Batch Active Learning for Text Classification and Sentiment Analysis

MARKO SAHAN, Dept. of Computer Science, FEE, CTU in Prague, Czech Republic
VACLAV SMIDL, Dept. of Computer Science, FEE, CTU, Czech Republic
RADEK MARIK, Dept.of Telecommunication Engineering, FEE, CTU, Czech Republic

Supervised learning of classifiers for text classification and sentiment analysis relies on the availability of labels that may be either difficult or expensive to obtain. A standard procedure is to add labels to the training dataset sequentially by querying an annotator until the model reaches a satisfactory performance. Active learning is a process that optimizes unlabeled data records selection for which the knowledge of the label would bring the highest discriminability of the dataset. Batch active learning is a generalization of a single instance active learning by selecting a batch of documents for labeling. This task is much more demanding because plenty of different factors come into consideration (i. e. batch size, batch evaluation, etc.). In this paper, we provide a large scale study by decomposing the existing algorithms into building blocks and systematically comparing meaningful combinations of these blocks with a subsequent evaluation on different text datasets. While each block is known (warm start weights initialization, Dropout MC, entropy sampling, etc.), many of their combinations like Bayesian strategies with agglomerative clustering are first proposed in our paper with excellent performance. Particularly, our extension of the warm start method to batch active learning is among the top performing strategies on all datasets. We studied the effect of this proposal comparing the outcomes of varying distinct factors of an active learning algorithm. Some of these factors include initialization of the algorithm, uncertainty representation, acquisition function, and batch selection strategy. Further, various combinations of these are tested on selected NLP problems with documents encoded using RoBERTa embeddings. Datasets cover context integrity (Gibberish Wackerow), fake news detection (Kaggle Fake News Detection), categorization of short texts by emotional context (Twitter Sentiment140), and sentiment classification (Amazon Reviews). Ultimately, we show that each of the active learning factors has advantages for certain datasets or experimental settings.

Additional Key Words and Phrases: active learning, BALD, batch active learning, dropout mc, entropy sampling, HAC min-margin, natural language processing, RoBERTa, sensitivity study, text classification, warm start

ACM Reference Format:
Marko Sahan, Vaclav Smidl, and Radek Marik. 2022. Batch Active Learning for Text Classification and Sentiment Analysis. 1, 1 (August 2022), 8 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 [20]. The most

Authors' addresses: Marko Sahan, Dept. of Computer Science, FEE, CTU in Prague, Karlovo náměstí 13, Prague, Czech Republic, 12135, sahanmar@fel.cvut.cz; Vaclav Smidl, Dept. of Computer Science, FEE, CTU, Karlovo náměstí 13, Prague, Czech Republic, 12135, smidlva1@fel.cvut.cz; Radek Marik, Dept.of Telecommunication Engineering, FEE, CTU, Technicka 2, Prague, Czech Republic, 16627, radek.marik@fel.cvut.cz.

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.

Various comparative studies have been performed recently with various focus and results. Batch active learning was studied in [5] for only one size of the batch (50 documents per query). Large sensitivity to the type of dataset was reported in [17], where different methods won for different data. Large variability of the results was also observed in [11]. In [19], 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 [16] 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 [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 [18].

In this contribution, we take a different approach to benchmarking of the batch active learning algorithms. Specifically, we decompose the algorithms into their building blocks: i) the size of the minibatch, ii) acquisition function, iii) representation of uncertainty of the classifier, and iii) initialization of the network. This approach allows us to quantify contribution of each of the building-block and combine them in previously untested versions. This allows us to demonstrate the following contribution:

(1) We present an extension of the Hierarchical Agglomerative Clustering (HAC) approach [2] to the Bayesian setting by replacing the min-margin with a Bayesian acquisition function, such as BALD [9]. This is a novel combination that has not been tested before.
(2) We show that performance of various methods is clustered based on particular building blocks of the method. Thus indicating that active learning methods may be tailored for each target application. A good example of Bayesian methods lies in estimating the distribution of neural networks which performed the best on Fake news but showed the same results on other datasets.
(3) In a large scale study we demonstrate that warm start is often beneficial and even simple methods (such as single neural network with entropy acquisition) provide results competitive to, or better than, complex active learning schemes. We also show that cold start approaches reach the same or even worse results than the aforementioned warm start techniques. This is encouraging for practitioners that are interested in the methodology.

