Incremental Machine Learning for Text Classification in Comment Moderation Systems

Anna Wolters[1(✉)], Kilian Müller[2], and Dennis M. Riehle[1]

[1] University of Koblenz-Landau, Koblenz, Germany
{awolters,riehle}@uni-koblenz.de
[2] University of Münster - ERCIS, Münster, Germany
kilian.mueller@ercis.uni-muenster.de

Abstract. Over the last decade, researchers presented (semi-)automated comment moderation systems (CMS) based on machine learning (ML) and natural language processing (NLP) techniques to support the identification of hateful and offensive comments in online discussion forums. A common challenge in providing and operating comment moderation systems is the dynamic nature of language. As language evolves over time, continuous performance evaluations and resource-inefficient model retraining are applied to ensure high-quality identification of hate speech in the long-term use of comment moderation systems. To study the potentials of adaptable machine learning models embedded in comment moderation systems, we present an incremental machine learning approach for semi-automated comment moderation systems. This study shows a comparison of incrementally-trained ML models and batch-trained ML models used in comment moderation systems.

Keywords: Incremental learning · Text classification · Comment moderation systems

1 Introduction

Increasing online communication confronts journalists in media and news corporations with a task that is "not historically part of a journalist's daily routines" [15, p. 1022]: comment moderation. Journalists feel responsible for eliminating hate speech and other forms of abusive language in order to prevent discussions from deviating or escalating, as well as to fulfill legal obligations [5,28]. Manual comment moderation, however, becomes a time-consuming task, as journalists are usually facing large amounts of comment data [5]. For this reason, some organizations take the measure of completely banning comment sections from their websites or outsource the moderation activities to overcome the issues of comment moderation [15]. In addition to the unmanageable number of comments, comment moderation is a challenging task in itself, since journalists are

F. Spezzano et al. (Eds.): MISDOOM 2022, LNCS 13545, pp. 138–153, 2022.
https://doi.org/10.1007/978-3-031-18253-2_10

in the dilemma of eliminating hateful comments, while guaranteeing freedom of speech [28]. To find a viable solution that appropriately addresses the challenges of comment moderation, researchers and platform operators are investigating automated classification mechanisms that use ML and NLP methods to identify abusive comments [16,29,39]. These automated methods offer the possibility to keep discussion forums running, while classifying and removing hateful comments automatically [38].

Providing and operating (semi-)automated comment moderation system include resource-intensive tasks ranging from data acquisition and labeling to constant maintenance and evaluation of the system. For the system development, labeled datasets are required as one of the key resources for training machine learning models. Initial data acquisition and labeling, however, are costly and time-inefficient tasks [36]. In addition to that, there is a need for repeated evaluation of the models' performances, since language evolves continuously [17]. As a consequence, we identify the need for action to study resource-efficient and more dynamic techniques that adequately address the presented obstacles for the operation of comment moderation systems. Additionally, we expect a comment moderation system to be able to adjust to changes in the data pattern, as any real-world application must be capable of doing [4]. A concept that supports the continuous adjustment of machine learning models is *incremental learning*. In the given research, we study how incrementally-trained ML models perform in comparison to batch-trained models in the context of semi-automated comment moderation. We present a learning strategy that aligns with human-in-the-loop ML techniques, as our solution continuously integrates domain knowledge provided by moderators [43].

This paper is structured as follows. In Sect. 2, we provide background information on comment moderation systems, as well as fundamental definitions of batch learning and incremental learning. Additionally, we present the results of a structured literature search on incremental learning techniques used in text classification tasks. The applied research method is presented in Sect. 3. Next, we describe the objectives for our development process and explain the performed development steps in more detail in Sect. 4. Section 5 covers the demonstration of our learning strategy and presents the results of our iterative evaluation. Our paper closes with a concluding discussion of the research findings and limitations in Sect. 6.

2 Theoretical Background

2.1 Comment Moderation and Comment Moderation Systems

Comment moderation and, in particular, automated comment moderation with the help of information technology finds increasing interest in news outlets, online communities, and academia [16,38]. Journalists perform comment moderation in order to eliminate *hate speech* or *abusive language* from their organization's website [5]. However, definitions of terminology, such as abusive language, hate speech, or harassment, are not clearly established in academia [7]. Removing comments

from the comment thread can potentially be compared to censorship [5], leaving journalists in the dilemma of removing uncivil and hateful comments, while ensuring freedom of speech [28]. Comment moderation is usually performed as pre- or post-moderation and is regulated by laws, ethics, and further guidelines [28].

Supporting technology in the form of comment moderation systems help journalists and community managers to better cope with the large amount of data and frees the journalists from their additional work as comment moderators. [39], for instance, present a data mining approach to automatically identify hateful or offensive comments. They combine textual data patterns to elements derived from the social network the comments were posted in, such as the interaction of users. [16] discuss the role of automated comment moderation on the social platform *Reddit* and demonstrate the effect of comment moderation in an online community. [5] investigate a simplified approach to automate comment moderation, in which they used a pre-defined list of swear words to identify comments of abusive language. However, contrary to their initial assumption, this approach did not present itself as a useful mechanism. Thus, more advanced techniques are necessary to support the automation of comment moderation.

