SN Computer Science (2021) 2:178
https://doi.org/10.1007/s42979-021-00480-4

ORIGINAL RESEARCH

Simple Baseline Machine Learning Text Classifiers for Small Datasets

Martin Riekert[1] · **Matthias Riekert**[2] · **Achim Klein**[1]

Received: 30 September 2020 / Accepted: 20 January 2021 / Published online: 30 March 2021

Abstract
Text classification is important to better understand online media. A major problem for creating accurate text classifiers using machine learning is small training sets due to the cost of annotating them. On this basis, we investigated how SVM and NBSVM text classifiers should be designed to achieve high accuracy and how the training sets should be sized to efficiently use annotation labor. We used a four-way repeated-measures full-factorial design of 32 design factor combinations. For each design factor combination 22 training set sizes were examined. These training sets were subsets of seven public text datasets. We study the statistical variance of accuracy estimates by randomly drawing new training sets, resulting in accuracy estimates for 98,560 different experimental runs. Our major contribution is a set of empirically evaluated guidelines for creating online media text classifiers using small training sets. We recommend uni- and bi-gram features as text representation, btc term weighting and a linear-kernel NBSVM. Our results suggest that high classification accuracy can be achieved using a manually annotated dataset of only 300 examples.

Keywords Social media · Manual annotation · Small datasets · Machine learning · Text classification

Introduction

To study online media content, researchers use methods of text classification to analyze large volumes of text data [1]. Text classifiers using supervised machine learning can be adapted to new classes and texts without modifying the algorithm, requiring an annotated training dataset only [2]. However, such training datasets are often not available for a certain class or topic of interest and a custom dataset needs to be manually annotated.

When annotating texts, the generated classifier's accuracy should increase with every additional text sample [3]. However, statistically each additional text increases the accuracy less than the previously added text, because of the asymptotical shape of the learning curve [4]. Therefore, annotating more texts decreases annotation efficiency. To minimize human annotation effort, an optimal-sized training set that provides the best trade-off between classification accuracy and manual effort should be annotated and the text classifier with the highest expected accuracy should be selected.

A major problem is that experimentally evaluating a multitude of text classifier designs for accuracy and selecting the classifier with the highest accuracy will result in overfitting of the selected classifier [5]. Consequently, both the training set size and the text classifier design should be pre-determined on empirically tested guidelines to avoid the necessity of model selection, which would require further annotation effort to create out-of-sample test data for estimating unbiased classification accuracy [6]. The objective of this work is to empirically work out a baseline recommendation for practitioners and researchers for designing as accurate as possible text classifiers given very limited resources for creating training datasets.

Previous work has concentrated on optimizing text classifiers for large datasets with more than 1000 texts or estimating accuracies for classifiers on one small dataset. Both of these approaches cannot be generalized to other small datasets, because the former approach does not take the effect of the training set size on the chosen text classifier into account and the latter may suffer from random errors in accuracy estimates due to the small training and test set size. Few studies have altered the training set size and estimated the effect on the accuracy of the classifiers. These studies

✉ Martin Riekert
martin.riekert@uni-hohenheim.de

[1] Present Address: University of Hohenheim, Stuttgart, Germany

[2] Eberhard Karls University of Tübingen, Tübingen, Germany

focused on comparing Support Vector Machines (SVMs), Naïve Bayes (NB) and other machine learning algorithms and did not evaluate the design factors of the feature vector. Moreover, these previous studies do not indicate the required dataset size that is necessary to train an accurate text classifier. Previous work concerning learning curves has provided methods to estimate the shape of the learning curve by fitting an inverse power law model to accuracy estimates. These approaches require that such a dataset is already available and are therefore not useful for researchers that intend to analyze new data.

We contribute by proposing a guideline for online media researchers and practitioners for designing text classifiers and efficiently creating custom datasets. We select a baseline classifier design based on empirical experiments using a four-way full-factorial, repeated-measures design with 32 design factors and 22 training set sizes. Furthermore, we quantify the effect of training set size on classifier accuracy. We find that a small dataset of 300 documents provides high accuracy and adding more training examples rarely substantially increases the classification performance.

Related Work

Dictionary-Based Approaches for Text Classification

Dictionaries might be chosen for small dataset sizes because they require no training set. The disadvantage of dictionary-based classifiers is that they are often not directly evaluated for the classification problem due to the lack of a labeled dataset [7]. The improvement of the dictionary is difficult due to the available words that might be added to a class, and the classifiers using dictionaries usually achieve lower accuracies [8]. A reason for the low accuracy of dictionary-based approaches is that these dictionaries are either developed for a broad application (e.g., General Inquirer [9]), achieving only rather low classification accuracy for a specific domain, or are specific dictionaries (e.g., dictionaries designed by Henry [10], Loughran and McDonald [11]) that can be applied only in a specific context. Furthermore, dictionary-based classifiers use equal weighting for each word in the dictionary because information regarding the importance of dictionary words is missing.

Effects of Training Set Sizes on Design Factors

A large body of literature improved the accuracy of text classifiers on large datasets, which typically contain over 2000 examples [12–14]. However, the training set size can play an important role for the decision of the optimal text classifier configuration [4]. The approach to select a classifier that performs well on large training set sizes neglects the effect of the training set size on the performance of the classifier, which might result in selecting the wrong classifier for the small training set [4].

Similarly, text classifiers are evaluated on smaller domain datasets in several studies [15], although accuracy estimates on small datasets suffer from random errors and results of individual experiments for one dataset might not generalize to other datasets. Therefore, one classifier might perform better than the other classifier just by chance, which might lead to contradictory results. Additionally, it cannot be estimated from such studies how big the effect of a specific training sample is on the accuracy, which is important to estimate the amount of training samples that need to be annotated.

Furthermore, there are pre-trained deep learning models like BERT [16] and ULMFIT [17] that can be fine-tuned to small datasets using transfer learning. Previous work has shown that these approaches can achieve higher accuracy than SVM text classifiers for small datasets [18]. However, these approaches require larger amounts of computational resources for training and application. For example, Usherwood et al. [18] state that BERT-base requires 12 GB of VRAM. Furthermore, the inference time of BERT-base is slower than the inference time for SVMs [19]. Therefore, substantial computational resources would be required if BERT-base is used to analyze large online media datasets [18].

Previous work on text classification has studied feature selection for small datasets by comparing feature selection methods for SVM, k-nearest neighbors (KNN) and NB classifiers on ten datasets [20]. Although this work helps to identify better feature selection methods for small datasets, classifiers were only compared on a fixed training sample of 1000 documents. Therefore, this previous work provides no information on the effect of training set size on selecting a text classifier for smaller datasets.

Furthermore, the effect of training set size on SVM and NB has been compared on a twitter dataset consisting of 4269 training and 782 test examples [21]. This study compares training set sizes from 10% (427 examples) to 100% (4269 examples) in 10% steps. Resampling of the training set was not applied and therefore the standard deviation of the accuracy is not available and random errors might affect the reported accuracy estimates. Furthermore, this study includes only one dataset with tweets and results on datasets with larger documents might vary.

Another work compared training set sizes on SVM, Multinomial NB and Decision Trees for the training set sizes of 50–500 examples in steps of 50 examples [22]. They compare their classifiers on four sentiment datasets. However, they use unigrams as text representation and term frequency as term weighting approach. They do not experiment with different term weightings and text representations.

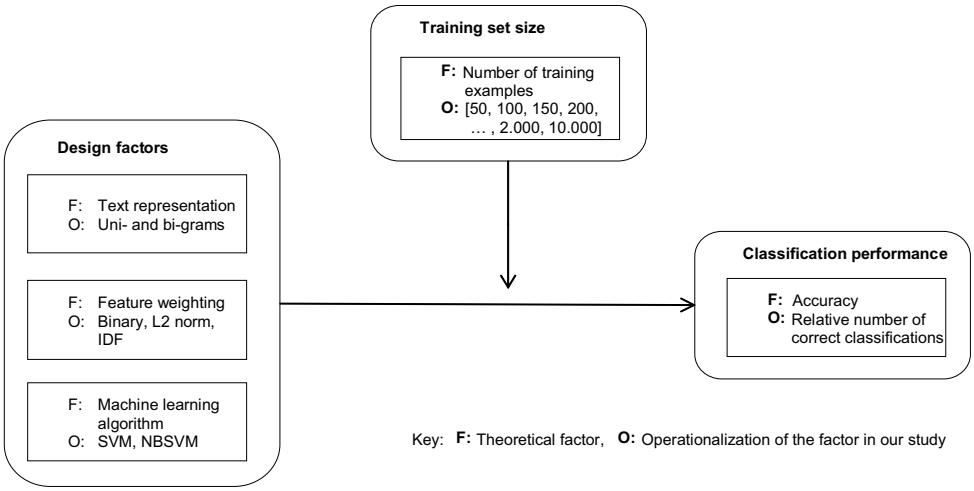

Fig. 1 Research model

The related work highlights an important gap in literature, i.e., identifying the optimal design factors for text classifiers for small training set sizes. The model selection should be conducted on several large-scale datasets to support the generalizability of the reported design factors. Additionally, the training sets should be resampled several times to reduce random errors and the standard deviations of the mean accuracies should be reported to identify the impact of a randomly annotated training sample on the accuracy.

