Towards More Reliable Text Classification on Edge Devices via a Human-in-the-Loop

Jakob Smedegaard Andersen and Olaf Zukunft
Hamburg University of Applied Sciences, Department of Computer Science, Hamburg, Germany

Keywords: Text Classification, Interactive Machine Learning, Time Efficiency.

Abstract: Reliably classifying huge amounts of textual data is a primary objective of many machine learning applications. However, state-of-the-art text classifiers require extensive computational resources, which limit their applicability in real-world scenarios. In order to improve the application of lightweight classifiers on edge devices, e.g. personal work stations, we adapt the Human-in-the-Loop paradigm to improve the accuracy of classifiers without re-training by manually validating and correcting parts of the classification outcome. This paper performs a series of experiments to empirically assess the performance of the uncertainty-based Human-in-the-Loop classification of nine lightweight machine learning classifiers on four real-world classification tasks using pre-trained SBERT encodings as text features. Since time efficiency is crucial for interactive machine learning pipelines, we further compare the training and inference time to enable rapid interactions. Our results indicate that lightweight classifiers with a human in the loop can reach strong accuracies, e.g. improving a classifier's F1-Score from 90.19 to 97% when 22.62% of a dataset is classified manually. In addition, we show that SBERT based classifiers are time efficient and can be re-trained in < 4 seconds using a Logistic Regression model.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

Maximizing the accuracy of automatic classifiers is a key goal of machine learning (LeCun et al., 2015). State-of-the-art text classifiers reach remarkable accuracy across many domains (Devlin et al., 2018; Sachan et al., 2019; Yang et al., 2019). Especially, transformer based classifiers such as BERT (Devlin et al., 2018) or XLNet (Yang et al., 2019) have demonstrated to be the best performing approaches in many text classification tasks. However, such strong classifiers are usually highly complex and consist of millions of parameters limiting their applicability on weak computational infrastructure, i.e. edge devices. The increasing energy consumption of state-of-the-art classifiers also creates environmental concerns (Strubell et al., 2019; Schwartz et al., 2020). If such strong models are not applicable, practitioners are excluded from their application and have to switch to less resource-intensive models that come at the cost of less reliable outcomes. Corazza et al. (Corazza et al., 2020) for example report a F1-Score of 82% for detecting hate speech in online forums using a traditional Word-Embedding-based classifier, which might not satisfy the demand of forum providers.

The need for reliable and trustful classifiers has recently risen in attention (Kendall and Gal, 2017; Holzinger, 2016; Sacha et al., 2015). Human-in-the-Loop machine learning (Holzinger, 2016) aims to overcome the obstacles of pure automatic classifiers by involving domain experts into the machine learning loop. Letting experts' correct classification outcomes during their daily work, e.g. Journalist-in-the-Loop (Karmakharm et al., 2019), is a promising way to increase the accuracy of classification outcomes without re-training (Pavlopoulos et al., 2017). In particular, uncertainty-based approaches have shown to be capable of detecting highly unreliable outcomes which are worth checking manually (He et al., 2020; Hendrycks and Gimpel, 2016).

The success of Human-in-the-Loop classification approaches do not only depend on a model's initial performance (e.g. F1-Score). An uncertainty-based semi-automated text classification approach requires accurate uncertainty estimations able to indicate misclassifications. Estimating reliable uncertainty scores in classification models is difficult, especially using Neural Networks (Hernández-Lobato and Adams, 2015). The question arises whether simpler models provide more accurate uncertainty esti-

Andersen, J. and Zukunft, O.
Towards More Reliable Text Classification on Edge Devices via a Human-in-the-Loop.
DOI: 10.5220/0010980600003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 2*, pages 636-646
ISBN: 978-989-758-547-0; ISSN: 2184-433X
Copyright © 2022 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

mations, which lead to higher F1-Scores when a certain number of the most uncertain instances are decided by a human rather than an automatic classifier. Furthermore, Human-in-the-Loop machine learning pipelines require rapid interaction cycles to e.g. retrain the model from time to time when additional human feedback is available (Amershi et al., 2014). Since the applicability of strong classifiers, i.e. BERT, is very limited on time dependent tasks and on weak computational infrastructure, we aim for a time efficient use of computational resources to enable rapid Human-in-the-Loop interactions. Especially, as classifiers benefit from being frequently re-trained when additional labeled data-instances are available (Arnt and Zilberstein, 2003; Haering et al., 2021).

In this paper, we empirically examine the quality of predicted probabilities and the macro F1-Score of nine commonly used and lightweight machine learning text classification models when a certain amount of the data is decided by humans instead of a machine. We perform several experiments on four publicly available benchmark datasets in the domain of text classification. As feature representations, we use semantic meaningful SBERT encodings (Reimers and Gurevych, 2019), which have shown to be efficiently computable while outperforming other recent pre-trained language models such as the Universal Sentence Encoder (Cer et al., 2018) or averaged GloVe embeddings (Pennington et al., 2014). To ensure rapid interaction cycles, we additionally compare the time needed to perform training and inference on a weak computational infrastructure. We focus on the following research questions:

- **RQ1:** How accurate do different lightweight classifiers estimate predicted probabilities?
- **RQ2:** Which lightweight classifier can capture the highest proportion of misclassifications via uncertainty-sampling, which after removal leads to the highest macro F1-Score?
- **RQ3:** How much of the most uncertain classification outcomes have to be manually annotated to reach a certain level of macro F1-Score?
- **RQ4:** How efficient are different classifiers regarding their training and inference time?

The remainder of the paper is structured as follows. Section 2 outlines the task of classification and its extension to the Human-in-the-Loop paradigm. Further, several classification models and techniques to estimate the uncertainty of individual classifications are described. In Section 3 we outline our research design and Section 4 reports our experimental results. Section 5 discusses our findings and Section 6 states related work. Finally, in Section 7 we draw our conclusions.

2 SEMI-AUTOMATIC TEXT CLASSIFICATION

We first outline the task of text classification (Rasmussen and Williams, 2006) and afterwards introduce the uncertainty-based semi-automatic classification of text.