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 [15] to represent documents in the feature space. RoBERTa is a modified BERT transformer model [4] that achieved comparable performance to BERT in [19] and outperformed all other embeddings in [18]. 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 on a training set $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^{K}$ by matching

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 $\mathbf{X}^{(i+1)} = \mathbf{X}^{(i)} \cup \tilde{\mathbf{X}}, \mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} \cup \tilde{\mathbf{y}}, i = i + 1$.

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* [7], where ensemble members are generated by random draws of the dropout layers, and ii) *deep ensembles* [14] 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 [2, 5, 19]. 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 [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 [18].

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

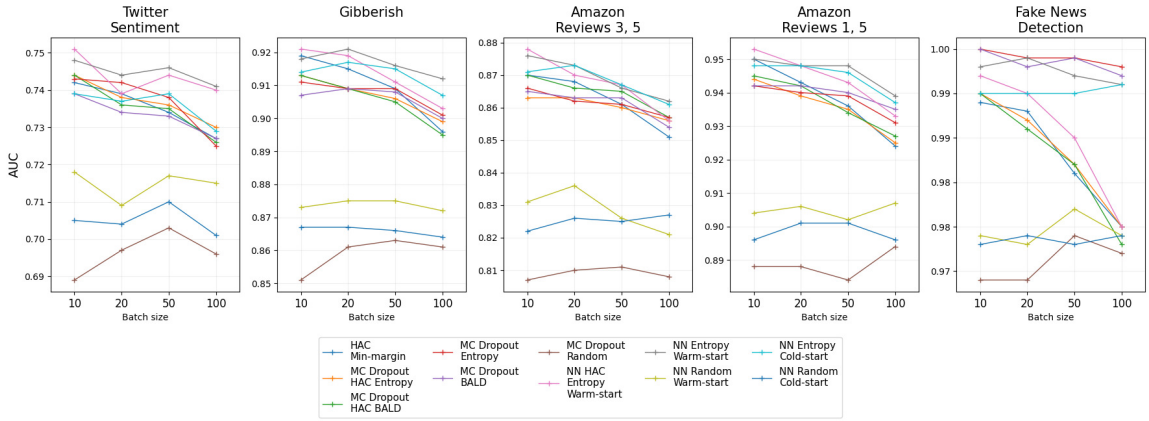

Fig. 1. AUC metrics for seven active learning and two random strategies after 1000 acquired samples given datasets and batch size. Prediction of the MC dropout classifiers is an average over ensemble members.

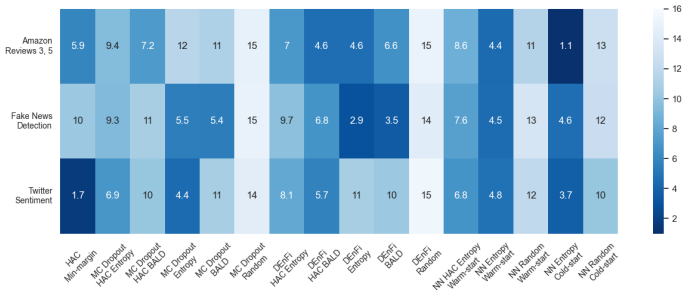

Fig. 2. Aggregated mean rank for 14 tuples of learning algorithms and acquisition functions given Amazon Reviews 3,5, Fake News Detection, and Twitter Sentiment for 50 active learning iterations with batch size 20.

strategy has to select a batch of b documents for off-line labeling. Strategies that try to approximate this selection using greedy search [13] 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_I. This approach may select samples close to each other, thus being redundant. **HAC** is a strategy based on the hierarchical clustering of a_I proposed in [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 [2], MC dropout with Entropy and BALD criteria using warm start [7], warm start ensemble learning with Entropy and BALD called DEnFi [18], 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 [8], Fake News Detection [1], two pairs of categories from Amazon Reviews Keung et al. [12], and Gibberish [21] 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. The only exception is Fake News Detection where only 4000 documents are available (2000 text documents per category).

