstract. In many practical applications, machine learning models are embedded into a pipeline involving a human actor that decides whether to trust the machine prediction or take a default route (e.g., classify the example herself). *Selective classifiers* have the option to abstain from making a prediction on an example they do not feel confident about. Recently, the notion of the *value* of a machine learning model has been introduced as a way to jointly consider the benefit of a correct prediction, the cost of an error, and that of abstaining. In this paper, we study how active learning of selective classifiers is affected by the focus on *value*. We show that the performance of the state-of-the-art active learning strategies drops significantly when we evaluate them based on *value* rather than accuracy. Finally, we propose a novel value-aware active learning strategy that outperforms the state-of-the-art ones when the cost of incorrect predictions substantially outweighs that of abstaining.

Keywords. active learning, value-based learning, cost-sensitive learning, selective classifier

1. Introduction

In most real-world applications, machine learning (ML) models are not used as standalone systems, but rather they are incorporated into processing pipelines that involve some kind of human contribution. In these systems, ML models are typically employed as selective classifiers [1,2], that can abstain from providing predictions when they do not feel confident enough (or when somebody else decides that this is the case) and a default route is chosen instead, which most often involves asking humans. In our recent papers [3,4,5], we showed that what follows from these simple observations is that standard evaluation metrics, like accuracy or F1-score, are inadequate to capture the *value* an ML model brings to a use case. This is because they fail to account for the presence of an "I do not know" option, and its implication in terms of the relative costs of different alternatives and their potential errors. To overcome this limitation, we proposed an alternative evaluation metric that measures the *value* of an ML model in a processing pipeline [4,5]. In the simplest and most popular approach to selective classification, namely rejecting instances with prediction confidence below a given threshold, we showed how to compute a threshold maximizing value as a function of the cost factor of errors, assuming that predictions come from a calibrated classifier. Results showed that evaluating threshold-based selective classifiers in terms of value rather than standard metrics leads to substantial changes in deciding which model is the most appropriate for a given scenario.

[1] Corresponding Author: Burcu Sayin, burcu.sayin@unitn.it

In this paper, we study whether these findings have an impact on active learning (AL) scenarios. In particular, we investigate whether uncertainty sampling [6], the de-facto standard in AL, is still a reasonable strategy when evaluating models in terms of value. The intuition that drives our work is as follows: the rationale of uncertainty sampling is to focus on examples on which the model is most uncertain, so as to improve the separation boundary between classes. However, a threshold-based selective classifier works by rejecting all examples with confidence below the rejection threshold. If the cost of errors is high, this threshold can be quite far from the boundary separating classes. This implies that uncertainty sampling concentrates learning on areas of the space that could be irrelevant to the final decision because instances located there are always rejected for being predicted with insufficient confidence. To overcome this potential limitation, we proposed *threshold-oriented sampling* as a simple value-aware AL strategy that focuses on the most uncertain examples *around the rejection threshold*. An experimental evaluation on various multiclass classification tasks from the NLP domain confirms our intuition, showing that:

- uncertainty sampling performs well in terms of model value in standard settings where the classifier never abstains, but it is often worse than simple random sampling in selective classification settings, even for fairly small cost ratios,
- threshold-oriented sampling strikes the best balance between uncertainty and confidence, consistently outperforming all alternatives across datasets and AL settings.

Our source code is freely available on Github[2] to support reproducibility .

2. Model value

Assume that our model operates on items and returns either a predicted class or rejects predicting. Following our papers [4,5], we define the *value* V of this model as follows:

$$V = \rho V_r + (1-\rho)(\alpha V_c + (1-\alpha)V_w) \tag{1}$$

where V_r, V_c, and V_w are the value of rejecting an item, classifying it correctly, and classifying it wrongly, respectively, ρ is the proportion of predictions that are rejected and α is the accuracy for predictions above the threshold. Reality can be more complex and account for different costs associated with different types of errors (see the original papers for the details [4,5]), but this simplified model is both good enough for our purposes and simple enough to be easily grasped by business process owners when thinking of the benefit of ML in production. If we set the value of not having AI (before we put our ML model into production) to 0, and we consider that not having ML is equivalent to having a model that rejects all predictions, then we have $V_r = 0$. We can then define V_w in terms of V_c without loss of generality:

