Early Forecasting of Text Classification Accuracy and F-Measure with Active Learning

Thomas Orth
Department of Computer Science
The College of New Jersey
Ewing, NJ 08628
Email: ortht2@tcnj.edu

Michael Bloodgood
Department of Computer Science
The College of New Jersey
Ewing, NJ 08628
Email: mbloodgood@tcnj.edu

***Abstract*— When creating text classification systems, one of the major bottlenecks is the annotation of training data. Active learning has been proposed to address this bottleneck using stopping methods to minimize the cost of data annotation. An important capability for improving the utility of stopping methods is to effectively forecast the performance of the text classification models. Forecasting can be done through the use of logarithmic models regressed on some portion of the data as learning is progressing. A critical unexplored question is what portion of the data is needed for accurate forecasting. There is a tension, where it is desirable to use less data so that the forecast can be made earlier, which is more useful, versus it being desirable to use more data, so that the forecast can be more accurate. We find that when using active learning it is even more important to generate forecasts earlier so as to make them more useful and not waste annotation effort. We investigate the difference in forecasting difficulty when using accuracy and F-measure as the text classification system performance metrics and we find that F-measure is more difficult to forecast. We conduct experiments on seven text classification datasets in different semantic domains with different characteristics and with three different base machine learning algorithms. We find that forecasting is easiest for decision tree learning, moderate for Support Vector Machines, and most difficult for neural networks.**

I. Introduction

Text classification has been used in many different applications and is an important task in semantic computing [1], [2], [3], [4], [5]. Using machine learning yields text classification systems with high performance, however, the major bottleneck in constructing new text classification systems is the cost of producing the training data. There has been a great deal of interest in reducing the annotation bottleneck for constructing new text classification systems through the use of active learning [6], [7], [8]. Active learning works by having the learner actively select the data that will be labeled with the goal of optimizing learner efficiency by requesting labeling effort where it is expected to be most useful [9], [10], [11], [12], [13]. To realize the potential benefits of active learning, it is crucial to stop the learning process when additional labels will no longer be useful. Determining when to stop active learning is an area of active research [14], [15], [16], [17], [18], [19], [20], [21].

A related area of interest is to devise methods that can predict, or forecast, the performance of a machine learning model during learning. Accurate performance forecasting can improve our ability to determine when to stop seeking additional labeled data during active learning. Model forecasting can be done by performing regression on the performance of a machine learning model as more data is given to it. Prior work has shown the learning curve of a machine learning model has a shape similar to that of certain families of equations [22], [23].

Figure 1 shows an example learning curve. Some part of the data is needed to create the forecasting model. The amount of points used to create the forecaster is determined by a Training Percent Cutoff (TPC). However, it is an open question where a good TPC would be. In past work, 15% has emerged as a pseudo-standard for setting the value of TPC [22], [23]. However, setting the TPC at 15% might gather more labeled data than is necessary, wasting annotation effort. Figure 2 illustrates this with a hypothetical stopping point, shown by the leftmost vertical line in the figure. In this case the cutoff of 15% would be wasteful of annotations because we would want to have stopped learning before we even are able to create the forecast that is supposed to help us determine when to stop learning. Section IV shows actual stopping points for text classification, using a state of the art stopping method for active learning, are often well before 15% of the data has been annotated.

In this paper, we explore the impact of different values for the TPC to see how early we can forecast performance without losing a lot of accuracy in our forecasting. We also compare the TPC value to the stopping percent found by a leading state of the art stopping method for active learning. We found that a smaller TPC could be used for developing forecasting models than the 15% that is currently widely used. While this earlier forecasting is an improvement, we find that forecasting models are still not effective until too late compared to the stopping percent determined by stopping methods. This indicates that further research to make even earlier forecasting more accurate would be productive. We investigate forecasting model performance in terms of accuracy and F-Measure in section IV-B and find that F-measure is significantly harder to forecast than accuracy. We explore the impact of using different batch percents as compared to using just 1% and find that it makes little difference to forecasting

This paper was published in the Proceedings of the 2020 IEEE 14th International Conference on Semantic Computing (ICSC), pages 77-84, San Diego, CA, USA, 2020. ©*2020 IEEE*
Link to article abstract in IEEE Xplore: https://doi.org/10.1109/ICSC.2020.00018

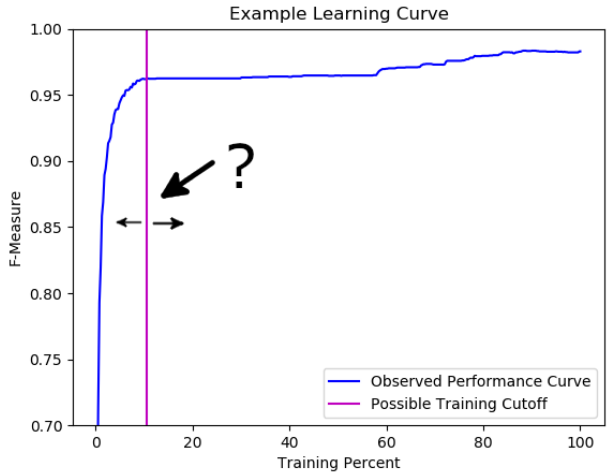

Fig. 1: Example Learning Curve showing the uncertainty of what value to use for the TPC