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 [22], 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.

Evaluation:

All algorithms were compared on the area under the curve (AUC) [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 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 [3] using order statistics.

4 RESULTS AND DISCUSSIONS

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. This behavior is a result of a good neural network parameters distribution estimate. However, in other datasets, DEnFi performed as good as a single neural network, and is not worth the computational cost. As a result, the algorithm 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

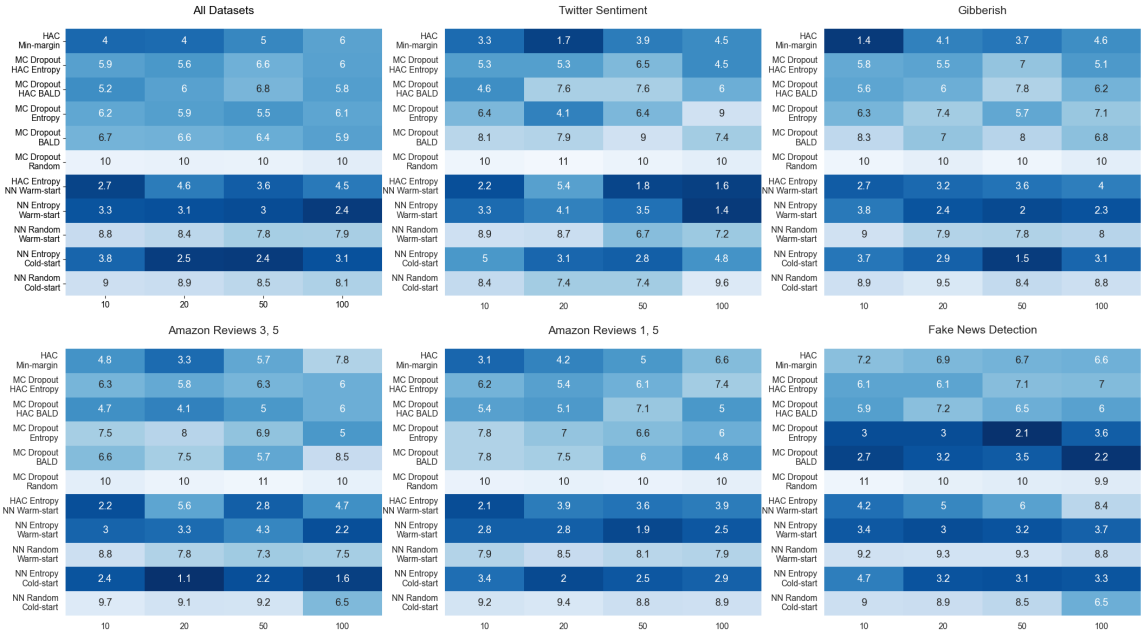

Fig. 3. Aggregated rank for 7 active learning algorithms and two random strategies averaged over datasets as a function of different batch sizes.

parametric uncertainty (now represented only by the warm start 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 our generalization of Entropy and BALD methods for the ensemble techniques seems insignificant, Figure 3. HAC methods perform well for some cases, but a simple entropy approach shows more stable results for the majority of problems, batch sizes, and initializations.

Initialization of the training: The proposed modification on warm start strategies (HAC Entropy, NN Entropy, and NN Random warm start) are better or comparable in performance to the cold start (HAC Min-margin, NN Entropy, NN Random); 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 [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. Smaller batch sizes and the proposed generalization to warm start HAC outperforms the cold start approach in most of cases. The batch advantage diminishes for sizes of 50 and 100 where the top selection strategy achieves comparable (ensembles) or better (single NN) results. We project that the most informative samples in our datasets are clustered in small groups, hence the selection of a batch with a large enough size contains all important samples.

The contribution of this paper lies in a comparative study where we decomposed different algorithms into building blocks and generalized various approaches. For a better understanding of the methodology, we selected some approaches for demonstration. In Figure 4 a comparison of various active learning sequences is displayed. More specifically, the methods proposed by us to well studied HAC min-margin approach. The uncertainty bounds in the figure are illustrated

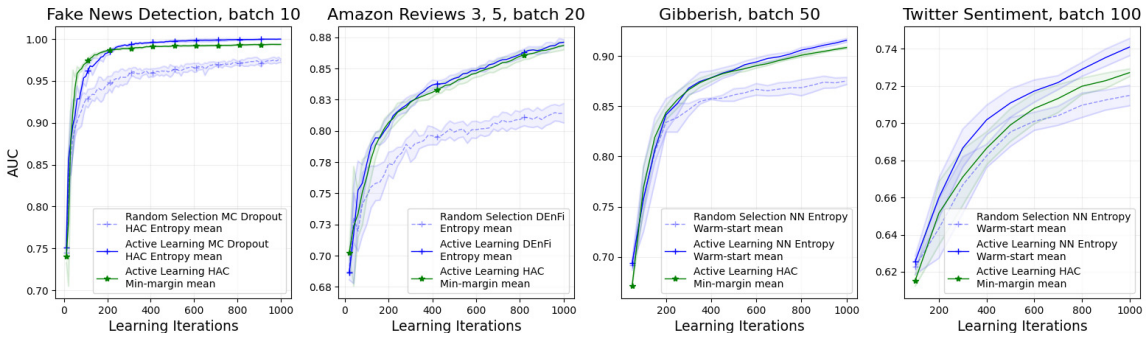

Fig. 4. Evolution of the mean AUC with a growing number of requests for the best algorithms representative vs HAC Min-margin given the batch size and dataset.

as one standard deviation from the mean value with respect to 5 runs. All algorithms were initially trained on 10 labeled text documents before sequential learning strategies were initialized.

5 CONCLUSION

We have studied the influence of various factors (i. e. acquisition functions, batch sizes, neural networks initialization) of active learning algorithms and their performance on cover context integrity, fake news detection, and sentiment classification tasks. While complex algorithms such as deep ensembles (DEnFi and Dropout MC) sometimes achieve good performance (Fake News detection), the winner, on average, is the classical prediction entropy of a single neural network with a few proposed modifications like warm start. Although the performance of the warm start method can sometimes be the same assume the cold start, the undeniable benefit is a lower computational cost. The selection of the batch size for annotation is also important. The agglomerative clustering improves performance for smaller batch sizes and may show better results than a more general method like entropy sampling.

ACKNOWLEDGMENTS

This research was supported by the project TL05000057, The Technology Agency of the Czech Republic www.tacr.cz, within the ETA Programme.

We would like to also express our gratitude to Deep Discovery company for computational power provision.

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*, volume 34, pages 11933–11944. Curran Associates, Inc.

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

[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*.

[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)*, pages 7949–7962.

[6] Tom Fawcett. 2006. An introduction to roc analysis. *Pattern recognition letters*, 27(8):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*, pages 1183–1192. JMLR. org.

[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*.

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

[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*, pages 3–29. Springer.

[12] Phillip Keung, Yichao Lu, György Szarvas, and Noah A Smith. 2020. The multilingual amazon reviews corpus. *arXiv preprint arXiv:2010.02573*.

[13] 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.

[14] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413.

[15] 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*.

[16] Katerina Margatina, Loic Barrault, and Nikolaos Aletras. 2021. Bayesian active learning with pretrained language models. *arXiv preprint arXiv:2104.08320*.

[17] Sumanth Prabhu, Moosa Mohamed, and Hemant Misra. 2021. Multi-class text classification using bert-based active learning. *arXiv preprint arXiv:2104.14289*.

[18] 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)*, pages 87–94. IEEE.

[19] Christopher Schröder, Andreas Niekler, and Martin Potthast. 2021. Uncertainty-based query strategies for active learning with transformers. *CoRR*, abs/2107.05687.

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

[21] John Wackerow. 2020. Gibberish text classification. https://www.kaggle.com/datasets/johnwdata/gibberish-text-classification, accessed on 16 may 2022.

[22] Christopher John Cornish Hellaby Watkins. 1989. Learning from delayed rewards.