Effect of Training Set Size on Accuracy

The effect of training set size on the accuracy of a text classifier can be represented by the learning curve [4]. The learning curve shows the relationship between expected performance and the number of training examples. The literature on learning curves for machine learning classifiers has highlighted that the test error is higher than the training error and both asymptotically reach a common value with increasing training set size [4]. The learning curve may be used to estimate the sample size that is necessary to obtain a specific minimal performance by fitting an inverse power law model to accuracy estimates using a small training set [23]. This approach supports the decision for choosing if more data should be annotated. However, it does not provide information on the necessary training set size if no training set is available yet, because organizing the collection and annotation of a random training sample requires already a large effort. To be able to decide the viability of such an effort, the best configuration and classifier must be selected based on prior studies [24]. For this purpose, it is necessary that on several text classification datasets the performance of the text classifiers is reported and that, if possible, a training set size is identified that achieves high accuracy.

Research Model

Figure 1 shows an overview of our research model. The dependent variable is classification performance measured by accuracy. Accuracy is the percentage of all documents in the test set that were classified with the class that matches the annotation of a human annotator [25]. The independent variables are grouped in design factors and training set size, which are described as follows.

The main task of text classification is based on a set of documents $D = \{d^{(1)},.., d^{(n)}\}$ and a set of classes $Y = \{1,..., m\}$, whereby a given document $d^{(i)} \in D$ is assigned a label $y^{(i)} \in Y$ [13, 26]. Generally, a class can be any conceptual entity and the number of classes could thus be arbitrarily high. However, no more than 20 classes were used in most previous datasets [14, 27].

The feature vector and the machine learning algorithm are our main design factors. The feature vectors are the input for the machine learning classifier. The feature vectors are constructed in the following text classification pipeline. First the input document $d^{(i)}$ is converted into the text representation, where n-grams (i.e., words or word sequences) contained in documents of the training set establish the dimensions x_j of the feature vector $x^{(i)}$. Second, the applied feature weighting approach calculates the values $x_j^{(i)}$ in the feature vector. Third, the feature vector $x^{(i)}$ is used as the input for the machine learning algorithm that estimates the label $y^{(i)}$. Prior to the application, the machine learning algorithm was trained on a training set that is composed of human-annotated examples. In the following we describe the three design factors that have been evaluated on the ACL IMDB dataset in our previous work [28].

Table 1 Feature weighting components

Component	Term frequency		Inverse document frequency		Length normalization	
Code	*n*	*b*	*n*	*t*	*n*	*c*
Formula	*tf*	sign(*tf*)	1	$\log \frac{N}{df}$	1	$\frac{1}{\|x^{(i)}\|_2}$

Text Representation

Text representation describes how the document will be represented in the feature vector $x^{(i)}$ for the machine learning algorithm [25]. Typically, *n*-grams are used with *n* not largely exceeding 3 [29]. However, due to the scale of our experiments, we limited the design factors to uni- and bi-grams, which have shown high accuracy in previous work [12]. Unigrams: each term is a feature, regardless of its arrangement and location in the text, e.g., ['the', 'new', 'Spielberg', 'film', 'is', 'all', 'good']. Bigrams: two sequential terms are a feature, e.g., ['the-new', 'new-Spielberg', 'Spielberg-film', …]. In our experiments we compare unigrams to a combination of uni- and bi-gram features. We consider that adding bigrams to unigrams features will increase the accuracy [12, 30]. The argument for adding bigrams to the feature vector is that bigrams allow for representing phrases in the feature vector [28].

Feature Weighting

This design factor defines the values in the feature vector. A three-letter code allows for convenient reference of each feature weight combination, e.g., ntn, bnn [31, 32]. Table 1 defines the formulas that are denoted by each letter of the code [28]. The code for the baseline feature weighting approach is indicated by *n*. For instance, nnn determines the absolute term frequency (tf), whereas ntc references the term frequency–inverse document frequency (TF–IDF), and then normalizes the vector to unit length with L2 normalization. The L2 normalization is calculated on the complete feature vector after the other feature weighting operations are calculated. Therefore, ntn is calculated as follows with *N* being the total number of documents and *df* being number of documents that contain the feature:

$$\text{ntn} = tf \times \log \frac{N}{df} \times 1$$

For the term frequency component, the binary representation (bxx) of features in the document has increased performance compared to absolute term frequency (nxx) for sentiment classification [33, 34]. A possible explanation is that word frequency per document has only limited impact on the sentiment of a document, and that the occurrence of features is more important [34]. The reason for applying inverse document frequency (xtx) stems from Zipf's Law, which states that few words occur often, whereas most words occur seldom [35]. Common words do not help in discriminating documents. Weighting features using IDF will decrease the values in the feature vector of common words [36, 37]. The argument for L2 normalization (xxc) is due to differences in the number of words per document. Then, shorter documents are represented by feature vectors with a lower L2 norm, while longer documents are represented by feature vectors with higher L2 norm vector length [31]. Dissimilar vector length potentially reduces classification accuracy because documents with similar content but different length will be represented differently. Therefore, inserting a normalization factor into the weighting formula can increase accuracy [34, 38].

Machine Learning Algorithms

Machine learning algorithms use annotated training data to learn a classification model for the application to unseen input documents. Linear-kernel Support Vector Machines (SVMs [39]) are frequently used and achieve high performance on text classification tasks with larger documents [12, 13]. An NBSVM is an SVM that uses Naïve Bayes (NB [40]) features and has been shown to achieve higher accuracies than SVM [12].

Training Set Size

Additional training examples increase the expected accuracy of the machine learning classifier. However, statistically, each additional example increases the accuracy less than the previous example because of the asymptotic shape of the learning curve [4]. Therefore, for each dataset, we generated training sets sizes of 50–1000 examples in intervals of 50 examples. Furthermore, we added training sets with 2000 and 10,000 examples for reference. We were mostly interested in the smaller training set sizes, but from a practical perspective the intervals of 50 examples seem sufficient, because annotating 50 documents can be achieved in a reasonable time frame. Each of these training sets were stratified, i.e., the same relative number of documents per class were present in the training set.

Table 2 Datasets

Name	Type	Domain	Classes	Training set size (k)	Test set size (k)
ACL IMDB	Sentiment	Movies	2	25	25
AG's news	Topic	News	4	120	8
DBPedia	Topic	Wiki	14	560	70
Sogou news	Topic	News	5	450	60
Yelp polarity	Sentiment	Restaurants	2	560	38
Yelp review full	Sentiment	Restaurants	5	650	50
Yahoo answers	Topic	Q&A	10	1400	60

Method

Experimental Setup

The experimental setup is described in the following. Our experiment had a four-way factorial repeated-measures design with accuracy as the dependent variable. Treatments were carried out by combining 22 training set sizes and 32 design factor combinations, i.e., 2 text representations, 8 feature weightings and 2 machine learning algorithms. The $22\times2\times8\times2$ factorial experiment allowed us to compare results obtained for a total of 704 treatment conditions for each of the 7 datasets, which resulted in 4928 combinations in total (design factors × training sets × datasets). Each of these 4928 combinations were repeated 20 times with resampled training sets resulting in 98,560 experimental runs. The 20 repetitions for each combination were used to reduce random errors for each combination and to obtain a standard deviation for each combination. In the machine learning experiments the hold-out method was used to evaluate the performance of the classifiers. The training and test set necessary for each hold-out method were generated as follows. The resampled training sets used in our study were drawn from the training set of the respective dataset. The test set to estimate the accuracy using the hold-out method was always the full test set of each dataset [14, 27].

Datasets

The datasets used in our study are described in Table 2. This table shows the different domains and problem types of the used online media datasets as well as their test and training set sizes. The first dataset is the ACL IMDB dataset by Maas et al. [27]. The other datasets are large-scale datasets that have been created by Zhang et al. [14] during an evaluation of deep learning algorithms, which require large training sets. The datasets analyzed during the this study are publicly available in online repositories.[1]

Table 2 indicates that there are various numbers of classes, domains, and classification types. All datasets contain a large number of training examples and were generated automatically or by the authors of the document, e.g., the ACL IMDB dataset classifies polarity by linearly mapping the 10 star rating {0, 1, … ,10} to sentiment polarity [27]. Manually annotated datasets are generally much smaller due to their expensive annotation process.