$$V_w = -kV_c \tag{2}$$

[2] https://github.com/burcusayin/value-aware-al

where k is a cost ratio of how bad an error with respect to how good is a correct prediction. By further setting $V_c = 1$, i.e., rescaling costs in terms of "units of V_c dollars", the *value* formula simplifies as follows:

$$V = (1-\rho)(\alpha - k(1-\alpha)) \tag{3}$$

and ML is helpful (i.e., it brings positive value) only if $(\alpha - k(1-\alpha))$ is positive. In the following, we assume that rejection of predictions is performed by having the classifier emit a prediction score or confidence, and by having the ML solution workflow filter predictions with a confidence lower than a threshold τ. In this setting, if the model is well calibrated [7] so that its confidence corresponds to the probability of being correct, we have that $\alpha = \tau$, and requiring $(\tau - k(1-\tau)) > 0$ implies setting τ as follows:

$$\tau = \frac{k}{k+1} \tag{4}$$

If the model is not calibrated, we can either recalibrate it with standard techniques like temperature scaling [8] and then apply the threshold in Eq. 4, or directly adjust the threshold to maximize value as computed on a validation set. Note that Eq. 4 implies that the greater the cost of errors k, the more selective we should be in accepting predictions. k is a parameter of the business problem, not of the model, and in general, given an ML model, the *value* will decrease linearly as k increases if τ remains constant. If we adjust τ as k increases, then *value* can decrease sub-linearly.

3. Value-aware active learning

In this section, we propose a simple solution to make existing AL strategies value-aware. The rationale is that examples to be labeled should be selected having in mind the fact that the classifier will be deployed as a selective classifier, and that learning should aim at maximizing value rather than standard classification metrics.

Let $f : \mathscr{X} \rightarrow \Delta^c$ be a probabilistic multiclass classifier producing a probability distribution $f(x)$ over the set of candidate classes $\mathscr{Y}$, with $c = |\mathscr{Y}|$. Let the *confidence score* $s(x)$ of f on x be the score of the highest scoring class according to $f(x)$, i.e.:

$$s(x) = \max_{y \in \mathscr{Y}} f_y(x)$$

Let's assume that after training, the classifier f will be employed as a threshold-based selective classifier with threshold τ, i.e., instances with a confidence score lower than τ will be rejected. This implies that instances with a confidence score *close* to τ are the most promising candidates to try increasing the value of such a classifier, by learning to increase the confidence score of correctly classified instances with a lower score, and decrease one of the classification errors. This is the same rationale that uncertainty sampling applies (in multiclass classification) to the *margin score*, i.e., the difference between the confidence score and the score of the second best. We thus adapt uncertainty sampling to become value-aware by sampling examples with confidence margin closest to the rejection threshold:

$$x^* = argmin_{x \in \mathscr{D}} |\tau - s(x)| \tag{5}$$

where $\mathscr{D}$ is the pool of unlabelled examples to choose from. Batch AL can be implemented by repeating the process in Eq. 5 for the desired number of examples, as customary in AL practice. We name this strategy ***threshold-oriented sampling (TOS)***. Note that despite their similarity, the two strategies tend to select rather different examples. Indeed, the confidence score of the most uncertain example is usually far away from the rejection threshold, especially for non-negligible values of the cost factor k. This implies that employing uncertainty sampling as an AL strategy to train a selective classifier tends to focus on areas of the space that are irrelevant to the final decision. As discussed in Section 4, our experimental evaluation confirms this intuition, showing that uncertainty sampling performs worse than simple random sampling for positive values of k. *TOS*, on the other hand, allows to focus the model on the area that is most critical to improve value, and substantially outperforms existing alternatives, especially when few active learning iterations are available.

Even in standard supervised classification settings, uncertainty sampling has some problems when applied in a batch way, which is due to the lack of diversity in the batch [9]. This problem can affect *TOS* too, especially for large values of k when the number of examples with confidence score over the threshold substantially shrinks. We thus introduce a simple variant of *TOS*, named ***below-threshold-oriented sampling (BTOS)***, that selects only rejected samples with confidence closest to τ:

$$x^* = argmin_{x \in \mathscr{D}, s(x) \leq \tau} (\tau - s(x)) \tag{6}$$

As our experimental evaluation will show, this simple strategy provides minor but consistent improvements over *TOS*. More advanced diversification strategies can further improve these threshold-based value-aware methods.

