Active Sampling for Text Classification with Subinstance Level Queries

Shayok Chakraborty, Ankita Singh

Department of Computer Science, Florida State University

Abstract

Active learning algorithms are effective in identifying the salient and exemplar samples from large amounts of unlabeled data. This tremendously reduces the human annotation effort in inducing a machine learning model as only a few samples, which are identified by the algorithm, need to be labeled manually. In problem domains like text mining and video classification, human oracles peruse the data instances incrementally to derive an opinion about their class labels (such as reading a movie review progressively to assess its sentiment). In such applications, it is not necessary for the human oracles to review an unlabeled sample end-to-end in order to provide a label; it may be more efficient to identify an optimal subinstance size (percentage of the sample from the start) for each unlabeled sample, and request the human annotator to label the sample by analyzing only the subinstance, instead of the whole data sample. In this paper, we propose a novel framework to address this challenging problem, in an effort to further reduce the labeling burden on the human oracles and utilize the available labeling budget more efficiently. We pose the sample and subinstance size selection as a constrained optimization problem and derive a linear programming relaxation to select a batch of exemplar samples, together with the optimal subinstance size of each, which can potentially augment maximal information to the underlying classification model. Our extensive empirical studies on six challenging datasets from the text mining domain corroborate the practical usefulness of our framework over competing baselines.

Introduction

A common bottleneck in developing supervised learning algorithms is the requirement of large amounts of hand-annotated data. However, while unlabeled data is cheap and easily available, obtaining class labels requires extensive human effort, often from experts with very limited availability. *Active Learning (AL)* algorithms alleviate this challenge by intelligently querying the labels of the most informative samples. This drastically reduces the human annotation effort, as the oracles only need to label the samples selected by the algorithm [1]. Further, it exposes the underlying machine learning model to the exemplar instances in the data population; the model thus typically depicts better generalization than a passive learner, where the training data is sampled at random. Active learning has depicted commendable performance in a variety of applications, including computer vision (Yoo and Kweon 2019), text mining (Tong and Koller 2001), anomaly detection (Pimentel et al. 2020) and medical diagnosis (Gorriz et al. 2017) among others.

Annotating data in applications like text or video classification is much more tedious and labor-intensive, as the oracles need to meticulously review each sample before providing a label. Thus, the paucity of human labor and the need to use it more efficiently is even more pronounced for such applications. This necessitates specialized and more user-friendly query mechanisms for the AL algorithms to be useful in a real-world setting. We explore one such query mechanism in this research. In the aforementioned applications, users form their notion about the label of a sample incrementally, with the notion becoming more concrete as a larger percentage of the sample is reviewed and analyzed. Thus, it may not be always necessary to analyze each data sample end-to-end to figure out its class label; depending on the data sample, it may be possible to gauge its label after reviewing only a certain percentage of the sample from the start. For instance, in a movie review classification application, the human annotator need not read through the entire review to provide a label (positive or negative); depending on the review in question, it may be possible to provide a label after reading, say, the first 40% of the text. Similarly, in video genre classification, a human oracle may be in a position to annotate a sample after watching only the first 50% of a given video. If the optimal subinstance size (percentage of a sample from the start) is accurately identified for each data sample, it can result in substantial savings in terms of time and human effort and result in a better utilization of the available query budget. However, this also poses the challenge that if the subinstance size is too small, it may not provide enough information to an annotator to make an informed decision about its label; in that case, he can abstain from labeling resulting in a wastage of query budget. We attempt to answer the following research question in this paper: *we are given a small amount of labeled training data L and a large amount of unlabeled data U. Each unlabeled data sample can be split into a predefined number (K) of*

[1] we use the terms *oracle, annotator, labeler* and *user* synonymously in this paper

subinstances from the start and can be queried by showing upto a certain percentage (such as $20\%, 40\%$) to the human annotator. Each subinstance size corresponds to a cost, denoting the price to be paid to get a subinstance of that size labeled; higher subinstance sizes correspond to higher costs. A budget B is imposed which denotes the maximum allowable total cost that can be incurred. Which unlabeled samples should we select for manual annotation and what is the optimal subinstance size for each queried sample, so as to induce a model with maximal generalization capability within the given budget?

We propose a novel algorithm to address this challenging problem, with the objective of further reducing the human annotation effort in AL applications. The active sample and subinstance size selection problem is solved using a single integrated framework to derive a batch of informative unlabeled samples together with their optimal subinstance sizes. Although validated on text classification in this paper, the proposed framework is generic and can be used in any application where the data has a temporal component (such as video classification, sensor readings to detect motion abnormalities etc.). We hope this research will motivate the development of other AL algorithms for temporal data, where annotation is much more strenuous and time-consuming.

Related Work

In this section, we first present a survey of active learning in general, followed by a survey of AL algorithms with novel query and annotation mechanisms geared toward further reducing the labeling burden on human oracles, which is the focus of this research.