The datasets were used as they were provided in the original studies. This includes but was not limited to: no cleansing procedures (e.g., no stop word removal) and no preprocessing (including no stemming or lemmatization). This is because the results of such cleaning procedures are context specific and may have a negative impact on the generalizability of our results for some classification tasks. We did not apply feature selection, because it comes with various hyperparameters, which are out of scope for this study. Furthermore, the effect of feature selection on accuracy is not finally determined if SVMs are used due to the built-in regularization term of the SVM, which has a similar effect as feature selection [37, 41].

Machine Learning Configuration

We applied the machine learning algorithms with their default configurations as provided in the scikit-learn machine learning library [42]. The SVM implementation by scikit-learn used LIBLINEAR, which is a publicly available SVM implementation [43]. We applied the default configuration (L2-regularized and L2-loss dual-form SVM with linear kernel, penalty $C=1$, and margin of tolerance $\varepsilon=0.01$). Similarly, we used the default configuration of the NBSVM algorithm (with $\beta=0.25$) [12]. We used the hold-out method with the complete test set of the datasets to calculate all our accuracy estimates [44].

Results

Model Selection

In the following, we analyze how different design factors affect the accuracy for both small and large training set sizes. We defined the following groups of training set sizes: small training sets consist of 50–500 training examples, large training sets consist of 550–1000 examples and, additionally, training set sizes of 2000 and 10,000 training examples are grouped individually (see Fig. 2).

[1] The datasets are accessible via following URLs https://github.com/zhangxiangxiao/Crepe and https://ai.stanford.edu/~amaas/data/sentiment/.

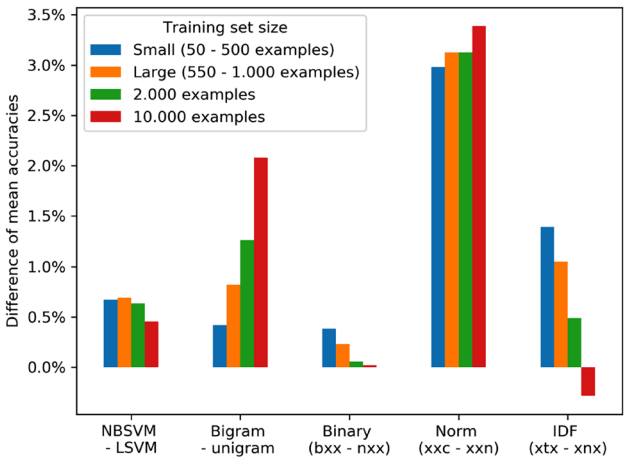

Fig. 2 Difference of mean accuracies for the design factors averaged over all datasets

Averaged over the seven datasets, the effects on accuracy for the design factors were in positive direction for all training set sizes except for IDF using a training set size of 10,000 examples (see Fig. 2). Additionally, Fig. 2 indicates that for NBSVM, binary and IDF, the effect on accuracy is inversely related to the training set size. Using uni- and bigrams always increased the accuracy compared to using only unigrams and the effect increases with training set size. Furthermore, we found that applying L2 normalization to the feature weights has the largest effect on accuracy. The effect seems to increase with more examples.

However, Fig. 2 averages the results over several datasets and it is likely that some of the design factors do not only depend on training set size but also on the dataset. To check whether the effect of the training set size was stable for the seven datasets used in this study, we analyzed the results for each dataset individually for different training set sizes (Fig. 3).

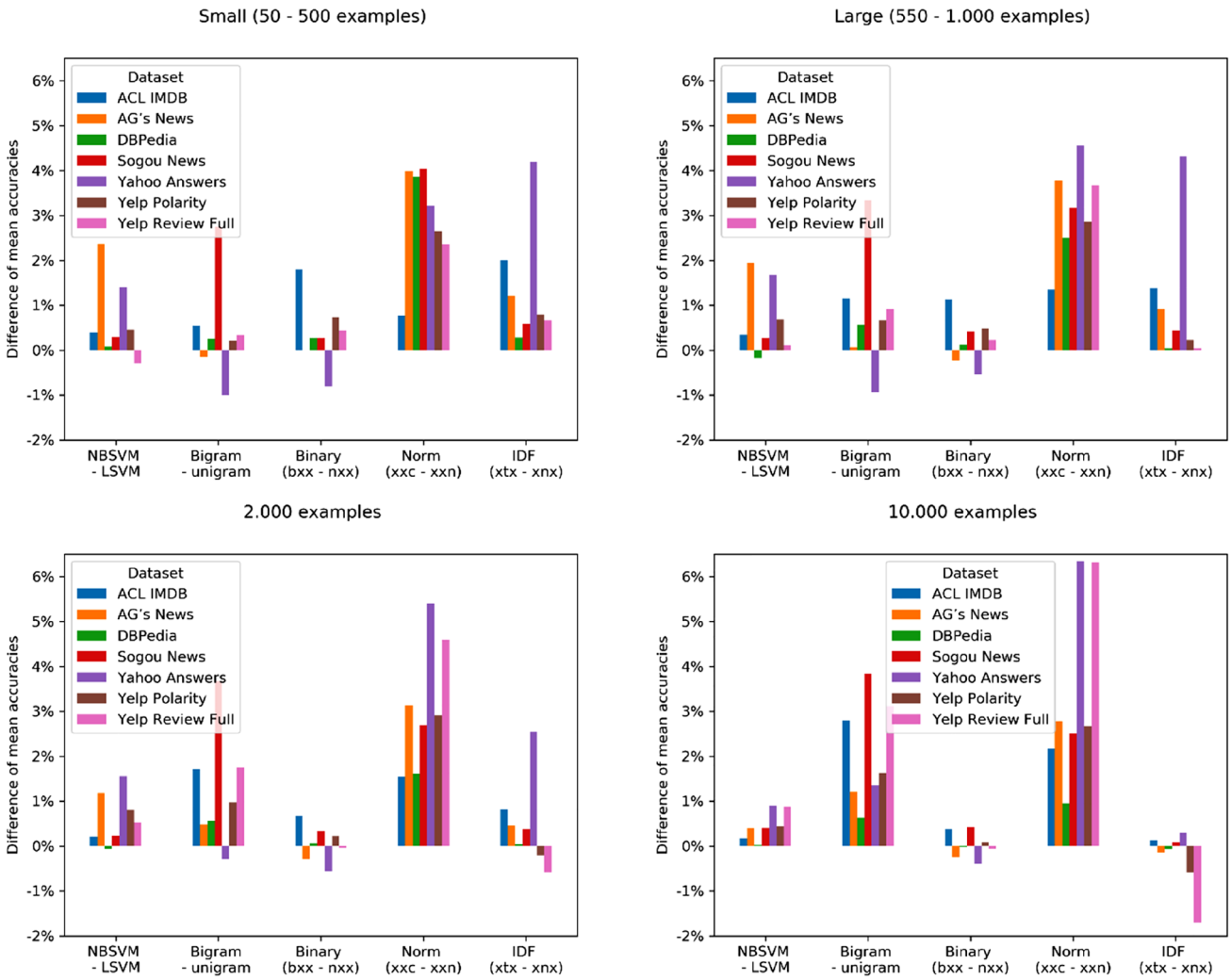

Fig. 3 Difference of mean accuracy for the design factors for the datasets used in this study

Figure 3 depicts the results for different training set sizes and datasets. NBSVM overall increased the performance for all datasets except for DBPedia. Bigrams had a negative effect on AG's News and Yahoo! Answers, but increased the performance for all other datasets. However, with increasing training set size the effect for both datasets turned positive eventually. The effect of binary features (bxx) decreased in larger training sets and was negative for AG's News Yahoo! Answers. L2 Normalization (xxc) had an overall positive effect on all datasets. IDF had a positive effect for small

