Batch Active Learning for Text Classification and Sentiment Analysis

Anonymous ACL submission

Abstract

The performance of text classification with supervised models is tied to quality and diversity of the data. The process of data collection and labeling may involve a lot of resources. The intuitive and the most standard approach is to sequentially extend a dataset with labeled data until reaching satisfactory metrics. Active learning techniques optimize the process of sequential unlabeled data selection, so that the annotations would provide the most information about the dataset. The problem of active learning becomes more complex when the sampling is done in batches. In this paper we show a study of advanced batch sampling techniques on text data and the problem of text classification and sentiment analysis. The study compares i) baseline algorithm based on agglomerative instance clustering with the subsequent sampling from clusters given minimum margin of class probabilities ii) wam start modifications of baseline techniques, and iii) Bayesian active learning baseline modification thanks to their ability of better representation of the classification uncertainty. The latter method in warm-start version too.

Transformers encoders show the state-of-the-art results in majority of NLP tasks. In this article, we use RoBERTa for text encoding. The methods are tested on three types datasets, context integrity (Kaggle Gibberish dataset), fake news detection (Kaggle Fake News Detection dataset) and sentiment classification (Twitter Sentiment140 and Amazon Review Classification datasets). We show that both warm-start and Bayesian baseline algorithm modifications outperform the state-of-art approach.

1 Introduction

The development of a text classifier on a new problem requires the availability of the training data and their labels. Labeling involves human annotators and a common practice is to label as many text documents as possible, train a classifier and search for new data and labels if the performance is unsatisfactory. Random choice of the documents for the data set extension can be costly because the new documents may not bring new information for the classification. Active learning strategy aims to select among available unlabeled documents those that the classifier is most uncertain about and queries an annotator for their labels. Therefore, it has the potential to greatly reduce the effort needed for the development of a new system. Its advantages have been demonstrated even for classical methods such as SVM (Tong and Koller, 2001), but recent shift to deep learning methods motivates revisiting of the topic. The most conventional active learning strategies select one unlabeled document after each training round to query due to simplicity of its selection. The next query document is selected only after the first on is labeled and the model re-trained. However, this strategy is impractical in text classification where annotators cannot wait for the models. Therefore, we pay special attention to strategies that select a batch of documents for querying.

Novel methods for *batch active learning* appear frequently, each demonstrating advantages on their benchmark data. One of the recent approaches demonstrate effectiveness even for batches of 5000 samples (Citovsky et al., 2021), combining the min-margin acquisition function with clustering under name Hierarchical Agglomerative Clustering (HAC). In this contribution, we investigate novel modifications of this idea using alternative acquisition functions. Specifically we propose to extend the HAC approach to Bayesian setting by replacing the min-margin by Bayesian acquisition function, BALD (Houlsby et al., 2011). However, the size of the minibatch is only one of many factors in performace of the active learning algorithms. Other factors are: i) selected acquisition function, ii) representation of uncertainty of the classifier, and iii) strategy of retraining of the network. The key contribution of our work is 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 various results. Batch active learning was studied in (Dor et al., 2020) for only one size of the batch (5o documents per query). Large sensitivity to the type of dataset was reported in

(Prabhu et al., 2021), where different method won for different data. Large variability of the results was also observed in (Jacobs et al., 2021). In (Schröder et al., 2021), the min-margin strategy was shown to be competitive to prediction entropy based method on a range of embeddings. The comparative studies shared similar properties, such as fixed network for embeddings (improvement with re-training can be expected (Margatina et al., 2021) but maybe too costly). All studies also assume cold start, i.e. completely new initialization of the classifier after each round of querying. This is motivated by the fear of over fitting which was demonstrated in (Hu et al., 2018) for hot start, i.e. continuation of training of the classifier. A compromise in the form of warm-start, i.e. initialization of the classifier weights by those of the previous classifier with added noise, was proposed in (Sahan et al., 2021).

