Activle Learning for Batch Text Classification

Marko Sahan
sahanmar@fel.cvut.cz
Dept. of Computer Science, FEE, CTU in Prague
Prague, Czech Republic

Václav Šmidl
smidlva1@fel.cvut.cz
Dept. of Computer Science, FEE, CTU in Prague
Prague, Czech Republic

Radek Mařík
radek.marik@fel.cvut.com
Dept.of Telecommunication Engineering, FEE, CTU in Prague
Prague, Czech Republic

ABSTRACT

Supervised learning of classifiers for text classification and sentiment analysis relies on the availability of labels that may be difficult/expensive to obtain. Active learning techniques optimize the process of obtaining labels by sequentially selecting documents from the unlabeled set for which the labels would be most valuable. Batch active learning selects a batch of documents for labeling. In this paper, we propose new methods for batch active learning by combining Bayesian strategies with agglomerative clustering. We study the effect of this proposal in a large-scale study comparing the effect of varying distinct factors (initialization of the algorithm, uncertainty representation, acquisition function, and batch selection strategy). Various combinations of these are tested on selected NLP problems with documents encoded using RoBERTa. Datasets cover context integrity, fake news detection, and sentiment classification. We show that each of the active learning factors has advantages for certain datasets or experimental settings.

ACM Reference Format:
Marko Sahan, Václav Šmidl, and Radek Mařík. 2022. Activle Learning for Batch Text Classification. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Supervised learning of classifiers relies on the availability of class labels which often involves a human annotator for a majority of NLP tasks. This can be costly for large datasets. Active learning is a strategy designed to minimize this cost by automatic selection of those unlabeled documents that are expected to bring useful information for the classifier. Advantages of this approach have been demonstrated even for classical methods such as SVM Tong and Koller [19]. The most conventional active learning strategies select only one unlabeled document after each training round to query due to the simplicity of its selection. The next query document is selected only after the first one is labeled and the model retrained, which means that the annotator has to wait for retraining. This impractical strategy can be avoided if the active learning algorithm selects a batch of documents. Novel methods for batch active learning appear frequently, each demonstrating advantages on their benchmark data.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

One of the recent approaches demonstrates effectiveness even for batches of 5000 samplesCitovsky et al. [2], combining the min-margin acquisition function with clustering under the name Hierarchical Agglomerative Clustering (HAC). In this contribution, we propose novel modifications of this idea using alternative acquisition functions and investigate their performance. Specifically, we propose to extend the HAC approach to the Bayesian setting by replacing the min-margin with a Bayesian acquisition function, BALD Houlsby et al. [9]. However, the size of the minibatch is only one of many factors in the performance of the active learning algorithms. Other factors are: i) selected acquisition function, ii) representation of uncertainty of the classifier, and iii) initialization of the network. The key contribution of our work is a sensitivity study of the classification task to these factors over a range of datasets from various text classification tasks.

Various comparative studies have been performed recently with various focus and results. Batch active learning was studied in Dor et al. [5] for only one size of the batch (50 documents per query). Large sensitivity to the type of dataset was reported in Prabhu et al. [16], where different methods won for different data. Large variability of the results was also observed in Jacobs et al. [11]. In Schröder et al. [18], the min-margin strategy was shown to be competitive with the prediction entropy-based method on a range of embeddings. The comparative studies shared similar properties, such as a fixed network for embeddings (improvement with retraining can be expected Margatina et al. [15] but may be too costly). All studies also assume a cold start, i.e. completely new initialization of the classifier after each round of querying. This is motivated by the fear of overfitting, which was demonstrated in Hu et al. [10] for hot start, i.e. continuation of training of the classifier. A compromise in the form of warm-start, i.e. adding noise to the weights of the previous classifier, was proposed in Sahan et al. [17].

The paper is organized as follows. In Section 2, we briefly review all tested factors of batch active learning. The experimental setup of the sensitivity study is described in Section 3 and the results are reported in Section 4.

2 BATCH ACTIVE LEARNING METHODS

Throughout the paper, we will use the RoBERTa embedding Liu et al. [14] to represent documents in the feature space. RoBERTa is a modified BERT transformer model Devlin et al. [4] that achieved comparable performance to BERT in Schröder et al. [18] and outperformed all other embeddings in Sahan et al. [17]. Representation of the k-th text document $\mathbf{x}_k$ is calculated as the mean value from sentence embeddings of all sentences in the text.