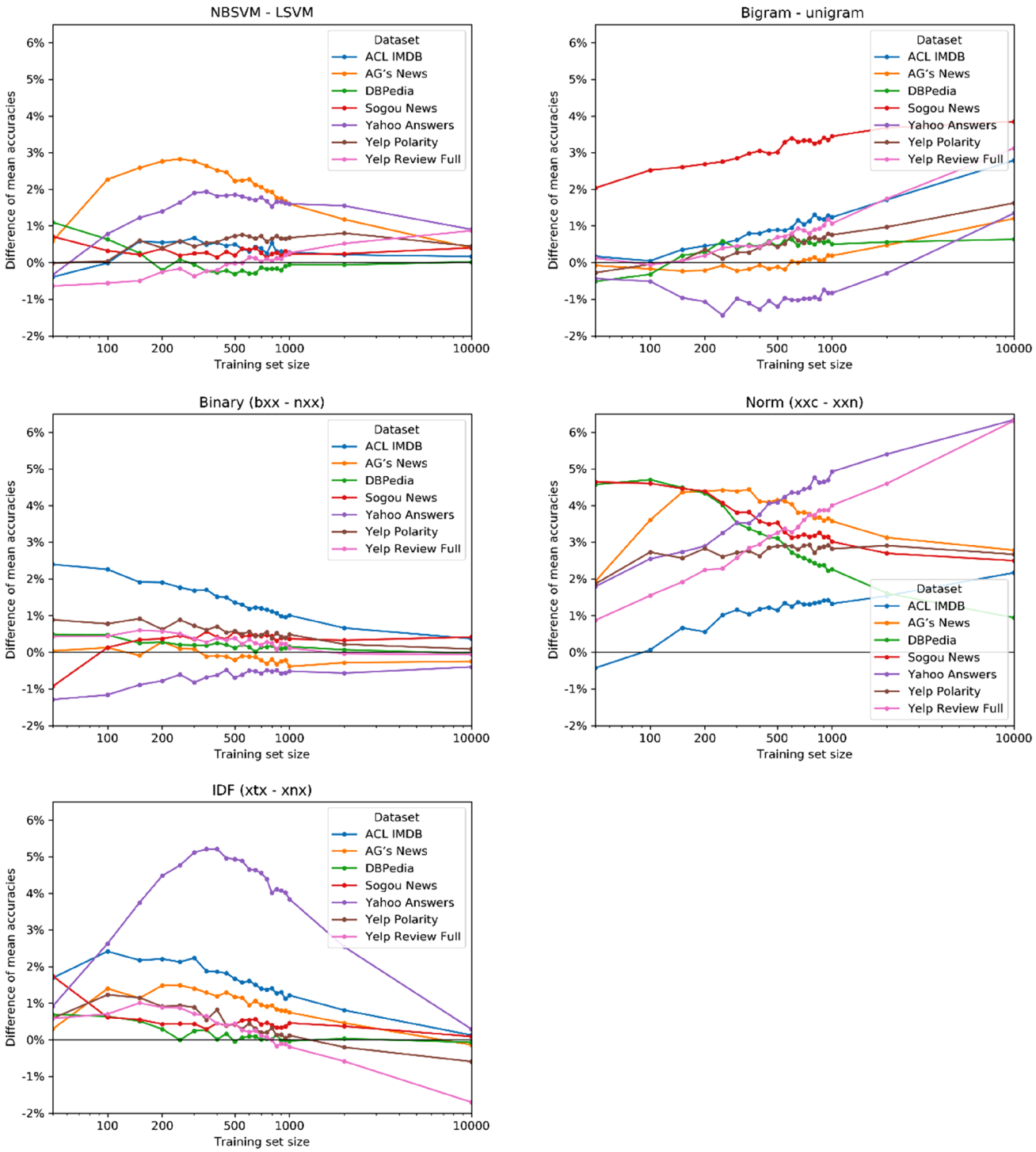

Fig. 4 Difference of mean accuracies for the design factors. The *x*-axis depicts the training set size in logarithmic scale

training set sizes. However, the positive impact of IDF (xtx) degraded and became negative or non-existent with increasing training set size. For training set sizes exceeding 2000 examples Yelp Review Full and Yelp polarity were both negatively affected by the application of IDF term weighting. To sum up, these results indicate that most factors were mainly affected by training set size.

Figure 4 displays changes of the effect of a design factor with respect to training set size and dataset. Figure 4 indicates that the direction of the effect for bigrams, visible at the top-right in Fig. 4 for Yahoo Answers changes with more than 2,000 training examples. Similar effects can be observed for IDF (Fig. 4, on the bottom-left).

Given the previous results, we analyze if there are any interaction effects between the three design factors. For this purpose, we provide tables containing all factor combinations for small (Table 4), large (Table 5) and all (Table 7) training set sizes and a training set size of 10,000 examples (Table 6) in the appendix of this study. These tables indicate that there are most likely no large interaction effects that would make the combination of all factors disadvantageous. Furthermore, Tables 4, 5, 6 and 7 indicate that the combination of all factors (i.e., uni- and bigrams, NBSVM, btc) achieves in most cases the highest accuracy, making further interaction effects irrelevant, because we are only interested in the most accurate factor combination. Therefore, for small datasets we select the design factor combination of uni- and bigrams, NBSVM and btc. Datasets with more than 1000 examples might use the same factor combination except for the term weighting ntc and datasets with more than 10,000 examples could use the term weighting nnc.

Training Set Size

Figure 5 depicts the accuracy for the selected factor combination for small datasets (uni- and bigrams, btc and NBSVM). The standard deviation of the accuracy is displayed by means of the error bars on each data point over the 20 repetitions per experiment. Figure 5 indicates that for the provided design factors a dataset with more than 300 examples increases the accuracy only moderately. Note that the number of classes of the dataset does not have a large impact for the accuracy improvement even though datasets with more classes contained fewer documents per class. For example, the DBPedia dataset consists of 14 classes and did not benefit from using more examples, but the Yahoo Answers dataset, consisting of 10 classes, might benefit slightly from more examples. However, in both cases the

Table 3 Average *M* and standard deviation SD of the accuracy both in percent for the factor combination uni- and bigrams, btc and NBSVM

Train set size	ACL IMDB		AG's news		DBPedia		Sogou news		Yahoo answers		Yelp polarity		Yelp review full	
	M [%]	SD [%]	*M* [%]	SD [%]	*M* [%]	SD [%]	*M* [%]	SD [%]	*M* [%]	SD [%]	*M* [%]	SD [%]	*M* [%]	SD [%]
50	65.50	1.97	38.82	3.01	57.87	3.20	78.33	2.54	18.13	1.59	69.08	2.39	29.02	1.35
100	71.79	1.57	52.83	2.69	69.28	2.20	83.73	1.28	28.25	2.23	75.23	1.25	33.71	1.00
150	74.99	1.64	60.83	1.99	77.57	1.48	85.26	0.64	34.04	1.92	78.13	1.16	36.74	1.14
200	76.74	0.96	66.46	2.03	80.04	0.98	86.32	0.68	37.41	1.76	80.02	1.15	38.46	0.95
250	78.39	1.14	69.47	1.29	82.58	1.01	87.11	0.62	40.57	1.42	81.03	0.85	39.57	0.94
300	79.30	0.85	71.96	1.07	83.92	0.74	87.89	0.60	43.24	1.24	82.21	0.65	40.92	0.75
350	79.91	0.71	74.15	0.80	85.50	0.42	88.24	0.52	45.14	1.10	82.58	1.00	41.83	0.87
400	80.49	0.94	75.32	0.70	86.52	0.61	88.60	0.58	46.68	1.34	83.45	0.59	42.10	0.55
450	81.26	0.81	76.57	0.65	87.34	0.42	88.79	0.74	48.09	1.00	84.00	0.66	42.81	0.70
500	81.40	0.87	77.76	0.91	88.16	0.48	89.24	0.49	49.04	0.85	84.55	0.75	43.50	0.70
550	81.91	0.48	78.52	0.78	88.77	0.47	89.59	0.40	50.14	0.93	85.08	0.72	44.20	0.67
600	82.26	0.40	79.35	0.65	89.16	0.26	89.83	0.36	50.67	0.73	85.33	0.57	44.23	0.51
650	82.40	0.40	79.78	0.60	89.35	0.45	89.96	0.45	51.13	0.97	85.81	0.60	44.85	0.45
700	83.03	0.38	80.49	0.46	90.07	0.43	90.23	0.28	51.97	0.80	85.81	0.58	45.30	0.38
750	82.98	0.43	80.92	0.52	90.35	0.33	90.35	0.35	52.43	0.59	86.27	0.55	45.49	0.50
800	83.53	0.47	81.33	0.49	90.70	0.41	90.51	0.36	53.04	0.85	86.38	0.43	45.89	0.51
850	83.33	0.48	81.66	0.51	90.91	0.34	90.71	0.28	53.33	0.61	86.68	0.32	46.04	0.38
900	83.54	0.45	81.92	0.41	91.27	0.33	90.92	0.26	53.86	0.50	86.83	0.47	46.14	0.46
950	83.85	0.40	82.42	0.46	91.38	0.39	91.07	0.26	54.32	0.58	87.05	0.53	46.76	0.44
1000	83.93	0.60	82.61	0.50	91.66	0.39	91.17	0.30	54.81	0.57	87.22	0.31	46.78	0.59
2000	85.99	0.17	85.62	0.27	94.00	0.17	92.51	0.20	59.38	0.46	89.37	0.20	49.55	0.28
10,000	89.58	0.10	89.50	0.22	96.95	0.04	94.85	0.06	66.74	0.19	92.73	0.09	54.67	0.25
All	91.05	–	92.62	–	98.87	–	97.47	–	75.34	–	96.10	–	61.35	–