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

2 Batch Active Learning Methods

Throughout the paper, we will use the RoBERTa embedding (Liu et al., 2019) to represent documents in the feature space. RoBERTa is a modified BERT transformer model (Devlin et al., 2018) that achieved comparable performance to BERT in (Schröder et al., 2021) and outperformed all other embedding in (Sahan et al., 2021). Representation of the i-th text document $\mathbf{x}_i$ 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 $\hat{\mathbf{y}}_k = \mathbf{y}(\theta, \mathbf{x}_k)$ predicting the class label for each document representation $\mathbf{x}_k$. In supervised setting, the classifier parameters are found by matching the prediction with 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 to the labeled, $\mathbf{X}^{(0)}$, and unlabeled parts, $\mathbf{X}_u^{(0)} = \mathbf{X} \backslash \mathbf{X}^{(0)}$. 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 factor as specified next.

1. Start of the algorithm: **Cold** initializing by random numbers, $\theta_{\text{init}}^{(i)} = \mathcal{N}(0, \sigma)$, where σ is given by the standard network init strategy, used most often (Dor et al., 2020; Citovsky et al., 2021; Schröder et al., 2021). **Hot** initailized by previous estimate, $\theta_{\text{init}}^{(i)} = \theta^{(i-1)}$,criticized in (Hu et al., 2018). **Warm** a combination of the above, $\theta_{\text{init}}^{(i)} = \theta^{(i-1)} + \mathcal{N}(0, \sigma)$, where σ is a hyper-parameter. Acvocated in (Sahan et al., 2021).

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, $a_k = A(\mathbf{x}_k, \theta^{(i)}), \forall \mathbf{x}_k \in \mathbf{X}_u^{(i)}$
3. select a batch of documents, $\tilde{\mathbf{X}} \subset \mathbf{X}$, for labeling using a_k
4. obtain $\tilde{\mathbf{y}}$ for $\tilde{\mathbf{X}}$ and extend the training set $\mathbf{X}^{(i+1)} = \mathbf{X}^{(i)} \cup \tilde{\mathbf{X}}, \mathbf{y}^{(i+1)} = \mathbf{y}^{(i)} \cup \tilde{\mathbf{y}}, i = i + 1$.

2. Uncertainty representation: Single network modeling uncertain prediction of the label only estimation of one hot encoding by a single network. This captures aleatoric uncertainty.

Ensemble of networks, each with different parametrization, capturing both aleatoric and epistemic uncertainty. We consider two method for generating the ensemble members: i) MC dropout (Gal et al., 2017), where ensemble members are generated by random draws of of the dropout layers, and ii) deep ensembles (Lakshminarayanan et al., 2017) where emsembles are trained independently. Note that MC dropout is computationally much cheaper.

3. Acquisition function **Entropy** exists in two forms, entropy of the prediction $a_k = \mathbb{H}(y|\mathbf{x}_k, \theta)$ for a single network, or expected entropy $a_k = \mathbb{E}_\theta \mathbb{H}(y|\mathbf{x}_k, \theta)$ for ensembles.

BALD is a mutual information metric, $a_k = \mathbb{E}_\theta \mathbb{H}(y|\mathbf{x}_k, \theta) - \mathbb{H}(y|\mathbf{x}_k)$ that is meaningful only for the ensembles.

Min-margin is a minimum of difference between class predictions $a_k = -\min_{c,d \in [1,C]}(y_c - y_d)$ where c is the number of clasess. Not that extreme of this criteria is equivalent to maximim entropy for binary classification.

4. Batch selection strategy **Top** selects top b samples from sorted values of a_k. This approach may select samples close to each other which are redundant.

HAC is a strategy proposed in (Citovsky et al., 2021), based on clustering of a_k and selecting top b samples from different clusters.

BatchBALD ???

Not From the range
========== ToDO

2.1 Deep Ensemble Filter (DEnFi):

is a deep ensemble method with 5 neural networks in the ensembles and warm-start training strategy [13] us-

ing weights of the ensemble members in the previous iteration as initial conditions for the new ensemble. Each weight is perturbed by an additive Gaussian noise of variance $q = 0.3$ which is a hyperparameter. In our experiments, the ensemble is trained with parameters $\text{initialization\_epochs} = 2500$ on the initial data and with additional $\text{warm\_start\_epochs} = 700$ epochs after each extension of the learning data set.

2.2 Dropout MC

is the standard algorithm [10] that trains only a single network with sampled dropout indices and uses the sampling even in the prediction step. Generation of the Monte Carlo prediction is obtained by sampling different values of the dropout binary variable and one forward pass of the network for each sample. We study warm-start with weights from the previous iteration perturbed by an additive noise of variance $q = 0.3$ with 700 epochs. Dropout rate is 0.2.

2.3 Softmax uncertainty

The simplest approach to uncertainty representation is a single neural network with a softmax output layer that considers uncertainty as the output of the softmax score. We add it to comparison since it is a baseline approach used in state-of-art in cold start version and has also been applied to active learning earlier in [4] as hot-start method without noise pertrubation. The model is trained to run 2500 epochs in every iteration for cold start and with additional 200 epochs after each extension of the learning data set for hot-start.

3 Experiment Setup

4 Results

4.1 Simulation

The methods were compared on different datasets and different batch size. The used datasets are positive/negative tweets from the Tweets [11] , Fake News Detection [8] , two pairs of categories from Amazon Reviews and Gibberish datasets. The batch sizes are 10, 20, 50 and 100 instances per active learning or random sampling iteration. Specifically, we compared active learning and random sampling strategies for different settings of algorithms, batch sizes and different representations of uncertainty. Each experiment was initiated by random choice of the initial training set of $l_0 = 10$ samples from 10000 text documents (5000 text documents per category), which were the initial 10000 documents of the datasets given categories. For each experiment we continue the active learning simulation until we sample 1000 with querying iterations. Hence, L ranges from 10 to 100 requests for annotation. The batch selection follows the ϵ -greedy approach [25] , i.e. the samples given the acquisition function is accepted with probability $\epsilon = \frac{\exp(l-3)}{\exp(l-3)+1}$. A batch of random documents is selected for labeling if not accepted. After each request, the classification performance is evaluated on the remaining part of the selected dataset (i.e. on the 9990 text documents in the first evaluation) using the area under the ROC curve (AUC) metrics [9] . In order to make the results statistically valid, we repeat the described simulation loop 5 times for all datasets.

4.2 Methods

We compare results for 12 different tuples of algorithms and acquisition functions that include i) five MC Dropout simulations with all acquisition functions, random sampling, all datasets, and all batch sizes, ii) three NN Warm-Start runs based on dropout algorithm with point-wise parameters distribution estimate for HAC Entropy, Entropy, random sampling acquisition functions, all datasets, and all batch sizes, iii) five DEnFi simulations with all acquisition functions, random sampling, three datasets, and only one batch size (due to the computational complexity), and iv) HAC Min-margin run with softmax uncertainty cold-start active learning strategy and HAC Entropy computed for all datasets and all batch sizes. The cold start strategy choice for HAC Min-margin is based on the experimental results from , where the hot start-methods (fine tuning without noise pertrubation) scored the worst. Hence, for the sake of not having computational biases we decided to train HAC Min-margin algorithm from scratch in every active learning iteration.

4.3 Text Classification

The comparison of AUC results is done with the ranking technique presented in . We compute the mean value over five simulations for each algorithm. Next, we calculate ranks in every iteration of active learning for algorithms with the same batch size and datasets. The methods are compared to the aggregated mean ranks. The mean values are computed for ranks of $100, 200, \ldots, 1000$ sampled instances. The reason of aggregation through a subset of all ranks is the understanding if more active learning iterations with smaller batch size or one with a larger one is better. The aggregated mean ranks for different datasets and batch sizes are in figure ...

References

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.

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*.