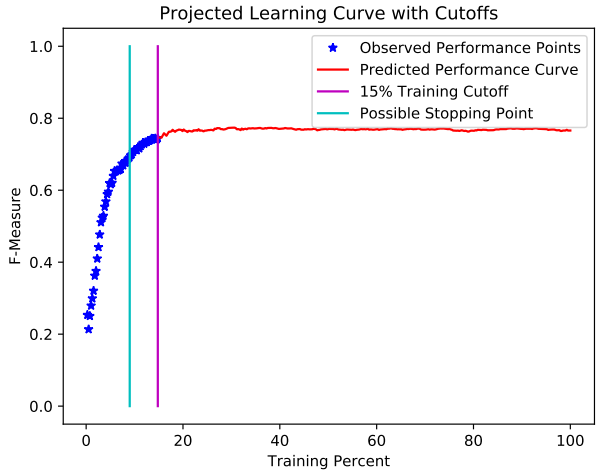

Fig. 2: Example curve demonstrating the prediction of performance using regression

capabilities. Additionally, we compare different base machine learning models and find that neural networks are more difficult to forecast than decision trees and SVMs (Support Vector Machines) are of medium difficulty to forecast. Finally, we compare passive and active learning and find that it is harder to forecast performance when using active learning.

II. Related Work

There has been a lot previous work in the area of using forecasting models to forecast the performance of base learners. Of these models, linear, exponential, logarithmic and power were the most popular. Another model was proposed by Weiss and Tian, which has no name so hereafter will be referred to as the Weiss and Tian model [24].

There has been some research into creating systems that use specific parameters related to a base machine learning model in order to forecast performance. Past work has explored using hyperparameters of bayesian neural networks to forecast accuracy [25], [26]. This methodology does not work for our setting. We forecast the performance of machine learning models based off the training data.

Other systems forecast the performance of machine learning models using task specific information. For the task of machine translation, past work has investigated forecasting performance by using a feature vector of information such as average sentence length of the test set [27]. These methods utilize a lot of properties that are specific to machine translation, which cannot easily be adapted to other NLP tasks such as text classification.

There have been systems that use training set percentage to forecast machine learning model performance. This was first done by Frey and Fisher in 1999 by forecasting decision trees [22]. Frey and Fisher used 15% of the points of a learning curve to train their forecasting models and tested the model on the other 85%. They came to the conclusion that power law was the best way to forecast a learning curve. We explore whether the TPC can be varied and find that in practical active learning situations a TPC less than 15% would be desired.

Singh investigated forecasting further and used different machine learning models in his experiments and found that logarithmic models worked better for forecasting than the power models [23]. Hence, we focus on logarithmic models in our paper. Finally, there has been some work from other areas such as using projective sampling to reduce the total cost of data mining that looks at using different amounts of training points for forecasting machine learning performance [28]. We investigate this as well, but with a more fine-grained examination of how the TPC can be varied for forecasting text classification performance as measured by different performance metrics with both active learning and passive learning.

There has been some prior work in predicting performance in active learning. Figueroa et al. performed a comparison between passive and active learning for predicting performance [29]. In our experiments, we not only compare passive and active learning, but we also examine how much data to use for the forecasting process.

There has been some prior work in stopping active learning using mathematical guidance on how much performance can be expected to change. For example, some past work has investigated how mathematical bounds on the amount of possible change in F-Measure from iteration to iteration can be used to stop the training process during active learning [15], [21]. Their method doesn't require labeled data to measure performance. In contrast, our method in the current paper requires some labeled data to obtain the initial points we use to regress our models. However, our models can then forecast levels of performance in addition to only changes. Future work includes developing algorithms to combine the mathematical bounds approach of [15], [21] with the regression approach in the current paper.

III. Experimental Setup

We now will describe our experimental setup. All of our experiments are conducted in an iterative learning setting, which we describe in section III-A.