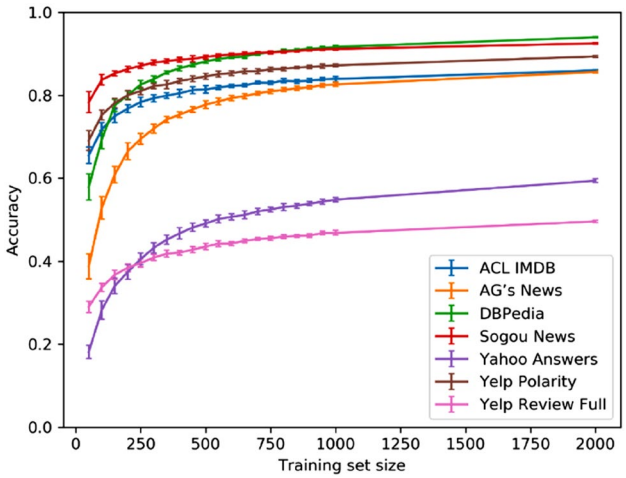

Fig. 5 Accuracy for the factor combination bigrams, btc and NBSVM. The error bars depict the standard deviation of the 20 repetitions

effect was rather small. Additionally, Fig. 5 indicates that the standard deviation for the training set size is small.

The following Table 3 provides the averaged accuracies from Fig. 5 in tabular form. The last column "all" displays the accuracy estimate of the selected model trained on the complete training set as displayed in Table 2. Note that for each dataset the training set size varies. Table 3 shows that for each dataset the individual training set sample had only a minor impact. For only 100 training examples the standard deviation was below 3% for each of the 7 datasets. This result indicates that annotating only a small dataset will already help to predict the accuracy of a model trained on a larger dataset.

Discussion

Findings

In general, our results indicate that increasing the training set size above 300 examples has only a minor effect on increasing a classifiers' accuracy for most datasets. Furthermore, this effect seems to hold independent of the number of annotated classes in the dataset. The findings are largely consistent across different datasets, although some interaction effects with the training set size were found:

- NBSVM improved the accuracy in most cases. Increasing the training set sizes reduced the positive effect of NBSVM.
- Bigrams increased the accuracy in most experiments. The positive effect of bigrams on accuracy increased with increasing training set size. This effect can be explained by model complexity [45]. First, many of the additional features that bigrams provide can only be utilized by larger datasets, because bigrams are less likely to occur. Second, bigrams increase the number of features, which results in the necessity to fit more weights of the machine learning algorithm. Therefore, using bigrams requires larger training sets to reduce overfitting of the additional weights.
- Binary features (bxx) increase the accuracy for small training sets, but should have no effect for large training sets. Similar results have been found in previous work for sentiment classification [34].
- IDF (xtx) has a positive effect on small training sets but the effect diminishes for large training sets. IDF increases the weight of terms that appear in fewer documents.
- L2 normalization (xxc) of the feature weights had consistently a positive effect on the accuracy. These results suggest that normalization should be applied in all text classifier designs.

Limitations

The generalizability of our results is subject to limitations. First, note that our results are based only on seven datasets and the labeling was generated by the authors or automatic processes, which is not the same as manual labeling from non-author annotators. Therefore, our results might not generalize to all manually generated datasets.

Second, for each design factor we compared all experiments including the factor and all experiments not including the factor to measure the effect on accuracy of the specific design factor. This experimental approach implies that design factors do not interact among each other. Although this is a naïve assumption, further work could study several effective design factor combinations that we already found like btc, ntc and bigrams to check for interaction effects among them.

Third, in this context another remark must be made upon our findings for the NBSVM classifier. All prior applications of NBSVM were restricted to a particular feature weighting schema (bnn) [12]. We acknowledge that the NBSVM algorithm might be tailored to this schema, which in turn could explain some of the interaction effects.

Fourth, we did not apply deep learning in our experiments, even though deep learning has recently generated breakthroughs in the fields of text understanding and conversational AI (e.g., the breakthrough results by BERT [16, 46]). Furthermore, the deep learning approach ULMfit has been suggested for text classification [17]. ULMfit uses transfer learning to improve text classification performance. We did not integrate ULMfit or BERT in our experiments.

At the time of writing, the added complexity and computational resource requirements to use these methods seemed not practical for many social media researchers and practitioners. Therefore, we focused on simpler, but yet effective approaches.

Implications

The implications of the laid-out limitations regarding future work are as follows. First, further work may study how much training data for transfer learning is necessary to achieve certain levels of classification accuracy, especially if BERT or ULMFiT is used. A focus on finding the right number of training epochs and fine-tuning parameters should be made, because in a short experiment we used the BERT-base model[2] and could not achieve high accuracy for a training set size of 100 or 300 examples for the ACL IMDB dataset using reported parameter configurations [18].

Second, further work is needed to fully understand the interactions between training set sizes and classifier design factors explored in this work. We found some interactions between design factor and dataset (e.g., for term presence and uni- and bigrams). However, we could not pinpoint the dataset characteristics that lead to these interactions. None of the characteristics of the dataset that were known to us (i.e., source, classification type, domain, or number of classes) seemed to be related to these interactions. Understanding these interactions would allow for small training sets, while increasing the classifiers' accuracy further.

Third, as pointed out in the limitations section above, we assume linear relationships between the design factors. We believe there is potential to further improve the classification performance by further studying interactions among effective design factors such as the different feature weighting schemes.

Fourth, in another study the interactions between accuracy and feature selection for large datasets between 10 thousand and 1.6 million tweets were investigated and it was found that common feature selection algorithms can play a role to increase accuracy [41]. Additionally, feature selection is the recommended approach by practitioners and researchers [48]. In contrast, previous work also indicates that the performance tends to increase if more features are used [47]. It would be interesting to investigate if feature selection can improve the accuracy for small datasets using our experimental approach [37].

Fifth, the already high accuracy for more than 300 examples is consistent with previous research that used a gene expression dataset [49]. Therefore, additional research could investigate whether the high accuracy for more than 300 examples is a more general phenomenon.

Conclusion

This paper reports an experimental study, examining the design factors that affect the accuracy of machine learning text classifiers for small, manually annotated datasets. We contribute insights on how text classifiers should be designed and how training sets should be sized to both achieve high classification accuracy and to also minimize the amount of human labor required.

We observed several interaction effects between design factors, training set size and dataset, which corroborate the need for further research. However, we find that overall, the theoretical design factors for machine learning-based classifier design generalize well among different training set sizes and datasets. Online media researchers and practitioners can use this knowledge as guidance for more efficiently designing custom datasets and they can readily use our proposed baseline design factor choice for their text classifiers. Thus, researchers and practitioners can reduce human labor and increase the accuracy of their classifiers without setting up a large number of experiments.

As a baseline for classifier training on small datasets, we recommend uni- and bi-gram features as text representation, btc term weighting and a linear-kernel NBSVM as the machine learning algorithm. Our results suggest that a manually annotated training set may contain only 300 examples and still achieve high accuracy. Accuracy could be measured by cross-validation to avoid an additional dataset. Additionally, one might measure the performance at smaller training set sizes to get a first indication on the feasibility of the pursued classification task, because the standard deviation of the accuracy for different training set examples is rather small even for small training set sizes.

Our experiments also indicate that the number of classes has a minor role for the relationship between training set size and accuracy, which is surprising, because the number of examples per class is lower for datasets with more classes, given equal training set size. However, further research is required to study the effect of the number of classes on the accuracy.

Appendix

The following Tables 4, 5, 6 and 7 show mean accuracies for all evaluated design factors. The color patterns in the tables indicate that design factors behave similarly in terms of accuracy for different data sets.

[2] https://www.tensorflow.org/tutorials/text/classify_text_with_bert.