The aim of document classification is to find a classifier $\hat{\mathbf{y}} = \mathbf{y}(\theta, \mathbf{x})$ predicting the class label for each document representation $\mathbf{x}$. In a supervised setting, the classifier parameters are found

Algorithm 1 General batch active learning

Initialize: set classifier structure $\mathbf{y} = \mathbf{y}(\mathbf{x}, \theta)$, iteration counter $i = 0$, initial data $\mathbf{Y}^{(0)}, \mathbf{X}^{(0)}, \mathbf{X}_u^{(0)}$

Iterate until a stopping condition:

1. Train a classifier parameter $\theta^{(i)}$ on $\mathbf{Y}^{(i)}, \mathbf{X}^{(i)}$, starting from $\theta_{\text{init}}^{(i)}$

2. Compute the value of a label for all documents in the unlabeled dataset, $a_l = A(\mathbf{x}_l, \theta^{(i)}), \forall \mathbf{x}_l \in \mathbf{X}_u^{(i)}$

3. Select a batch of documents, $\tilde{\mathbf{X}} \subset \mathbf{X}$, for labeling using a_l

4. Query labels $\tilde{\mathbf{y}}$ for $\tilde{\mathbf{X}}$ and extend the training set $\mathbf{X}^{(i+1)} = \mathbf{X}^{(i)} \cup \tilde{\mathbf{X}}, \mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} \cup \tilde{\mathbf{y}}, i = i + 1$.

on a training set $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^K$ by matching the prediction $\mathbf{y}(\theta, \mathbf{x}_k)$ with the provided label $\mathbf{y}_k$ for each document. We are concerned with binary classification for simplicity, however, an extension to multiclass is straightforward.

We assume that for the full corpus of text documents $\mathbf{X}$, only a small initial set of labels $\mathbf{Y}^{(0)}$, is available. The full set $\mathbf{X}$ is thus split into the labeled, $\mathbf{X}^{(0)}$, and unlabeled parts, $\mathbf{X}_u^{(0)} = \mathbf{X} \backslash \mathbf{X}^{(0)}$, the training set in the first round is then $\mathbf{X}^{(0)}, \mathbf{Y}^{(0)}$. Active learning is defined as a sequential extension of the training data set following a simple iterative strategy in algorithm 1.

The general algorithm can be specialized to many variants depending on various factors as specified next. We will introduce several choices labeled by the step in which they appear in algorithm 1.

1a. Uncertainty representation: The uncertainty can be represented by a maximum likelihood estimate, represented by a single network, or a Bayesian probabilistic estimate, represented typically by an ensemble of networks. We will consider the following options: **Single network** with a softmax output layer predicting the normalized probability of each class in one hot encoding. This probability is conditioned on the parameter, and thus captures only aleatoric uncertainty. Uncertainty in parameters is not represented. **Ensemble of networks**, represent uncertainty in parameters by different parameter value in each ensemble thus capturing both aleatoric and epistemic uncertainty. We consider two methods for generating the ensemble members: i) *MC dropout* Gal et al. [7], where ensemble members are generated by random draws of the dropout layers, and ii) *deep ensembles* Lakshminarayanan et al. [13] where ensembles are trained independently. Note that MC dropout is computationally much cheaper.

1b. Initialization of the training: Each training in step 1 is a new task. However, the data set typically overlaps with the one from the previous iteration, which motivates the following strategies of reusing results from the previous iteration. The **Cold start** strategy is not reusing any information, the networks are initialized by random numbers, $\theta_{\text{init}}^{(i)} = \mathcal{N}(0, \sigma)$, where σ is given by the standard network init strategy, used most often Citovsky et al. [2], Dor et al. [5], Schröder et al. [18]. The **Hot start** strategy reuses all information, setting the estimate from the previous iteration as a starting point, $\theta_{\text{init}}^{(i)} = \theta^{(i-1)}$,criticized in Hu et al. [10]. The **Warm start** strategy a combination of the above, $\theta_{\text{init}}^{(i)} = \theta^{(i-1)} + \mathcal{N}(0, \sigma)$, where σ is a hyper-parameter Sahan et al. [17].