4. Experimental Evaluation

We investigate the performance of different AL strategies in terms of the value of the resulting classifier when used as a selective classifier with threshold τ. For the sake of

Algorithm 1 Value-aware experimental protocol

Input: M, Q, N, I, I_{test}, k

1: Build the initial training set I_{train}
2: $I \leftarrow I \setminus I_{train}$
3: **while** stopping criterion **do**
4: Train M on I_{train}
5: Make M a selective classifier with threshold $\tau \leftarrow k/(k+1)$
6: Evaluate M on I_{test}
7: Compute value $V \leftarrow (1-\rho)(\alpha - k(1-\alpha))$
8: $I_B \leftarrow$ select N items from I via Q
9: $I \leftarrow I \setminus I_B$, $I_{train} \leftarrow I_{train} \cup I_B$
10: **end while**

clarity, the learning and evaluation process is described in Algorithm 1. The algorithm takes as input: (i) an ML model M to be trained via AL; (ii) an AL strategy Q; (iii) an AL batch size N; (iv) a pool of unlabelled examples I from which to choose the AL batches; (v) a test set I_{test} to evaluate M in terms of value; (vi) a cost factor k to compute model value according to Eq. 3. The model M is bootstrapped with an initial training set made by randomly selecting one example per class. At each iteration, the model is trained with standard supervised learning on the current training set but evaluated as a threshold-based selective classifier with the threshold τ maximizing value (Eq. 4). Its value on the test set I_{test} is recorded as the value-aware performance metric. The next batch of examples from the unlabelled pool is selected according to the AL strategy Q and labeled by querying the oracle, and the procedure is iterated until 30% of the training set I has been labeled.

As an ML model we use Logistic Regression (LogReg) on top of pre-trained text encoders because (i) it is simple and can learn from small datasets (typical of AL scenarios), and (ii) it learns calibrated predictions [10] satisfying our assumption for using the theoretical threshold in Equation 4. We use two different text encoders: (i) the default tf-idf vectorizer from scikit-learn[3] with ngram (1, 3), and (ii) MPNet [11] from Hugging Face[4]. We run experiments for increasing values of the cost factor $k \in \{0, 2, 4, 8\}$, and varying the AL batch size $N \in \{5, 10, 20, 30\}$.

AL strategies. We compare *TOS* and *BTOS* with the following strategies:

- *Random sampling*: the simplest strategy that samples instances randomly from the unlabelled pool.
- *Uncertainty sampling*: the most popular strategy in the AL literature, that samples instances for which the model is most uncertain [6]. We use the margin-based uncertainty sampling recommended for multi-class classification [12,13]: for each instance in the unlabeled pool, we measure the margin between the confidence score of the two most likely classes and then select instances with the minimum margin.
- *Certainty sampling*: the opposite of uncertainty sampling: we pick instances for which the margin between the confidence score of the two most likely classes is maximized. As our experimental evaluation will show, certainty sampling ends up being the best strategy in some degenerate cases.

Datasets. We consider four standard multi-class text classification tasks from the NLP literature (see Table 1 for the statistics of the datasets we use):

- *Twitter US Airline Sentiment* dataset is a 3-class unbalanced dataset. The task is to detect the sentiments of tweets about US Airlines into negative, neutral, and positive, with negative being the dominant class.
- *Clinc150* is a 150-class balanced dataset. The task is to classify intents (e.g. changing the volume, finding the phone, suggesting a meal, etc.) from the text.
- *DBPedia* is a highly unbalanced 9-class dataset where we extract content (e.g. agent, place, and species, corresponding to the dominant classes) from Wikipedia.
- *Hate Speech* is a 3-class unbalanced dataset of tweets. The task is to classify tweets as hate speech, offensive, or neither.

[3] https://tinyurl.com/sklearn-tfidf-vectorizer
[4] huggingface.co/docs/transformers/model_doc/mpnet

Table 1. Statistics of the datasets used in the experiments

Dataset	Classes	Class Distribution (%)	Train/Val/Test size
Twitter US Airline	3	63%, 21%,16%	8784/2928/2928
Clinc150	150	Balanced	15000/5000/5000
DBPedia	9	3 dominant classes: 52.3%,8.49%,18.8%	15000/5000/5000
Hate Speech	3	5.77%, 77.43%,16.8%	14869/4957/4957

4.1. Results

In this section, we present experimental results aimed at answering the following research questions:

- **Q1** Are the good performance of uncertainty sampling confirmed when the cost ratio changes?
- **Q2** Do our proposed threshold-oriented strategies outperform existing alternatives?