Active Learning: Active Learning is a well-researched problem in the machine learning literature (Settles 2010). In a typical setup, the learner is exposed to a pool of unlabeled samples and it iteratively queries samples for annotation, until some stopping condition is satisfied. Several criteria have been studied to quantify the usefulness of a batch of samples, including entropy based uncertainty sampling (Bhattacharya, Liu, and Chakraborty 2019; Guo and Schuurmans 2007), distance from the decision boundary in the feature space for SVM classifiers (Tong and Koller 2001), the extent of disagreement among a committee of classifiers (Freund et al. 1997), the Fisher information matrix (Hoi et al. 2008), mutual information (Guo 2010) and the representativeness of samples (Huang, Jin, and Zhou 2014) among others. With the advent and popularity of deep neural networks, researchers have studied the problem of deep active learning, where the goal is to automatically learn discriminating features and simultaneously select the informative samples for manual annotation (Sener and Savarese 2018; Yoo and Kweon 2019). Deep AL methods based on adversarial learning have particularly shown promising performance (Zhang et al. 2020; Sinha, Ebrahimi, and Darrell 2019; Deng et al. 2018; Zhu and Bento 2017). A body of research has focused on novel extensions of AL, such as a combination of active learning and transfer learning (Su et al. 2020), actively completing an incomplete data matrix (Ruchansky, Crovella, and Terzi 2015), actively summarizing a video (Molino et al. 2017) and AL in the presence of noisy oracles (Chakraborty 2020; Huang et al. 2017) among others. Cost-sensitive AL techniques have also been studied, which incorporates annotation cost in evaluating the informativeness of an unlabeled sample (Wei et al. 2019; Bloodgood and Callison-Burch 2010).

AL with Novel Query Mechanisms: The fundamental premise of active learning is to reduce human annotation effort in inducing a machine learning model. In keeping with this objective, a few research efforts have focused on the design of novel algorithms to further reduce the onus on human oracles, in a variety of ways. Joshi *et al.* (Joshi, Porikli, and Papanikolopoulos 2010) proposed a binary feedback algorithm where the active learner queried an unlabeled image together with a sample training image, and the human labeler had to merely provide the binary feedback as to whether the two images belonged to the same category. This is useful in applications where there is a large number of concept classes (such as ImageNet) and it may not be feasible for a human expert to be knowledgeable about all of them. Hu *et al.* (Hu et al. 2019) also proposed an active query mechanism with binary user feedback and a strategy to learn from partial labels, in order to simplify user annotation. Clustering algorithms have also been developed with active binary feedback, which query a pair of samples and the oracles need to specify whether or not the samples in a pair correspond to the same cluster (Biswas and Jacobs 2012). Qian *et al.* (Qian et al. 2013) proposed an AL framework, which queried the ordering of the importance of an instance's neighbors, rather than its label. Thus, a non-expert can place an ordering (or a partial ordering) on the similarity of the neighbors of a queried sample, instead of providing its absolute label. Along similar lines, Xiong *et al.* (Xiong et al. 2015) developed an AL framework which queried an unlabeled triplet (x_i, x_j, x_k) and posed the question: *is instance x_i more similar to x_j than to x_k?*

Annotating a data sample in a text / video classification application is extremely time-consuming and laborious. This necessitates specialized query mechanisms which are more user-friendly in order to further alleviate the labeling burden and make more efficient usage of the available query budget. However, this problem has received considerably less attention in the AL literature. Loaiza *et al.* (Loaiza, Culotta, and Bilgic 2014) proposed the *Anytime Active Learning (AAL)* algorithm, which studied the problem of training text classifiers by requesting human annotation on examples before inspection is fully complete. To the best of our knowledge, this is the only published research where the query mechanism is similar to our framework. However, both the *Static AAL* and the *Dynamic AAL* algorithms proposed in the paper queried only a single unlabeled instance in each AL iteration. This may result in inefficient usage of labeling resources, as only a single annotator is being utilized at any given point of time (in a crowdsourcing platform like the Amazon Mechanical Turk, multiple annotators are present to label samples simultaneously); myopically extending the single-instance selection to multi-instance selection produces sub-optimal results (as depicted in our empirical studies). Further, the single instance selection requires frequent model updates which is computationally inefficient. In contrast, our framework

queries a batch of samples simultaneously and identifies the optimal subinstance size for each queried sample. Through its batch selection strategy, our framework can exploit the presence of multiple annotators in any crowdsourcing platform, and can potentially be a step toward the development of efficient active query strategies for temporal data. We now describe our framework.

Proposed Framework

In our problem setup, we are given a labeled training set L and an unlabeled set U. The size of the unlabeled set is much larger than the labeled set ($|L| \ll |U|$). Let m be the model trained on L and C be the number of classes in the problem. We are given an integer K, which denotes the number of discrete time points when an oracle can be interrupted and asked for a label, in the process of analyzing each unlabeled sample. In other words, each unlabeled sample x_i is split into K subinstances $\{x_i^k\}$ from the start, $k = 1, 2, \ldots K$. A cost vector $Q = \{q_1, q_2, \ldots q_K\}$ is given, where q_k denotes the price to be paid to get subinstance $\{x_i^k\}$ labeled by the annotator. A query budget B is also given, which denotes the maximum budget that can be expended for label query. Our objective is to select a batch of unlabeled samples together with a subinstance size for each sample, such that the total cost incurred does not exceed the budget, and the selected samples with the provided labels (obtained from the oracles who attempt to label the samples by analyzing only the queried subinstance) augment maximal information to the classification model. However, if the selected subinstance size is too small and it does not provide sufficient information to label the sample, the oracle can abstain from labeling. Regardless of whether a label is obtained from the oracle, the price of the subinstance is deducted from the query budget, since the oracle has expended effort to analyze the sample, even if he abstains from labeling. We assume that when a label is obtained from an oracle, it is correct. To address this problem, we propose to perform active selection of both samples and subinstance sizes. These are detailed below.