The objective of classification is to predict class labels $y \in Y \subset \mathbb{N}$ for new data instances $x \in X \subset \mathbb{R}^n$ e.g. text encodings, which are related according to an unknown conditional class probability $p(y = c|x)$. Classification models aim to learn a function of the form $f : X \to Y$ or $f : X \to p(Y|X)$ from a set of labeled training examples $D \subset X \times Y$. Given an instance x, a probability based model f reports the label which receives the highest conditional class probability $y^* = f(x) = \arg\max_c p(y = c|x)$ over all classes c. Since not all classifiers are able to report probabilities, fractions of majority votes or scaling techniques are carried out to transform classification outcomes, e.g. distance functions, into probability distributions (Platt et al., 1999).

A common method to assess the uncertainty of classifiers is by calculating Shannon's Entropy (Shannon, 2001) of the conditional class probabilities, that is:

$$H(x) = -\sum_c P(y = c|x) \log_2 P(y = c|x) \qquad (1)$$

Shannon's Entropy estimates uncertainty as a lack of confidence in all class outcomes. The most uncertain instance u can be identified as $u = \arg\max_x H(x)$. A prediction $f(x)$ maximizes $H(x)$ when all class outcomes are equally certain, e.g. $p(0|x) = p(1|x) = 0.5$ in a binary classification task and minimizes $H(x)$ when either $p(0|x)$ or $p(1|x)$ are equal to 1. Sampling a subset of the most uncertain data instances is commonly referred as *uncertainty sampling* (Lewis and Gale, 1994).

Manually annotating text is a typical labeling task, where humans are asked to manually infer labels for some data instances. Since manual labeling is cost intensive and time-consuming, it makes sense to let humans only observe instances where a model provides unreliable and probable wrong outcomes. Human efforts should be focused on the most uncertain predictions to maximize the efficiency of their participation (Hendrycks and Gimpel, 2016). Especially, since uncertainty inherent in data instances cannot be explained by classifiers causing unreliable model behaviour (Kendall and Gal, 2017). In order to spend human efforts most rewarding and efficient, a classifier has to provide a decent ranking of misclassifications in regard to the reported uncertainty scores.

2.1 Machine Learning Classifiers

Several classification approaches are successfully used to classify text documents (Lai and Tsai, 2004; Liu and Chen, 2017; Stanik et al., 2019). However, previous work mostly focuses on pure automatic approaches and does not cover the objective of semi-automated classification. It remains unclear which model is most efficient when humans are involved in the classification process while saving time and computational costs. In our experiments, we consider the following lightweight machine learning models for classification (Bishop, 2006; Hastie et al., 2009) and outline how to obtain conditional class probabilities for the assessment of uncertainties.

These are: (1) a **Decision Tree (DT)** which estimates its conditional class probabilities by reporting pre-calculated fractions of correct class outcomes of each leaf node during training (Neville et al., 2003). (2) A **Random Forest (RF)** which reports conditional class probability as the fraction of trees voting for a certain class outcome. Further, we consider (3) a ***k*-Nearest Neighbour (*k*NN)** classifier where new documents are classified according to the k most similar documents of the training dataset. A majority vote is carried out to determine the final class outcome. Analogously to a Random Forest, we consider the fraction of votes as the conditional class probability. Naive Bayes classifiers are a family of conditional probability models which use Bayes rule to infer conditional class probabilities. Since SBERT encodings consist of continuous and also negative attributes, we apply (4) **Gaussian Native Bayes (GNB)**, a variation which makes the assumption that attributes of the feature vector are distributed according to a normal distribution. (5) A **Support Vector Machine (SVM)** classifies data by searching an optimal linear hyperplane which separates features with a maximal margin. The classification rule is based on which side of the hyperplane a data point occurs. In this paper, we apply Platt scaling (Platt et al., 1999) to obtain conditional probabilities from SVM outcomes. (6) **Logistic Regression (LR)** is a commonly used classifier which is capable of additionally predicting conditional class probabilities. A Logistic Regression model uses a sigmoid function to squeeze the output of a linear predictor function between 0 and 1 to represent class probabilities. (7) A **Multilayer Perceptron (MLP)** is a Neural Network-based classifier consisting of layers of interconnected computational units performing summation and thresholding. Similar to Logistic Regression, the class activation scores are normalized to obtain pseudo class probabilities.

Further, we consider a Bayesian approach to enable rich uncertainty interpretations (Gal and Ghahramani, 2016; Siddhant and Lipton, 2018). A Bayesian classifier replaces the models' weights ω with distributions, i.e. a Gaussian prior $\omega \sim N(0,1)$. Since the posterior probability $p(\omega|X,Y)$ cannot be evaluated analytically, several approximation techniques are used in practise (Blundell et al., 2015; Gal and Ghahramani, 2016). Sample-based approximations aim to fit the posterior $p(\omega|X,Y)$ with a simple to compute distribution $q^*(\omega)$. The conditional class probability can then be approximated by averaging T Monte Carlo samples over possible weights. In this paper, we consider (8) a **Bayesian** variation of the **Multilayer Perceptron (B-MLP)**. Bayesian models are of particular interest since they also capture uncertainty inherent in the models parameters (Kendall and Gal, 2017), while conventional deterministic classifiers do only assess uncertainties inherent in the data. A holistic uncertainty assessment of Bayesian classifiers can be carried out by calculating Shannon's Entropy (Eq. 1) on the mean conditional class probabilities obtained by averaging the results of multiple model runs.