2. Acquisition function: Is a measure of the expected utility of the knowledge label, $\boldsymbol{y}_l$, for each document, $\boldsymbol{x}_l$, in the unlabeled data set. Different running index l is used to indicate that we operate on the unlabeled set. While many different utilities are proposed, we will study only the most popular ones. **Entropy** exists in two forms, entropy of the prediction $a_l = \mathbb{H}(y|\mathbf{x}_l, \theta)$ for a single network, or expected entropy $a_l = \mathbb{E}_\theta \mathbb{H}(y|\mathbf{x}_l, \theta)$ for ensembles. **BALD** is a mutual information metric, $a_l = \mathbb{E}_\theta \mathbb{H}(y|\mathbf{x}_l, \theta) - \mathbb{H}(y|\mathbf{x}_l)$ that is meaningful only for the ensembles. **Min-margin** is a minimum difference between class predictions $a_l = -\min_{c,d \in [1,C]}(y_c - y_d)$, where C is the number of classes. Note carefully that the extreme of this criteria is equivalent to maximum entropy for binary classification with a single network.

3. Batch selection strategy When only one sample is to be selected, it is optimal to choose the one with maximum utility given by the acquisition function. However, the complexity of the maximum utility grows exponentially when the strategy has to select a batch of b documents for off-line labeling. Strategies that try to approximate this selection using greedy search Kirsch et al. [12] are still too computationally expensive for large batches. Therefore, we select two batch selection strategies that scale well with b. **Top** selects top b samples from sorted values of a_l. This approach may select samples close to each other, thus being redundant. **HAC** is a strategy based on the hierarchical clustering of a_l proposed in Citovsky et al. [2], and selecting top b samples from different clusters.

Tested algorithm variants: From the range of all possibilities, we will study the combinations that exist in the literature: HAC min-margin using cold start Citovsky et al. [2], MC dropout with Entropy and BALD criteria using warm start Gal et al. [7], warm start ensemble learning with Entropy and BALD called DEnFi Sahan et al. [17], and conventional single-network with prediction Entropy a with warm start. If HAC is not in the name, the Top strategy is used.

Since HAC strategy is an orthogonal factor to the remaining ones, we propose its combination with other approaches, giving rise to: HAC Entropy for the single neural network and both ensemble methods (MC dropout and DEnFi) and HAC BALD for the ensembles.

3 EXPERIMENT SETUP

The methods were compared on different datasets and different batch sizes. The used datasets are positive/negative tweets from the Tweets Go et al. [8], Fake News Detection fak [1], two pairs of categories from Amazon Reviews, and Gibberish Wackerow [20] datasets. From all datasets, we select from 10000 text documents (5000 text documents per category, selecting only two categories for binary classification, e.g. 1 and 5 in Amazon reviews), which were the initial 10000 documents of the datasets given categories.

All algorithms were compared on the area under the curve (AUC) Fawcett [6] on the test data (i.e. the documents not present in the training set). The algorithms were compared after 1000 acquired labels. The algorithms with smaller batch sizes thus benefited the from higher number of retrainings. To reduce the influence of stochastic initialization and training, the AUCs were run 5 times and averaged. Even then, the difference between the algorithms