Table 2 reports the results of our experiments using LogReg with a simple TF-IDF encoding. Each "cell" reports results for a given dataset (column) and AL strategy (row). Within each cell, model values are computed for increasing the value of the cost factor k, and the size of the AL batch N. All results are computed on the test set of each dataset using a value-aware selective classifier that rejects predictions according to the threshold in Eq. 4. In the following, we discuss how these results answer our research questions.

Q1: Uncertainty sampling performs poorly in a selective classification setting As expected, uncertainty sampling tends to outperform the alternatives in the standard active learning setting, in which the model never abstains ($k = 0$). However, it is almost always outperformed by simple random sampling for positive values of k, i.e., in a selective classification setting. Certainty sampling, again as expected, often performs rather poorly, occasionally producing models with *negative* value (i.e., one should rather ignore them altogether). There is however an exception, which is the Clinc150 dataset. Here both

Table 2. Performance of AL strategies with a value-aware selective LogReg classifier & TF-IDF encoding.

AL Strategy	N	US Airline				Clinc150				DBPedia				Hate Speech			
		Value				Value				Value				Value			
		k=0	k=2	k=4	k=8	k=0	k=2	k=4	k=8	k=0	k=2	k=4	k=8	k=0	k=2	k=4	k=8
Uncertainty	5	**0.719**	0.14	0.03	0.002	0.173	0.0	0.0	0.0	**0.835**	0.125	0.014	0.002	**0.796**	0.456	0.366	0.064
	10	**0.732**	0.243	0.066	0.01	0.451	0.0	0.0	0.0	**0.898**	0.271	0.058	0.011	**0.869**	0.537	0.249	0.073
	20	**0.752**	0.321	0.149	0.054	**0.665**	0.0	0.0	0.0	**0.934**	0.536	0.221	0.057	0.891	0.632	0.35	0.128
	30	0.761	0.363	0.21	0.095	**0.721**	0.001	0.0	0.0	**0.942**	0.699	0.431	0.182	**0.896**	0.68	0.45	0.191
Random	5	0.694	0.276	0.081	0.003	**0.405**	0.0	0.0	0.0	0.753	0.36	0.147	0.018	0.795	0.51	**0.493**	0.198
	10	0.72	0.304	0.165	0.054	**0.541**	0.0	0.0	0.0	0.834	**0.506**	0.289	0.117	0.83	0.565	0.521	0.332
	20	0.738	0.353	**0.24**	0.119	0.658	0.006	0.0	0.0	0.883	**0.639**	**0.448**	0.246	0.85	0.632	0.551	0.385
	30	0.748	0.367	**0.254**	0.135	0.704	0.023	0.005	0.0	0.905	0.706	**0.53**	0.33	0.865	0.66	0.574	0.408
Certainty	5	0.635	0.064	0.056	0.032	0.052	**0.026**	**0.016**	**0.01**	0.157	0.055	0.042	0.031	0.777	0.35	0.053	0.247
	10	0.631	-0.063	-0.336	0.067	0.075	**0.04**	**0.03**	**0.016**	0.143	0.081	0.082	0.069	0.778	0.351	0.01	0.015
	20	0.632	-0.027	-0.238	-0.016	0.14	**0.055**	**0.036**	**0.022**	0.625	0.142	0.119	0.095	0.777	0.338	-0.042	-0.479
	30	0.635	0.101	0.036	0.126	0.207	**0.073**	**0.047**	0.021	0.436	0.193	0.172	0.144	0.777	0.335	-0.062	-0.734
TOS	5	0.689	0.237	0.14	0.049	0.141	0.004	0.006	0.005	0.794	0.337	0.28	0.023	0.787	0.578	0.454	0.226
	10	**0.732**	0.31	0.193	0.102	0.387	0.011	0.016	0.014	0.869	0.437	0.381	0.197	0.856	0.628	0.549	0.41
	20	0.746	0.346	0.212	**0.15**	0.643	0.034	0.032	0.015	0.93	0.596	0.416	0.376	**0.893**	0.683	0.574	0.484
	30	**0.762**	0.367	0.244	**0.176**	0.669	0.059	0.045	0.02	0.94	**0.72**	0.438	**0.392**	**0.896**	0.708	**0.585**	**0.502**
BTOS	5	0.255	**0.277**	**0.159**	**0.077**	0.361	0.002	0.003	**0.011**	0.283	**0.396**	**0.315**	**0.1**	0.388	**0.602**	0.463	**0.375**
	10	0.255	**0.313**	**0.204**	**0.11**	0.361	0.014	0.006	**0.017**	0.283	0.442	**0.393**	**0.302**	0.388	**0.658**	**0.558**	**0.455**
	20	0.255	**0.361**	0.217	0.125	0.361	0.027	0.023	**0.022**	0.283	0.617	0.423	**0.378**	0.388	**0.696**	**0.582**	**0.487**
	30	0.255	**0.377**	**0.254**	0.155	0.361	0.058	0.04	**0.03**	0.283	**0.726**	0.479	0.381	0.388	**0.717**	**0.587**	0.497