3 BENCHMARK DESIGN

To answer our research questions, we first assess the quality of predicted probabilities provided by the classifiers outlined in Section 2.1. We measure the Brier score (Brier et al., 1950) of each classifier applied to each dataset. The Brier score is a proper scoring rule to measure the accuracy of predicted probabilities. It is calculated as the squared error of the predicted probabilities and true class outcomes, that is:

$$BS = |Y|^{-1} \sum_{y \in Y} \sum_{c \in C} \big(p(y=c|x) - I(\hat{y}=c)\big)^2 \qquad (2)$$

where $I(\hat{y}=c) = 1$ if the true class of x represented by $\hat{y}$ is equal to c else 0. The lower the Brier score the better are the conditional class probabilities calibrated. Calibrated class probabilities are desired to reliably assess the true probability of predictions leading to more accurate quantification of predictive uncertainties. Second, we compare the macro F1-Score of these classifiers when a certain amount of the most uncertain data instances, in our case 0, 10, 20, and 30%, are removed from the test dataset. We use the macro-average since some used datasets are highly imbalanced and we aim to treat all classes equally important. Third, we estimate the amount of manual effort a human has to spend, i.e. by correcting classifier outcomes, in order to reach a specific target macro F1-Score. We measure human efforts in terms of instances a human has to decide manually. In our experiments, we simulate human annotations by selecting the ground truth label for each annotation request, a common approach when evaluating interactive machine learning approaches (Siddhant and Lipton, 2018). Since human annotations are known to be noisy, we simulate three different human noise levels. We assign a randomly selected class label with a probability of 0, 5, or 10% respectively to each annotation request instead of the ground truth label. The macro F1-Score of a combined human-automatic classifier is calculated based on the unified sets of manually corrected and automatically inferred labels. Fourth, we measure the training and inference time to assess the computational efficiency of different text classifiers. All experiments are run on an Intel® Xeon® Gold 5115 CPU @ 2.40GHz using 1 core and 4 GB of memory. All reported measurements are the mean of five stratified cross fold data sets with a 50% training-test split. In the following, we use the shortened term ”F1-Score” to refer to the macro F1-Score.

3.1 Datasets

We consider four different publicly available real-world datasets covering heterogeneous classification tasks in our experiments. Key statistics of the datasets are summarized in Table 1.

Table 1: Statistics of the datasets including size, number of classes and its distribution as well as the mean number and standard deviation of words per text instance.

Dataset	Size	$\|C\|$	Class Distribution	#Words ($\mu \pm \sigma$)
IMDB	50000	2	25000:25000	234 ± 173
App Store	5752	3	3472:1286:994	24 ± 29
Reuters	8614	8	3930:2319:527:499: 458:425:290:166	117 ± 129
Hate Speech	24783	2	19190:5593	15 ± 7

First, we use the **IMDB** dataset (Maas et al., 2011), a commonly used benchmark for sentiment analysis. The dataset consists of highly polarized film reviews, which are either labeled as positive or negative. Second, we consider a corpus of app reviews from the domain of participatory requirements engineering. The **App Store** dataset (Maalej et al., 2016) contains user reviews, which are manually labeled as feature request, bug report or praise. Third, we take a dataset collected from the **Reuters** financial newswire service (Lewis et al., 2004). Documents are labeled regarding their topic. In our experiments, we use a subset of the 8 most frequent topics with unambiguous labels. Lastly, we consider the **Hate Speech** dataset (Davidson et al., 2017) which comes with the task of identifying toxic tweets (hate speech or offensive language). For each dataset, we apply a stratified split of 50% for training and the remaining for testing.

3.2 Document Features

Text documents consist of sequences of characters and have to be transformed to a vector space before passing them to machine learning models. As the feature representation for text documents, we consider Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) encodings. SBERT is a modification of the *pre-trained Bidirectional Encoder Representations from Transformers* (BERT) (Devlin et al., 2018) model and provides semantically meaningful encodings for unlabeled text documents without the need of domain specific pre-training and fine-tuning. Studies show that SBERT encodings outperform out-of-the-box BERT encoding in several text classification tasks (Reimers and Gurevych, 2019). Furthermore, SBERT encodings are resource efficient to compute.

SBERT, like other BERT variations, encodes a document d as an n-dimensional vector of continuous attributes $x = (a_i, ..., a_n)$. We employ the pre-trained *bert-base-nli-mean-tokens*[1] model, which computes encodings of length $n = 768$. Since BERT uses subword tokenization, BERT encodings are limited to 512 tokens, which are around 300 to 400 words. Therefore, we use the mean SBERT encoding of each individual sentence for the IMDB dataset to avoid truncation. In a preliminary investigation, we found that mean SBERT encodings have a positive effect only on the F1-Score of the IMDB classifiers.

3.3 Classifier Implementations

For the majority of classifiers, we rely on the default implementation provided by the Scikit-learn library[2] since these are commonly used for machine learning experiments. For the Random Forest classifier, we use $T = 100$ Decision Trees and set $k = 25$ for the kNN classifier. The structure of the MLP takes the shape [768, 500, 500, C]. We do not perform hyper-parameter tuning. Since Scikit-learn does not offer a Bayesian-MLP, we employ Tensorflow[3] version 2.4.1 for the implementation (B-MLP*). We approximate the posterior using a dropout variational distribution

[1] https://huggingface.co/sentence-transformers/bert-base-nli-mean-tokens

[2] https://scikit-learn.org/stable/index.html

[3] https://www.tensorflow.org/

(Gal and Ghahramani, 2016) and apply $T = 100$ forward passes. Since an identical recreation of Scikit-learn's MLP in Tensorflow is difficult, we additionally develop a (9) conventional non-Bayesian-MLP (MLP*) to compare the impact of Bayesian modelling. In comparison to Scikit-learn's MLP implementation, our MLP model applies dropout similarly to the Bayesian MLP, but only during training. Further, we select 10% of the training data as validation data for all MLP implementations to enable early stopping. The source code of all models, parameters and experiments are publicly available.[4]

4 RESULTS

In this section, we present the results of our experiments and answer the four research questions.

4.1 Quality of Predicted Probabilities (RQ1)

The Brier scores of our experiments covering nine classifiers and four datasets are shown in Table 2. The lower the Brier score, the more accurate the predicted conditional class probabilities. A Brier score of 0 indicates a perfectly accurate classifier, whereas a score of 1 indicates a highly inaccurate one.

