Impact of Stop Sets on Stopping Active Learning for Text Classification

Luke Kurlandski
Department of Computer Science
The College of New Jersey
Ewing, New Jersey 08618
Email: kurlanl1@tcnj.edu

Michael Bloodgood
Department of Computer Science
The College of New Jersey
Ewing, New Jersey 08618
Email: mbloodgood@tcnj.edu

Abstract—Active learning is an increasingly important branch of machine learning and a powerful technique for natural language processing. The main advantage of active learning is its potential to reduce the amount of labeled data needed to learn high-performing models. A vital aspect of an effective active learning algorithm is the determination of when to stop obtaining additional labeled data. Several leading state-of-the-art stopping methods use a stop set to help make this decision. However, there has been relatively less attention given to the choice of stop set than to the stopping algorithms that are applied on the stop set. Different choices of stop sets can lead to significant differences in stopping performance. We investigate the impact of different stop set choices on different stopping methods. This paper shows the choice of the stop set can have a significant impact on the performance of stopping methods and the impact is different for stability-based methods from that on confidence-based methods. Furthermore, the unbiased representative stop sets suggested by original authors of methods work better than the systematically biased stop sets used in recently published work, and stopping methods based on stabilizing predictions have stronger performance than confidence-based stopping methods when unbiased representative stop sets are used. We provide the largest quantity of experimental results on the impact of stop sets to date. The findings are important for helping to illuminate the impact of this important aspect of stopping methods that has been under-considered in recently published work and that can have a large practical impact on the performance of stopping methods for important semantic computing applications such as technology assisted review and text classification more broadly.

I. INTRODUCTION

Active learning (AL) is a widely used strategy for reducing data labeling costs. AL has been used for many semantic computing applications within natural language processing (NLP), such as text classification (TC) [1]–[7], named entity recognition (NER) [8], [9], and machine translation (MT) [10]–[12]. Problem settings and recent developments in AL are discussed further in [13].

The essential concept behind AL is for the machine learning (ML) model to take an operational role in deciding which data instances should be labeled for learning. The active learning process begins with an unlabeled pool of data, an initially empty pool of labeled training data, and a batch size. In the first step of AL, the model queries an oracle (a human data labeler) for the labels of an initial batch of training data. Following this, the algorithm enters an iterative process: the model is trained on the existing set of labeled data; the model queries the oracle to label a new batch of training examples; and the labeled and unlabeled sets are adjusted accordingly. This process is shown in Algorithm 1, where $b$ refers to the batch size, $L$ the set of labeled examples, $U$ the unlabeled pool of examples, and $B$ the batch set of examples queried at each iteration of AL.

Algorithm 1 Active Learning Algorithm

```
Require: $U$, $b$ {input to algorithm}
        $L \leftarrow$ select and annotate $b$ random examples from $U$
        $U \leftarrow U - L$
repeat
    Train model using $L$
    $B \leftarrow$ select and annotate $b$ examples from $U$ using the selection algorithm
    $U \leftarrow U - B$
    $L \leftarrow L \cup B$
until stopping criterion is met
```

An essential component to any active learning setup is an effective stopping method that halts the data labeling process. Without a stopping method, the model would request annotations for every training example, defeating the purpose of AL. Past work has used the term “aggressive” to characterize stopping methods that tend to prioritize saving annotations and the term “conservative” to characterize stopping methods that tend to maximize model performance [8]. In general, it is desirable for stopping methods to be:

- capable of reducing the number of labels required without sacrificing final performance,
- capable of adjustment to be more or less aggressive, and
- applicable and consistent across many AL settings, base learners, tasks, and datasets.

A simple stopping method would be to stop after an ad-hoc predetermined labeling budget has been exhausted. However, this runs a large risk of either obtaining a significantly lower-performing model or wasting large amounts of data labeling effort.