uncertainty sampling and random sampling fail, producing a model with zero value in most cases. On the other hand, certainty sampling always manages to produce a model with a positive value. This is because CLINC150 is the only balanced dataset, and certainty sampling manages to evenly pick (high-confidence) items from every class. Overall, these results seem to indicate that the best value-unaware strategy is the simplest one, namely random sampling. We believe that the poor performance of uncertainty sampling in a selective classification setting is due to the fact that it focuses on an area (the uncertain one) that is not relevant to the final decision, as the prediction will most likely be rejected because of its low confidence.

Q2: Threshold-oriented sampling strategies outperform value-unaware alternatives in a selective classification setting The TOS strategies we introduced in this work have been designed precisely to adapt the concept of uncertainty sampling to the selective classification setting, by considering uncertainty *around the rejection threshold*. Indeed, results in Table 2 show that *TOS* and *BTOS* have a robust behavior across all datasets, showing the overall best performance for all positive values of k. *BTOS* tends to slightly outperform *TOS* in most cases, most likely because of the lack of diversity of *TOS* in the area above the rejection threshold. Indeed, the average distance between sampled items in each batch is lower for *TOS* than for *BTOS*. The lack of diversity in batch AL is a well-known problem of uncertainty sampling too [9], and diversification strategies developed in the AL community could prove useful to further improve the performance of threshold-oriented sampling strategies. Note that in the CLINC150 dataset, where certainty sampling is the best overall strategy, both threshold-oriented strategies are close runners-up, especially for large values of k when they are almost indistinguishable. This is not by chance. In this balanced dataset, confidence scores are rather high regardless of which is the most confident class, the confidence score becomes similar to the margin, and threshold-oriented sampling with high values of k boils down to picking the most certain prediction. It is indeed interesting to highlight that TOS strategies manage to (approximately) recover the behavior of certainty sampling when this happens to be the best strategy.

Figure 1 reports value curves as functions of the number of batches up to 30% of the unlabelled pool, which corresponds to the results shown in Table 2, for the case of $k = 8$ (highest cost factor) and $N = 30$. Results confirm how *TOS* and *BTOS* consistently outperform value-unaware alternatives. Furthermore, while each of the value-unaware strategies shows extremely poor performance in some settings (Clinic 150 for uncertainty and random sampling, Hate Speech for certainty sampling), *TOS* and *BTOS* present stable performance in all cases (apart from a single outlier in the first batch in US Airline).

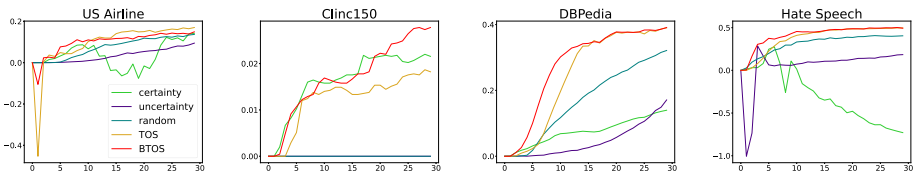

Figure 1. Value curves (y-axes) as functions of the number of batches (x-axes) for different AL strategies with a value-aware selective LogReg classifier & TF-IDF encoding ($k = 8$, $N = 30$). The legend is shown on the leftmost plot only for the sake of readability.

Table 3 reports results when replacing TF-IDF encoding with a richer encoding produced by MPNet [11]. While values are higher on average, the relative behavior of the different AL strategies is the same as in Table 2, confirming the generality of the findings.

Table 3. Performance of AL strategies with a value-aware selective LogReg classifier & MPNet encoding.