Table 2: Brier scores of different classifiers and datasets measuring the accuracy of the predicted conditional class probabilities. The lower the Brier score the better are the conditional class probabilities calibrated.

Classifier	**IMDB**	**App Store**	**Reuters**	**Hate Speech**	AVG
DT	0.4259	0.5443	0.4869	0.4659	0.4806
RF	0.2059	0.2629	0.2247	0.2221	0.2289
*k*NN	0.2076	0.2629	0.1794	0.2302	0.2200
GNB	0.3174	0.4615	0.3919	0.4962	0.4168
SVM	**0.1460**	0.2435	0.1043	0.1934	0.1718
LR	**0.1464**	0.2398	0.0954	0.1919	0.1684
MLP	0.1592	0.2098	0.1029	0.2045	0.1691
MLP*	0.1542	**0.2060**	**0.0939**	**0.1879**	**0.1605**
B-MLP*	0.1513	**0.2036**	**0.0920**	**0.1819**	**0.1572**

The table reveals huge differences between the classifiers regarding their quality of predicted probabilities. A DT and GNB provide the worst calibrated probabilities with an average Brier score of 0.48 and 0.42 respectively. RF and *k*NN reach nearly equally calibrated probabilities, with an average of > 0.22. SVM, LR and MLP as well as its variations receive the best Brier scores. LR followed by SVM obtains the best scores on the IMDB dataset, whereas MLP* and B-MLP* receive the best scores on the App Store, Reuters and Hate-Speech Dataset. Overall, a Bayesian MLP (B-MLP*) followed by a dropout based MLP (MLP*) provide the most accurate probabilities with an average Brier score of ~ 0.16.

4.2 Classifier Performance under Stepwise Removal of Uncertain Instances (RQ2)

Table 3 lists the F1-Scores of the classifiers applied to each of the datasets. The columns represent the F1-Scores which are obtained when a certain number (0, 10, 20 and 30%) of the most uncertain instances are removed from the test set. Each cell additionally states the relative improvement of F1-Score in relation to the previous removal ratio. For example, a SVM on the IMDB dataset reaches a F1-Score of 90.24% on the entire test dataset. If 10% of the most uncertain instances are removed, the F1-Score increases to 93.60% which is a relative improvement of 3.72%.

Our experiment shows that on the whole test dataset, i.e. using a removal ratio of 0%, a DT and GNB provide the worst F1-Score followed by the *k*NN and RF classifiers. LR, SVM and the MLP reach the highest initial F1-Scores. Scikit-learn's MLP implementation provides a worse performance compared to our Tensorflow implementation. In our setting, Bayesian modelling (B-MLP*) shows no improvement in F1-Score compared to a deterministic MLP. Overall, SVM and LR classifiers perform best on the IMDB and Reuters datasets, whereas a deterministic MLP with dropout performs best on the App Store and Hate Speech datasets.

When a certain number of the most uncertain instances are removed from the test dataset, the F1-Score generally increases. Only the uncertainty estimates of a Decision Tree classifier are unable to detect misclassifications since the F1-Score is not improving when removing highly uncertain instances. Further, the relative F1-Score improvements decrease with larger removal ratios indicating a decreasing human efficiency when large amounts of removed data are passed to human annotators. Overall, classifiers with high initial F1-Scores also reach the best F1-Scores after removing uncertain instances from the test dataset. Only the initially better performing LR gets outperformed by the MLP* when removing uncertain instances on the Reuters dataset.

4.3 Semi-automated Classification Performance (RQ3)

Table 4 shows how much of the most uncertain instances from the unseen test set have to be classified

[4] https://github.com/jsandersen/MRTviaHIL

Table 3: F1-Scores of different classifiers when a certain number of the most uncertain predictions were removed from the test dataset.

	IMDB				App Store				Reuters				Hate Speech			
Classifier	0%	10%	20%	30%	0%	10%	20%	30%	0%	10%	20%	30%	0%	10%	20%	30%
DT	78.70	78.75 +0.05	78.77 +0.03	78.77 -0.00	64.80	65.02 +0.33	64.94 -0.11	65.00 +0.09	58.11	58.30 +0.34	58.15 -0.27	58.30 +0.27	66.99	66.95 -0.01	67.03 +0.11	67.05 +0.02
RF	86.37	89.69 +3.84	92.33 +2.94	94.34 +2.18	76.02	79.41 +4.47	81.94 +3.18	82.61 +0.82	75.47	81.41 +7.87	85.42 +4.93	89.80 +5.13	73.47	74.07 +0.82	73.23 -1.14	70.91 -3.16
*k*NN	85.54	88.88 +3.91	91.61 +3.07	93.71 +2.30	76.65	80.70 +5.28	83.85 +3.90	86.69 +3.38	76.46	81.36 +6.41	83.66 +2.82	78.27 -6.43	68.48	67.54 -1.37	67.06 -0.72	67.26 +0.30
GNB	83.95	87.25 +3.93	90.16 +3.33	92.55 +2.65	72.86	76.26 +4.66	79.51 +4.26	82.66 +3.96	69.06	73.98 +7.12	80.37 +8.63	87.62 +9.03	69.95	72.82 +4.11	75.76 +4.03	78.95 +4.21
SVM	**90.24**	**93.60** +3.72	**95.74** +2.29	**97.04** +1.35	76.84	80.83 +5.19	84.84 +4.96	88.45 +4.26	88.38	93.98 +6.34	96.77 +2.97	98.44 +1.73	79.51	82.16 +3.33	84.42 +2.76	86.15 +2.05
LR	**90.19**	**93.53** +3.70	**95.71** +2.33	97.02 +2.37	78.89	83.33 +5.62	87.59 +5.12	91.19 +4.11	**88.75**	94.98 +7.02	97.56 +2.72	98.42 +0.88	80.17	83.54 +4.20	86.44 +3.47	88.42 +2.29
MLP	89.55	92.96 +3.81	95.32 +2.53	96.85 +1.61	80.09	84.65 +5.69	88.55 +4.62	92.10 +4.00	87.77	94.32 +7.46	97.57 +3.44	98.64 +0.11	80.06	83.53 +4.33	86.54 +3.60	88.82 +2.64
MLP*	89.94	93.36 +3.80	95.59 +2.39	96.96 +1.43	**81.08**	**86.12** +6.21	**89.89** +4.38	**92.97** +3.42	88.68	**95.24** +7.39	**97.87** +2.76	**99.15** +1.31	**81.12**	**84.77** +4.50	**88.24** +4.10	**90.68** +2.77
B-MLP*	89.95	93.39 +3.83	95.66 +2.43	**97.08** +1.48	**80.99**	**86.02** +6.21	**89.74** +4.32	**92.97** +3.61	**88.76**	**95.14** +7.19	**98.09** +3.10	**99.24** +1.17	**81.12**	**84.80** +4.54	**87.99** +3.76	**90.42** +2.76