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

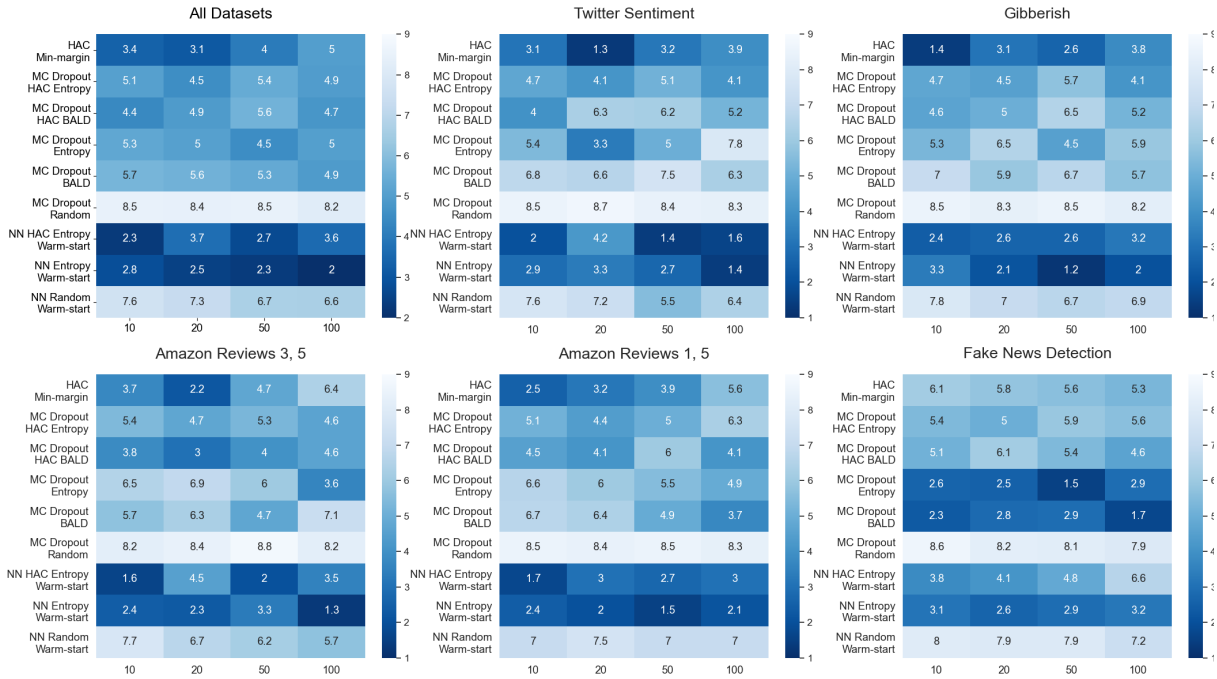

Figure 1

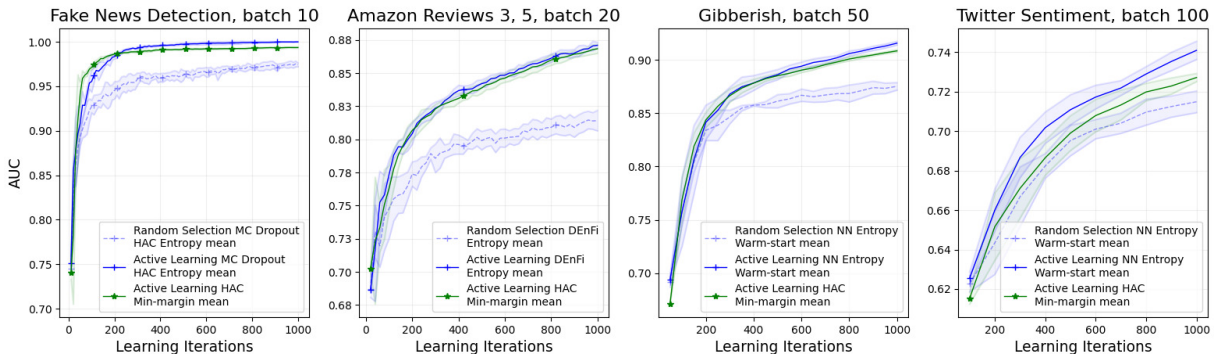

Figure 2

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.

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.

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

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

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.

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.

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*.

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

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

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.

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

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.