Active Sample Selection

In order to identify the optimal set of samples to be queried, we need a metric to quantify the utility score of a batch of unlabeled samples. We used the *uncertainty* criterion to compute the information content of a batch of samples. However, if two samples are individually informative, but furnish the same information, then the knowledge gained by querying both of them is not optimal. We therefore incorporated a *diversity* metric to quantify the diversity between every pair of samples. A sample selection framework driven by these two conditions ensures that the selected samples are individually informative and they have minimal redundancy (duplication) among them. Such criteria has been used in previous active learning research (Chakraborty et al. 2015).

Computing uncertainty: The information content of an unlabeled sample x_i was computed as the classification uncertainty of x_i using the model m. We used the Shannon's entropy to compute the uncertainty of an unlabeled sample:

$$H(x_i) = -\sum_{j=1}^{C} p_{ij} \log p_{ij} \tag{1}$$

where p_{ij} is the posterior probability of x_i with respect to class j, computed by the current model m. A high value of entropy denotes high classification uncertainty, and thus a more informative sample from an active learning perspective. Note that the uncertainty is computed with respect to the entire unlabeled sample x_i and not any subinstance x_i^k; this is because the base model is trained on the labeled training set, which consists only of complete instances.

Computing diversity: We also computed a diversity matrix $D \in \Re^{|U| \times |U|}$, where D_{ij} denotes the diversity between samples x_i and x_j in the unlabeled set. We used the kernelized distance to quantify the diversity between a pair of samples, where a high value of the diversity denotes low redundancy. The matrix D was computed as follows:

$$D(i, j) = \phi(x_i, x_j) \tag{2}$$

where $\phi = (.,.)$ denotes a kernel in the Reproducing Kernel Hilbert Space (RKHS) (Sriperumbudur et al. 2010).

Active Subinstance Selection

For each unlabeled sample, we estimated the optimal subinstance size based on two conditions: the *probability of obtaining a label from the oracle*; and the overall *labeling cost* incurred to label the subinstance. These are detailed below.

Computing labeling probability of an oracle: In a real-world application, each unlabeled sample contains varying degrees of information. For some instances, inspecting the first few words may provide a concrete idea about the label of the sample; for others, a more thorough and extensive inspection may be necessary to provide a label. We therefore exploited a data-driven strategy to compute this probability. The oracles were asked to label a part of the dataset, where each data sample was split into all the allowed subinstance sizes. For each sample and for each subinstance size, we noted whether the oracle provided a label or abstained from labeling. A binary SVM classifier was then trained to model the labeling pattern. Given a particular sample and a subinstance size, the trained SVM returns the probability of obtaining a label from the oracle. Let $\widehat{p_{ik}}$ denote the probability of obtaining a label from the oracle for a given subinstance x_i^k. We refer to this SVM model as the *neutrality model*, as it denotes whether the oracle provides a label for a subinstance or remains neutral.

Computing labeling cost: The labeling cost of a subinstance is directly obtained from the given cost vector Q; the cost of labeling a subinstance of size k is q_k. The cost vector can be computed based on the available resources of a given application and is assumed to be a known parameter.

Active (Sample-Subinstance) Selection

Given the uncertainty vector H, the cost vector Q and the labeling probabilities $\widehat{p_{ik}}$, we compute a matrix $P \in \Re^{K \times |U|}$ (K is the number of subinstance sizes), where each column represents an unlabeled sample and each row represents

a subinstance size. Our objective is threefold: (i) select a batch of unlabeled samples which furnish high entropy values (high uncertainties); (ii) select a subinstance size for each sample which maximizes the probability of obtaining a label from the oracle; and (iii) minimize the labeling cost of the subinstance. The matrix P is defined to capture all these conditions:

$$P(k,i) = \frac{H(x_i) * \widehat{p_{ik}}}{q_k}, \quad i = 1, \dots |U|, \quad k = 1, \dots K \tag{3}$$

Also, we would like to maximize the diversity among the selected samples, as given by the entries in the matrix D. We define a binary matrix $Z \in \{0,1\}^{|U| \times K}$ where each row corresponds to an unlabeled sample and each column corresponds to a subinstance size. A value of 1 in a row denotes that the sample should be selected for annotation, and the position of 1 in a particular row of Z denotes the subinstance size for that sample that should be used for query. The active (sample-subinstance) selection is thus posed as the following optimization problem:

$$\begin{aligned} \max_{Z} \quad & \text{trace}(ZP) + \lambda (Ze)^\top D(Ze) \\ \text{s.t.} \quad & Z_{ik} \in \{0,1\}, \forall i,k \\ & Z_i.e \leq 1, \forall i \\ & \langle Z, E \rangle \leq B \end{aligned} \tag{4}$$

where λ is a weight factor governing the relative importance of the two terms, e is a vector of length K with all entries 1, Z_i denotes row i of matrix Z, $\langle \cdot, \cdot \rangle$ denotes the matrix inner product operator, E is a matrix of the same dimension as Z ($|U| \times K$) with the cost value q_k in the entire column k, and B is the labeling budget. The first term in the objective function denotes that the selected samples have high entropy, high probability of obtaining a label from the oracle and low labeling cost; the second term ensures that the selected samples have maximal diversity among them. The first constraint denotes that Z is a binary matrix; the second constraint signifies the each row of Z can have at most one entry as 1, since each selected unlabeled sample can be queried with exactly one subinstance size; and the third constraint denotes that the total cost incurred by labeling the selected samples is within the specified budget B. Such a formulation enables us to utilize the presence of multiple labeling oracles simultaneously (corroborating its usefulness in real-world applications), contrary to the methods proposed in (Loaiza, Culotta, and Bilgic 2014), which query only a single unlabeled sample and utilize a single labeling oracle in each AL iteration. We now establish an important property, which enables us to efficiently solve this optimization problem.