AL Strategy	N	US Airline				Clinc150				DBPedia				Hate Speech			
		Value				Value				Value				Value			
		κ=0	κ=2	κ=4	κ=8	κ=0	κ=2	κ=4	κ=8	κ=0	κ=2	κ=4	κ=8	κ=0	κ=2	κ=4	κ=8
Uncertainty	5	**0.8**	0.391	0.219	0.082	0.608	0.0	0.0	0.0	**0.933**	0.369	0.083	0.003	**0.87**	**0.641**	0.456	0.237
	10	**0.814**	0.451	0.298	0.148	**0.869**	0.0	0.0	0.0	**0.961**	0.566	0.212	0.037	**0.883**	0.659	0.495	0.297
	20	**0.815**	0.485	0.336	0.224	**0.943**	0.0	0.0	0.0	**0.971**	**0.776**	0.487	0.187	**0.885**	0.683	0.545	0.353
	30	0.819	**0.511**	0.362	0.271	**0.96**	0.0	0.0	0.0	**0.972**	**0.863**	**0.702**	0.452	**0.888**	**0.699**	0.578	0.411
Random	5	0.784	0.405	0.273	0.153	0.519	0.0	0.0	0.0	0.852	**0.502**	0.324	0.191	0.853	0.616	0.502	0.348
	10	0.795	0.455	0.314	0.22	0.716	0.0	0.0	0.0	0.899	**0.643**	**0.468**	0.304	0.868	0.642	0.532	0.401
	20	0.802	0.488	**0.362**	0.256	0.882	0.002	0.0	0.0	0.927	0.74	**0.605**	**0.434**	0.873	0.663	0.556	0.432
	30	0.805	0.502	0.365	0.278	0.919	0.015	0.0	0.0	0.942	0.79	0.672	**0.513**	0.877	0.676	0.571	0.453
Certainty	5	0.653	0.199	0.177	0.139	0.095	0.022	**0.018**	0.009	0.442	0.098	0.076	0.055	0.784	0.389	0.217	0.201
	10	0.639	0.112	-0.036	-0.003	0.121	**0.051**	**0.039**	**0.021**	0.443	0.078	0.071	0.073	0.781	0.371	0.11	0.036
	20	0.635	0.084	-0.079	-0.045	0.16	**0.104**	**0.076**	**0.03**	0.609	0.359	0.302	0.226	0.778	0.355	0.032	-0.193
	30	0.644	0.206	0.161	0.178	0.219	**0.153**	**0.11**	**0.048**	0.609	0.336	0.329	0.297	0.778	0.354	0.04	-0.206
TOS	5	0.796	0.373	**0.313**	**0.233**	0.284	0.015	0.013	**0.015**	0.909	0.394	0.333	0.064	0.855	0.631	**0.51**	0.341
	10	0.805	0.444	0.328	0.241	0.62	0.036	0.026	**0.021**	0.952	0.584	0.452	0.119	0.876	**0.671**	0.523	0.396
	20	0.813	0.493	0.355	**0.27**	0.9	0.075	0.057	0.027	**0.97**	0.742	0.557	0.148	**0.885**	**0.693**	0.567	**0.441**
	30	**0.822**	**0.511**	**0.376**	0.274	0.94	0.123	0.079	0.034	**0.971**	0.857	0.592	0.42	**0.888**	**0.697**	**0.588**	0.451
BTOS	5	0.485	**0.409**	0.307	0.213	**0.701**	**0.024**	0.009	**0.012**	0.342	0.451	**0.354**	**0.263**	0.216	**0.641**	0.503	**0.361**
	10	0.485	**0.46**	**0.331**	**0.249**	0.701	0.048	0.017	**0.022**	0.342	0.592	0.428	**0.35**	0.216	**0.671**	**0.536**	**0.41**
	20	0.485	**0.501**	**0.362**	**0.271**	0.701	0.085	0.051	**0.03**	0.342	0.746	0.525	0.415	0.216	0.691	**0.573**	**0.44**
	30	0.485	**0.512**	**0.378**	**0.291**	0.701	0.129	0.087	0.041	0.342	0.858	0.6	0.43	0.216	**0.699**	0.586	**0.461**

5. Limitations and Conclusions

In this work-in-progress paper, we are just scratching the surface of value-aware AL. We tested a limited number of NLP datasets, encoders, and algorithms. For this reason, even if TOS seems superior we cannot claim this as a fact with any sort of generality. What we can say however is that we believe we have provided enough evidence to reconsider the superiority of uncertainty sampling, propose TOS as a valid contender, and lay down the motivations for larger-scale investigation on both outcomes and reasons behind the outcomes. This is important because i) uncertainty sampling is very widely adopted, but ii) model thresholding is also ubiquitous in practice, and from what we have seen even random sampling is often preferable already at rather low cost ratios.