2.2 Batch Learning vs. Incremental Learning

Machine learning is usually performed using *batch learning*, which is also referred to as *offline learning* [4], *single-task learning* [44], or *isolated learning* [11]. Batch learning works under the assumption that the data distribution of training and test data is static and given in the training phase of the model creation [22]. After executing the training phase and deploying the machine learning instance to production, the model is applied to classify or cluster incoming data, i.e., there is a clear distinction between the training and testing phase. In a dynamic environment, however, where the entire data distribution is not given in the training phase, machine learning instances must be able to learn continuously in order to adjust to their environment [4]. To reflect the dynamic environment in the learning instance, an incremental learning technique can be applied.

Incremental learning techniques aim at creating machine learning instances that adapt to their environments while retaining previously learned knowledge [30]. Further, incremental learning approaches are characterized by their ability to learn new classes as well as their independence from previously learned data [27]. Naturally, incremental learning is also better suited to address dynamic learning problems, in which loading large datasets into memory is not feasible [4]. *Online learning* [20], *lifelong learning* [27], *evolutionary learning* [20], or *continual learning* [23] can be used as synonyms for incremental machine learning. Frequently, however, these terms are also distinguished from one another. A learning strategy is described as *online learning* to stress that it is able to process one instance at a time, thus may be applied to data streams [1]. The term *lifelong learning*, however, is used to emphasize the idea to replicate human learning [11]. *Evolutionary learning* can be understood as a synonym to lifelong learning [20]. Incremental learning techniques can be further distinguished between batch

incremental learning and single-instance learning, describing the portion of data that is processed at a time [22].

Besides their ability to work in dynamic environments, incremental learning techniques have further advantages over batch learning approaches. Incremental learning methods offer computational benefits and lead to a reduced demand in storage capacity, since a large dataset does not have to be stored in memory throughout the entire training process [30]. The key challenge in incremental learning, however, is finding the right balance between learning new knowledge and retaining previously learned information, which is known as the *stability-plasticity dilemma* [8]. In the stability-plasticity dilemma, the balance of learning and retaining knowledge is described as a design question for learning systems [8]. If a system is designed to tend to forget previously learned knowledge, it is prone to *catastrophic forgetting* [23,33].

2.3 Incremental Learning in Text Classification

Among several use cases, incremental machine learning techniques also find application in text classification (TC) tasks. In a structured literature search based on the methodology proposed by [40], the literature databases *Scopus* and *Web of Science* were searched for research that cover the usage of incremental learning techniques in text classification tasks. The search string was set to a combination of the terms *incremental learning* and *text classification*. We did not make any further restrictions in order to obtain a broader range of results. After excluding duplicate literature, a total of 19 papers were identified, which were published between 2004 and 2020. Within the identified set of literature, we recognize a shared understanding of the benefits of incremental machine learning in a variety of domains as well as a wide range of tasks. Additionally, similar observations with regard to the challenges of incremental learning are given. Examples of domains covered are the identification of spam in e-mails [37], or the detection of evidence of breast cancer in medical reports [9].

A wide range of approaches to implement incremental learning strategies for TC tasks was identified in the literature search. While the majority of the literature covers supervised machine learning, unsupervised and semi-supervised learning techniques are also included. Within the literature on supervised incremental machine learning for text classification, well-known baseline algorithms for text classification such as Naïve Bayes (NB) [18] or Support Vector Machines (SVM) [46] are studied. A multi-class approach was presented in [4], which is based on the work by [32]. In [31], a non-probabilistic approach based on the *Winnow* algorithm is investigated. Further, deep learning techniques such as neural networks, particularly Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN), are also applied to implement an incremental learning strategy for TC tasks [9,12,30]. Within the set of literature focusing on unsupervised techniques, different clustering strategies are presented [14,21,37]. Additionally, ensemble learning is covered, which yields the advantage to better balance learning new information and forgetting formerly learned knowledge by continuously evaluating and exchanging classifiers within the ensemble [35].

Lastly, semi-supervised methods are also presented, which additionally offer the resource-efficient advantage to include labeled as well as unlabeled data for a TC task using incremental learning [19,36,41]. Further, in [3], a bibliometric survey on incremental learning in text classification is presented.