Theorem 1. *The optimization problem defined in Equation (4) can be expressed as an equivalent linear programming (LP) problem.*

Proof. The first term in the objective function can be expressed as a linear term in the variable Z: $trace(ZP) = \sum_{i,j} P_{ij}.Z_{ji}$. Using the properties of inner product operations, and that the matrix $ee^\top$ contains all entries as 1, the second term can be simplified as follows:

$$\begin{aligned} (Ze)^\top D(Ze) &= \sum_{i,j} D_{ij} (Ze)_i (Ze)_j = \sum_{i,j} D_{ij} \langle Z_i.e, Z_j.e \rangle \\ &= \sum_{i,j} D_{ij} \langle Z_i, Z_j.ee^\top \rangle = \sum_{i,j} D_{ij} \langle Z_j^\top Z_i, ee^\top \rangle \\ &= \sum_{i,j} D_{ij} \sum_{a,b} Z_{ia}.Z_{jb} = \sum_{i,j} \sum_{a,b} D_{ij} Z_{ia}.Z_{jb} \\ &= \sum_{i,j} \sum_{a,b} D_{ij} W_{ijab} \end{aligned}$$

where $W_{ijab} = Z_{ia}.Z_{jb}$. We now attempt to write this quadratic equality as a linear term. Since Z is a binary matrix with only 0 and 1 entries, W_{ijab} will equal 1 when both Z_{ia} and Z_{jb} are 1 and will equal 0 otherwise. The quadratic equality $W_{ijab} = Z_{ia}.Z_{jb}$ can thus be expressed as the following equivalent linear inequality:

$$-Z_{ia} - Z_{jb} + 2W_{ijab} \leq 0 \tag{5}$$

A simple analysis of the inequality reveals that when Z_{ia} and Z_{jb} are both 0, or when one of them is 0 and the other is 1, W_{ijab} is forced to be 0. When Z_{ia} and Z_{jb} are both 1, W_{ijab} is free to be 0 or 1. However, we are solving a maximization problem, where one of the terms in the objective function is $\sum_{i,j} \sum_{a,b} D_{ij} W_{ijab}$, and the matrix D can have only non-negative entries. Thus the nature of the problem will force W_{ijab} to be 1, as that will produce a better (higher) value of the objective. Hence, the values of W_{ijab} obtained from the quadratic equality $W_{ijab} = Z_{ia}.Z_{jb}$ and the linear inequality $-Z_{ia} - Z_{jb} + 2W_{ijab} \leq 0$ are exactly the same under all possible conditions. The optimization problem in Equation (4) can thus be expressed as follows:

$$\begin{aligned} \max_{Z,W} \quad & \sum_{i,j} P_{ij}.Z_{ji} + \lambda \sum_{i,j} \sum_{a,b} D_{ij} W_{ijab} \\ \text{s.t.} \quad & Z_{ij}, W_{ijab} \in \{0,1\}, \forall i,j,a,b \\ & Z_i.e \leq 1, \forall i \\ & \langle Z, E \rangle \leq B \\ & -Z_{ia} - Z_{jb} + 2W_{ijab} \leq 0, \forall i,j,a,b \end{aligned} \tag{6}$$

In this optimization problem, both the objective function and the constraints are linear in the variables Z and W. It is thus a linear programming (LP) problem.

□

We relax the integer constraints on Z and W into continuous constraints and solve the problem using an off-the-shelf LP solver. After obtaining the continuous solution, we recover the integer solution of our variable of interest Z, using a greedy approach where the highest entries in each row of Z are reconstructed as 1 and the other entries as 0, observing the constraints. While this may introduce approximation errors, our algorithm still comprehensively outperforms the baseline methods, as demonstrated in the following section.

Experiments and Results

Datasets: We used 6 challenging datasets from the text mining domain to study the performance of our framework: (i) **Hotel Reviews** [2]. Each rating varies from 1 to 5; we used 1 and 2 as the negative class, 4 and 5 as the positive class and discarded samples where the rating was 3; (ii) **IMDB** (Maas et al. 2011); (iii) **SRAA** (Nigam, Thrun, and Mitchell 1998); (iv) **Review Polarity** (Pang and Lee 2004); (v) **Sentence Polarity** (Pang and Lee 2005); and (vi) **Wikipedia Movie Plots** [3]. We used the top three frequent genres – action, drama and comedy and discarded samples labeled with more than one genre. The TF-IDF features were extracted, which are extensively used in text mining research (Maas et al. 2011).

Oracle Simulation: Following the setup proposed in (Loaiza, Culotta, and Bilgic 2014), we used an L_1 regularized logistic regression (LR) classifier to simulate the labeling oracle. The oracle model was trained on a part of each dataset and was tested on another part; the prediction uncertainty (defined as 1 – the maximum class probability) was computed for every test sample and the 75^{th} percentile in the vector of uncertainties was selected as the threshold T. In our empirical studies, if the prediction uncertainty of the oracle on a given unlabeled sample exceeded this threshold, no label was returned and the oracle was assumed to abstain from labeling. The algorithms, however, were not given any information about the functioning of the oracle and this information was not used in the development of any of the algorithms. We assume the presence of multiple oracles, who can label batches of unlabeled samples simultaneously, and each oracle was modeled in this manner (trained on different subsets of the dataset).

Experimental Setup: Each dataset was divided into 6 parts: (i) oracle training data (to train the oracle model); (ii) oracle testing data (to test the oracle and compute the oracle prediction threshold T); (iii) neutrality training data (to train the SVM neutrality model); (iv) initial training set L; (v) unlabeled set U; and (vi) test set. In all the datasets, the size of the initial training set was kept very small (only 20 samples), which appropriately simulates a real-world scenario, where labeled data is difficult to obtain. We selected $K = 5$ as the number of subinstance sizes and split each unlabeled sample at $20\%, 40\%, 60\%, 80\%$ and 100% granularities from the start. The cost vector was defined as $Q = [1, 2, 3, 4, 5]$, and depicts that the cost is directly proportional to the subinstance size that is being analyzed. A query budget B was imposed in each AL iteration; each algorithm queried a batch of unlabeled samples together with a corresponding subinstance size, such that the total cost of purchasing the labels from the oracles did not exceed the budget B. For each queried sample, the oracle uncertainty was computed and if it was less than the uncertainty threshold T, the complete sample (100% granularity) was appended to the training set together with its label; if the oracle uncertainty exceeded the threshold for a particular sample, it was not added to the training set. In either case, the corresponding price was charged, since the oracles had to spend time and resources to analyze the queried samples (as mentioned before, we assume that the labels provided by the oracles are all correct). The process was continued iteratively until a stopping condition was satisfied (taken as 25 iterations, except for the Hotel Reviews dataset where it was taken as 10 iterations, due to the size of the unlabeled set). The objective was to study the increment in accuracy on the test set with increasing number of iterations. The query budget B was selected as 50 for each AL iteration. All the results were averaged over 3 runs to rule out the effects of randomness. The weight parameter λ was taken as 0.5 and the Gaussian kernel was used to compute the diversity in Equation (2). Logistic regression was used as the base model in all our studies to be consistent with (Loaiza, Culotta, and Bilgic 2014).

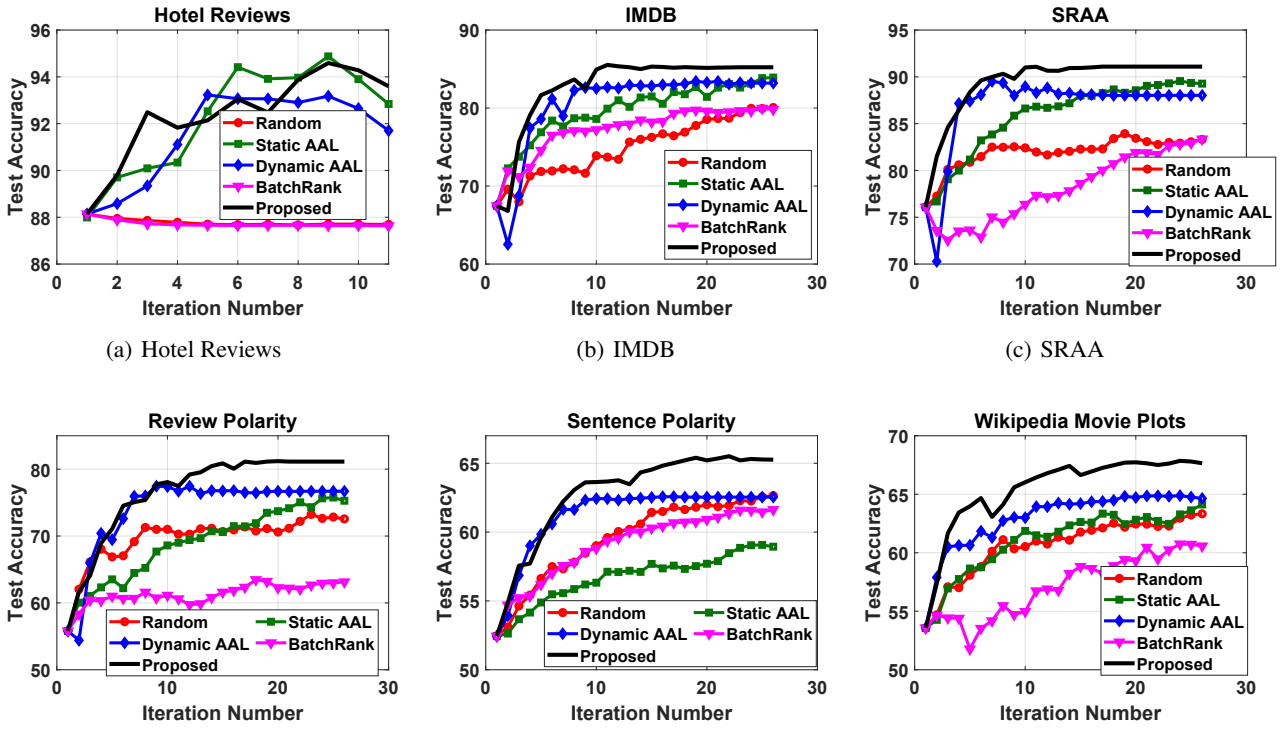
Comparison Baselines: We used the following comparison baselines in our work: **Static AAL**, the static anytime active learning algorithm proposed by Loaiza *et al.* (Loaiza, Culotta, and Bilgic 2014) that decides a subinstance size apriori and queries samples based on an uncertainty based utility function; **Dynamic AAL**, the dynamic AAL algorithm proposed by the same authors (Loaiza, Culotta, and Bilgic 2014), which dynamically computes the subinstance size and queries samples based on an uncertainty based criterion. To the best of our knowledge, these are the only two published algorithms to address the problem in question. We also used **Random Sampling** as a baseline where the unlabeled samples as well as the subinstance sizes were both selected at random. In addition, we compared our method against the **BatchRank** algorithm (Chakraborty et al. 2015) which also selects samples by optimizing an uncertainty and diversity based criterion (with LR as the base model); however, it queries samples as a whole and does not permit subinstance level queries. This was included to assess the comparative performance of an AL algorithm which only queries complete data samples, in order to understand the usefulness of subinstance level queries. Since LR was used as the base model in our experiments to be consistent with the relevant literature (Loaiza, Culotta, and Bilgic 2014) and to enable a fair comparison across all methods, we did not include any of the deep active learning methods as our comparison baselines.

Active Learning Performance

The active learning performance results are depicted in Figure 1. In each figure, the x-axis denotes the iteration number and the y-axis denotes the accuracy on the test set. *Random Sampling* in general depicts inferior performance. For the Hotel Reviews dataset, for instance, it depicts almost constant accuracies and fails to show any accuracy growth. However, sometimes it depicts good performance, as in the Sentence Polarity and Wikipedia Movie Plots datasets. The *Static AAL* and *Dynamic AAL* techniques mostly outperform *Random Sampling*, with *Dynamic AAL* outperforming *Static AAL* in general (except in the Hotel Reviews dataset). This is consistent with the observations made in the paper proposing these algorithms (Loaiza, Culotta, and Bilgic 2014). The *BatchRank* method does not depict good performance (at

[2] https://www.kaggle.com/datafiniti/hotel-reviews

[3] https://www.kaggle.com/jrobischon/wikipedia-movie-plots



(a) Hotel Reviews (b) IMDB (c) SRAA

(d) Review Polarity (e) Sentence Polarity (f) Wikipedia Movie Plots

Figure 1: Active Learning performance comparison. The x-axis denotes the iteration number and the y-axis denotes the accuracy on the test set. Best viewed in color.

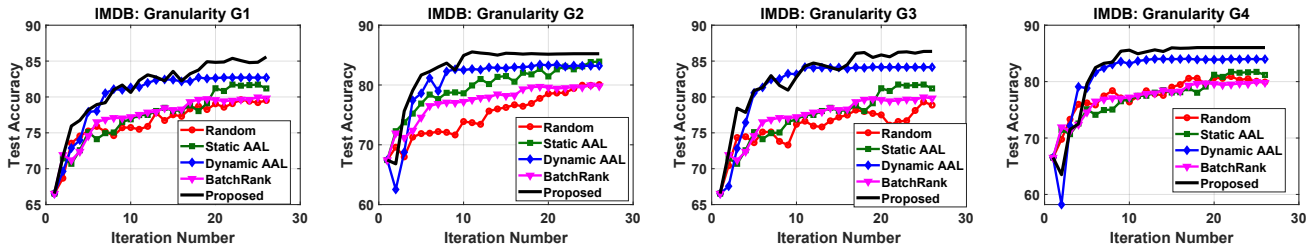
par or worse than *Random Sampling* for most datasets); this is because, even though *BatchRank* has depicted impressive performance (Chakraborty et al. 2015), it queries data samples as a whole, incurring a high labeling cost per sample and is unable to leverage the potential of subinstance queries. The proposed method consistently depicts impressive peformance across all the datasets. At any given iteration number in any dataset, it depicts the highest accuracy compared to the baselines, most of the times. Our method identifies the exemplar samples, together with an optimal subinstance size for each sample; it thus makes efficient usage of the available query budget and outperforms the baselines. We also note that both the *Static AAL* and *Dynamic AAL* methods query only a single unlabeled sample in each iteration; thus, myopically extending the single instance selection to multi-instance selection produces sub-optimal results. The results unanimously corroborate the potential of our method to efficiently utilize the available query budget and induce a model with good generalization capability in applications such as text mining, where annotating a single data instance can be tedious and labor-intensive.

Study of Subinstance Granularity

In this experiment, we studied the effect of the subinstance granularity (number of subinstance sizes K) on the AL performance. We studied 4 different granularities: $G1 = \{30\%, 70\%, 100\%\}$, $G2 = \{20\%, 40\%, 60\%, 80\%, 100\%\}$, $G3 = \{20\%, 30\%, 40\%, 60\%, 70\%, 90\%, 100\%\}$ and $G4 = \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%, 100\%\}$. Each granularity denotes the set of sizes at which the unlabeled sample can be queried. For instance, in case of $G1$, each unlabeled sample can be queried after the user has analyzed $30\%, 70\%$ or 100% of the sample. The results on the IMDB dataset are presented in Figure 2. Our method outperforms the baselines at all 4 granularity levels. This shows the robustness of our framework to subinstance granularity. This further corroborates the usefulness of our framework to appropriately select the optimal subinstance size from a number of available options, so as to efficiently use the available query budget and produce a model with good generalization capability. The *Dynamic AAL* depicts the best performance among the baseline methods.