Another simple approach would be to use a small set of labeled data to evaluate performance during learning and stop when performance levels off. However, previous work has...
In this paper, we investigate the impact of using the remaining unlabeled pool as a stop set. We experiment with different stop sets and observe that the choice of stop set has a substantial impact on performance for both stability-based stopping and confidence-based stopping. Furthermore, the impact is different depending on whether the stop set is used in a stability-based method or a confidence-based method.

In extensive experiments with multiple freely and publicly available widely used benchmark datasets, which will allow for others to reproduce our experiments or run further experiments, we find that when SP is used with an unbiased representative stop set, the remaining unlabeled pool was used, which is a shrinking and systematically biased stop set.

In this paper, we investigate the impact of using the remaining unlabeled pool as a stop set and various randomly selected stop sets of different sizes. We show that the choice of stop set has a substantial impact on performance for both stability-based stopping and confidence-based stopping. Furthermore, the impact is different depending on whether the stopping method is a stability-based method or a confidence-based method.

In this paper, we investigate the impact of using the remaining unlabeled pool as a stop set and various randomly selected stop sets of different sizes. We show that the choice of stop set has a substantial impact on performance for both stability-based stopping and confidence-based stopping. Furthermore, the impact is different depending on whether the stopping method is a stability-based method or a confidence-based method.
method to investigate because it is widely applicable with few if any restrictions on when it can be used, and it has been found to work well in many settings [8, 14, 19, 24]. The method stops when successively trained models have high agreement in terms of their predictions on the examples in a large randomly selected set of examples. The authors call this set of examples a stop set [8]. In the main experiments, a large randomly selected fixed set of examples is used as the stop set. There is a small section of the paper that explores how stopping performance is impacted if the stop set is made smaller and it’s found that if the set becomes too small the performance could begin to degrade, but it’s concluded that the size required to be representative could vary depending on dataset and task so future work is needed to determine how large the stop set should be. We experiment with this method using different stop sets and measure the impact of the different stop sets on performance. We refer to this method as Stabilizing Predictions (SP).

A theoretical analysis of SP stopping is presented in [15]. An equation for the variance in the agreement estimator between successive models is derived, which shows that the variance is inversely proportional to the size of the stop set. The variance estimator is suggested as a principled way to check stop set size in order to make sure reliable estimates of agreement can be obtained.

Using labeled data versus using unlabeled data to inform the stopping decision is considered in [14]. The tradeoff is that although the labels might give some extra information, that has to be balanced with the fact that using a stop set of labeled data for many applications will by necessity have to be kept relatively small whereas a stop set of unlabeled data can be large. It has been found that stopping methods that use unlabeled data in their decision are more effective than methods that use labeled data.

Two stopping methods are presented in [18]. They are similar to the Declining Confidence (DC) method from [17]. The difference is that whereas the DC method requires a consistent drop in confidence, the methods from [18] require a consistent period where confidence is not increasing, i.e., it could be staying the same or dropping. Another difference is that the methods were implemented with two different stop sets: the batch set, \( B \), and the unlabeled pool, \( U \), instead of a large representative set as suggested by [17]. We refer to these methods as Non-increasing Confidence (NC).

In [18] the performance of the NC methods are compared with current state-of-the-art methods, including SP, on a private technology assisted sensitivity review dataset. The conclusion reached in [18] is that the leading method, SP, stops too aggressively on their private dataset and does not produce a model as well-learned as when using their NC methods. However, it is important to observe that SP was not implemented correctly in the experiments in [18]. Instead of using a large unbiased representative stop set, the remaining unlabeled pool was used as the stop set. We hypothesize that this choice can cause SP to stop more aggressively and confidence-based methods such as DC and NC to stop more conservatively. In the current paper, we show that this is indeed the case and explain the reasons why. We compare the performance of NC with DC, which is not considered in [18], and quantify how much more conservative DC is on several datasets. We also compare the performance of NC with SP implemented in the normal way, which is not considered in [18], and find that then SP has compellingly stronger performance.