A related but somehow complementary research line in the AL literature is the one focusing on cost-sensitive AL [14,15,16,17] where the model is charged a cost for every query and it should learn how to improve the model by compensating the cost of asking labels. This research line is complementary to our notion of value-aware AL as it still assumes that the resulting classifier never abstains. As we showed for cost-sensitive errors in static classification [4], failing to account for the reject option prevents cost-sensitive learning strategies from maximizing value. Nonetheless, developing value-aware AL strategies that are also cost-sensitive is a promising direction for further research.

Acknowledgements. This research was partially supported by TAILOR, a project funded by the EU Horizon 2020 research and innovation program under GA No 952215. The work of Burcu Sayin and Andrea Passerini was partially supported by the project AI@Trento (FBK-Unitn).

References

[1] Callaghan W, Goh J, Mohareb M, Lim A, Law E. Mechanicalheart: A human-machine framework for the classification of phonocardiograms. In: The 21st ACM Conference on Computer-Supported Cooperative Work and Social Computing; 2:28:1–17; 2018.

[2] Geifman Y, El-Yaniv R. Selective classification for deep neural networks. In: Advances in Neural Information Processing Systems; Volume 30; 2017.

[3] Casati F, Noël P, Yang J. On the value of ML models. ArXiv; abs/2112.06775; 2021.

[4] Sayin B, Casati F, Passerini A, Yang J, Chen X. Rethinking and recomputing the value of ML models. ArXiv; abs/2209.15157; 2022.

[5] Sayin B, Yang J, Passerini A, Casati F. The science of rejection: A research area for human computation. In: The 9th AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2021); 2021; AAAI Press.

[6] Lewis DD, Gale WA. A sequential algorithm for training text classifiers. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 1994; p. 3–12.

[7] Filho TMS, Song H, Perelló-Nieto M, Santos-Rodríguez R, Kull M, Flach PA. Classifier Calibration: How to assess and improve predicted class probabilities: a survey. ArXiv; abs/2112.10327; 2021.

[8] Guo C, Pleiss G, Sun Y, Weinberger KQ. On Calibration of Modern Neural Networks. In: Proceedings of the 34th International Conference on Machine Learning - Volume 70 (Sydney, NSW, Australia) (ICML'17); 2017; p. 1321–1330; https://doi.org/10.48550/ARXIV.1706.04599.

[9] Dasgupta S, Kalai AT, Monteleoni C. Analysis of perceptron-based active learning. Learning Theory; Berlin, Heidelberg; 2005; p.249–263.

[10] Kull M, Filho TMS, Flach PA. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. Electronic Journal of Statistics. 2017; 11:5052–5080.

[11] Song K, Tan X, Qin T, Lu J, Liu T. Mpnet: Masked and permuted pre-training for language understanding. In: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20; Red Hook, NY, USA; 2020.

[12] Settles B. Active learning literature survey. In: University of Wisconsin; Madison; volume 52; 2010.

[13] Sayin B, Krivosheev E, Yang J, Passerini A, Casati F. A review and experimental analysis of active learning over crowdsourced data. Journal of Artificial Intelligence Review. 2021; 54:5283–5305.

[14] Greiner R, Grove AJ, Roth, D. Learning cost-sensitive active classifiers. Artificial Intelligence. 2002; 139(2):137–174.

[15] Tomanek K, Hahn U. A comparison of models for cost-sensitive active learning. In: Coling 2010: Posters; 2010; Beijing, China; p. 1247–1255.

[16] Demir B, Minello L, Bruzzone L. A cost-sensitive active learning technique for the definition of effective training sets for supervised classifiers. In: IEEE International Geoscience and Remote Sensing Symposium; 2012; Munich, Germany; p. 1781-1784, doi: 10.1109/IGARSS.2012.6351169.

[17] Xie K, Chang C, Ren L, Chen L, Yu, K. Cost-sensitive active learning for dialogue state tracking. In: Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue; 2018; Melbourne, Australia; p. 209–213.