A. Iterative Learning Setup

We use the 20NewsGroups dataset[1], the Reuters dataset, in particular the Reuters-21578 Distribution 1.0 ModApte split[2] as done in [30] and [31], the WebKB dataset[32], the spamassassin corpus[33], the IMDB sentiment dataset[3], TrecSpam 2005 ham25[34], and the first 20000 entries of Ohsumed[4] for our experiments. We report the results for the four largest categories of the WebKB dataset as done in past work [32], [20], [17]. We used 10-fold cross validation and present the averages for SpamAssassin, WebKB, Trec and Ohsumed. For the other datasets, we used the standard train-test split provided by the dataset. For text classification, we use a bag of words approach with a frequency cutoff of three, meaning that each feature is a word and we don't create features for words that occur fewer than three times. We use binary feature values, meaning the value of the feature is a 1 if the feature (word) occurs in the document and 0 if the feature (word) does not occur in the document. There are also words that hold very little to no value for classification, called stop words. We remove stop words that appear in the *Long Stopword List* from https://www.ranks.nl/stopwords. We use SVM, decision tree, and multi-layer perceptron neural network as our main classifiers. For SVM, we use a linear kernel. For our neural network, we use a densely connected layer for the input layer with 64 hidden units, a dropout layer with 20% dropout and another densely connected layer for the output with a hidden unit. We used a passive selection algorithm which chooses random samples with all three base learners. We also use the closest-to-hyperplane selection algorithm with SVM for active learning [35], [36], [16]. This is because previous work has shown that it has better performance over other selection algorithms used [37]. For each iteration of training, the number of samples used is determined by a batch percent, bp. The bp percent of the total amount of unlabeled data originally available is the amount of data that will be added to the labeled training data during each iteration of learning. We use different batch percents in our experiments. We first used 1.0% to compare to Frey and Fischer [22]. We also used 0.25% to test whether more fine-grain sampling would have any impact on forecasting performance.

7 datasets

B. Overview of Predicting Learning Curves

Figure 3 shows an example, using the described setup to forecast the performance of a decision tree on our Ohsumed dataset, where in this case our performance metric is accuracy. Linear, Weiss and Tian, and exponential were the least accurate when used to forecast performance. Power and logarithmic models do the best. The equations are shown in Table I. In past literature, logarithmic was found to be the best forecaster [23]. In our experiments, we also observed logarithmic to be the best forecasting model. Therefore, for all the rest of the experiments in this paper, we use logarithmic as our forecasting model.

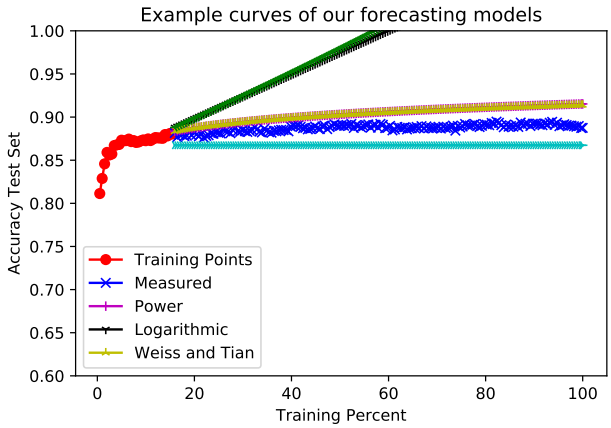

Fig. 3: Example curves of the different forecasting models

Name	Equation
Linear	$y = ax + b$
Power	$y = a * x^b$
Logarithmic	$y = alog(x) + b$
Exponential	$y = a10^{bx}$
Weiss and Tian	$y = a + bx/(x + 1)$

TABLE I: List of equations used

C. Forecasting Performance Setup

Once we run the iterative learning process described in Section III-A and record the performance of the learned classifier on held-out test data at each iteration, we then process all of the data to forecast the performance of the machine learning models used. We use the equations specified in Table I to forecast on the given data. Notice in the equations the variables y, x, a, and b. The first variable, y, is our classification performance metric. We experimented with two performance metrics, Accuracy and F-Measure[5] The next variable, x, is the parameter from the data we collected that we are using to perform our prediction. For our experiments, we used the training percent for the current iteration, consistent with past work [22], [23]. The final two parameters, a and b, are the learned coefficients from performing regression on the given data.

[1]Downloaded the "bydate" version from http://qwone.com/~jason/20Newsgroups/. This version does not include duplicate posts and is sorted by date into train and test sets.

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[3]http://ai.stanford.edu/~amaas/data/sentiment

[4]Downloaded from http://disi.unitn.it/moschitti/corpora.htm on July 13, 2017

[5]Both Accuracy and F-Measure are commonly used performance metrics for evaluating text classification performance. Accuracy is the percentage of classifications that are correct, while F-Measure is the harmonic mean of Precision and Recall, with Precision defined as the percentage of predicted positive instances that are truly positive instances and Recall defined as the percentage of truly positive instances that are predicted as positive instances.