Table 4: Performance of Human-in-the-Loop classifiers. Each cell shows how much of the most uncertain classification outcomes in percent have to be manually annotated in order to reach a certain F1-Score given a specific noise level of the human annotator. Unobtainable F1-Scores due to a high number of committed human errors are marked as ”-”.

	IMDB				App Store				Reuters				Hate Speech				
Classifier	.93	.95	.97	.99	.89	.91	.93	.95	.93	.95	.97	.99	.89	.91	.93	.95	
DT	67.18	76.55	85.94	95.12	68.46	74.03	80.16	86.01	83.49	87.76	92.76	97.42	66.58	72.76	78.56	84.68	0% human noise
RF	17.00	24.84	35.94	57.04	22.53	28.41	33.82	40.43	14.21	18.34	24.40	38.91	25.67	30.76	37.44	48.14	
*k*NN	18.87	26.94	38.44	60.21	21.96	26.35	31.41	37.48	15.37	18.50	23.31	34.50	23.31	26.97	31.75	38.55	
GNB	22.87	30.97	42.78	64.06	29.79	34.01	38.61	44.06	28.47	32.39	37.87	51.78	41.68	48.01	54.70	62.52	
SVM	**6.33**	**12.71**	**22.46**	**44.42**	18.43	22.34	27.28	32.42	5.11	8.94	13.47	24.75	15.03	19.60	24.95	32.46	
LR	**6.60**	**12.88**	**22.62**	**43.95**	15.21	19.24	23.56	29.22	**3.90**	**7.24**	12.19	23.91	14.17	18.56	23.72	30.88	
MLP	8.00	14.60	24.18	46.25	13.49	17.68	21.62	27.00	5.32	8.31	12.91	22.31	13.83	18.24	23.55	31.25	
MLP*	7.08	13.40	23.19	45.91	**11.42**	**15.39**	**19.77**	**25.34**	**4.16**	**6.99**	**11.75**	**20.64**	**12.19**	**16.33**	**21.26**	**28.74**	
B-MLP*	7.02	13.17	22.77	44.16	**11.67**	**15.68**	**20.06**	**25.56**	4.23	7.43	**11.52**	**19.80**	**12.23**	**16.28**	**21.06**	**27.95**	
DT	75.81	86.38	97.79	-	7.75	84.07	90.30	96.53	-	-	-	-	73.77	80.46	87.39	94.54	5% human noise
RF	18.34	28.06	44.64	-	26.91	33.54	40.93	51.50	18.11	27.58	-	-	23.48	28.30	35.02	46.43	
*k*NN	20.73	30.62	48.88	-	24.31	29.22	36.23	43.37	18.20	24.96	-	-	25.06	29.64	36.20	48.12	
GNB	25.23	35.41	55.46	-	33.01	37.92	43.80	53.35	33.78	-	-	-	46.49	54.17	63.57	76.75	
SVM	**6.77**	**14.05**	28.26	-	20.18	25.53	31.04	40.08	5.94	10.68	23.17	-	16.50	21.76	28.70	41.51	
LR	**7.06**	**14.18**	**27.76**	-	16.49	20.96	26.44	33.95	**4.27**	**8.80**	19.13	-	15.41	20.34	27.28	38.63	
MLP	8.55	16.23	29.50	-	14.52	19.15	23.84	31.48	6.11	10.47	18.53	-	15.03	20.34	27.33	39.24	
MLP*	7.54	14.79	28.60	-	**12.48**	**16.71**	**22.59**	**29.32**	4.99	9.01	**18.02**	-	**13.15**	**17.87**	**24.35**	**35.26**	
B-MLP*	7.51	14.56	**27.40**	-	**12.77**	**17.21**	**22.68**	**29.76**	**4.97**	**8.61**	**16.32**	-	**13.14**	**17.88**	**24.09**	**34.36**	
DT	88.38	-	-	-	87.58	95.03	-	-	-	-	-	-	83.17	91.45	99.62	-	10% human noise
RF	20.64	33.76	-	-	29.79	37.17	46.46	-	-	-	-	-	25.70	32.11	42.97	-	
*k*NN	23.60	37.63	-	-	27.32	34.39	42.77	-	-	-	-	-	27.36	33.37	44.30	-	
GNB	28.74	44.38	-	-	36.42	42.99	53.82	-	-	-	-	-	51.97	63.28	82.17	-	
SVM	**7.34**	**16.02**	-	-	22.40	28.60	37.67	-	7.53	-	-	-	18.13	24.54	35.16	-	
LR	**7.51**	**16.00**	-	-	18.18	23.37	30.48	-	**5.32**	13.49	-	-	16.71	22.67	32.15	-	
MLP	9.17	18.06	-	-	16.46	21.96	30.16	-	7.22	14.60	-	-	16.20	22.43	32.46	-	
MLP*	8.19	16.61	-	-	**14.05**	**19.46**	**27.00**	-	**5.71**	**13.03**	-	-	**14.34**	**19.83**	**29.56**	-	
B-MLP*	8.10	16.50	-	-	**14.49**	**19.52**	**26.47**	**47.47**	5.99	**12.03**	-	-	**14.39**	**19.78**	**28.04**	-	

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

manually in order to raise the semi-automatic classification outcomes to a certain F1-Score. Each sub-table represents a different human noise level as introduced in Section 3. For example, on the IMDB dataset 12.71% of the most uncertain prediction have to be manually corrected to improve the model’s F1-Score (from initial 90.24%) to 95% using a SVM. The table indicates that models with a high initial F1-Score require less manual efforts to raise the F1-Score to a certain target level. Overall, models with lower initial F1-Score scores do rarely overtake better performing classifiers in regard to the final F1-Score when human annotators are in the loop.