Despite the differences in the usage of incremental learning in text classification, similar motivational factors and a shared understanding of the benefits were identified. Incremental learning is used to address high memory utilization, which is a common concern in batch learning [4,30,46]. Further, the lack of adaptability of batch learning techniques motivates the use of incremental learning techniques as well as its inefficiency, particularly with regard to maintenance tasks such as completely retraining machine learning models [4,34]. Incremental learning is used to prevent performance degradation of machine learning models used in production, i.e., to guarantee high classification performance over time [34,46]. Similarly, a set of common concerns was identified. As textual data is usually linked to a high dimensionality of the feature space, the constant increase of the feature space due to continuous learning of new knowledge is a common concern [10,18]. In the set of literature examined in our literature search, catastrophic forgetting is also considered as a potential issue for incremental learning in text classification [30].

3 Research Approach

Our research connects incremental learning in text classification tasks with comment moderation systems. This paper is based on the architecture and moderation process of a research project presented by [29]. In this research, we aim at developing dynamic machine learning models for semi-automated comment moderation systems using incremental machine learning. Our research follows a design science approach based on the Design Science Research Methodology (DSRM) proposed by [26], using a problem-centered approach. The artifacts of our research are machine learning models that are able to learn incrementally when embedded in a semi-automated comment moderation system. Additionally, we present a testing environment to support the comparison of incrementally-trained and batch-trained ML models in the context of comment moderation systems.

First, according to [26], researchers are demanded to identify the research problem in order to capture the complexity of the problem. In line with that, our research is firstly motivated and introduced: semi-automated text classification approaches used in comment moderation systems are affected by the dynamic nature of language. We study a flexible learning technique that continuously adjusts to the environment and contrast it to traditional batch learning (Sects. 1 and 2). Second, objectives for the research outlet must be defined. Third, the artifact's design is planned, and the actual development is performed. We define requirements for a dynamic learning approach embedded in comment moderation systems and apply the requirements to the artifact development (Sect. 4). Next, the developed artifacts are demonstrated and evaluated with regard to

their ability to continuously learn new knowledge and in comparison to techniques based on batch learning (Sect. 5). We perform an iterative performance evaluation based on common performance metrics to compare both learning techniques. Thus, this section also covers the evaluation phase according to the methodology by [26]. Last, the paper closes with a discussion of the research findings and an outlook on future research (Sect. 6).

4 Incremental ML in Comment Moderation Systems

4.1 Design Objectives

For our research, we consider a semi-automated comment moderation system where the extent of incoming comments that are automatically moderated is controlled by the level of certainty of the machine learning instance as measured by the prediction probability. Manual moderation is only performed if the machine learning instance is not able to assign the comment a label with a probability above a pre-defined threshold. Thus, a certain amount of moderation must still be performed manually, which will be used as the foundation for incrementally updating the machine learning instance. During the operation of a CMS, each manual moderation activity generates a labeled data instance as a by-product. Our goal is to incorporate a single-instance incremental learning technique, in which the machine learning model is updated after processing manual moderation decisions performed by a user. In this way, a resource-efficient and continuous learning strategy is created, and human knowledge provided in the form of the labeled comment is continuously integrated.

As a testing environment, we create a simplified comment moderation system. We simulate a steady data flow and manual moderation activities with the help of a labeled dataset. A manual moderation decision is simulated by deriving the true label of a data instance from the labeled dataset. Given that, the application does not rely on user input, but makes use of the knowledge provided in the labeled dataset. A continuous arrival of new data is replicated by processing one data instance at a time, chronologically ordered based on the creation timestamp of the data instances. Further, we aim at contrasting an incremental learning technique with batch learning. To do so, we additionally create batch-trained classifiers and use them as a reference for the subsequent evaluation and incremental training procedure. In order to better observe the adaptation of the models and the effect of incremental learning, time-intensive training of the models in advance is deliberately avoided, while we perform proper batch learning on the benchmark models. For the ML models that will subsequently be trained incrementally, we train the models in advance to guarantee that their initial performance is at least better than random guessing (i.e., validation accuracy: 50%).

4.2 Development

We study incremental learning approaches based on three different algorithms: Naïve Bayes (NB), Adaptive Boosting (AdaBoost), and Logistic Regression

(LR). The algorithms were selected because of their simplicity and suitability for incremental learning [21,24,45]. NB is a prominent algorithm for incremental text classification tasks, as its simplicity allows for a straightforward update of the classifier [21]. An incremental learning variant of a NB classifier updates its prior probability by turning the posterior probability to the new prior probability [21]. While in regular adaptive boosting, the weight of an observation is changed based on the entire training set, incremental AdaBoost uses sampling with replacement based on a Poisson distribution [24]. As a base estimator, we use simple tree-based classifiers. For incremental LR, the incremental adjustment is performed by updating the logistic regression parameters using stochastic gradient descent [45]. For each algorithm, we develop a batch-trained and incrementally-trained ML model. We implemented the algorithms that will be trained incrementally using the Python library *river* and thereby fulfill the goal to create ML models based on single-instance incremental learning [22].