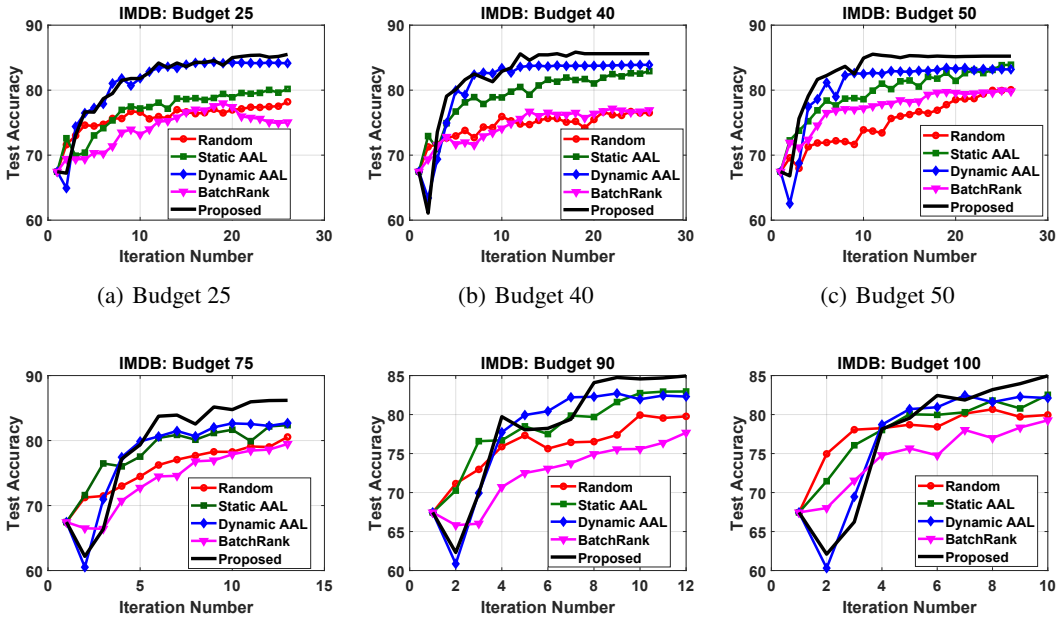
Study of Query Budget

The objective of this experiment was to study the effect of query budget on the AL performance. We studied 6 different budgets $B = \{25, 40, 50, 75, 90, 100\}$. The number of subinstance sizes K was fixed at $5(20\%$ to 100% in steps of $20\%)$. The results on the IMDB dataset are shown in Figure 3. Our method once again depicts impressive performance consistently across all budgets. This result is particularly significant for real-world applications, where the budget is



(a) Granularity G1 (b) Granularity G2 (c) Granularity G3 (d) Granularity G4

Figure 2: Study of subinstance granularity on the IMDB dataset. Each granularity level contains the percentage values upto which a sample can be shown to the human annotators and queried for its label. Please refer to the text for more details. Best viewed in color.



(a) Budget 25 (b) Budget 40 (c) Budget 50

(d) Budget 75 (e) Budget 90 (f) Budget 100

Figure 3: Study of query budget on the IMDB dataset. Best viewed in color.

governed by the available time and other resources, and is different for different applications.

Conclusion and Future Work

In this paper, we proposed a novel active sampling algorithm for text classification which permits subinstance level queries. The active selection of samples and subinstance sizes was posed as a constrained optimization problem based on the uncertainty, diversity, labeling cost and labeling probability criteria, and an LP relaxation was derived to solve the same. Our extensive empirical studies on six challenging datasets from the text mining domain depicted the usefulness of our framework over competing baselines. We hope this research will foster the development of other AL algorithms for temporal data where the labeling convenience of the human oracles is of critical importance. As part of future work, we plan to extend our algorithm to handle noisy oracles, who can provide incorrect labels in addition to abstaining from labeling. While we used LR as the base model in this work to be consistent with (Loaiza, Culotta, and Bilgic 2014), we also plan to study the performance of our algorithm with deep neural networks, which have depicted impressive results in text classification (Minaee et al. 2021).

Acknowledgments

This research is supported in part by the AWS Machine Learning Research Awards Program.

References

Bhattacharya, A.; Liu, J.; and Chakraborty, S. 2019. A Generic Active Learning Framework for Class Imbal-

ance Applications. In *British Machine Vision Conference (BMVC)*.

Biswas, A.; and Jacobs, D. 2012. Active Image Clustering: Seeking Constraints from Humans to Complement Algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bloodgood, M.; and Callison-Burch, C. 2010. Bucking the Trend: Large-Scale Cost-Focused Active Learning for Statistical Machine Translation. In *Association for Computational Linguistics (ACL)*.

Chakraborty, S. 2020. Asking the Right Questions to the Right Users: Active Learning with Imperfect Oracles. In *AAAI Conference on Artificial Intelligence*.

Chakraborty, S.; Balasubramanian, V.; Sun, Q.; Panchanathan, S.; and Ye, J. 2015. Active Batch Selection via Convex Relaxations with Guaranteed Solution Bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(10): 1945–1958.

Deng, Y.; Chen, K.; Shen, Y.; and Jin, H. 2018. Adversarial Active Learning for Sequence Labeling and Generation. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Freund, Y.; Seung, S.; Shamir, E.; and Tishby, N. 1997. Selective Sampling Using the Query by Committee Algorithm. *Machine Learning*, 28(2-3): 133–168.

Gorriz, M.; Carlier, A.; Faure, E.; and i Nieto, X. G. 2017. Cost-Effective Active Learning for Melanoma Segmentation. In *Neural Information processing Systems (NeurIPS) Workshop*.

Guo, Y. 2010. Active Instance Sampling via Matrix Partition. In *Neural Information Processing Systems (NeurIPS)*.

Guo, Y.; and Schuurmans, D. 2007. Discriminative Batch Mode Active Learning. In *Neural Information Processing Systems (NeurIPS)*.

Hoi, S.; Jin, R.; Zhu, J.; and Lyu, M. 2008. Semi-supervised SVM batch mode active learning for image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu, P.; Lipton, Z.; Anandkumar, A.; and Ramanan, D. 2019. Active Learning with Partial Feedback. In *International Conference on Learning Representations (ICLR)*.

Huang, S.; Chen, J.; Mu, X.; and Zhou, Z. 2017. Cost-Effective Active Learning from Diverse Labelers. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Huang, S.; Jin, R.; and Zhou, Z. 2014. Active Learning by Querying Informative and Representative Examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36.

Joshi, A.; Porikli, F.; and Papanikolopoulos, N. 2010. Breaking the Interactive Bottleneck in Multi-class Classification with Active Selection and Binary Feedback. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Loaiza, M.; Culotta, A.; and Bilgic, M. 2014. Anytime Active Learning. In *AAAI Conference on Artificial Intelligence*.

Maas, A.; Daly, R.; Pham, P.; Huang, D.; Ng, A.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *Association for Computational Linguistics (ACL)*.

Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; and Gao, J. 2021. Deep Learning based Text Classification: A Comprehensive Review. *ACM Computing Surveys*, 54(3): 1–40.

Molino, A.; Boix, X.; Lim, J.; and Tan, A. 2017. Active Video Summarization: Customized Summaries via On-line Interaction with the User. In *Association for the Advancement of Artificial Intelligence (AAAI)*.

Nigam, K.; Thrun, A. M. S.; and Mitchell, T. 1998. Learning to Classify Text from Labeled and Unlabeled Documents. In *Innovative Applications of Artificial Intelligence (IAAI)*.

Pang, B.; and Lee, L. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Association for Computational Linguistics (ACL)*.

Pang, B.; and Lee, L. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Association for Computational Linguistics (ACL)*.

Pimentel, T.; Monteiro, M.; Veloso, A.; and Ziviani, N. 2020. Deep Active Learning for Anomaly Detection. In *IEEE International Joint Conference on Neural Networks (IJCNN)*.

Qian, B.; Wang, X.; Wang, F.; Li, H.; Ye, J.; and Davidson, I. 2013. Active Learning from Relative Queries. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Ruchansky, N.; Crovella, M.; and Terzi, E. 2015. Matrix Completion with Queries. In *ACM Conference on Knowledge Discovery and Data Mining (KDD)*.

Sener, O.; and Savarese, S. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *International Conference on Learning Representations (ICLR)*.

Settles, B. 2010. Active Learning Literature Survey. In *Technical Report 1648, University of Wisconsin-Madison*.

Sinha, S.; Ebrahimi, S.; and Darrell, T. 2019. Variational Adversarial Active Learning. In *IEEE International Conference on Computer Vision (ICCV)*.

Sriperumbudur, B.; Gretton, A.; Fukumizu, K.; Scholkopf, B.; and Lanckriet, G. 2010. Hilbert Space Embeddings and Metrics on Probability Measures. *Journal of Machine Learning Research (JMLR)*, 11: 1517 – 1561.

Su, J.; Tsai, Y.; Sohn, K.; Liu, B.; Maji, S.; and Chandraker, M. 2020. Active Adversarial Domain Adaptation. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Tong, S.; and Koller, D. 2001. Support Vector Machine Active Learning with Applications to Text Classification. *Journal of Machine Learning Research (JMLR)*, 2: 45–66.

Wei, Q.; Chen, Y.; Salimi, M.; Denny, J.; Mei, Q.; Lasko, T.; Chen, Q.; Wu, S.; Franklin, A.; and Cohen, T. 2019. Cost-aware Active Learning for Named Entity Recognition in Clinical Text. *Journal of the American Medical Informatics Association (JAMIA)*, 26(11): 1314 – 1322.

Xiong, S.; Pei, Y.; Rosales, R.; and Fern, X. 2015. Active Learning from Relative Comparisons. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 27(12).

Yoo, D.; and Kweon, I. 2019. Learning Loss for Active Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, B.; Li, L.; Yang, S.; Wang, S.; Zha, Z.; and Huang, Q. 2020. State-relabeling Adversarial Active Learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhu, J.; and Bento, J. 2017. Generative Adversarial Active Learning. In *Workshop at Neural Information processing Systems (NeurIPS-W)*.