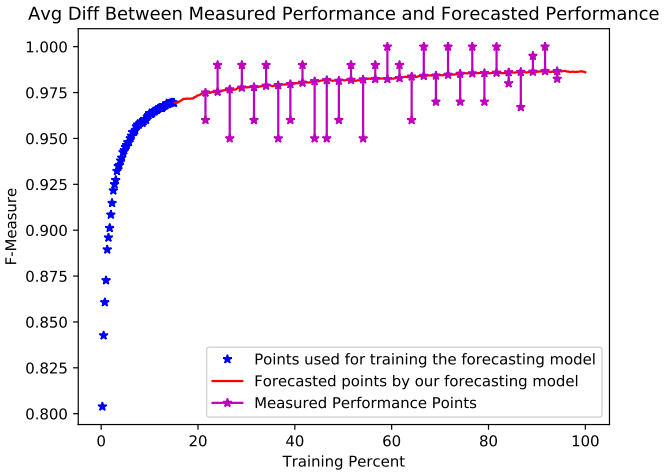

Fig. 4: Illustration of Average Difference

In order to evaluate the performance of forecasting models, we define a measurement, which we call Average Difference, that captures how much the forecasted values differ on average from the observed values. Average Difference is defined in equation 1 below.

$$\text{Average Difference} = \frac{\sum_{i=1}^{n} |f(x_i) - y_i|}{n} \quad (1)$$

where f is the forecasting function, x_i is the training percent at the i^{th} test point, y_i is the observed performance at the i^{th} point, and n is the number of test points, as defined in equation 2 below.

$$n = \frac{100 - TPC}{bp} \quad (2)$$

where TPC is the Training Percent Cutoff and bp is the batch percent, both as defined earlier in this paper.

Average Difference is illustrated in Figure 4. The points lying around our predicted curve are example measured points. Each point is of the form (x_i, y_i). We use equation 1. We go over all points from $i = 1$ to n, with x_1 being the x-coordinate of the first point after the TPC. We take the difference of the forecasted performance and the observed performance at each point and average these differences. When the Average Difference measurement is smaller, that means our forecasting model is performing better.

IV. Results

This section discusses the results of our experiments. We show the impact of batch percent on forecasting performance, we compare the performance of forecasting classification performance in terms of Accuracy versus in terms of F-Measure, we analyze varying the TPC, we show how the choice of base learner impacts forecasting, and finally we show the difference between forecasting in a passive learning setting versus in an active learning setting.

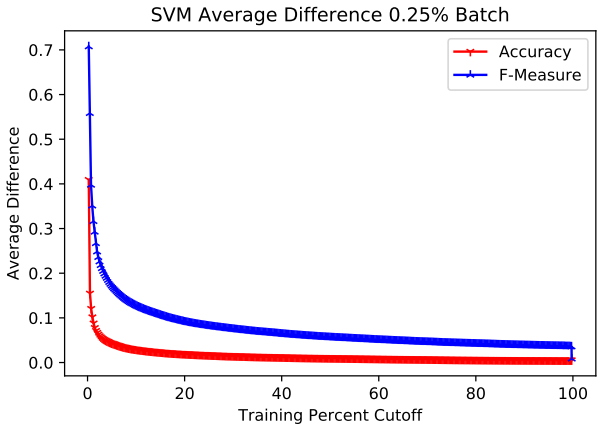

Fig. 5: Quality of forecasting (as measured by Average Difference) for varying TPC values when text classification performance is measured in terms of Accuracy and in terms of F-Measure. A lower Average Difference means a higher quality forecast.

A. Impact of Batch Percent

We performed a comparison of forecasting performance when batch percent is 0.25% versus when batch percent is 1%, by computing the Average Difference as defined in equation 1. We compute an overall average difference by averaging the individual average differences over all datasets and base machine learning models. We use 15% as our TPC as it is a commonly used TPC [22], [23]. For Accuracy, the overall average difference was 0.0256 for 0.25% batch percent and 0.0208 for 1.0%. We can see that there is not much difference between the change in batch percent for accuracy. For F-Measure, the overall average difference was 0.167 for 0.25% batch percent and 0.129 for 1.0% batch percent. Again, we see there is not much difference in performance of our forecasting system when we vary batch percent. It is possible that with much larger batch percents we would see a change in forecasting performance, but using larger batch percents is known to have various negative effects on active learning [38], [39], so we did not investigate the impact with larger batch percents that are less likely to be used in practice.

B. Accuracy vs. F-Measure

In this section we compare forecasting when text classification performance is measured in terms of Accuracy versus in terms of F-Measure. In these experiments we vary the TPC to go through all possible TPC values. Figure 5 shows the overall average difference for Accuracy and F-Measure using SVM as the base learner over all datasets using 0.25% as the batch percent. The results are compelling: Accuracy has a much lower average difference than F-Measure. This shows that new forecasting methods are needed when classification performance is measured in terms of F-Measure.

C. TPC Analysis

Informally it is expected that forecasting is more useful if it can be done earlier in the iterative training process, however, it is also expected to be more difficult to create high quality forecasts earlier in the process. In this section, we examine these issues in detail, with experiments illuminating more specifically the value of forecasting by certain points in the iterative learning process and the expected changes of the quality of the forecasts due to changes in when the forecast is created. Specifically, we experiment with changing the TPC. Frey and Fisher used 15% for the TPC [22]. There was no analysis done of possibly changing the TPC. We use a 0.25% batch percent for the experiments in this section.

Figure 5 shows the Average Difference using SVM as the base learner and varying the TPC from 0.25% of our data to 99.75% of our data to show a fine-grained look at the impact of changing the TPC on forecast quality when Accuracy is used as the classification performance metric and when F-Measure is used as the classification performance metric. Figure 5 shows that as TPC is increased, our forecasting quality improves, or in other words, our average difference gets smaller. However, we can see that the rate of improvement in forecasting quality is very different at different points in the iterative learning process, or in other words, at different TPC values. In particular, there is a very sharp improvement in forecasting quality up to about ten percent TPC and then the rate of improvement is much slower, with forecasts improving only by small amounts for larger settings of the TPC. This shows that the TPC can potentially be pushed back a bit lower than 15% without sacrificing too much forecast quality, especially for Accuracy. For F-Measure, the shape is not as much of an elbow dip, but as discussed in section IV-B, current forecasting methods don't work well for F-Measure and are in need of improvement.

Table II shows the stopping points automatically determined during active learning for all of our datasets. These results were obtained by using the state-of-the-art stopping method for active learning described in [14], hereafter referred to as the Stabilizing Predictions (SP) method. The stopping point percents in Table II were determined using an active learning (or in other words, selective sampling) setting with SVM as the base learner and closest-to-hyperplane sampling as the selection algorithm. Figure 6 shows the situation for the TREC dataset. In Figure 6 the TPC is reduced to 10% from the previously used 15% since our results showed it could be pushed back to about 10% without sacrificing large amounts of forecast quality. However, the stopping percent is even smaller than this reduced TPC, showing that it would be practically valuable to develop new forecasting methods that can forecast with higher quality earlier than the current state-of-the-art approach.

D. Impact of Base Learner on Forecasting Performance

This section shows the impact of the base learner (SVM, decision tree, neural network) used during iterative learning. Results are only shown for Accuracy as F-Measure curves represented similar results. Figure 7 shows the overall average difference of the forecasts for the different base machine learning models for varying TPC values. As shown, decision tree classifiers are the easiest to forecast, neural network classifiers are the hardest to forecast, and SVM classifiers are in the middle.

Dataset	Stopping Percent
20NewsGroups	5.922
Reuters	4.795
Ohsumed	11.824
SpamAssassin	5.109
Trec	2.605
WebKB	11.765
IMDB	15.996

TABLE II: Stopping Percents automatically determined during active learning by using the Stabilizing Predictions (SP) method from [14]

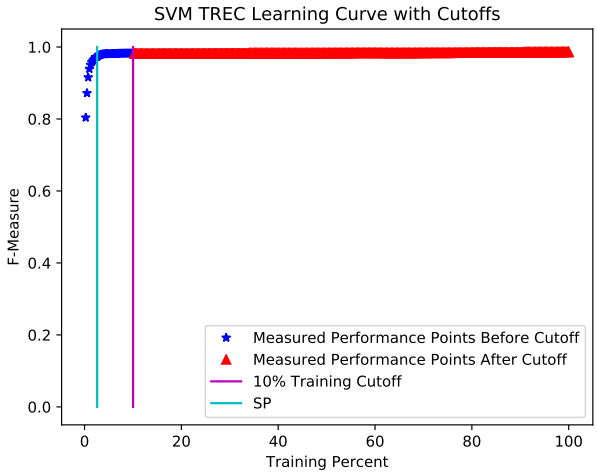

Fig. 6: Learning Curve using active learning with SVM and closest-to-hyperplane sampling on the TREC dataset. The TPC is set to 10%, about the earliest the current state-of-the-art can be set to without sacrificing large amounts of forecast quality. The stopping percent automatically determined during active learning by the Stabilizing Predictions (SP) method from [14] is shown by the SP vertical line.

E. Impact of Passive Learning vs Active Learning

The results in this section show how well forecasting can be done in a passive learning setting versus in an active learning setting. For passive learning, we randomly select the next batch of data to be labeled at each iteration of the iterative learning process described in section III-A. This is the standard setting under which most forecasting methods have been developed and tested [23], [22].

For active learning, an algorithm actively selects the next batch of data it wants to have labeled at each iteration of the iterative learning process. The idea is that by selectively sampling the examples the algorithm expects to be most

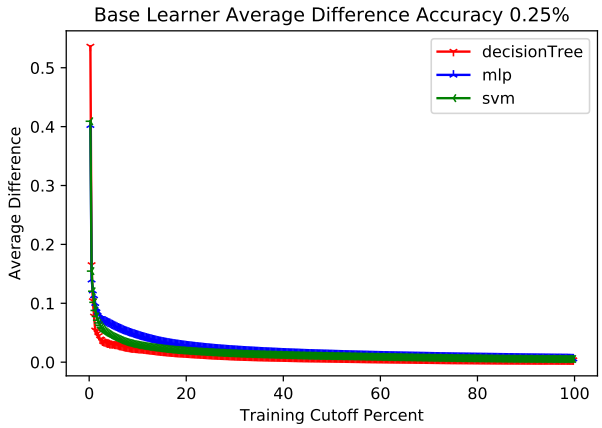

Fig. 7: Overall Average Difference over all datasets when classification performance is measured by Accuracy for different base learners. Decision tree classifiers are the easiest to forecast, neural network classifiers are the hardest to forecast, and SVM classifiers are in the middle.

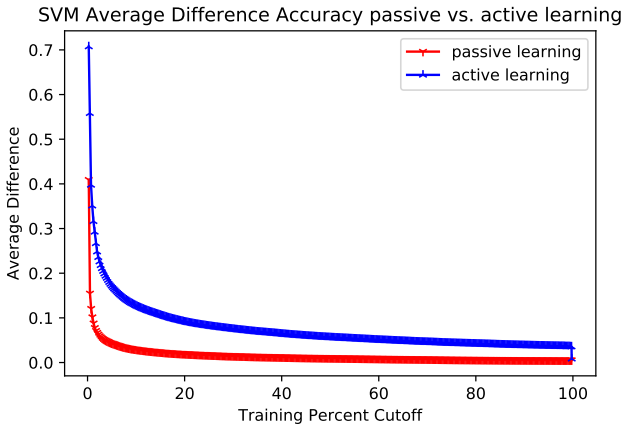

Fig. 8: Overall Average Difference over all datasets when classification performance is measured by Accuracy for SVM base learner in a passive learning setting (random selection of examples at each iteration) and an active learning setting (closest-to-hyperplane selection of examples at each iteration). The results show that current forecasting methods work much better in a passive learning setting. Lower Average Difference means higher quality forecast.

valuable to have labeled, an effective model will be able to be learned from smaller amounts of data, thereby reducing data labeling cost. Since forecasting is intended to be used to help provide guidance on when to stop labeling additional data so that data labeling efforts are not wasted, it is a natural fit that forecasting could be of particular value and interest in active learning settings. However, investigations of forecasting effectiveness in active learning settings have been limited.

We have already seen that SVM is the middle base learner in terms of forecasting difficulty. Furthermore, active learning has been well studied with SVMs and a well known successful algorithm is to sample the examples that are closest to the current model's learned hyperplane as was discussed in section III-A. For these reasons the results we present in this section are for SVM with passive learning versus for SVM with active learning as implemented by the closest-to-the-hyperplane selection algorithm. Also, all results in this section are for forecasting classification performance in terms of Accuracy since forecasting performance in terms of F-Measure is an area in need of future work.

Figure 8 shows compelling results: current state-of-the-art forecasting methods perform much better when using passive learning than when using active learning. To see why this is the case, we show the learning curves for each setting. Figure 9 shows the learning curves for the 20NewsGroups dataset when using passive learning and active learning. The active learning curve has a different shape, deviating from the shape of a logarithmic curve. Because of this, it's harder to forecast the performance of SVM with active learning by assuming that a learning curve shape follows the shape of a logarithmic function. This shows the need for improving the state-of-the-art so that we can forecast effectively in active learning settings. Future work that could be promising for accomplishing this includes developing algorithms to combine

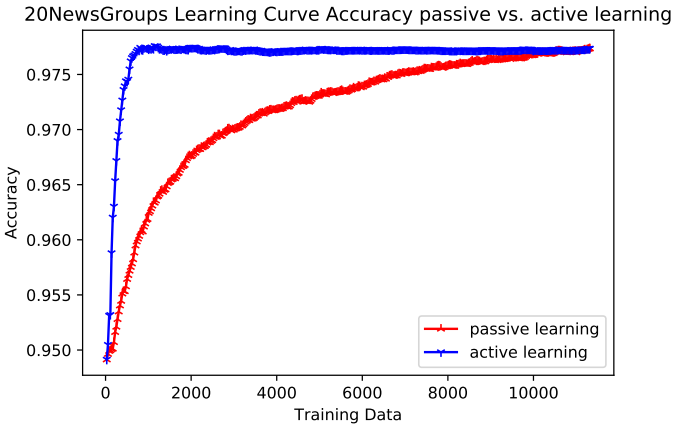

Fig. 9: Learning curves for 20NewsGroups dataset when classification performance is measured by Accuracy for SVM base learner for passive learning (random selection of examples at each iteration) and active learning (closest-to-hyperplane selection of examples at each iteration). The active learning curve deviates from a logarithmic shape making it difficult for existing state-of-the-art forecasting methods to generate high quality forecasts in active learning settings.

the mathematical bounds approach of [15], [21] with the regression approach in the current paper.

V. Conclusion & Future Work

An area of interest in text classification is being able to forecast the performance of base learners in an iterative learning process. Past work has shown that forecasting models can be developed by regressing on a subset of the data that

occurs before a cutoff we refer to as the TPC and forecasting on the rest of the data. A critical question is what TPC to use, which controls how early a forecast can be developed. In past work forecasts have been developed with a TPC of fifteen percent of the data. We show in this paper that earlier forecasting would be beneficial. In many cases for text classification, forecasts can be developed with a TPC between ten percent and fifteen percent of the data. However, analysis with active learning and stopping methods for active learning revealed that even earlier forecasting is still desired. We also found that forecasting is more difficult with some base learners than others, with decision tree classifiers being forecast the easiest, with SVM classifiers being in the middle, and neural network text classifiers being the hardest to forecast. We also found that using active learning algorithms made it harder to forecast due to the shape of the learning curve not matching current state of the art forecasting methods' expectations about the shape of learning curves. Finally, we found that forecasting performance is more difficult with some performance metrics than others. In particular, we found that forecasting performance measured by accuracy is much easier than forecasting performance measured by F-measure. Future work includes devising methods for even earlier forecasting of performance that are more accurate than the forecasting models currently used and better integrating forecasting methods with active learning stopping methods.

Acknowledgment

This work was supported in part by The College of New Jersey Support of Scholarly Activities (SOSA) program. The authors acknowledge use of the ELSA high performance computing cluster at The College of New Jersey for conducting the research reported in this paper. This cluster is funded by the National Science Foundation under grant number OAC-1828163.

References

[1] A. Mishler, K. Wonus, W. Chambers, and M. Bloodgood, "Filtering tweets for social unrest," in *Proceedings of the 2017 IEEE 11th International Conference on Semantic Computing (ICSC)*. San Diego, CA, USA: IEEE, January 2017, pp. 17–23. [Online]. Available: https://doi.org/10.1109/ICSC.2017.75

[2] S. C. H. Hoi, R. Jin, and M. R. Lyu, "Large-scale text categorization by batch mode active learning," in *Proceedings of the 15th International Conference on World Wide Web*, ser. WWW '06. New York, NY, USA: ACM, 2006, pp. 633–642. [Online]. Available: http://doi.acm.org/10.1145/1135777.1135870

[3] M. Janik and K. J. Kochut, "Wikipedia in action: Ontological knowledge in text categorization," in *2008 IEEE International Conference on Semantic Computing*. IEEE, 2008, pp. 268–275.

[4] M. Allahyari, K. J. Kochut, and M. Janik, "Ontology-based text classification into dynamically defined topics," in *Semantic Computing (ICSC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 273–278.

[5] M. Kanakaraj and R. M. R. Guddeti, "Performance analysis of ensemble methods on twitter sentiment analysis using nlp techniques," in *Semantic Computing (ICSC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 169–170.

[6] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: Springer-Verlag New York, Inc., 1994, pp. 3–12.

[7] M. Bloodgood and K. Vijay-Shanker, "Taking into account the differences between actively and passively acquired data: The case of active learning with support vector machines for imbalanced datasets," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 137–140. [Online]. Available: http://www.aclweb.org/anthology/N/N09/N09-2035.pdf

[8] G. Beatty, E. Kochis, and M. Bloodgood, "The use of unlabeled data versus labeled data for stopping active learning for text classification," in *Proceedings of the 2019 IEEE 13th International Conference on Semantic Computing (ICSC)*. Newport Beach, CA, USA: IEEE, January 2019, pp. 287–294. [Online]. Available: https://doi.org/10.1109/ICOSC.2019.8665546

[9] S. Hantke, Z. Zhang, and B. Schuller, "Towards intelligent crowdsourcing for audio data annotation: Integrating active learning in the real world," *Proc. Interspeech*, pp. 3951–3955, 2017.

[10] M. Bloodgood and C. Callison-Burch, "Bucking the trend: Large-scale cost-focused active learning for statistical machine translation," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 854–864. [Online]. Available: http://www.aclweb.org/anthology/P10-1088

[11] S.-W. Lee, D. Zhang, M. Li, M. Zhou, and H.-C. Rim, "Translation model size reduction for hierarchical phrase-based statistical machine translation," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Jeju Island, Korea: Association for Computational Linguistics, July 2012, pp. 291–295. [Online]. Available: http://www.aclweb.org/anthology/P12-2057

[12] F. Mairesse, M. Gasic, F. Jurcicek, S. Keizer, B. Thomson, K. Yu, and S. Young, "Phrase-based statistical language generation using graphical models and active learning," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 1552–1561. [Online]. Available: http://www.aclweb.org/anthology/P10-1157

[13] A. Miura, G. Neubig, M. Paul, and S. Nakamura, "Selecting syntactic, non-redundant segments in active learning for machine translation," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 20–29. [Online]. Available: http://www.aclweb.org/anthology/N16-1003

[14] M. Bloodgood and K. Vijay-Shanker, "A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*. Boulder, Colorado: Association for Computational Linguistics, June 2009, pp. 39–47. [Online]. Available: http://www.aclweb.org/anthology/W09-1107

[15] M. Bloodgood and J. Grothendieck, "Analysis of stopping active learning based on stabilizing predictions," in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Sofia, Bulgaria: Association for Computational Linguistics, August 2013, pp. 10–19. [Online]. Available: http://www.aclweb.org/anthology/W13-3502

[16] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *Proceedings of the 17th International Conference on Machine Learning (ICML)*, 2000, pp. 839–846.