Involving humans with higher noise levels requires more manual efforts to reach a specific F1-Score, which is straightforward, since more misclassifications are committed. However, our results indicate that Human-in-the-Loop text classification can reach a higher F1-Score compared to its pure machine and human parts on their own. For example, an LR classifier with an initial F1-Score of 90.19% on the IMDB dataset and a 10% noisy human can reach an F1-Score of >95% (max. 96.77%) when >18.6% of the dataset is classified manually.

Our results reveal that lightweight classifiers can reach strong accuracies with a human in the loop even if the annotator commits several errors. Using the best performing classifier, an F1-Score of 95% (+4.81), 91% (+9.92), 95%(+6.24), 91% (+9.88) can be reached with a manual effort of 16.02, 19.46, 12.03, and 19.78% respectively considering a human noise level of 10%. Compared to a 100% accurate human annotator, this is an increase in manual efforts of 24.37%, 26.45%, 61.91%, and 21.50% respectively. Our results also demonstrate that top F1-Scores, e.g. 95-99%, are not reachable in all Human-in-the-Loop settings. If the human annotations are too noisy, the F1-Score is starting to decrease after a certain amount of human assistance. As less uncertain predictions are more often annotated by humans during larger workloads, the accuracy decreases. This phenomenon occurs because noisy humans incorrectly annotate instances that the machine would have correctly decided by itself reducing the overall accuracy of the semi-automated classification outcomes.

4.4 Runtime Comparison (RQ4)

The training time of the classifiers is illustrated in Figure 1. The x-axis lists the classifiers and the y-axis represents the average training-time measured in seconds in log-scale. The *k*NN classifier is not listed since it is a memory-based learning algorithm that requires no training. A GNB has the fastest training time taking an average of 0.2 Seconds for 25000 instances (IMDB). The LR has the second-shortest training time being 3.7 seconds. A DT, MLP, and RF perform much slower with around 64.28, 73.42, and 91.14 seconds respectively. The dropout based-MLP* implementation took 138.66 seconds which is nearly double the time of MLP. MLP* and B-MLP* require the same time for training as they share the same training procedure. The SVM is the only classifier which shows an exponential growth in training time in regard to the size of the training data ranging from 12.81 seconds for the App Store (size 2876) to 27.18 minutes for the IMDB (size 25000) dataset.

Figure 2 depicts the time needed to perform inference. The DT, RF, GNB, LR and the MLP classifiers take less than one second to infer the labels for 25000 instances (IMDB). The MLP* implementation is slightly slower with an average of 1.33 seconds on the same dataset. The *k*NN classifier is much slower with an inference time of 48.87 seconds (IMDB).

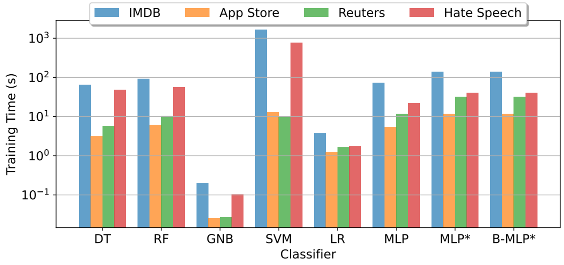

Figure 1: Total training time of the experiments.

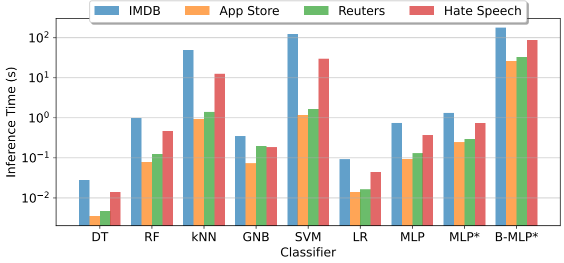

Figure 2: Total inference time of the experiments.

The inference time of a SVM and *k*NN grows exponentially in regarding the number of predicted texts. The SVM needs 1.15 seconds for the App Store and 123.84 seconds for the IMDB dataset. The *k*NN classifier requires 0.91 seconds for the App Store and 48.87 seconds for the IMDB dataset. Sampling-based Bayesian approximations require more time for inference since multiple forward passes have to be carried out to approximate the condition class probability. By performing 100 forward passes, a Bayesian MLP takes 176.41 seconds (IMDB).

5 DISCUSSION

Our results indicate that manually annotating parts of the outcomes of lightweight text classifiers using SBERT can lead to substantial improvements with a manageable manual effort. A Human-in-the-Loop approach can increase the F1-Score to at least 95% on all datasets by manually validating less than 28% of the data. Our findings are especially important for domains in which annotation tasks are still carried out purely manually in case applicable automatic classifiers do not provide a required F1-Score out of the box. While solving a classification task by hand can be an alternative solution, many domains are confronted with an overwhelming amount of data exceeding human capabilities. Human-in-the-Loop classification aims to overcome the accuracy limitations of pure automatic classification with the cost of human involvement. Human efforts are usually wasted when used to perform tasks a cheap artificial model can

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

perform equally. The effectiveness of Human-in-the-Loop emerges by focusing human efforts on instances where an automatic classifier mostly fails. Overall, the applicability of Human-in-the-Loop text classification depends on whether human efforts are affordable during a classifiers operational use and whether more reliable classification outcomes are needed.