Table 4 Small training set size (50 – 500 examples)

NBSVM	Bigram	Binary	Norm	IDF	ACL IMDB	AG's News	DBPedia	Sogou News	Yahoo Answers	Yelp Polarity	Yelp Review Full	mean
TRUE	TRUE	TRUE	TRUE	TRUE	0.770	0.664	0.799	0.864	0.391	0.800	0.389	0.668
FALSE	TRUE	FALSE	TRUE	TRUE	0.748	0.643	0.813	0.872	0.400	0.798	0.387	0.666
TRUE	FALSE	FALSE	TRUE	TRUE	0.752	0.666	0.801	0.843	0.415	0.795	0.383	0.665
		TRUE	TRUE	TRUE	0.767	0.668	0.793	0.830	0.408	0.801	0.384	0.664
	TRUE	FALSE	TRUE	TRUE	0.750	0.668	0.806	0.853	0.394	0.791	0.378	0.663
FALSE	TRUE	TRUE	TRUE	TRUE	0.766	0.641	0.803	0.863	0.375	0.801	0.389	0.663
	FALSE	FALSE	TRUE	TRUE	0.746	0.645	0.807	0.853	0.405	0.792	0.382	0.662
		TRUE	TRUE	TRUE	0.759	0.645	0.801	0.827	0.386	0.796	0.381	0.656
TRUE	FALSE	TRUE	TRUE	FALSE	0.744	0.649	0.792	0.831	0.344	0.783	0.369	0.644
FALSE	TRUE	TRUE	TRUE	FALSE	0.743	0.617	0.791	0.851	0.334	0.783	0.378	0.642
TRUE	TRUE	TRUE	FALSE	FALSE	0.750	0.618	0.770	0.848	0.347	0.779	0.365	0.639
FALSE	FALSE	TRUE	TRUE	FALSE	0.737	0.619	0.795	0.823	0.337	0.779	0.372	0.637
TRUE	TRUE	TRUE	TRUE	FALSE	0.743	0.644	0.791	0.841	0.297	0.782	0.358	0.637
		FALSE	FALSE	FALSE	0.749	0.618	0.770	0.826	0.354	0.775	0.361	0.636
FALSE	TRUE	FALSE	TRUE	FALSE	0.703	0.616	0.785	0.856	0.344	0.769	0.370	0.635
TRUE	FALSE	TRUE	FALSE	FALSE	0.742	0.623	0.770	0.806	0.354	0.776	0.360	0.633
FALSE	FALSE	FALSE	TRUE	FALSE	0.701	0.619	0.787	0.843	0.345	0.766	0.367	0.633
TRUE	FALSE	FALSE	FALSE	FALSE	0.742	0.618	0.767	0.796	0.356	0.776	0.361	0.631
			TRUE	FALSE	0.690	0.646	0.784	0.835	0.327	0.762	0.350	0.628
FALSE	TRUE	TRUE	FALSE	TRUE	0.750	0.599	0.760	0.829	0.321	0.764	0.357	0.626
TRUE	TRUE	FALSE	FALSE	TRUE	0.731	0.612	0.757	0.827	0.352	0.754	0.345	0.626
FALSE	TRUE	FALSE	FALSE	TRUE	0.739	0.599	0.754	0.817	0.341	0.757	0.357	0.623
TRUE	TRUE	FALSE	TRUE	FALSE	0.692	0.642	0.777	0.829	0.295	0.758	0.345	0.620
		TRUE	FALSE	TRUE	0.723	0.608	0.752	0.830	0.331	0.749	0.343	0.620
FALSE	FALSE	TRUE	FALSE	TRUE	0.740	0.597	0.751	0.791	0.328	0.757	0.348	0.616
TRUE	FALSE	FALSE	FALSE	TRUE	0.722	0.611	0.749	0.787	0.348	0.752	0.340	0.615
FALSE	FALSE	FALSE	FALSE	TRUE	0.732	0.598	0.747	0.782	0.340	0.752	0.345	0.614
	TRUE	TRUE	FALSE	FALSE	0.728	0.586	0.761	0.820	0.289	0.755	0.352	0.613
TRUE	FALSE	TRUE	FALSE	TRUE	0.715	0.607	0.746	0.769	0.338	0.748	0.339	0.609
FALSE	FALSE	TRUE	FALSE	FALSE	0.717	0.586	0.762	0.789	0.294	0.749	0.344	0.606
	TRUE	FALSE	FALSE	FALSE	0.709	0.587	0.747	0.785	0.293	0.746	0.346	0.602
	FALSE	FALSE	FALSE	FALSE	0.700	0.587	0.744	0.765	0.293	0.743	0.342	0.596

Table 5 Large training set size (550–1000 examples)

NBSVM	Bigram	Binary	Norm	IDF	ACL IMDB	AG's News	DBPedia	Sogou News	Yahoo Answers	Yelp Polarity	Yelp Review Full	mean
TRUE	TRUE	TRUE	TRUE	TRUE	0.831	0.809	0.904	0.904	0.526	0.862	0.456	0.756
FALSE	TRUE	FALSE	TRUE	TRUE	0.819	0.792	0.911	0.908	0.538	0.860	0.449	0.754
		TRUE	TRUE	TRUE	0.832	0.786	0.909	0.905	0.523	0.866	0.452	0.753
TRUE	FALSE	FALSE	TRUE	TRUE	0.819	0.811	0.902	0.880	0.551	0.859	0.446	0.753
	TRUE	FALSE	TRUE	TRUE	0.821	0.812	0.905	0.898	0.526	0.858	0.447	0.753
	FALSE	TRUE	TRUE	TRUE	0.825	0.810	0.897	0.872	0.550	0.859	0.443	0.751
FALSE	FALSE	FALSE	TRUE	TRUE	0.810	0.792	0.908	0.886	0.542	0.851	0.434	0.746
		TRUE	TRUE	TRUE	0.818	0.788	0.906	0.872	0.533	0.853	0.433	0.743
TRUE	FALSE	TRUE	TRUE	FALSE	0.812	0.799	0.894	0.870	0.495	0.848	0.437	0.736
	TRUE	TRUE	FALSE	FALSE	0.821	0.768	0.888	0.896	0.484	0.848	0.424	0.733
FALSE	TRUE	TRUE	TRUE	FALSE	0.808	0.767	0.899	0.899	0.469	0.848	0.438	0.733
TRUE	TRUE	FALSE	FALSE	FALSE	0.817	0.768	0.887	0.877	0.489	0.844	0.422	0.729
FALSE	TRUE	FALSE	TRUE	FALSE	0.782	0.773	0.896	0.894	0.481	0.836	0.435	0.728
TRUE	FALSE	FALSE	TRUE	FALSE	0.776	0.798	0.893	0.878	0.484	0.835	0.425	0.727
	TRUE	TRUE	TRUE	FALSE	0.811	0.792	0.892	0.893	0.426	0.848	0.422	0.726
FALSE	FALSE	FALSE	TRUE	FALSE	0.778	0.774	0.896	0.879	0.487	0.832	0.429	0.725
		TRUE	TRUE	FALSE	0.797	0.771	0.897	0.869	0.476	0.838	0.426	0.725
TRUE	FALSE	TRUE	FALSE	FALSE	0.800	0.765	0.882	0.851	0.482	0.839	0.412	0.719
FALSE	TRUE	TRUE	FALSE	TRUE	0.818	0.753	0.881	0.885	0.458	0.827	0.409	0.718
TRUE	TRUE	FALSE	TRUE	FALSE	0.775	0.794	0.889	0.880	0.429	0.833	0.414	0.716
FALSE	TRUE	FALSE	FALSE	TRUE	0.809	0.754	0.877	0.875	0.472	0.820	0.406	0.716
TRUE	FALSE	FALSE	FALSE	FALSE	0.799	0.764	0.880	0.839	0.479	0.837	0.412	0.716
	TRUE	FALSE	FALSE	TRUE	0.793	0.763	0.876	0.882	0.484	0.812	0.391	0.714
		TRUE	FALSE	TRUE	0.787	0.760	0.875	0.887	0.469	0.813	0.391	0.712
FALSE	TRUE	TRUE	FALSE	FALSE	0.792	0.738	0.880	0.873	0.409	0.819	0.405	0.702
	FALSE	TRUE	FALSE	TRUE	0.796	0.749	0.870	0.836	0.455	0.812	0.390	0.701
		FALSE	FALSE	TRUE	0.790	0.751	0.861	0.831	0.459	0.807	0.389	0.698
	TRUE	FALSE	FALSE	FALSE	0.781	0.743	0.875	0.846	0.415	0.816	0.404	0.697
TRUE	FALSE	FALSE	FALSE	TRUE	0.776	0.755	0.861	0.827	0.469	0.809	0.380	0.697
		TRUE	FALSE	TRUE	0.771	0.754	0.860	0.822	0.464	0.807	0.380	0.694
FALSE	FALSE	TRUE	FALSE	FALSE	0.776	0.740	0.877	0.833	0.410	0.809	0.392	0.691
		FALSE	FALSE	FALSE	0.771	0.741	0.871	0.821	0.412	0.809	0.392	0.688

Table 6 10,000 training examples

NBSVM	Bigram	Binary	Norm	IDF	ACL IMDB	AG's News	DBPedia	Sogou News	Yahoo Answers	Yelp Polarity	Yelp Review Full	mean
TRUE	TRUE	FALSE	TRUE	TRUE	0.893	0.898	0.970	0.945	0.675	0.926	0.545	0.836
		TRUE	TRUE	TRUE	0.896	0.895	0.970	0.948	0.667	0.927	0.547	0.836
FALSE	TRUE	TRUE	TRUE	TRUE	0.895	0.886	0.970	0.946	0.672	0.926	0.536	0.833
		FALSE	TRUE	TRUE	0.889	0.891	0.971	0.945	0.679	0.923	0.533	0.833
TRUE	TRUE	TRUE	TRUE	FALSE	0.889	0.890	0.964	0.942	0.642	0.921	0.537	0.827
FALSE	TRUE	FALSE	TRUE	FALSE	0.880	0.887	0.966	0.937	0.665	0.915	0.533	0.826
		TRUE	TRUE	FALSE	0.884	0.882	0.966	0.943	0.652	0.918	0.530	0.825
TRUE	FALSE	FALSE	TRUE	TRUE	0.878	0.893	0.966	0.920	0.680	0.915	0.521	0.825
	TRUE	FALSE	TRUE	FALSE	0.875	0.892	0.965	0.936	0.653	0.916	0.534	0.824
	FALSE	FALSE	TRUE	FALSE	0.867	0.891	0.965	0.917	0.684	0.907	0.532	0.823
		TRUE	TRUE	FALSE	0.878	0.887	0.962	0.914	0.678	0.910	0.531	0.823
				TRUE	0.879	0.890	0.964	0.918	0.675	0.913	0.515	0.822
FALSE	FALSE	FALSE	TRUE	FALSE	0.870	0.885	0.964	0.917	0.669	0.906	0.521	0.819
TRUE	TRUE	TRUE	FALSE	FALSE	0.888	0.875	0.964	0.942	0.635	0.914	0.505	0.818
FALSE	FALSE	FALSE	TRUE	TRUE	0.868	0.886	0.967	0.919	0.669	0.908	0.502	0.817
TRUE	TRUE	FALSE	FALSE	FALSE	0.883	0.875	0.964	0.930	0.634	0.912	0.502	0.814
FALSE	FALSE	TRUE	TRUE	TRUE	0.869	0.881	0.965	0.916	0.663	0.907	0.496	0.814
				FALSE	0.869	0.879	0.962	0.912	0.656	0.905	0.510	0.813
	TRUE	TRUE	FALSE	TRUE	0.886	0.871	0.961	0.934	0.619	0.894	0.480	0.806
		FALSE	FALSE	TRUE	0.881	0.874	0.959	0.928	0.628	0.890	0.473	0.805
TRUE	TRUE	TRUE	FALSE	TRUE	0.874	0.868	0.961	0.935	0.625	0.888	0.468	0.803
FALSE	TRUE	TRUE	FALSE	FALSE	0.872	0.868	0.961	0.926	0.601	0.901	0.489	0.803
		FALSE	FALSE	FALSE	0.871	0.870	0.960	0.916	0.604	0.900	0.489	0.802
TRUE	TRUE	FALSE	FALSE	TRUE	0.873	0.866	0.961	0.930	0.627	0.887	0.466	0.801
	FALSE	TRUE	FALSE	FALSE	0.847	0.859	0.957	0.898	0.608	0.892	0.470	0.790
		FALSE	FALSE	FALSE	0.843	0.857	0.957	0.884	0.602	0.893	0.468	0.786
FALSE	FALSE	TRUE	FALSE	FALSE	0.840	0.852	0.955	0.881	0.583	0.881	0.451	0.778
		FALSE	FALSE	FALSE	0.839	0.854	0.955	0.874	0.581	0.884	0.453	0.777
		TRUE	FALSE	TRUE	0.841	0.845	0.951	0.877	0.579	0.868	0.427	0.770
		FALSE	FALSE	TRUE	0.839	0.849	0.948	0.870	0.579	0.867	0.427	0.768
TRUE	FALSE	TRUE	FALSE	TRUE	0.828	0.843	0.947	0.876	0.582	0.869	0.423	0.767
		FALSE	FALSE	TRUE	0.827	0.844	0.946	0.874	0.576	0.873	0.424	0.766

Table 7 Accuracy values of all training set sizes

NBSVM	Bigram	Binary	Norm	IDF	ACL IMDB	AG's News	DBPedia	Sogou News	Yahoo Answers	Yelp Polarity	Yelp Review Full	mean
TRUE	TRUE	TRUE	TRUE	TRUE	0.807	0.749	0.861	0.889	0.474	0.839	0.431	0.721
FALSE	TRUE	FALSE	TRUE	TRUE	0.791	0.731	0.871	0.894	0.485	0.836	0.426	0.719
TRUE	FALSE	FALSE	TRUE	TRUE	0.793	0.751	0.861	0.866	0.498	0.834	0.422	0.718
FALSE	TRUE	TRUE	TRUE	TRUE	0.806	0.727	0.865	0.889	0.466	0.840	0.429	0.718
TRUE	TRUE	FALSE	TRUE	TRUE	0.793	0.753	0.865	0.880	0.476	0.832	0.422	0.717
	FALSE	TRUE	TRUE	TRUE	0.802	0.751	0.855	0.856	0.494	0.836	0.421	0.716
FALSE	FALSE	FALSE	TRUE	TRUE	0.785	0.732	0.866	0.873	0.488	0.828	0.415	0.713
		TRUE	TRUE	TRUE	0.795	0.730	0.862	0.855	0.475	0.831	0.414	0.709
TRUE	FALSE	TRUE	TRUE	FALSE	0.785	0.737	0.852	0.855	0.439	0.822	0.412	0.700
FALSE	TRUE	TRUE	TRUE	FALSE	0.783	0.707	0.855	0.880	0.420	0.823	0.417	0.698
TRUE	TRUE	TRUE	FALSE	FALSE	0.793	0.707	0.840	0.877	0.432	0.821	0.403	0.696
		FALSE	FALSE	FALSE	0.791	0.707	0.839	0.857	0.437	0.818	0.400	0.693
FALSE	TRUE	FALSE	TRUE	FALSE	0.753	0.709	0.851	0.880	0.431	0.811	0.412	0.692
TRUE	TRUE	TRUE	TRUE	FALSE	0.785	0.732	0.851	0.872	0.382	0.823	0.401	0.692
FALSE	FALSE	TRUE	TRUE	FALSE	0.774	0.710	0.855	0.851	0.425	0.816	0.407	0.691
		FALSE	TRUE	FALSE	0.749	0.711	0.851	0.866	0.435	0.807	0.407	0.689
TRUE	FALSE	FALSE	TRUE	FALSE	0.743	0.736	0.848	0.861	0.427	0.807	0.398	0.688
		TRUE	FALSE	FALSE	0.777	0.707	0.836	0.834	0.433	0.814	0.392	0.685
FALSE	TRUE	TRUE	FALSE	TRUE	0.792	0.691	0.831	0.863	0.406	0.802	0.390	0.682
TRUE	FALSE	FALSE	FALSE	FALSE	0.776	0.704	0.834	0.822	0.431	0.813	0.393	0.682
FALSE	TRUE	FALSE	FALSE	TRUE	0.782	0.692	0.827	0.852	0.423	0.795	0.388	0.680
TRUE	TRUE	FALSE	FALSE	TRUE	0.770	0.701	0.828	0.860	0.434	0.791	0.375	0.680
			TRUE	FALSE	0.744	0.732	0.843	0.861	0.383	0.805	0.390	0.680
		TRUE	FALSE	TRUE	0.764	0.699	0.825	0.864	0.417	0.789	0.375	0.676
FALSE	TRUE	TRUE	FALSE	FALSE	0.768	0.678	0.832	0.852	0.367	0.795	0.387	0.668
	FALSE	TRUE	FALSE	TRUE	0.773	0.687	0.821	0.818	0.406	0.791	0.374	0.667
		FALSE	FALSE	TRUE	0.767	0.688	0.816	0.811	0.413	0.786	0.371	0.665
TRUE	FALSE	FALSE	FALSE	TRUE	0.755	0.696	0.816	0.812	0.422	0.787	0.365	0.665
FALSE	TRUE	FALSE	FALSE	FALSE	0.754	0.681	0.823	0.823	0.372	0.790	0.383	0.661
TRUE	FALSE	TRUE	FALSE	TRUE	0.750	0.694	0.814	0.802	0.415	0.785	0.364	0.660
FALSE	FALSE	TRUE	FALSE	FALSE	0.754	0.678	0.830	0.816	0.369	0.787	0.374	0.658
		FALSE	FALSE	FALSE	0.743	0.679	0.819	0.799	0.369	0.784	0.373	0.653

Acknowledgements We thank the machine learning community for providing many free resources for our research. Work by Dr. Achim Klein was partially supported by Bundesministerium für Wirtschaft und Energie (Grant number 03EGSBW498).