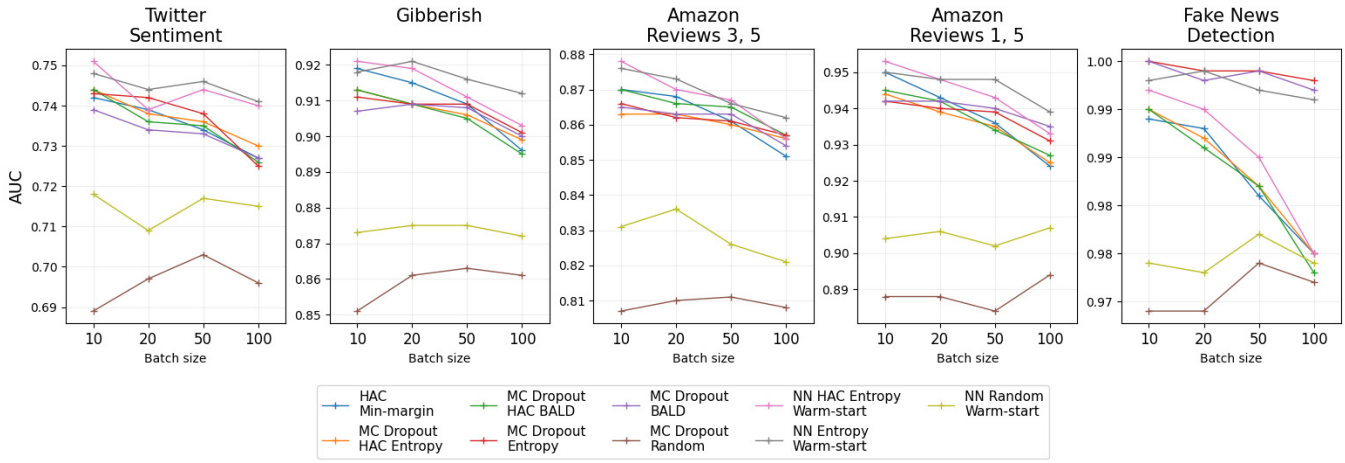

Figure 1: AUC metrics for seven active learning and two random strategies after 1000 acquired samples given datasets and batch size. Prediction of the MC dropout classifiers is an average over ensemble members.

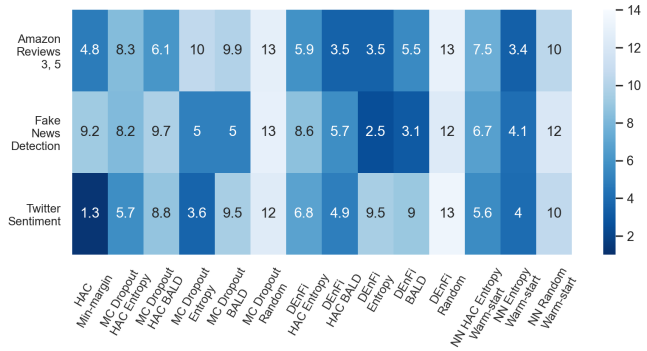

Figure 2: Aggregated mean rank for 14 tuples of learning algorithms and acquisition functions given Amazon Reviews 3,5, Fake News Detection, and Twitter Sentiment for 50 active learning iterations with batch size 20.

was sometimes marginal. To show the effect of various factors on the performance, we sorted the AUC and assigned a rank of each method accordingly. I.e. the best performing method has rank 1, second rank 2, etc. This approach allows the comparison of various methods across multiple datasets Demšar [3] using order statistics.

4 RESULTS

Parameter uncertainty: The effect of parameter uncertainty (Bayesian approach) is the most costly to evaluate, due to the high computational demand of the ensemble approach (DEnFi). Therefore, we have evaluated all algorithm variants only for batch size $b = 20$. The results are displayed in Figure 2. The advantage of the Bayesian approach is evident only for the Fake News dataset; in other datasets, DEnFi is not worth the computational cost and was omitted from subsequent large-scale studies.

A summary of the performance of all tested methods for various batch sizes is displayed in Figure 1 via AUC after 1000 samples for all methods, and via relative rank for all methods in Figure 3 averaged over ranks after each 100 label requests. Note that the datasets follow a similar pattern, with the exception of the Fake News data sets, where the parametric uncertainty (now represented only by the MC dropout strategy) is beneficial, and HAC batch selection strategy has a negative effect (probably due to preference of large clusters).

Acquisition functions: Due to binary classification, the min-margin and entropy approaches coincide for a single network function. The difference between Entropy and BALD for the ensemble methods seems insignificant, Figure 3.

Initialization of the training: The warm start strategy (HAC Entropy warm start) is better or comparable in performance to the cold start (HAC Min-margin); this is achieved at a fraction of the training cost. This indicates that the additive noise is sufficient to avoid overfitting of the hot start Hu et al. [10].

Batch selection strategy: The HAC batch selection has a clear advantage for smaller batch sizes (10 and 20). This is consistent when comparing HAC and Top variants of all methods except Fake News Detection. This advantage diminishes for batch sizes of 50 and 100 where the top selection strategy achieves comparable (ensembles) or better (single NN) results. We conjecture that the most informative samples in our datasets are clustered in small clusters, hence the selection of a batch with a large enough size contains all important samples.

5 CONCLUSION

We have studied the influence of various factors of active learning algorithms on their performance on cover context integrity, fake news detection, and sentiment classification tasks. While complex algorithms such as deep ensembles sometimes achieve good performance (Fake News detection), the winner, on average, is the classical prediction entropy of a single neural network with few proposed modifications. Specifically, the warm start of the network training achieves good performance at a lower computational cost,

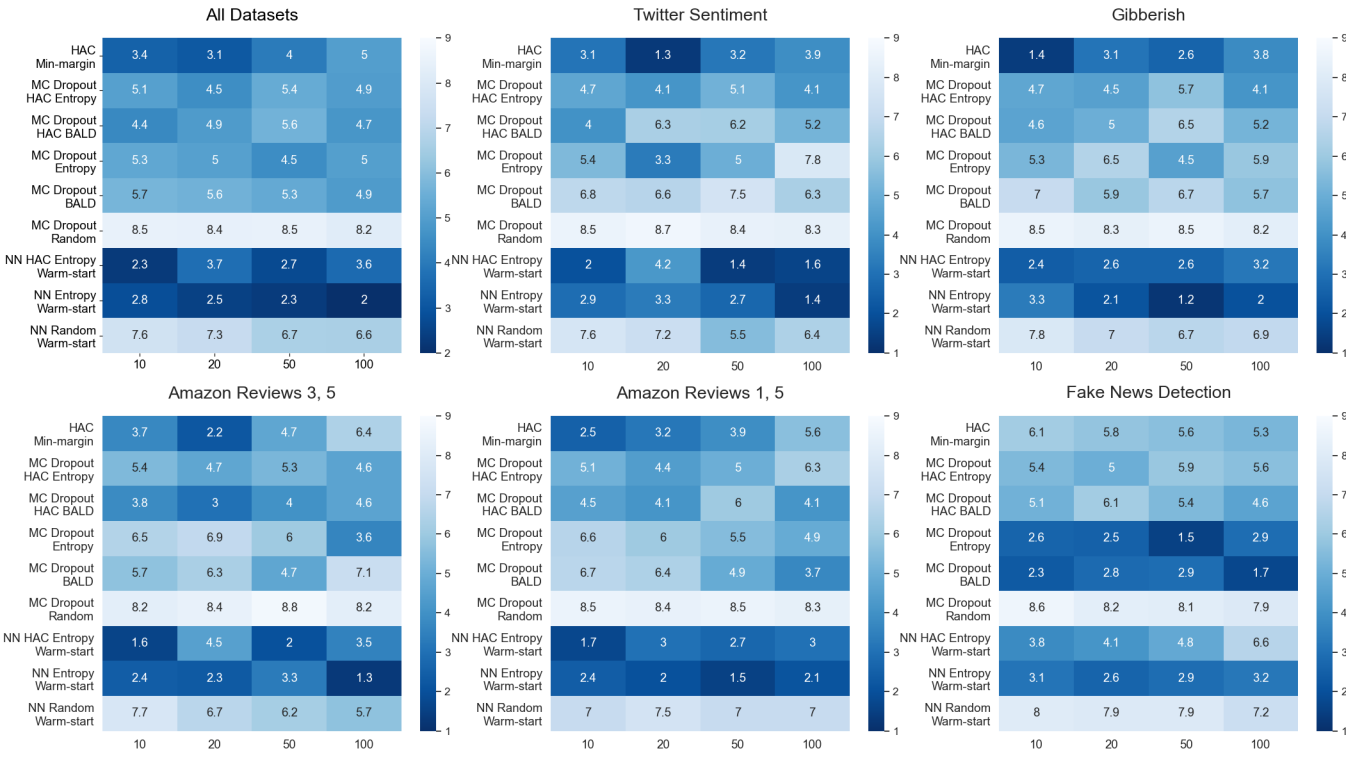

Figure 3: Aggregated rank for 7 active learning algorithms and two random strategies averaged over datasets as a function of different batch sizes.

and the selection of the batch for annotation using agglomerative clustering improves performance for smaller batch sizes.

A APPENDIX: EXPERIMENT PARAMETERS

Each active learning experiment was initialized by the training set $\mathbf{X}^{(0)}, \mathbf{Y}^{(0)}$ of 10 samples. The active learning strategy was set to sample b samples with a discrete set of variants, $b = 10, 20, 50, 100$. The active learning was run until 1000 samples were labeled, i.e. making a different number of steps for each batch size (10 iterations for $b = 100$, 20 for $b = 50$, etc.). The batch selection follows the ϵ -greedy approach Watkins [21], i.e. the samples selected by the acquisition function are accepted with probability $\epsilon = \frac{\exp(l-3)}{\exp(l-3)+1}$. A batch of random documents is selected for labeling if not accepted. The AUC is evaluated on the remaining part of the selected dataset (i.e. on the 9990 text documents in the first evaluation). The reported AUC values are averaged over 5 independent runs.

The initial number of epochs for the first iteration is 2500 for all algorithms. The same number is used for the cold start strategy in each iteration. The training of the warm start strategies is run for 150 epochs, with weights perturbation noise of variance $\sigma = 0.3$ for both MC dropout and DEnFi. Both DEnFi and MC Dropout generate 5 ensemble members. The key difference is in computational complexity, while DEnFi has to tune the parameters for each ensemble member, the MC dropout does it for only one network and generated ensemble members by 5 different realizations of the dropout mask.

REFERENCES

[1] 2018. Gibberish Text Classification. https://www.kaggle.com/jruvika/fake-news-detection, accessed on 16 may 2022.

[2] Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. 2021. Batch Active Learning at Scale. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 11933–11944. https://proceedings.neurips.cc/paper/2021/file/64254db8396e404d9223914a0bd355d2-Paper.pdf

[3] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7 (2006), 1–30.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] 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)*. 7949–7962.

[6] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27, 8 (2006), 861–874.

[7] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1183–1192.

[8] Alec Go, Richa Bhayani, and Lei Huang. 2009. *Twitter sentiment classification using distant supervision*. Technical Report. Stanford. http://help.sentiment140.com/for-students/ accessed on 26 june 2020.

[9] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745* (2011).

[10] Peiyun Hu, Zachary C Lipton, Anima Anandkumar, and Deva Ramanan. 2018. Active learning with partial feedback. *arXiv preprint arXiv:1802.07427* (2018).

[11] Pieter Floris Jacobs, Gideon Maillette de Buy Wenniger, Marco Wiering, and Lambert Schomaker. 2021. Active learning for reducing labeling effort in text classification tasks. In *Benelux Conference on Artificial Intelligence*. Springer, 3–29.

[12] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. 2019. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems* 32 (2019).
[13] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*. 6402–6413.
[14] 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. *arXiv preprint arXiv:1907.11692* (2019).
[15] Katerina Margatina, Loic Barrault, and Nikolaos Aletras. 2021. Bayesian active learning with pretrained language models. *arXiv preprint arXiv:2104.08320* (2021).
[16] Sumanth Prabhu, Moosa Mohamed, and Hemant Misra. 2021. Multi-class Text Classification using BERT-based Active Learning. *arXiv preprint arXiv:2104.14289* (2021).
[17] Marko Sahan, Vaclav Smidl, and Radek Marik. 2021. Active Learning for Text Classification and Fake News Detection. In *2021 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*. IEEE, 87–94.
[18] Christopher Schröder, Andreas Niekler, and Martin Potthast. 2021. Uncertainty-based Query Strategies for Active Learning with Transformers. *CoRR* abs/2107.05687 (2021). arXiv:2107.05687 https://arxiv.org/abs/2107.05687
[19] Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research* 2, Nov (2001), 45–66.
[20] John Wackerow. 2020. Gibberish Text Classification. https://www.kaggle.com/datasets/johnwdata/gibberish-text-classification, accessed on 16 may 2022.
[21] Christopher John Cornish Hellaby Watkins. 1989. Learning from delayed rewards. (1989).