[17] J. Zhu, H. Wang, and E. Hovy, "Multi-criteria-based strategy to stop active learning for data annotation," in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester, UK, August 2008, pp. 1129–1136. [Online]. Available: http://www.aclweb.org/anthology/C08-1142

[18] F. Laws and H. Schütze, "Stopping criteria for active learning of named entity recognition," in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester, UK, August 2008, pp. 465–472. [Online]. Available: http://www.aclweb.org/anthology/C08-1059

[19] A. Vlachos, "A stopping criterion for active learning," *Computer Speech and Language*, vol. 22, no. 3, pp. 295–312, 2008.

[20] J. Zhu, H. Wang, and E. Hovy, "Learning a stopping criterion for active learning for word sense disambiguation and text classification," in *In Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, 2008, pp. 366–372.

[21] M. Altschuler and M. Bloodgood, "Stopping active learning based on predicted change of f measure for text classification," in *Proceedings of the 2019 IEEE 13th International Conference on Semantic Computing (ICSC)*. Newport Beach, CA, USA: IEEE, January 2019, pp. 47–54. [Online]. Available: https://doi.org/10.1109/ICOSC.2019.8665646

[22] L. J. Frey and D. H. Fisher, "Modeling decision tree performance with the power law." in *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, Florida, 1999.

[23] S. Singh, "Modeling performance of different classification methods: Deviation from the power law," Department of Computer Science, Vanderbilt University, USA, Tech. Rep., April 2005.

[24] G. M. Weiss and Y. Tian, "Maximizing classifier utility when there are data acquisition and modeling costs," *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 253–282, Oct 2008. [Online]. Available: https://doi.org/10.1007/s10618-007-0082-x

[25] T. Domhan, J. T. Springenberg, and F. Hutter, "Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves," in *IJCAI*, vol. 15, 2015, pp. 3460–8.

[26] J. T. S. Aaron Klein, Stefan Falkner and F. Hutter, "Learning curve prediction with bayesian neural networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, Toulon, France, April 2017.

[27] P. Kolachina, N. Cancedda, M. Dymetman, and S. Venkatapathy, "Prediction of learning curves in machine translation," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Jeju Island, Korea: Association for Computational Linguistics, jul 2012, pp. 22–30.

[28] M. Last, "Improving data mining utility with projective sampling," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 487–496. [Online]. Available: http://doi.acm.org/10.1145/1557019.1557076

[29] R. L. Figueroa, Q. Zeng-Treitler, S. Kandula, and L. H. Ngo, "Predicting sample size required for classification performance," *BMC Medical Informatics and Decision Making*, vol. 12, no. 1, 2012.

[30] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *ECML*, ser. Lecture Notes in Computer Science, C. Nedellec and C. Rouveirol, Eds., vol. 1398. Springer, 1998, pp. 137–142.

[31] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in *CIKM '98: Proceedings of the seventh international conference on Information and knowledge management*, Bethesda, Maryland, United States, 1998, pp. 148–155.

[32] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," in *Proceedings of AAAI-98, Workshop on Learning for Text Categorization*, 1998. [Online]. Available: http://www.kamalnigam.com/papers/multinomial-aaaiws98.pdf

[33] D. Sculley, "Online active learning methods for fast label-efficient spam filtering," in *CEAS*, 2007.

[34] G. Cormack and T. Lynam, "Trec 2005 spam track overview," in *TREC-14*, 2005.

[35] C. Campbell, N. Cristianini, and A. J. Smola, "Query learning with large margin classifiers," in *Proceedings of the 17th International Conference on Machine Learni ng (ICML)*, 2000, pp. 111–118.

[36] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research (JMLR)*, vol. 2, pp. 45–66, 2001.

[37] M. Bloodgood, "Support vector machine active learning algorithms with query-by-committee versus closest-to-hyperplane selection," in *Proceedings of the 2018 IEEE 12th International Conference on Semantic Computing (ICSC)*. Laguna Hills, CA, USA: IEEE, January 2018, pp. 148–155. [Online]. Available: https://doi.org/10.1109/ICSC.2018.00029

[38] G. Beatty, E. Kochis, and M. Bloodgood, "Impact of batch size on stopping active learning for text classification," in *Proceedings of the 2018 IEEE 12th International Conference on Semantic Computing (ICSC)*. Laguna Hills, CA, USA: IEEE, January 2018, pp. 306–307. [Online]. Available: https://doi.org/10.1109/ICSC.2018.00059

[39] K. Brinker, "Incorporating diversity in active learning with support vector machines," in *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, T. Fawcett and N. Mishra, Eds. AAAI Press, 2003, pp. 59–66.