Author contributions MR: writing—original draft, review and editing, conceptualization, methodology, investigation, formal analysis, data curation, software, validation. MR: writing—original draft, review and editing, conceptualization, methodology, formal analysis, validation. AK: writing—original draft, review and editing, methodology.

Funding Open Access funding enabled and organized by Projekt DEAL. Dr. Achim Klein acknowledges funding via the project Crypto-Captain (BMWi). There were no other third-party sources of funding for the research reported.

Data availability The datasets analyzed during the this study are available in the https://github.com/zhangxiangxiao/Crepe and https://ai.stanford.edu/~amaas/data/sentiment/ repository.

Compliance with Ethical Standards

Conflict of interests Martin Riekert declares that he has no conflict of interest. Matthias Riekert declares that he has no conflict of interest. Achim Klein declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Samuel J, Ali GGMN, Rahman MM, Esawi E, Samuel Y. COVID-19 public sentiment insights and machine learning for tweets classification. Information. 2020;11:1–23.
2. Sebastiani F. Machine learning in automated text categorization. ACM Comput Surv. 2002;34:1–47.
3. Mitchell TM. Machine learning, vol. 45, No. 37. Burr Ridge, IL: McGraw Hill; 1997. p. 870–7.
4. Cortes C, Jackel LD, Solla SA, Vapnik V, Denker JS. Learning curves: asymptotic values and rate of convergence. In: 6th International conference on neural information processing system, vol. 6, pp 327–334, 1994
5. Cawley GC, Talbot NLC. On over-fitting in model selection and subsequent selection bias in performance evaluation. J Mach Learn Res. 2010;11:2079–107.
6. Bottou L, Curtis FE, Nocedal J. Optimization methods for large-scale machine learning. SIAM Rev. 2016;60:223–311.
7. Tetlock PCP, Content G, Sentiment I, Role T, Author SM, Source PCT, Journal T. Giving content to investor sentiment: the role of media in the stock market. J Finance. 2007;62:1139–68.
8. Hartmann J, Huppertz J, Schamp C, Heitmann M. Comparing automated text classification methods. Int J Res Mark. 2019;36:20–38.
9. Stone PJ, Bales RF, Namenwirth JZ, Ogilvie DM. The general inquirer: a computer system for content analysis and retrieval based on the sentence as a unit of information. Behav Sci. 2007;7:484–98.
10. Henry E. Are investors influenced by how earnings press releases are written? J Bus Commun. 2008;45:363–407.
11. Loughran T, McDonald B. Textual analysis in accounting and finance: a survey. J Acc Res. 2016;54:1187–230. https://doi.org/10.1111/1475-679X.12123.
12. Wang S, Manning CD. Baselines and bigrams: simple, good sentiment and topic classification. In: Proceedings of the 50th annual meeting of the association for computational linguistics, vol. 2. Jeju, South Korea, pp 90–94, 2012
13. Tang H, Tan S, Cheng X. A survey on sentiment detection of reviews. Expert Syst Appl. 2009;36:10760–73.
14. Zhang X, Zhao J, LeCun Y. Character-level convolutional networks for text classification. In: Proceedings of the 28th International Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press; 2015. p. 649–57. https://doi.org/10.5555/2969239.2969312.
15. Klein A, Riekert M, Kirilov L, Leukel J. Increasing the explanatory power of investor sentiment analysis for commodities in online media. Lect Notes Bus Inf Process. 2018;320:321–32.
16. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. 2018. arXiv:1810.04805.
17. Howard J, Ruder S. Universal language model fine-tuning for text classification. In: 56th Annual Meeting of the Association for Computational Linguistics. 2019. p. 328–39. https://www.aclweb.org/anthology/P18-1031/.
18. Usherwood P, Smit S. Low-shot classification: a comparison of classical and deep transfer machine learning approaches. 2019. arXiv:1907.07543.
19. Büyüköz B, Hürriyetoğlu A, Özgür A. Analyzing ELMo and DistilBERT on socio-political news classification. In: Proceedings of the workshop on automated extraction of socio-political events from news. 2020, pp. 9–18
20. Kou G, Yang P, Peng Y, Xiao F, Chen Y, Alsaadi FE. Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. Appl Soft Comput J. 2020;86:105836.
21. Abdelwahab O, Bahgat M, Lowrance CJ, Elmaghraby A. Effect of training set size on SVM and Naïve Bayes for Twitter sentiment analysis. In: 2015 IEEE International symposium on signal processing and information technology (ISSPIT). 2016, pp. 46–51
22. Choi Y, Lee H. Data properties and the performance of sentiment classification for electronic commerce applications. Inf Syst Front. 2017;19:993–1012.
23. Figueroa RL, Zeng-Treitler Q, Kandula S, Ngo LH. Predicting sample size required for classification performance. BMC Med Inform Decis Mak. 2012;12:8.
24. Meek C, Thiesson B, Heckerman D. The learning-curve sampling method applied to model-based clustering. J Mach Learn Res. 2002;2:397–418.
25. Manning CD, Raghavan P, Schütze H. Introduction to information retrieval. Cambridge: Cambridge University Press; 2008.
26. Tsytsarau M, Palpanas T. Survey on mining subjective data on the web. Data Min Knowl Discov. 2011;24:478–514.
27. Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C. Learning word vectors for sentiment analysis. In: ACL-HLT 2011 Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, vol. 1, 2011, pp. 142–150
28. Riekert M, Leukel J, Klein A. Online media sentiment: Understanding machine learning-based classifiers. In: 24th European conference on information systems. 2016
29. Joachims T. Learning to classify text using support vector machines. Norwell: Kluwer Academic Publishers; 2002.
30. Kennedy A, Inkpen D. Sentiment classification of movie reviews using contextual valence shifters. Comput Intell. 2006;22:110–25.
31. Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Inf Process Manag. 1988;24:513–23.
32. Paltoglou G, Thelwall M. A study of Information Retrieval weighting schemes for sentiment analysis. In: 48th Annual meeting of the association for computational linguistics. 2010, pp. 1386–1395
33. O'Keefe T, Koprinska I. Feature selection and weighting methods in sentiment analysis. In: 14th Australasian document computing symposium. 2009, pp. 67–74
34. Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of conference on empirical methods of Nat Lang Process, Philadelphia, PA, USA, 2002, pp. 79–86
35. Zipf GK. Human behavior and the principle of least effort. Eastford: Martino Publishing; 1949.
36. Robertson S. Understanding inverse document frequency: on theoretical arguments for IDF. J Doc. 2004;60:503–20.
37. Joachims T. Text categorization with support vector machines: learning with many relevant features. In: Proceedings of 10th European conference on machine learning Chemnitz, Germany, 1998, pp. 137–142
38. Ng V, Dasgupta S, Arifin N. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In: Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics. 2006, pp. 611–618
39. Boser B, Guyon I, Vapnik V. A training algorithm for optimal margin classifiers. In: 5th Annual ACM workshop on computational learning theory. 1992, pp. 144–152
40. McCallum A, Nigam K. A comparison of event models for naive bayes text classification. In: 15th National conference on artificial

intelligence of working, learning and text category. 1998, pp. 41–48
41. Wang Z, Lin Z. Optimal feature selection for learning-based algorithms for sentiment classification. Cognit Comput. 2020;12:238–48.
42. Pedregosa F, Grisel O, Weiss R, et al. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12:2825–30.
43. Fan R, Chang K, Hsieh C. LIBLINEAR: a library for large linear classification. J Mach Learn Res. 2008;9:1871–4.
44. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. Int Jt Conf Artif Intell. 1995;5:1–7.
45. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning. 2nd ed. New York: Springer; 2009.
46. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521:436–44.
47. Tang Z, Li W, Li Y. An improved term weighting scheme for text classification. Concurr Comput. 2020;32:1–19.
48. Deng X, Li Y, Weng J, Zhang J. Feature selection for text classification: a review. Multimed Tools Appl. 2019;78:3797–816.
49. Kim SY. Effects of sample size on robustness and prediction accuracy of a prognostic gene signature. BMC Bioinform. 2009;10:4–7. https://doi.org/10.1186/1471-2105-10-147.