In our experiments, we observed large variations between different classifiers regarding their suitability for Human-in-the-Loop text classification. We show that the quality of uncertainty estimates of simple models such as Decision Trees, Gaussian Naive Bayes, Random Forest and *k*-Nearest Neighbour classifier are fare less accurate compared to an Logistic Regression model, Support Vector Machine or Multilayer Perceptron, limiting their suitability for uncertainty assessments. We also show, that these simple models do not provide any advantages compared to a Logistic Regression model in Human-in-the-Loop classification settings, since they reach a lower F1-Score or require much more computational costs.

The Multilayer Perceptron and a Support Vector Machine have shown to provide similar or even stronger performance scores compared to Logistic Regression, but require much more computational resources. Although no classifier consistently outperforms the others in our experiments, a Multilayer Perceptron with dropout reaches on average the highest performance across all datasets. Overall, classifiers which reach a higher F1-Score pure automatically also require less effort to reach an even higher F1-Score when placing a Human-in-the-Loop. Further, our results indicate that Bayesian modelling i.e. Monte Carlo Dropout (Gal and Ghahramani, 2016) does slightly improve the quality of uncertainty estimated, but has not a great impact on the resulting F1-Score of Human-in-the-Loop classification using a small MLP and SBERT encodings as text features. Since classifiers with the highest F1-Score also provide the best Brier scores it is not necessary using one classifier to estimate uncertainties and another classifier to provide the classification decision.

To enable rapid or even real-time Human-in-the-Loop processing, a Logistic Regression model is the fastest approach in inference and training while providing a decent initial as well as human in the loop performance. It only requires < 4 seconds for training and inference on 25000 data instances. A Support Vector Machine is less applicable due to its comparable slow training time and it does not scale well to large datasets. A dropout-based Multilayer Perceptron has shown to provide on average a better performance, but comes with higher computational efforts of a total of < 139 seconds for training and inference.

We also demonstrate that humans and machines can work together to achieve even greater accuracy than their individual parts. Highly uncertain instances are most likely to be misclassified automatically, and even noisy human annotators have the potential to provide more accurate labels. By simulating different kinds of human behaviour, we demonstrate the performance of Human-in-the-Loop text classification across multiple domains and human performances. Practitioners in the loop have to judge about their own behaviour to draw insides about how much effort is worth to spend in the loop. Our study provides guidelines to support practitioners in choosing the most efficient classifier when strong classifiers are not applicable because of high computational costs, and humans are willing to label some part of the classification results.

SBERT-based classifiers clearly underperform state-of-the-art text classifiers such as BERT. For example, a fine-tuned BERT model has shown to reach a F1-Score of 93.46% (Sanh et al., 2019) on the IMDB dataset. However, BERT requires huge computational resources and takes multiple hours to days to be fine-tuned on a CPU. In comparison, a Logistic Regression model employed on the same task using SBERT encodings takes a few seconds on a CPU for training and inference to reach an F1-Score of 90.2%, which is a higher score than recent Word-Embedding based approaches (He et al., 2020; Hendrycks and Gimpel, 2016). As shown by our results, manually annotating 12.70% of the data leads to an F1-Score of 95%, which outperforms BERT's performance. Thus, SBERT-based classifiers with a human in the loop are an alternative or even a substitute of BERT if training and inference have to be carried out efficiently and human efforts are arrangeable.

This paper investigates Human-in-the-Loop classification with on-device training and inference. Alternatively to our approach, practitioners can also train classifiers on more powerful machines if available and afterwards transfer the parameters to weak edge devices in order to maintain applicability and save computational costs. However, the inference of state-of-the-art classifiers such as BERT is still very slow on weak computational infrastructure e.g. edge devices due to their high resource consumption. With our research, we follow a more personalized approach, where practitioners are capable to reach strong classification performances on weak infrastructure. Hereby, we aim to support practitioners to rapidly extract desired information from their textual data using classification on their own work stations.

6 RELATED WORK

The rising demand for interactive real-time processing (Amershi et al., 2014; Dudley and Kristensson, 2018; Zanzotto, 2019) and resource efficient machine learning (Al-Jarrah et al., 2015; Zhang et al., 2018) upraise the need of additional evaluation perspectives. In contrast to traditional performance driven benchmark studies (Chauhan and Singh, 2018; Luu et al., 2020; Stanik et al., 2019), we focus on the accuracy and time efficiency of semi-automatic and lightweight text classifiers.

Rattigan et al. (Rattigan et al., 2007) initially investigate the objective of maximizing the accuracy of classifiers while limiting human efforts. On the one side, related work in the domain of text classification focus on approaches based on estimating thresholds of conditional class probabilities to separate unreliable class outcomes similar to our work. Pavlopoulos et al. (Pavlopoulos et al., 2017) suggest identifying upper and lower class probability thresholds to determine a fixed sized slice of data instances which maximizes the accuracy of a model when manually annotated. In contrast, our approach is based on uncertainty estimates and does not require solving an optimization task. He et al. (He et al., 2020) seek to improve the quality of uncertainty estimates to enable a more efficient annotation process. However, their approach is only applicable to Deep Neural Networks while ours can be applied to any classifier. On the other side, training an additional reject function is another common approach to delegate unreliable instances to humans (Cortes et al., 2016; Geifman and El-Yaniv, 2017). An abstain option can either be modelled as an additional class outcome or is achieved by training a separate classifier leading to additional computational costs and effort.

Manually annotating classifier outcomes can also be considered as a special case of *Algorithm-in-the-Loop Decision Making* (Green and Chen, 2019), where humans rather than algorithms are making the final classification decision. In contrast, our approach seeks to only involve humans when the model is unable to provide reliable classification outcomes. Another closely related field to Human-in-the-Loop classification is *Active Learning* (Lewis and Gale, 1994). Active learning seeks to minimize human efforts in the creation of training data to reach highly accurate classifiers. In Active Learning a machine actively queries labels from human annotators to improve a model's learning behaviour. Similar to Human-in-the-Loop classification, both approaches can utilize uncertainty sampling to guide human involvement. In contrast, Active Learning is applied during the training step of the initial model while our approach seeks to further raise the accuracy of an already trained model during its operational use. Human-in-the-Loop classification aims to exceed the maximum achievable accuracy (Baram et al., 2004) of a pre-trained classifier with the cost of human participation during the classification process.