III. METHODS FOR STOPPING

In this section, we provide more details about the Stabilizing Predictions, Declining Confidence, and Non-increasing Confidence stopping methods, including details about their usage of stop sets.

A. Stabilizing Predictions

Stabilizing Predictions determines when to stop based upon the agreement between successively trained models [8]. The agreement is computed between the models’ two sets of predictions on a large, randomly selected, fixed stop set of training examples. There are several ways the agreement can be computed. In [8] Cohen’s Kappa statistic [25] is suggested to account for agreement due to chance.

We compute agreement in our implementation of SP as given by equation 1

\[
\text{agreement} = \frac{A_o - A_e}{1 - A_e} \tag{1}
\]

where \( A_o \) is the observed agreement between the two models and \( A_e \) is the agreement expected between the two models by chance. We compute \( A_e \) by the proportion of examples the two models agree upon. We compute \( A_e \) as in equation 2

\[
A_e = \sum_{k \in \{+1, -1\}} P(k|c_1)P(k|c_2) \tag{2}
\]

where \( c_i \) are the two models whose agreement is being computed and \( P(k|c_i) \) is the probability that model \( c_i \) labels an example as being in category \( k \) [8, 25].

The SP method determines when to stop based on whether the average of the agreements over a window of size \( k \) exceeds a threshold \( K \), where \( k \) and \( K \) are user-defined parameters. The SP method can be adjusted to be more aggressive by decreasing \( k \) and \( K \), and can be adjusted to be more conservative by increasing \( k \) and \( K \). Previous research has suggested that \( k = 3 \) and \( K = 0.99 \) work well as default values [8].

The stop set of examples on which SP computes the agreement between successively trained models is an important part of the stopping method [8, 14, 15]. Since SP uses this set of examples to make a generalization about the data distribution for the intended application of the learned model, the stop set should be an unbiased representation of the data. If it is systematically biased, the agreement between successively learned models on the elements in the stop set might not be indicative of the agreement between the learned models on application data as a whole. A larger stop set is more likely to be representative and unbiased, but a smaller set is more
computationally efficient for computing agreement between the learned models. An easy way to select the stop set is to allocate a randomly chosen fixed percent of the data to it, such as 50%, as in [14]. We denote such a randomly chosen unbiased stop set as \( S \). Previous research found that the variance in the Kappa statistic [15] is inversely proportional to the size of the stop set. High variance in Kappa could cause SP to behave unpredictably so the variance check from [15] is recommended as a principled way to ensure that the size is large enough to get low-variance estimates of agreement.

Recently, in [18] SP was implemented using the remaining unlabeled pool, \( U \), as the stop set. We will refer to this method as \( SP_U \) and the original method as SP or \( SP_S \). The choice of using the remaining unlabeled pool as the stop set is not explained in [18], providing evidence that the community could be under-appreciating the importance of stop set selection. When using the remaining unlabeled pool as the stop set, the set is changing at each iteration of AL. In particular, the size of this stop set is shrinking at each iteration of AL. Therefore, the number of examples in the remaining unlabeled pool at iteration \( i-1 \) and iteration \( i \) will be different. Cohen’s Kappa agreement needs to be computed on the same set of examples [25]. When we implement \( SP_U \), to ensure equally sized prediction lists, we compute the Kappa agreement between models trained at iteration \( i-1 \) and iteration \( i \), but only use the examples in the unlabeled pool at iteration \( i \). In contrast to \( S \), \( U \) is not representative of the application space, is systematically biased, and becomes increasingly more biased as AL proceeds. The extent and way that \( U \) is biased will depend on the details of the AL setting, including the base learner that is used and the query strategy that is used. In [18] a support vector machine (SVM) base learner is used with uncertainty sampling based on closest-to-hyperplane selection. This is known to result in a set of labeled data and a remaining set of unlabeled data that are biased [26]. Intuitively, this makes sense because the algorithm is systematically selecting the examples that are the most difficult to classify to be labeled, leaving \( U \) to consist increasingly of examples that are easier to classify as AL proceeds. We hypothesize that using this biased set as a stop set can be intuitively expected to yield higher agreement scores sooner than if an unbiased set were used. Our experimental results in section [V] confirm this hypothesis and help shed light on the extent of the impact.

B. Declining Confidence

The Declining Confidence method is based on the premise that once all the informative examples in the unlabeled pool are labeled, further labeling could mislead the model and decrease its confidence [17]. DC determines when to stop AL based on when the model’s confidence decreases for \( \epsilon \) consecutive iterations

\[
\text{Conf}(D) = \frac{1}{|D|} \sum_{d \in D} \text{confidence}(d) \tag{3}
\]

where \( \text{confidence}(d) \) is a measure of the model’s confidence on datapoint \( d \). In [17] when a support vector machine base learner is used, the confidence is measured by the distance of the datapoint \( d \) from the separating hyperplane learned by the current model.

Recently, in [18], two stopping methods were presented that are similar to DC from [17]. One difference is that the methods in [18] use the batch set, \( B \), and the remaining unlabeled pool, \( U \), as their stop sets. In [17], similarly to [8], a large representative set is suggested to be used with DC. To investigate the impact of these different stop set choices, we implement DC using \( B \) and \( U \), and a large randomly selected unbiased stop set \( S \). We call these methods DC\(_B\), DC\(_U\), and DC\(_S\), respectively [26]. Each method stops when \( \text{Conf}(B), \text{Conf}(U) \), or \( \text{Conf}(S) \) decreases for \( \epsilon = 3 \) iterations, respectively. We acknowledge that implementing DC with \( B \) and \( U \) is contrary to Vlachos’ guidance about how to select the stop set because these sets are biased, changing, and in the case of \( B \), small. We do so for illumination of the impact of stop set selection on the method since these sets were suggested for use as stop sets with the similar NC method in [18].

C. Non-increasing Confidence

The two stopping methods presented in [18], TotalConf and LeastConf, both stop AL if the model’s \( \text{Conf} \) score (equation [3]) measured over a stop set of training examples does not increase for \( \epsilon \) iterations, where \( \epsilon \) is a user-defined parameter. In [18] the value of \( \epsilon \) is set to three. Since the methods from [18] stop when the model’s confidence stops increasing, we refer to them as Non-increasing Confidence (NC). The difference between TotalConf and LeastConf is that TotalConf uses \( U \) to compute confidence and LeastConf uses \( B \). For brevity, we will call LeastConf \( \text{NC}_B \) and TotalConf \( \text{NC}_U \).

We also implement NC with a large fixed unchanging unbiased representative stop set as recommended in [9], [17] and refer to this implementation of NC stopping as \( \text{NC}_S \). We acknowledge that this was not suggested by the original authors in [18]. We do so to illuminate the impact of stop set choice on NC stopping methods and we also hypothesize that using such an unbiased fixed stop set will strengthen the performance of NC stopping, which is confirmed experimentally in section [V].

IV. EXPERIMENTAL SETUP

In our experiments, we perform text classification with seven widely used publicly available datasets. To make our results as reproducible as possible, we use the train-test split

\(^1\) Observe that neither [18] nor the current paper use \( B \) as a stop set with SP because \( B \) is a completely new set at each iteration of AL, and therefore, cannot be used with SP.
recognized by the community if such a split exists. Otherwise, we perform 10-fold cross validation.

We use the “bydate” version of the 20NewsGroup dataset and classify its newsgroup postings into 20 categories. We use the Reuters-21578 Distribution 1.0 ModApte split as in [27], [28] and classify the 10 largest categories of business news articles. We classify email spam from the SpamAssassin corpus and the ham25 version of the TREC corpus. We use the results from the four largest categories of academic webpages in the WebKB dataset as in [29], [30], [31], [8]. We use 10-fold cross validation for SpamAssassin, TREC, and WebKB; we macro average our results across folds and categories.

In order to compare directly with the technology assisted sensitivity review experiment conducted in [18] we tried to obtain the dataset that was used in that paper. Unfortunately, this dataset is private and cannot be shared with other researchers (personal communication with Graham McDonald). Instead, we perform TC on the publicly available RCV1-v2 dataset [32] that has been used frequently in past works to simulate TAR [33], [34], [35], [56], [37], [38]. Most of these authors use a subset of the 103 RCV1-v2 categories; we use all 103 of them. RCV1-v2 is computationally demanding, so we do not use it in all of our experiments.

We use a support vector machine with a linear kernel and C=1, as done by [8], [17], [18]. We use the closest-to-hyperplane selection algorithm as the query strategy during AL, as done by [8], [17], [18]. We also use random selection, that is, passive learning, for a comparison experiment. We allocate 50% of training examples to the randomly selected stop set, S, as in [14]. We use a batch size of 0.5%, consistent with the findings of [39]. We use a bag-of-words approach that only considers words that appear more than three times. For all datasets other than RCV1-v2 during preprocessing we remove 174 stop words using the Default English Stopwords List.

When comparing stopping methods, we report the number of annotations (i.e., labels) acquired (denoted by ANN in our tables), the proportion of annotations of the entire dataset acquired (ANN-P in tables), the learned model’s F-1 measure (F1), F-2 measure (F2), accuracy (ACC), and balanced accuracy (BAC), all at the time the stopping method indicates to stop AL. These performance measures were used in [18], which will help to facilitate comparison. We also display these statistics when the model is given all possible training data (denoted by the column “Final” in our tables). F-β Measure is a weighted harmonic mean between precision and recall, where β is a constant such that recall is β times more important than precision. Accuracy is the proportion of the model’s predictions that are correct. Balanced accuracy is the average of the recall scores for each class. We report these statistics for individual datasets, as well as a macro average across all datasets. In our tables we boldface entries that are the best-performing for each metric. Lower numbers are better for ANN and ANN-P because that reflects reducing the amount of labeled data required. Higher numbers are better for the other metrics since that reflects better system performance for end-users.

V. RESULTS AND ANALYSIS

This section displays the results of our experiments and analyzes the results. Table I shows the notation used throughout this section.

| Symbol | Type | Definition |
|--------|------|------------|
| SP     | stopping method | Stabilizing Predictions |
| DC     | stopping method | Declining Confidence |
| NC     | stopping method | Non-increasing Confidence |
| S      | stop set | Standard fixed unbiased set of examples |
| B      | stop set | Batch set of examples |
| U      | stop set | Unlabeled pool of examples |

TABLE I

NOTATION USED IN THIS SECTION. STOPPING METHODS AND STOP SET MAY BE COMBINED USING SUBSCRIPT NOTATION. FOR EXAMPLE, SP$_S$ INDICATES THAT THE STABILIZING PREDICTIONS STOPPING METHOD IS USED WITH A STANDARD FIXED UNBIASED STOP SET.

A. Impact of Stop Sets on Stabilizing Predictions

Table II shows the performance of SP$_S$ and SP$_U$ for SVM-closest and SVM-random on all datasets (except RCV1-v2 for computational feasibility).

As expected, active learning significantly outperforms passive learning, achieving similar performance with far fewer annotations. When the uncertainty sampling algorithm is used in active learning, we see that SP$_U$ is more aggressive than SP$_S$. We know from [15] that as the unlabeled pool shrinks, the variance in the Kappa statistic increases, and Stabilizing Predictions becomes more unstable. It was hypothesized in [18] that the unlabeled pool becoming small in the later stages of AL could cause the more aggressive behavior of SP$_U$.

However, when examining the size of the unlabeled pool at the point when SP$_U$ stops AL, we see that U is large at the stopping point for all datasets. At the stopping point, U contains between 85% and 96% of the total dataset (91% on average). Since U never gets small, the variance of the Kappa estimator never gets large. Therefore, the decreasing size of U is not the main cause of the more aggressive stopping behavior of SP$_U$.

When an uncertainty sampling query strategy is used, B and U become biased sets. Since uncertainty sampling selects the most uncertain training examples for labeling, elements in B are systematically biased such that they are difficult for the model to predict correctly. At each round of AL, the elements of B are removed from U. This in turn causes the elements in U to be biased such that they are easier for the model to predict correctly. Since the examples in U are easier to predict, the
Passive Learning and Comparison with the SP Method

During passive learning, U should be representative of the dataset except that it is by definition not permitted to have any representation in the training data. This will make U slightly harder to achieve agreement on when compared with an unbiased predetermined random set that is not artifically prevented from having representation in the training data. Therefore, we would expect SP_U to behave more conservatively during passive learning than SP_S. Table II confirms this, where we see that SP_U is more conservative than SP_S during passive learning.

SP_U requires a non-standard approach to compute Cohen’s Kappa agreement between successively trained models, was found to behave over-aggressively by [18] during AL, and uses a biased non-representative stop set. For these reasons, we suggest referring to the original authors of SP and implementing the method with a fixed unbiased representative stop set S. We compare SP_S with the confidence-based stopping methods in the next section.

B. Impact of Stop Sets on Confidence Based Stopping Methods and Comparison with the SP Method

In Table III we compare the stopping results from Declining Confidence and Non-increasing Confidence using the batch set, the unlabeled pool, and a large, randomly selected, fixed set as stop sets during AL. We also add the performance of SP_S as a column for ease of comparison of stabilizing predictions stopping performance with confidence-based stopping performance.

TABLE II

The number of annotations and model performance when SP_U and SP_S stop for active learning and passive learning for all datasets (except RCV1 for computational feasibility). Bolded entries are the best-performing value for the relevant statistic.

| Dataset       | Stat | SP_U   | DC_S   | DC_S   | NC_S   | NC_S   | NC_S   | NC_S   | Total   |
|---------------|------|--------|--------|--------|--------|--------|--------|--------|---------|
| News         | ANN-P | 0.061  | 0.122  | 0.970  | 0.970  | 0.970  | 0.970  | 0.970  | 0.970   |
|              | F1    | 0.786  | 0.786  | 0.786  | 0.786  | 0.786  | 0.786  | 0.786  | 0.786   |
|              | F2    | 0.246  | 0.246  | 0.246  | 0.246  | 0.246  | 0.246  | 0.246  | 0.246   |
|              | ACC   | 0.989  | 0.989  | 0.989  | 0.989  | 0.989  | 0.989  | 0.989  | 0.989   |
|              | BAC   | 0.556  | 0.556  | 0.556  | 0.556  | 0.556  | 0.556  | 0.556  | 0.556   |

TABLE III

The number of annotations and model performance when a stopping method stops. Bolded entries are the best-performing value for the relevant statistic.

Table III shows that an unbiased stop set, S, is the best stop set to use with DC. Table III shows that when implemented with U and B, DC uses far more annotations than is required and does not achieve any meaningful performance boost compared to DC_S. Comparing DC_S with SP, from the average row, we see that DC_S uses more than twice as many annotations as SP and all performance measures except ACC are worse than those of SP.

For NC, we see that the methods presented in [18] have very different behavior on different datasets. For the individual datasets in Table III, we see that NC_B does not learn an effective model for 20NewsGroups, Reuters, and WebKB, while it requests wastefully large numbers of annotations for RCV1, SpamAssassin, and TREC. NC_D does not learn a good model for 20NewGroups and requests many more annotations than are required for all other datasets. In both the individual dataset results and the overall average results, we see that NC performs best when implemented with a large representative stop set, S. This strategy was first implemented in this work and was not considered in [18], yet it appears to be the highest-performing variant of NC. In the average row, we see that NC_S attains essentially the same model performance as DC_S, but
uses about half of the annotations. Therefore, it is reasonable to conclude that NC<sub>S</sub> is the best out of all the confidence-based methods. However, its performance is still not as strong as SP. The average row shows that SP<sub>S</sub> requires less labeled data and produces higher-performing models across all performance measures than NC<sub>S</sub>.

Figure 1 shows a representative learning curve for the 20NewsGroups dataset. This learning curve captures common trends that occur at the level of an individual fold or category: the methods that use U fail to stop, the methods that use B stop too early, and the methods that use S stop at reasonable points. The methods that use S are more stable than the other methods. Out of these methods, SP<sub>S</sub> is the most efficient at saving annotations and producing a high performing model.

VI. CONCLUSION

Active learning is used to produce high-performing models while using significantly less labeled training data than when using passive learning. Without an effective stopping method, AL could stop too late, wasting unnecessary data labeling effort, or AL could stop too early, producing a poorly learned model. Several AL stopping methods determine when to stop on the basis of the results of computations performed on a set of examples called the stop set. The choice of the stop set has received relatively less attention than development of the computations that the stopping methods perform on the stop set. In this work, we experiment with several different possible options for a stop set and find that the choice of the stop set can have a significant impact on stopping active learning. It is important to communicate these findings to prevent suboptimal stop sets from being used and suboptimal stopping methods from being used.

Authors of the widely used Stabilizing Predictions method and the Declining Confidence method had originally suggested that representative unbiased sets be used as stop sets for their methods. Recent work used the remaining unlabeled pool and the batch set as stop sets in their new confidence-based methods and also used the remaining unlabeled pool as the stop set when implementing the Stabilizing Predictions method as a baseline for comparison. The remaining unlabeled pool and the batch set are systematically biased as active learning proceeds.

We find that different stop set choices impact different stopping methods in different ways. When Stabilizing Predictions is implemented with the unlabeled pool instead of an unbiased representative stop set, it becomes more aggressive compared with when it is implemented with an unbiased representative stop set. When confidence-based methods are implemented with the unlabeled pool instead of a large unbiased stop set, they become more conservative, sometimes failing to stop at all. When confidence-based methods are implemented with the batch set, they become less consistent in their behavior and sometimes may become overly conservative or overly aggressive. For both stabilizing predictions and confidence-based methods, we find that the best-performing stop set to use is a random unbiased set. Using one of the biased sets produces a less reliable and lower-performing method.

Fig. 1. Learning curve for the 20NewsGroups dataset, classifying articles as belonging to the talk-religion category or not. In this plot, DC<sub>B</sub> and NC<sub>B</sub> stopped at the same point, so their vertical lines are perfectly on top of each other. The same occurred for DC<sub>U</sub> and NC<sub>U</sub>.

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] 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.
[2] 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.
[3] C. Schröder and A. Niekler, “A survey of active learning for text classification using deep neural networks,” arXiv preprint arXiv:2008.07267, 2020.
[4] Y.-F. Yan, S.-J. Huang, S. Chen, M. Liao, and J. Xu, “Active learning with query generation for cost-effective text classification,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, 2020, pp. 6583–6590.
[5] Z. Zhan, Z. Hou, Q. Yang, J. Zhao, Y. Zhang, and C. Hu, “Knowledge attention sandwich neural network for text classification,” Neurocomputing, vol. 406, pp. 1–11, 2020.
[6] C. A. Flores, R. L. Figueroa, and J. E. Pezoa, “Active learning for biomedical text classification based on automatically generated regular expressions,” IEEE Access, vol. 9, pp. 38 767–38 777, 2021.