For the model training, we additionally use the Python packages *scikit-learn* [25] and *scikit-optimize*[1] as well as *spacy*[2] and *NLTK*[3] for preprocessing the textual data. The model training is based on a labeled dataset, some of which is publicly available [2]. The dataset contains around 430,000 German comments and covers the period from November 2018 to July 2021. A binary labeling was used to differentiate the comment data between *rejected (positive)*, i.e. hateful or offensive comments, and *accepted (negative)* instances [2]. With the help of community managers from a German newspaper as well as crowd workers the dataset was labeled. Only around 25,000 data instances, roughly 6% of the entire dataset, are labeled as positive, thus making the dataset highly imbalanced.

Prior to applying ML algorithms on the data, the dataset is preprocessed using common NLP techniques. We apply stop word removal, lemmatization, lowercase conversion and remove special and single characters. Further, we use the *term frequency-inverse document frequency* (TF-IDF) weighting scheme to vectorize the textual data due to its simplicity and suitability for incremental learning. The implementation of the vectorization for the incrementally-trained models supports a continuous extension of the feature space when new tokens appear in the comment data, which appropriately reflects a continuous learning behavior [6]. Next, hyperparameter tuning is performed using *GridSearch* and *Bayesian Optimization* using *scikit-learn* and *scikit-optimize*. We apply well-established and commonly used techniques and additionally align the data preparation steps with the suggestions by [2] in order to simplify the learning process and focus on the comparison of the learning techniques.

Based on the preprocessed data and optimal hyperparameter settings, we perform the (initial) training of the classifiers. For the batch-trained models, we created a balanced subset of the entire dataset (∼34,000 instances). For a reduced initial performance, only a subset of the newly created balanced dataset is used to initially train the models that will perform incremental learning in the

[1] https://scikit-optimize.github.io/.
[2] https://spacy.io/.
[3] https://www.nltk.org/.

subsequent step (∼4,000 instances). The data samples in the subset were randomly sampled. It is important to note that we utilized the oldest data available in the dataset for the (initial) training and newer data for the performance evaluation in order to correctly represent the time dimension, as we aim at understanding how the performance of ML models changes while language evolves over time. Further, there is no overlap between data used for the training and evaluation, and the dataset for the evaluation immediately follows the training dataset in time.

The final part of the development step covers the testing environment used for incrementally updating the classifiers, as well as to evaluate their performance. For the development, we used the Python library *streamlit*[4] that supports the creation of web applications for data science tasks. We use the application to simulate the constant data flow and moderation activities, as well as to trigger the incremental model updates. In the given case, the data flow is paused until a comment is processed and, if necessary, an update iteration of the machine learning model is performed. As specified above, the update of a model is triggered when the certainty of the ML instance for classifying an incoming comment based on the prediction probability falls below a pre-defined threshold. A false classification does not initiate an update.

5 Demonstration and Evaluation

For the performance evaluation, we apply *prequential evaluation*, or *test-then-train* evaluation, which is a well-known technique to evaluate the performance of data streams [1,4,13]. The term *prequential* is the short form for *predictive sequential* [1]. The first step in the prequential evaluation is computing the predictions for each incoming data instance. In a second step, every single instance is used to update the classifier [13]. We use a modification of the prequential evaluation since we restrict the update of the classifiers to the manual moderation decision, whereas we evaluate the performance by updating the performance metrics for each incoming instance. Further, we apply immediate feedback, i.e., whenever a manual moderation activity is simulated, the data flow is paused [4,37]. This ensures that we keep the chronological order of the comments based on their creation timestamp. We evaluate the performance outcome of the incremental model with regard to varying values for the threshold for the prediction probability as well as the observation period.

The performance evaluation is based on the metrics accuracy, the area under the receiver operating curve (ROCAUC), precision, and recall. In combination, these performance metrics allow for a comprehensive and appropriate evaluation of the classifiers' performances, when evaluated on imbalanced datasets. Based on simple cross-validation, we determine the starting performance of each classifier based on a validation set. As we study a single-instance incremental learning approach, we apply the API for metrics provided by the Python package *river*. The metrics are updated based on the true label and the predicted

[4] https://streamlit.io/.

Table 1. Initial performance per classifier

Classifier	Initial performance metrics			
	Accuracy	ROCAUC	Precision	Recall
NB (BL/IL)	0.70/0.63	0.77/0.71	0.74/0.62	0.61/0.67
AB (BL/IL)	0.66/0.53	0.72/0.52	0.66/0.51	0.66/0.55
LR (BL/IL)	0.70/0.60	0.76/0.64	0.74/0.60	0.61/0.54

label or the prediction probability. In Table 1, the initial performance values, i.e., the starting point for the subsequent evaluation, for the batch-trained models (BL) and models that will be incrementally-trained (IL) are depicted. In the subsequent performance evaluation, the metrics of both types of classifiers will be updated after processing a single instance. The results in Table 1 indicate a decent performance for each batch-trained classifier. These results justify the use of the models as performance benchmarks in the given research. Each classifier that will be incrementally trained in the subsequent step, shows a reduced initial performance as desired. Each of the weak classifiers performs better than random guessing, i.e., the previously defined design objective is fulfilled. In addition to the performance metrics, we also record the number of misclassifications performed by each model in the subsequent performance evaluation. Further, we additionally save the label of each instance that was used to perform an incremental update in order to get more insight into the training dataset.

We executed the performance evaluation in an iterative manner to study the effect of different parameter settings on the classifiers' performances. First, we focused on the effect of varying threshold values for the prediction probability, i.e., we regulated the amount of manual comment moderation in our system. Naturally, a higher prediction probability creates a larger number of comments that must be manually moderated in the semi-automated setting. A larger amount of manual moderation, however, also implies more frequent updates of the incrementally-trained models. We noticed, that a higher frequency of incremental updates of the model leads to fewer misclassifications performed by the incrementally-learned classifier. Additionally, a high increase in the accuracy as well as the ROCAUC score of the incremental models was recorded. However, in the given context, a high accuracy might give a misleading impression of the true performance, since the dataset is imbalanced. For the incrementally-trained models, we observe a strong decline in the recall and/or precision of the classifiers. As compared to the incrementally-trained models, the batch-trained models, however, rather show slight performance degradation with regard to the development of the classifiers' accuracy and ROCAUC score. However, they also demonstrate a constant value for their recall scores. Based on these initial observations, we came to the interim conclusion that the batch-trained classifiers outperform the incrementally-trained models with regard to their ability to properly identify the minority class. We attribute this observation to the differences in the distribution of the training data. While we used a balanced dataset

for the batch training, which is a common practice to allow for proper distinction between classes, the incrementally-trained models work on an imbalanced dataset. Thus, we continued our evaluation by incorporating sampling strategies to the incremental learning process.

First, we incorporated random undersampling of the majority class to the incremental learning process. The sampling strategy extends our set of parameters by the desired class ratio for the data sampling. We executed different performance evaluations using varying parameter settings for the desired class ratio, as well as the threshold for the prediction probability and the observation horizon. The results, however, indicated that undersampling the majority class does not improve the incremental learning process. Rather, undersampling showed a worse performance than in previous iterations of our evaluation. In particular, we noticed a strong increase in the number of misclassifications performed by the incrementally-trained models.

Second, we therefore applied random oversampling of the minority class instead of undersampling the majority class. We were able to see minor performance improvements as compared to previous iterations where no sampling strategy was used. Therefore, we continued applying oversampling to the incremental learning approach and observed that an equal ratio between both classes creates the best results. Most likely, oversampling outperforms undersampling in the given context, as undersampling reduces the training dataset for the incremental model updates [42]. Still, the loss in the classifiers' precision and recall value remain. In several cases, however, the significance of the loss is reduced when oversampling with an equal ratio between the classes was applied. Still, an indication of a potential competitiveness of incremental learning techniques with batch-trained models is the lower number of misclassifications performed by an incrementally-trained ML model. For the comparison between incremental learning and batch learning, we additionally observed that the incrementally-trained models are able to outperform their batch-trained counterpart with regard to the accuracy. However, the batch classifiers showed more stable results, particularly concerning the recall and precision value.

In Fig. 1, exemplary results of our research are presented. The figure shows the performance development of each incrementally-trained classifier based on the selected performance metrics. In the legend of the performance plot, the starting value for each metric of each classifier is given. The results for the AdaBoost, Naïve Bayes, and Logistic Regression classifier are depicted as dashed, dotted, and solid lines, respectively. For each incrementally-trained classifier, oversampling with an equal ration between both classes was applied. Additionally, the threshold for the prediction probability was set to 80% and the performance development was observed over the course of three months with equals an amount of roughly 36,500 comments. Figure 1 shows the performance development of the classifiers after processing each of the comments in the evaluation dataset. Interestingly, the development of the accuracy, ROCAUC, and recall score of each incrementally-trained classifier show a similar pattern. For the accuracy of each classifier, a rather strong increase can be observed at the

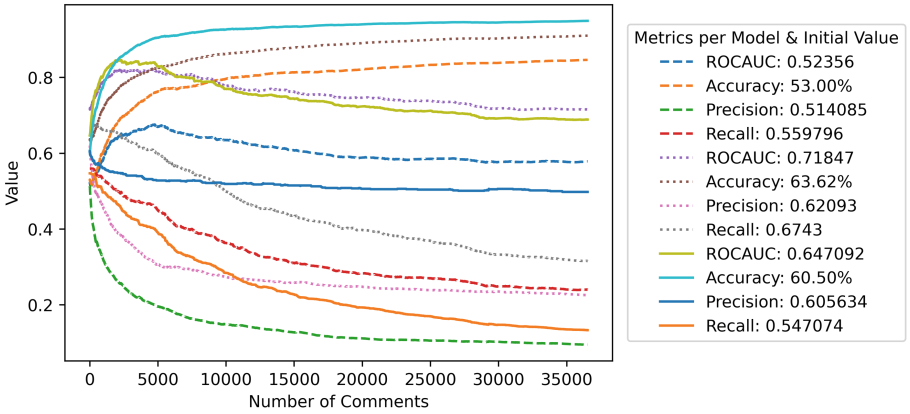

Fig. 1. Performance development of incrementally-trained models (--- AdaBoost, ····· Naïve Bayes, —— Logistic Regression)

beginning of the observation period, while the slope of the increase is reduced after around one third of the observation horizon. Similarly, each model shows an increase in its ROCAUC score at the beginning of the observation period and a slight decrease of the score after processing around 8,000 comments. In roughly the last third of the observation period, the values for the ROCAUC scores appear to be rather stagnant. Additionally, the final ROCAUC scores are very close to the initial performance scores. While we observe a constant decline of the classifiers' recall values, the development of the precision score differs. Here, the logistic regression model demonstrates a slight decrease at the beginning of the observation period, but shows a rather constant value after processing around 5,000 comments. Both remaining models, however, show a very strong decline in their precision at the beginning of the observation and a reduced but still constant decrease towards the end of the observation period. Still, neither the recall nor the precision value of any incrementally-trained classifier demonstrates an improvement during the observation.

In addition, we observed differences in the training datasets for the incremental learning for each classifier. These differences concern the size of the datasets as well as the given distribution between the classes, indicating different levels of confidence between the classifiers and among the classes. The Logistic Regression classifier appears to have a rather high level of confidence in its predictions, since the training dataset for incrementally updating the classifier was rather small. Additionally, data instances from the minority class are slightly less underrepresented, accounting for about 14% of the training data set. For the NB and AdaBoost classifier, however, the training dataset always roughly reflects the overall distribution of the classes. The training set for the incremental Naïve Bayes classifier contains around a third of the evaluation dataset, while more than half of the dataset was used for updating the incrementally-trained AdaBoost classifier.

We come to the conclusion that the batch-trained classifiers outperform incrementally-trained models with regard to the proper identification of hate speech, i.e., correctly classifying the positive class. These inclusive results indicate that more in-depth research on the reasons for the demonstrated developments is necessary. We conclude that the incrementally-trained models are not able to properly distinguish between classes and thus fail to properly identify hate speech in comment data. The monitoring of the development of the performance metrics over the course of time shows that the incrementally-trained classifier are not fully able to compete with batch-trained ML models.

6 Concluding Discussion and Future Work

In our research, we investigated how incrementally-trained ML models perform in comparison to batch-trained ML models in the context of semi-automated comment moderation systems. In a testing environment, we simulated a continuous data flow as well as manual moderation decisions, which we used to incrementally train the underlying machine learning model. In several performance evaluations, we compared incrementally-trained ML models to batch-trained models when embedded in a comment moderation system.

In each evaluation iteration, it became evident that the ability of a machine learning model which is continuously learned on incoming comments to compete against batch learning is limited. In several cases, we observed fewer misclassifications performed by the incrementally-trained classifiers than the batch-trained models. Still, the performance development of the incrementally-trained models showed insufficient improvements, since the incremental training of the classifier does not improve the precision and recall score of the model. Possibly, the lack of improvement is attributed to the underlying data the incremental training was performed on. Although we incorporated sampling strategies, we regard the imbalance in the dataset as a potential cause for the insufficient improvement. Additionally, the high dimensionality of the feature space and its constant increase might be a possible reasoning for the classifier's difficulties to properly distinguish and learn both classes. More advanced sampling strategies and feature selection techniques might be appropriate to improve the data and thus the incremental learning [18]. Further, our research is limited with regard to the evaluation metrics used. Nevertheless, it became evident that oversampling the minority class when updating the classifier incrementally improves the performance of the classifier with regard to properly identify abusive language. Contrary to that, undersampling caused weaker performances of the incremental classifiers due to the reduction in the training dataset. Given the current limitations of our research, we regard a combination of traditional batch learning and incremental updates of the classifier as an appropriate technique. It would ensure a constant level of classification quality in the long-term use of the semi-automated comment moderation system, as well as circumvent complete retraining steps of the classifier in the future.

Besides the limitations regarding the results of the incremental training, our study is also limited with regard to the simulation of semi-automated comment moderation. We assumed that manual moderation decisions are strictly made in the chronological order and without any delay. Still, our research opens up the debate on incremental machine learning techniques for comment moderation systems and introduces the use of more dynamic learning strategies in semi-automated comment moderation systems. Future work should aim at understanding the observed behavior in more detail, and finding solutions to improve the outcome of an incremental learning technique embedded in semi-automated comment moderation systems.

References

1. Ashfahani, A.: Autonomous deep learning: incremental learning of deep neural networks for evolving data streams. In: IEEE International Conference on Data Mining Workshops, ICDMW 2019, Beijing, China, pp. 83–90 (2019)
2. Assenmacher, D., Niemann, M., Müller, K., Seiler, M., Riehle, D.M., Trautmann, H.: RP-Mod & RP-Crowd: moderator- and crowd-annotated German news comment datasets. In: Proceedings of the NeurIPS Datasets and Benchmarks 2021, Virtual, pp. 1–14 (2021)
3. Barve, Y., Mulay, P.: Bibliometric survey on incremental learning in text classification algorithms for false information detection. Libr. Philos. Pract. **2020**, 2388–2392 (2020)
4. Bittencourt, M.M., Silva, R.M., Almeida, T.A.: ML-MDLText: an efficient and lightweight multilabel text classifier with incremental learning. Appl. Soft Comput. **96**, 1–15 (2020)
5. Boberg, S., Schatto-Eckrodt, T., Frischlich, L., Quandt, T.: The moral gatekeeper? Moderation and deletion of user-generated content in a leading news forum. Media Commun. **6**, 58–69 (2018)
6. Brants, T., Chen, F., Farahat, A.: A system for new event detection. In: Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval, Toronto, Canada, pp. 330–337 (2003)
7. Brunk, J., Niemann, M., Riehle, D.M.: Can analytics as a service save the online discussion culture? - The case of comment moderation in the media industry. In: Proceedings - 21st IEEE Conference on Business Informatics, CBI 2019, Moscow, Russia, pp. 472–481 (2019)
8. Carpenter, G.A., Grossberg, S.: The art of adaptive pattern recognition by a self-organizing neural network. Computer **21**, 77–88 (1988)
9. Chen, D., Qian, G., Shi, C., Pan, Q.: Breast cancer malignancy prediction using incremental combination of multiple recurrent neural networks. In: Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S. (eds.) ICONIP 2017. LNCS, vol. 10635, pp. 43–52. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70096-0_5
10. Chen, Z., Huang, L., Murphey, Y.L.: Incremental learning for text document classification. In: Proceedings 2007 International Joint Conference on Neural Networks, Orlando, USA, pp. 2592–2597 (2007)
11. Chen, Z., Liu, B.: Lifelong machine learning. In: Synthesis Lectures on Artificial Intelligence and Machine Learning, vol. 10, pp. 1–145 (2016)

12. D'Andecy, V., Joseph, A., Cuenca, J., Ogier, J.M.: Discourse descriptor for document incremental classification comparison with deep learning. In: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, Sydney, Australia, pp. 467–472 (2019)
13. Dawid, A.P.: Present position and potential developments: some personal views: statistical theory: the prequential approach. J. Roy. Stat. Soc. Ser. A (General) **147**, 278–292 (1984)
14. Doan, T., Kalita, J.: Overcoming the challenge for text classification in the open world. In: 2017 IEEE 7th Annual Computing and Communication Workshop and Conference, CCWC 2017, Las Vegas, USA, pp. 1–7 (2017)
15. Ferrucci, P., Wolfgang, J.D.: Inside or out? Perceptions of how differing types of comment moderation impact practice. Journal. Stud. **22**, 1010–1027 (2021)
16. Jhaver, S., Birman, I., Gilbert, E., Bruckman, A.: Human-machine collaboration for content regulation. ACM Trans. Comput. Hum. Interact. **26**(5), 1–35 (2019)
17. Karjus, A., Blythe, R., Kirby, S., Smith, K.: Quantifying the dynamics of topical fluctuations in language. Lang. Dyn. Change **10**, 86–125 (2020)
18. Katakis, I., Tsoumakas, G., Vlahavas, I.: On the utility of incremental feature selection for the classification of textual data streams. In: Bozanis, P., Houstis, E.N. (eds.) PCI 2005. LNCS, vol. 3746, pp. 338–348. Springer, Heidelberg (2005). https://doi.org/10.1007/11573036_32
19. Liu, L., Liang, Q.: A high-performing comprehensive learning algorithm for text classification without pre-labeled training set. Knowl. Inf. Syst. **29**, 727–738 (2011). https://doi.org/10.1007/s10115-011-0387-3
20. Losing, V., Hammer, B., Wersing, H.: Incremental on-line learning: a review and comparison of state of the art algorithms. Neurocomputing **275**, 1261–1274 (2018)
21. Ma, H., Fan, X., Chen, J.: An incremental Chinese text classification algorithm based on quick clustering. In: Proceedings 2008 International Symposiums on Information Processing (ISIP), Moscow, Russia, pp. 308–312 (2008)
22. Montiel, J., et al.: River: machine learning for streaming data in Python. J. Mach. Learn. Res. **22**, 1–8 (2020)
23. Moons, E., Moens, M.F.: Clinical report classification: continually learning from user feedback. In: Proceedings of the IEEE 34th Symposium on Computer-Based Medical Systems, CBMS, Virtual, pp. 455–460 (2021)
24. Oza, N.: Online bagging and boosting. In: Conference Proceedings - IEEE International Conference on Systems, Man, and Cybernetics, Waikoloa, USA, vol. 3, pp. 2340–2345 (2005)
25. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
26. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. J. Manag. Inf. Syst. **24**, 45–77 (2007)
27. Polikar, R., Upda, L., Upda, S.S., Honavar, V.: Learn++: an incremental learning algorithm for supervised neural networks. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **31**, 497–508 (2001)
28. Pöyhtäri, R.: Limits of hate speech and freedom of speech on moderated news websites in Finland, Sweden, The Netherlands and the UK. Ann. Ser. Hist. Sociol. **24**, 513–524 (2014)
29. Riehle, D.M., Niemann, M., Brunk, J., Assenmacher, D., Trautmann, H., Becker, J.: Building an integrated comment moderation system – towards a semi-automatic moderation tool. In: Meiselwitz, G. (ed.) HCII 2020. LNCS, vol. 12195, pp. 71–86. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49576-3_6

30. Shan, G., Xu, S., Yang, L., Jia, S., Xiang, Y.: Learn#: a novel incremental learning method for text classification. Expert Syst. Appl. **147**, 1–11 (2020)
31. Siefkes, C., Assis, F., Chhabra, S., Yerazunis, W.S.: Combining winnow and orthogonal sparse bigrams for incremental spam filtering. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) PKDD 2004. LNCS (LNAI), vol. 3202, pp. 410–421. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30116-5_38
32. Silva, R., Almeida, T., Yamakami, A.: MDLText: an efficient and lightweight text classifier. Knowl. Based Syst. **118**, 152–164 (2017)
33. Singh, B., Sun, Q., Koh, Y.S., Lee, J., Zhang, E.: Detecting protected health information with an incremental learning ensemble: a case study on New Zealand clinical text. In: Proceedings - 2020 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2020, Virtual, pp. 719–728 (2020)
34. Song, S., Qiao, X., Chen, P.: Hierarchical text classification incremental learning. In: Leung, C.S., Lee, M., Chan, J.H. (eds.) ICONIP 2009. LNCS, vol. 5863, pp. 247–258. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10677-4_28
35. Srilakshmi, V., Anuradha, K., Bindu, C.S.: Optimized deep belief network and entropy-based hybrid bounding model for incremental text categorization. Int. J. Web Inf. Syst. **16**, 347–368 (2020)
36. Tang, X.L., Han, M.: Ternary reversible extreme learning machines: the incremental tri-training method for semi-supervised classification. Knowl. Inf. Syst. **23**, 345–372 (2010). https://doi.org/10.1007/s10115-009-0220-4
37. Taninpong, P., Ngamsuriyaroj, S.: Tree-based text stream clustering with application to spam mail classification. Int. J. Data Min. Model. Manag. **10**, 353–370 (2018)
38. van Aken, B., Risch, J., Krestel, R., Löser, A.: Challenges for toxic comment classification: an in-depth error analysis. In: Proceedings of the Second Workshop on Abusive Language Online, ALW2, Brussels, Belgium, pp. 33–42 (2018)
39. Veloso, A., Meira Jr, W., Macambira, T., Guedes, D., Almeida, H.: Automatic moderation of comments in a large online journalistic environment. In: International AAAI Conference on Web and Social Media, ICWSM 2007, Boulder, USA, pp. 1–8 (2007)
40. vom Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., Cleven, A.: Reconstructing the giant: on the importance of rigour in documenting the literature search process. In: Proceedings of the 17th European Conference on Information Systems, ECIS 2009, Verona, Italy, pp. 1–12 (2009)
41. Wang, D., Al-Rubaie, A.: Incremental learning with partial-supervision based on hierarchical Dirichlet process and the application for document classification. Appl. Soft Comput. **33**, 250–262 (2015)
42. Wegier, W., Ksieniewicz, P.: Application of imbalanced data classification quality metrics as weighting methods of the ensemble data stream classification algorithms. Entropy **22**, 1–17 (2020)
43. Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., He, L.: A survey of human-in-the-loop for machine learning. Future Gener. Comput. Syst. **135**, 364–381 (2022)
44. Xia, R., Jiang, J., He, H.: Distantly supervised lifelong learning for large-scale social media sentiment analysis. IEEE Trans. Affect. Comput. **8**, 480–491 (2017)

45. Xie, Y., Willett, R.: Online logistic regression on manifolds. In: Proceedings - ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada, pp. 3367–3371 (2013)
46. Zhang, B., Su, J., Xu, X.: A class-incremental learning method for multi-class support vector machines in text classification. In: Proceedings - International Conference on Machine Learning and Cybernetics 2006, Dalian, China, pp. 2581–2585 (2006)