7 CONCLUSION

In this paper, we conduct several experiments to identify best performing and time efficient semi-automatic text classifiers using SBERT encodings. We investigate the quality of uncertainty estimates as well as the F1-Score of lightweight text classifiers, when a certain amount of the most uncertain classification outcomes is manually validated and corrected. Further, we assess the time needed to perform training and inference to assess a model's applicability on edge devices as well as enabling rapid human interaction cycles. Our study consists of nine different classification models and four real-world text classification tasks. Our results indicate that the initially best performing automatic classifiers (without human involvement) require less manual effort to achieve a strong F1-Score compared to initially weaker classifiers. We also show that SBERT-based classifiers are time efficient and only take seconds to a few minutes to be trained, enabling rapid interactive machine learning cycles. Our research provides guidelines for semi-automatic text classification approaches when conventional state-of-the-art classifiers are not applicable due to time constraints. As further work, we plan to perform more user experiments and investigate the acceptance of using Human-in-the-Loop text classification in real-world domains.

REFERENCES

Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Karagiannidis, G. K., and Taha, K. (2015). Efficient machine learning for big data: A review. *Big Data Research*, 2(3):87–93.

Amershi, S., Cakmak, M., Knox, W. B., and Kulesza, T. (2014). Power to the people: The role of humans in interactive machine learning. *Ai Magazine*, 35(4):105–120.

Arnt, A. and Zilberstein, S. (2003). Learning to perform moderation in online forums. In *Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003)*, pages 637–641. IEEE.

Baram, Y., Yaniv, R. E., and Luz, K. (2004). Online choice of active learning algorithms. *Journal of Machine Learning Research*, 5(Mar):255–291.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR.

Brier, G. W. et al. (1950). Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al. (2018). Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Chauhan, N. K. and Singh, K. (2018). A review on conventional machine learning vs deep learning. In *International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 347–352. IEEE.

Corazza, M., Menini, S., Cabrio, E., Tonelli, S., and Villata, S. (2020). A multilingual evaluation for online hate speech detection. *ACM Transactions on Internet Technology (TOIT)*, 20(2):1–22.

Cortes, C., DeSalvo, G., and Mohri, M. (2016). Learning with rejection. In *International Conference on Algorithmic Learning Theory*, pages 67–82. Springer.

Davidson, T., Warmsley, D., Macy, M., and Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dudley, J. J. and Kristensson, P. O. (2018). A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):1–37.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Geifman, Y. and El-Yaniv, R. (2017). Selective classification for deep neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4885–4894.

Green, B. and Chen, Y. (2019). Disparate interactions: An algorithm-in-the-loop analysis of fairness in risk assessments. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 90–99.

Haering, M., Andersen, J. S., Biemann, C., Loosen, W., Milde, B., Pietz, T., Stoecker, C., Wiedemann, G., Zukunft, O., and Maalej, W. (2021). Forum 4.0: An open-source user comment analysis framework. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 63–70.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

He, J., Zhang, X., Lei, S., Chen, Z., Chen, F., Alhamadani, A., Xiao, B., and Lu, C. (2020). Towards more accurate uncertainty estimation in text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8362–8372.

Hendrycks, D. and Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Hernández-Lobato, J. M. and Adams, R. (2015). Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869. PMLR.

Holzinger, A. (2016). Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131.

Karmakharm, T., Aletras, N., and Bontcheva, K. (2019). Journalist-in-the-loop: Continuous learning as a service for rumour analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 115–120.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5580–5590.

Lai, C.-C. and Tsai, M.-C. (2004). An empirical performance comparison of machine learning methods for spam e-mail categorization. In *Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*, pages 44–48. IEEE.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Liu, Z. and Chen, H. (2017). A predictive performance comparison of machine learning models for judicial cases. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6. IEEE.

Luu, S. T., Nguyen, H. P., Van Nguyen, K., and Nguyen, N. L.-T. (2020). Comparison between traditional machine learning models and neural network models for vietnamese hate speech detection. In *International Conference on Computing and Communication Technologies (RIVF)*, pages 1–6. IEEE.

Maalej, W., Kurtanović, Z., Nabil, H., and Stanik, C. (2016). On the automatic classification of app reviews. *Requirements Engineering*, 21(3):311–331.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Neville, J., Jensen, D., Friedland, L., and Hay, M. (2003). Learning relational probability trees. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 625–630.

Pavlopoulos, J., Malakasiotis, P., and Androutsopoulos, I. (2017). Deep learning for user comment moderation. In *Proceedings of the First Workshop on Abusive Language Online*, pages 25–35.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Rattigan, M. J., Maier, M., and Jensen, D. (2007). Exploiting network structure for active inference in collective classification. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, pages 429–434. IEEE.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sacha, D., Senaratne, H., Kwon, B. C., Ellis, G., and Keim, D. A. (2015). The role of uncertainty, awareness, and trust in visual analytics. *IEEE transactions on visualization and computer graphics*, 22(1):240–249.

Sachan, D. S., Zaheer, M., and Salakhutdinov, R. (2019). Revisiting lstm networks for semi-supervised text classification via mixed objective function. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6940–6948.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schwartz, R., Dodge, J., Smith, N. A., and Etzioni, O. (2020). Green ai. *Communications of the ACM*, 63(12):54–63.

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55.

Siddhant, A. and Lipton, Z. C. (2018). Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909.

Stanik, C., Haering, M., and Maalej, W. (2019). Classifying multilingual user feedback using traditional machine learning and deep learning. In *27th International Requirements Engineering Conference Workshops (REW)*, pages 220–226. IEEE.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zanzotto, F. M. (2019). Human-in-the-loop artificial intelligence. *Journal of Artificial Intelligence Research*, 64:243–252.

Zhang, Q., Yang, L. T., Chen, Z., and Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, 42:146–157.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS