Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence

Anytime Active Learning

Maria E. Ramirez-Loaiza, Aron Culotta, and **Mustafa Bilgic**
Illinois Institute of Technology
Chicago, IL 60616
mramire8@hawk.iit.edu, {aculotta, mbilgic}@iit.edu

Abstract

A common bottleneck in deploying supervised learning systems is collecting human-annotated examples. In many domains, annotators form an opinion about the label of an example incrementally — e.g., each additional word read from a document or each additional minute spent inspecting a video helps inform the annotation. In this paper, we investigate whether we can train learning systems more efficiently by requesting an annotation before inspection is fully complete — e.g., after reading only 25 words of a document. While doing so may reduce the overall annotation time, it also introduces the risk that the annotator might not be able to provide a label if interrupted too early. We propose an anytime active learning approach that optimizes the annotation time and response rate simultaneously. We conduct user studies on two document classification datasets and develop simulated annotators that mimic the users. Our simulated experiments show that anytime active learning outperforms several baselines on these two datasets. For example, with an annotation budget of one hour, training a classifier by annotating the first 25 words of each document reduces classification error by 17% over annotating the first 100 words of each document.

Introduction

Active learning is a machine learning approach that seeks to maximize classifier accuracy while minimizing the effort of human annotators (Settles 2012). This is typically done by prioritizing example annotation according to the utility to the classifier.

In this paper, we begin with the simple observation that in many domains human annotators form an opinion about the label of an example incrementally. For example, while reading a document, an annotator makes a more informed decision about the topic assignment as each word is read. Similarly, in video classification the annotator becomes more certain of the class label the longer she watches the video.

The question we ask is whether we can more efficiently train a classifier by interrupting the annotator to ask for a label, rather than waiting until the annotator has fully completed her inspection. For example, in document classification the active learner may request the label after the annotator has read the first 50 words of the document. For video classification, the active learner may decide to show only a short clip. We refer to this approach as *anytime active learning* (AAL), by analogy to anytime algorithms, whose execution may be interrupted at any time to provide an answer.

If the decision of when to interrupt the annotator is made optimally, we can expect to reduce total annotation effort by eliminating unnecessary inspection time that does not affect the returned label. However, the annotator may not be able to provide a label if interrupted too early – e.g., the annotator will not know how to label a document after seeing only the first word. AAL strategies, then, must balance two competing objectives: (1) the time spent annotating an instance (*annotation cost*); (2) the likelihood that the annotator will be able to produce a label (*annotation response rate*). In this paper, we propose and evaluate a number of anytime active learning strategies applied to the domain of document classification. In this domain, it is natural to implement this approach by revealing only the first k words to the annotator, which we refer to as a *subinstance*.

We first conduct user studies to estimate annotation times and response rates, and then create simulated oracles that mimic the human annotators. We perform simulated-oracle experiments on two document classification tasks, comparing two classes of anytime active learning strategies: (1) *static* strategies select subinstances of a fixed size; (2) *dynamic* strategies select subinstances of varying sizes, optimizing cost and response rate simultaneously. Our research questions and answers are as follows:

RQ1. How does subinstance size affect human annotation time and response rate? We conducted a user study in which each user labeled 480 documents from two domains under different interruption conditions (e.g., seeing only the first k words). We find that as subinstance sizes increase, both response rates and annotation times increase (non-linearly), and that the rate of increase varies by dataset.

RQ2. How do static AAL strategies compare with traditional active learning? We find that simple static strategies result in significantly more efficient learning, even with few words shown per document. For example, with an annotation budget of one hour, labeling only the first 25 words of each document reduces classification error by 17% compared with labeling the first 100 words of each document.

RQ3. How do dynamic AAL strategies compare with static strategies? The drawback of the static strategy is that we must select a subinstance size ahead of time; however, we find that the optimal size varies by dataset. Instead, we formulate a *dynamic* AAL algorithm to minimize cost while maximizing response rates. We find that this dynamic approach performs as well or better than the best static strategy, without the need for additional tuning.

The remainder of the paper is organized as follows: we first formalize the anytime active learning problem, then propose static and dynamic solutions. Next, we describe our user studies and how they are used to inform our simulation experiments. Finally, we present the empirical results and discuss their implications.

Anytime Active Learning (AAL)

In this section, we first review standard active learning and then formulate our proposed anytime extension.

Problem Formulation

Let $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ be a labeled dataset where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector and $y_i \in \{y^0, y^1\}$ is its class label.[1] Let $\mathcal{U} = \{\mathbf{x}_i\}_{i=l+1}^{m}$ be a set of unlabeled examples. Let $P_{\mathcal{L}}(y|\mathbf{x})$ be the conditional probability of y given $\mathbf{x}$ according to a classifier trained on $\mathcal{L}$.

Typical pool-based active learning selects instances $\mathcal{U}^* \subseteq \mathcal{U}$ to be labeled by a human annotator (*oracle*) and appended to $\mathcal{L}$. Assuming a prespecified annotation budget B and an annotation cost function $C(\mathbf{x})$, the goal of the active learning algorithm (*student*) is to select $\mathcal{U}^*$ to minimize the classifier's generalization error subject to the budget constraints:

$$\mathcal{U}^* \leftarrow \underset{\mathcal{U}_i \subseteq \mathcal{U}}{\operatorname{argmin}} \, Err(P_{\mathcal{L} \cup \mathcal{U}_i}(y|\mathbf{x})) \text{ s.t. } \sum_{\mathbf{x}_j \in \mathcal{U}_i} C(\mathbf{x}_j) \leq B \tag{1}$$

Equation 1 is typically optimized by greedy algorithms, selecting one or more examples at a time according to some heuristic criterion that estimates the utility of each labeled example. A common approach is to request a label for the unlabeled instance that maximizes benefit-cost ratio: $\mathbf{x}_i^* \leftarrow \arg\max_{\mathbf{x}_i \in \mathcal{U}} \frac{U(\mathbf{x}_i)}{C(\mathbf{x}_i)}$.

Various definitions of utility $U(\cdot)$ are used in the literature, such as expected error reduction (Roy and McCallum 2001) and classifier uncertainty (Lewis and Gale 1994).

We propose an alternative formulation of the active learning problem in which the student has the added capability of interrupting the human oracle to request a label while the annotation of $\mathbf{x}_i$ is being performed. For example, in video classification, the student may request a label after the oracle has spent only one minute watching the video. Similarly, in document classification the student may request a label after the oracle has read only the first ten words of a document.

[1] We assume binary classification for ease of presentation; this is not a fundamental limitation.

Let $\mathbf{x}_i^k$ indicate this abbreviated instance, which we call a *subinstance*. The nature of subinstances will vary by domain. For example, k could indicate the time allotted to inspect the instance. In this paper, we focus on document classification, where it is natural to let $\mathbf{x}_i^k$ be the first k words of document $\mathbf{x}_i$.

The potential savings from this approach arises from the assumption that $C(\mathbf{x}_i^k) < C(\mathbf{x}_i)$; that is, subinstances are less costly to label than instances. While the magnitude of these savings are data-dependent, our user studies below show substantial savings for document classification.

The immediate problem with this approach is that $\mathbf{x}_i^k$ may be considerably more difficult for the oracle to label. We therefore must account for imperfect oracles (Donmez and Carbonell 2008; Yan et al. 2011). There are at least two scenarios to consider — (1) a *noisy* oracle produces a label for any $\mathbf{x}_i^k$, but that label may be incorrect; (2) a *reluctant* oracle may decide not to produce a label for some examples, but labels that are produced are assumed to be correct. Our user studies below suggests that the latter case is more common; thus, in this paper, we restrict our attention to reluctant oracles, leaving noisy oracles for future work.

In each interaction between the student and oracle, the student presents a subinstance $\mathbf{x}_i^k$ to the oracle, and the oracle returns an answer $a \in \{y^0, y^1, n\}$, where the answer can be either the correct label y or neutral, n, which represents an "I don't know" answer. If the oracle returns a non-neutral answer a for $\mathbf{x}_i^k$, the student adds $\mathbf{x}_i$ and the returned label (y^0 or y^1) to its training data and updates its classifier. If n is returned, the labeled data is unchanged. In either case, the annotation cost $C(\mathbf{x}_i^k)$ is deducted from the student's budget because the oracle spends time inspecting $\mathbf{x}_i^k$ even if she returns a neutral label. To choose the optimal subinstance, the student must consider both the cost of the subinstance as well as the likelihood that a non-neutral label will be returned. Below, we propose two AAL strategies.

Static AAL Strategies

We first consider a simple, static approach to AAL that decides a priori on a fixed subinstance size k. For example, the student fixes $k = 10$ and presents the oracle subinstances $\mathbf{x}_i^{10}$ (please see Algorithm 1).

Let $\mathcal{U}^k = \{x_i^k\}_{i=l+1}^{m}$ be the set of all unlabeled subinstances of fixed size k. In SELECTSUBINSTANCE (line 3), the student picks $\mathbf{x}_i^{k*}$ as follows:

$$\mathbf{x}_i^{k*} \leftarrow \underset{\mathbf{x}_i^k \in \mathcal{U}^k}{\arg\max} \frac{U(\mathbf{x}_i)}{C(\mathbf{x}_i^k)} \tag{2}$$

Note that the utility is computed from the full instance $\mathbf{x}_i$, not the subinstance, since $\mathbf{x}_i$ will be added to our labeled set (line 8). In our experiments, we consider two utility functions: uncertainty (`static-k-unc`), which sets $U(\mathbf{x}_i^k) = 1 - \max_y P_{\mathcal{L}}(y|\mathbf{x}_i)$, and constant (`static-k-const`), which sets the utility of each subinstance to one, $U(\mathbf{x}_i^k) = 1$. We use `static-k-const` as a baseline for other AAL methods because it is an anytime version of random sampling.

Algorithm 1 Static Anytime Active Learning

1: **Input:** Labeled data $\mathcal{L}$; Unlabeled data $\mathcal{U}$; Budget B; Classifier $P(y|\mathbf{x})$; Subinstance size k
2: **while** $B > 0$ **do**
3: $\quad \mathbf{x}_i^k \leftarrow$ SELECTSUBINSTANCE($\mathcal{U}$, k)
4: $\quad \mathcal{U} \leftarrow \mathcal{U} \setminus \{\mathbf{x}_i\}$
5: $\quad a \leftarrow$ QUERYORACLE($\mathbf{x}_i^k$)
6: $\quad B \leftarrow B - C(\mathbf{x}_i^k)$
7: $\quad$ **if** $a \neq n$ **then** *// Non-neutral response*
8: $\quad\quad \mathcal{L} \leftarrow \mathcal{L} \cup (\mathbf{x}_i, a)$
9: $\quad\quad P(y|\mathbf{x}) \leftarrow$ UPDATECLASSIFIER($\mathcal{L}, P$)

Algorithm 2 Dynamic Anytime Active Learning

1: **Input:** Labeled data $\mathcal{L}$; Unlabeled data $\mathcal{U}$; Budget B; Classifier $P(y|\mathbf{x})$; Neutrality classifier $Q(z|\mathbf{x}_i^k)$; Neutrality labeled data $\mathcal{L}^z \leftarrow \emptyset$.
2: **while** $B > 0$ **do**
3: $\quad \mathbf{x}_i^k \leftarrow$ SELECTSUBINSTANCE($\mathcal{U}$)
4: $\quad \mathcal{U} \leftarrow \mathcal{U} \setminus \{\mathbf{x}_i\}$
5: $\quad a \leftarrow$ QUERYORACLE($\mathbf{x}_i^k$)
6: $\quad B = B - C(\mathbf{x}_i^k)$
7: $\quad$ **if** $a \neq n$ **then** *// Non-neutral response*
8: $\quad\quad \mathcal{L} \leftarrow \mathcal{L} \cup (\mathbf{x}_i, a)$
9: $\quad\quad P(y|\mathbf{x}) \leftarrow$ UPDATECLASSIFIER($\mathcal{L}, P$)
10: $\quad \mathcal{L}^z \leftarrow \mathcal{L}^z \cup (\mathbf{x}_i^k,$ ISNEUTRAL$(a))$
11: $\quad Q(z|\mathbf{x}_i^k) \leftarrow$ UPDATECLASSIFIER($\mathcal{L}^z, Q$)

Dynamic AAL Strategies

The static strategy ignores the impact that k has on the likelihood of obtaining a neutral label from the oracle. In this section, we propose a dynamic strategy that models this probability directly and uses it to guide subinstance selection (see Algorithm 2).

Let $Q(z|\mathbf{x}_i^k)$ be the probability distribution that models whether the oracle will return an "I don't know" answer (i.e. a neutral label) for the subinstance $\mathbf{x}_i^k$, where $z \in \{n, \neg n\}$. The objective of SELECTSUBINSTANCE (line 3) is to select the subinstance that maximizes utility and the probability of obtaining a non-neutral label, $\neg n$, while minimizing cost.

$$\mathbf{x}_i^{k*} \leftarrow \underset{\mathbf{x}_i^k \in \mathcal{U}^k \in \mathcal{S}}{\arg\max} \frac{U(\mathbf{x}_i) Q(z = \neg n|\mathbf{x}_i^k)}{C(\mathbf{x}_i^k)} \quad (3)$$

In contrast to the static approach, the dynamic algorithm searches over an expanded set of p different subinstance sizes: $\mathcal{S} = \{\mathcal{U}^{k_1} \ldots \mathcal{U}^{k_p}\}$.

The immediate question is how to estimate $Q(z|\mathbf{x}_k^i)$. We propose a supervised learning approach using the previous interactions with the oracle as labeled examples. That is, we maintain an auxiliary binary labeled dataset $\mathcal{L}^z$ containing $(\mathbf{x}_i^k, z_i)$ pairs (line 10), indicating whether subinstance $\mathbf{x}_i^k$ received a neutral label or not.[2] This dataset is used to train the neutrality classifier $Q(z|\mathbf{x}_k^i)$ (line 11). Algorithm 2 outlines this approach, where ISNEUTRAL maps the oracle answer to z (i.e., n or $\neg n$). As in the static strategy, we consider two settings of the utility function: uncertainty (`dynamic-unc`) and constant (`dynamic-const`). While both `dynamic-unc` and `dynamic-const` consider the cost of annotation, `dynamic-unc` balances utility with the chance of receiving a non-neutral label, while `dynamic-const` simply maximizes the chance of a non-neutral label.

[2]Mazzoni, Wagstaff, and Burl (2006) use a similar approach to identify "irrelevant" examples.

	Time (secs)		% Neutral	
Size	**IMDB**	**SRAA**	**IMDB**	**SRAA**
10	5.7	5.2	53	50
25	8.2	6.5	33	38
50	10.9	7.6	24	42
75	15.9	9.1	12	22
100	16.7	10.3	13	13

Table 1: User study results reporting average annotation time in seconds and percent of neutral labels by subinstance size.

Experimental Evaluation

Our experiments use two datasets: (1) **IMDB:** A collection of 50K reviews from IMDB.com labeled with positive or negative sentiment (Maas et al. 2011); (2) **SRAA:** A collection of 73K Usenet articles labeled as related to aviation or auto documents (Nigam et al. 1998).

User Studies

To estimate the real-world relationships among subinstance size, annotation time, and response rate, we first performed several user studies in which subjects were shown document subinstances of varying sizes and asked to provide a correct label or an "I don't know" answer (i.e., a neutral label).

Each user performed six classification tasks per dataset, labeling document subinstances of sizes $\{10, 25, 50, 75, 100, \text{All}\}$. For example, to create the 50-word task, we truncated the documents to the first 50 words. For each classification task, the users were asked to annotate 20 randomly-chosen documents from each class, resulting in 40 annotations per task. The documents were presented to the users in random order. For every subinstance, we recorded the annotation time, the number of words seen, and the label. We used the average over five users on the IMDB and three users on the SRAA data.

Table 1 shows the average annotation time (in seconds) and average percentage of neutral labels returned for each subinstance size. We find that the annotation time varies by subinstance size and dataset. For instance, in IMDB annotation time of subinstances of size 50 is 25% greater than for subinstances of size 25. These responses are influenced by the user experience and domain knowledge familiarity, among other factors.

Intuitively, the annotator will be more likely to provide a non-neutral label when he can see more of a document. This intuition was confirmed by our user studies. Table 1 shows that the percentage of neutral labels decreases as subinstance

size increases. However, the rate at which the neutral answer decreases differs by dataset. For example, there was approximately a 50% neutral rate on both datasets for subinstances with 10 words; yet for 75 words the neutral responses were 12% on IMDB and 22% on SRAA. We speculate that the SRAA dataset is a more specialized domain, whereas classifying movie reviews (IMDB) is easier for non-expert human annotators.

Simulations

We use the results of the user study to inform our large-scale studies on the two datasets.

Oracle In order to compare many AAL strategies at scale, it is necessary to simulate the actions of the human annotators. Specifically, we must simulate for which examples the annotator will return a neutral label. We wanted to better reflect the fact that the lexical content of each subinstance influences its neutrality instead of random neutrality — e.g., if a subinstance has strong sentiment words it is not likely to be labeled neutral. To accomplish this, we trained two oracles (one per dataset) that mimic the human annotators. We simulated the oracle with a classifier trained on held-out data; a neutral label is returned when the class posterior probability for a subinstance $\mathbf{x}_i^k$ is below a specified threshold. We tune this classifier so that the pattern of neutral labels matches that observed in the user study.

At the start of each experiment we fit a logistic regression classifier on a held-out labeled dataset (25K examples for IMDB; 36K for SRAA). We use L1 regularization controlled by penalty C to encourage sparsity. When the oracle is asked to label a subinstance $\mathbf{x}_i^k$, we compute the posterior probability with respect to this classifier and compute oracle's uncertainty on $\mathbf{x}_i^k$ as $1 - \max_y P(y|\mathbf{x}_i^k)$. If the uncertainty is greater than a specified threshold T, then the oracle returns a neutral label. Otherwise, the true label is returned.

For each of the datasets, we set C and T so that the distribution of neutral labels by subinstance size most closely matches the results of the user study. We searched values $C \in [0.001, 3]$ with 0.001 step and $T \in [0.3, 0.45]$ with 0.05 step, selecting $C = 0.3, T = 0.4$ for IMDB and $C = 0.01, T = 0.3$ for SRAA. Figures 1(a) and 1(b) show the simulated distribution of neutral labels by subinstance size over the same documents from the user study, indicating a close match with human behavior.

To simulate the cost of each annotation, we used a fixed cost equal to the average annotation time from the user study for subinstances of that size — e.g., for all subinstances of size 10 for IMDB, the cost is the average annotation time for all subinstances of size 10 in the IMDB user study. In future work, we will consider modeling annotation time as a function of the lexical content of the subinstance.

Student For the student, we use a logistic regression classifier with L1 regularization using the default parameter $C = 1$, seeded with a labeled set of two examples. At each round of active learning, a subsample of 250 examples are selected uniformly from the unlabeled set $\mathcal{U}$. Following the user study, subinstances of sizes $\{10, 25, 50, 75, 100\}$ are created for each example in the subsample and scored according to the appropriate strategy (Equation 2 for static; Equation 3 for dynamic). We reserve half of the data for testing, and use the remaining to simulate active learning. For all methods, we report the average result of 10 trials.

For both datasets, we used documents that at least contain 100 words. We created binary feature representations of the documents, using stemmed n-grams (sizes one to three), pruning n-grams appearing in fewer than five documents. In SRAA, we filtered header information, preserving only the subject line and body of the messages.

Results and Discussion

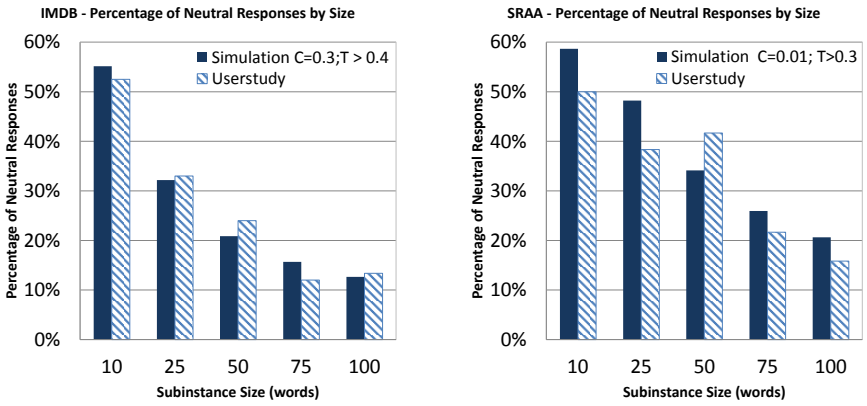
With an oracle simulation and annotation cost in place, we explored the performance of several AAL strategies. We examined learning curves for accuracy and area under the ROC curve (AUC) and observed the same trends and behaviors for each; therefore we include only AUC results here. We summarize our findings below.

Smaller subinstances generally outperform larger subinstances. Figure 2 shows the performance of `static-k-const` for IMDB and SRAA datasets. These results consistently show that savings can be achieved by selecting smaller subinstances. For example, after an hour of annotation (3600 seconds) on IMDB, inspecting the first 100 words of each document results in an AUC of .752; whereas labeling only the first 25 words results in an AUC of .792, a 17% reduction in error. This suggests that while a smaller k results in a high neutral percentage, the time saved by reading shorter documents more than make up for the losses. The results for `static-k-unc` are similar but are omitted due to space limitations.

The optimal subinstance size varies by dataset. Comparing Figure 2(a) to Figure 2(b) indicates that the optimal k^* varies by dataset ($k^* = 25$ for IMDB, $k^* = 50$ for SRAA). This follows from the observed differences between these datasets in the user study (Table 1); i.e., the annotation cost rises more slowly with subinstance size in SRAA. Thus, somewhat bigger subinstances are worth the small additional cost to reduce the likelihood of a neutral label.

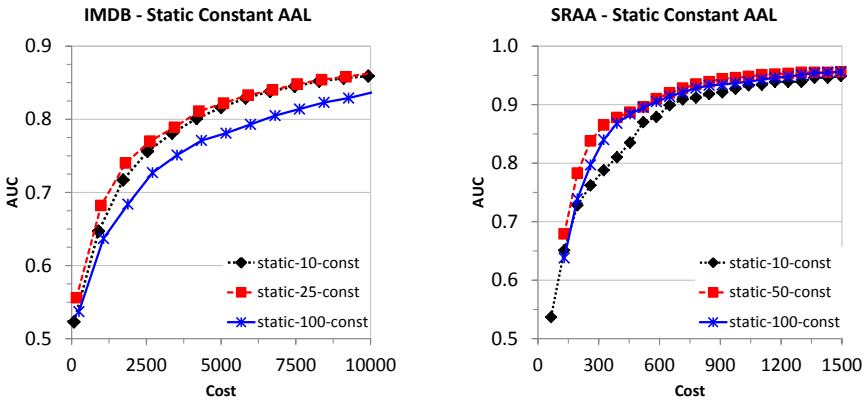
`dynamic-unc` does better than or equal to the best static AAL algorithm. Given the fact that the optimal subinstance size varies by dataset, we examine how the dynamic approach compares to the static approach. Figure 3 compares the dynamic approach with uncertainty and constant utility (`dynamic-unc`, `dynamic-const`) with the best static methods. We find that the `dynamic-unc` outperforms the best static method for the IMDB dataset (Figure 3(a)). In Figure 3(b), the static and dynamic approaches are comparable; however, the advantage of `dynamic-unc` is that there is no need to specify k ahead of time.

Dynamic approaches tend to pick a mixture of subinstance sizes. To better understand the behavior of the dynamic approaches, Figure 4 plots the distribution of subinstance sizes selected by both approaches. As we can see, the dynamic approaches select a mixture of subinstance sizes, but heavily favor smaller sizes. Combining this observation with the results that `dynamic-unc` is either able to outperform or perform comparable to static approaches, this sug-



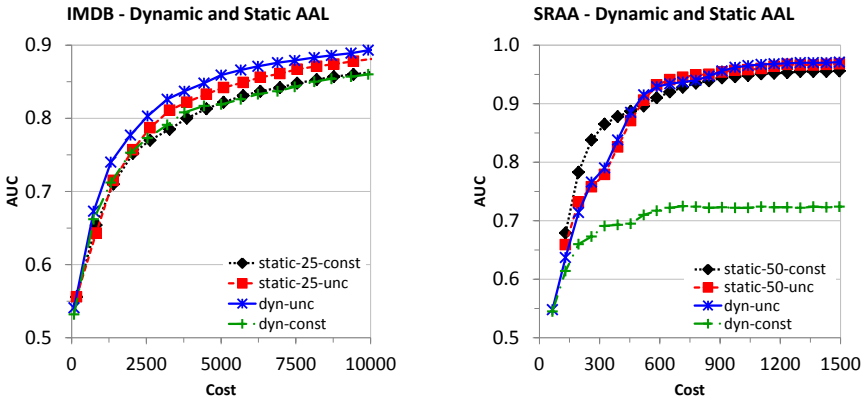
(a) IMDB (b) SRAA

Figure 1: A comparison of the proportion of neutral responses by subinstance size for the user studies and the simulated oracles.



(a) IMDB (b) SRAA

Figure 2: AUC learning curves for `static-k-const`. The trends for `static-k-unc` are similar. The optimal k^* depends on the domain. For IMDB, $k^* = 25$ while for SRAA, $k^* = 50$.

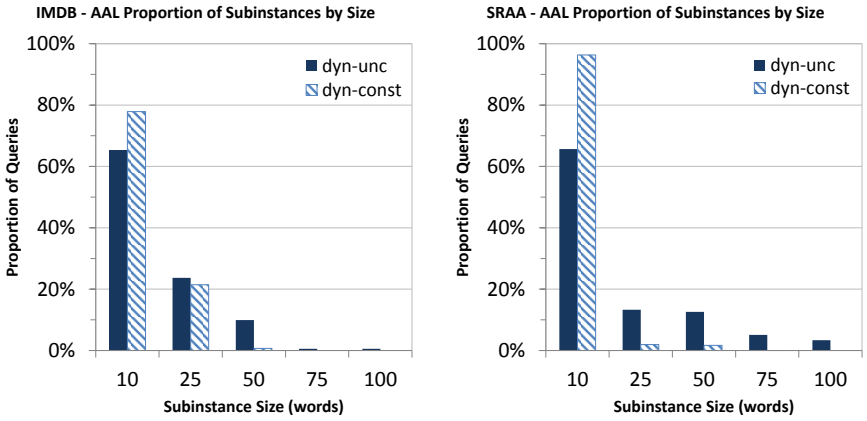


(a) IMDB (b) SRAA

Figure 3: Comparing dynamic AAL to the best of the static AAL approaches. `dynamic-unc` outperforms all methods for IMDB whereas it is comparable to the best static approaches for SRAA.

gests that the dynamic approach is able to pick small subinstances that receive non-neutral labels.

Table 2 further investigates this by comparing the proportion of neutral labels observed with the expected proportion based on the user study. That is, by combining the data from Table 1 and Figure 4, we compute the proportion of neutral labels we expect to see for the observed distribution of subinstance sizes. We can see that the neutrality classifier $Q(z|\mathbf{x}_i^k)$ enables the dynamic approach to select small subinstances while limiting the impact of neutral labels.



(a) IMDB (b) SRAA

Figure 4: Proportion of subinstance sizes selected by dynamic AAL.

		Obs.	Exp.
IMDB	`const`	15%	48%
	`unc`	34%	45%
SRAA	`const`	2%	50%
	`unc`	37%	45%

Table 2: The percentage of observed neutral labels for `dynamic-unc` and `dynamic-const`, compared with what is expected for subinstances of the observed sizes.

Uncertainty improves the exploration of AAL algorithms. In general, we find that using uncertainty outperforms constant utility. For example, in Figure 3, `dynamic-unc` outperforms `dynamic-const` on both datasets. This supports prior work showing the benefits of uncertainty sampling. Additionally, we find that the uncertainty term of the objective balances the neutrality term, enabling a better exploration of the sample space. For example, in the SRAA dataset, Figure 3(b) shows that `dynamic-const` stops learning after a few rounds of active learning. Interestingly, analysis of this result showed that the $Q(z|\mathbf{x}_i^k)$ model quickly learned in the first iterations a strong correlation between non-neutral response and certain terms, e.g. "GPL" (proper name indicating class "auto") in the subject line; `dynamic-const` selected subinstances with those terms thereafter obtaining non-neutral labels for those subinstances. Thus, $Q(z|\mathbf{x}_i^k)$ was further reinforced to predict the subinstances that contain the word "GPL" as non-neutral. This prevented the neutrality model from exploring other terms and the student did not receive a diverse set of documents to improve learning. Including the uncertainty term encouraged the student to seek a more diverse set of examples, thus avoiding this problem.

This is further supported by Table 2, which shows that `dynamic-const` primarily selects instances for which it is very likely to receive a non-neutral label. E.g., in SRAA, only 2% of instances receive a neutral label for `dynamic-const`; Figure 3(b) shows this to be an ineffective strategy, since the student observes only a homogeneous subset of examples. While `dynamic-unc` increases the rate of neutral labels, this additional cost is worth the greater diversity of labeled data.

Related Work

There has been a significant amount of work on noisy and reluctant oracles for traditional active learning scenarios without anytime capability (Donmez and Carbonell 2008; Donmez, Carbonell, and Schneider 2009; Fang, Zhu, and Zhang 2012; Wallace et al. 2011; Yan et al. 2011). Much of this work considered which instance to show to which oracle, and the oracle's quality is fixed. In the anytime framework that we propose in this paper, the student has control over how much time an oracle should spend on an instance, thus controlling the quality of the label.

There has also been a significant amount of work on cost-sensitive active learning (Settles, Craven, and Friedland 2008; Donmez and Carbonell 2008; Haertel et al. 2008; Kapoor, Horvitz, and Basu 2007; Tomanek and Hahn 2010). The common strategy is to use a utility-cost ratio to determine the most cost-effective instances. We follow the same strategy and use utility-cost ratio, with the additional multiplicative factor of probability of non-neutrality.

We build on our previous work (Ramirez-Loaiza, Culotta, and Bilgic 2013) about the problem of searching over subinstances with some notable differences: i) we conducted user studies to determining annotation time, whereas they assumed a linear cost function, ii) we allow the oracles to return a neutral label and model neutrality.

Conclusions and Future Work

We present an anytime active learning framework in which the student is allowed to interrupt the oracle to save annotation time. User studies were conducted to quantify the relationship between subinstance size, annotation time, and response rate. These were used to inform a large-scale simulated study on two document classification tasks, which showed that although interruption can cause the oracle return neutral labels, interrupting at the right time can lead to significantly more efficient learning. We found that optimal interruption time depends on the domain and proposed a dynamic AAL strategy that is better than or comparable to the best static strategy that uses a fixed interruption time.

In the future, we will expand our model of annotation time to account for lexical content. Moreover, we assumed in this paper that the annotator reads the document serially starting from the beginning and hence created subinstances that correspond to the first k words of the document. As future work, we will consider alternative techniques of interruption, such as structured reading (e.g., the first and last sections of a document) and text summarization to speed up annotation.

References

Donmez, P., and Carbonell, J. G. 2008. Proactive learning: : Cost-sensitive active learning with multiple imperfect oracles. In *Proceeding of the 17th ACM conference on Information and Knowledge Mining - CIKM '08*, 619.

Donmez, P.; Carbonell, J. G.; and Schneider, J. 2009. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, 259.

Fang, M.; Zhu, X.; and Zhang, C. 2012. Active Learning from Oracle with Knowledge Blind Spot. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence - ICML '12*.

Haertel, R.; Ringger, E.; Seppi, K.; Carroll, J.; and Mcclanahan, P. 2008. Assessing the Costs of Sampling Methods in Active Learning for Annotation. In *ACL*, 65–68.

Kapoor, A.; Horvitz, E.; and Basu, S. 2007. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *IJCAI*, volume 7, 877–882.

Lewis, D. D., and Gale, W. A. 1994. A sequential algorithm for training text classifiers. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 3–12.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142–150. Portland, Oregon, USA: Association for Computational Linguistics.

Mazzoni, D.; Wagstaff, K.; and Burl, M. 2006. Active learning with irrelevant examples. In *ECML*, 695–702.

Nigam, K.; McCallum, A.; Thrun, S.; and Mitchell, T. 1998. Learning to classify text from labeled and unlabeled documents. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, 792–799.

Ramirez-Loaiza, M. E.; Culotta, A.; and Bilgic, M. 2013. Towards anytime active learning: Interrupting experts to reduce annotation costs. In *KDD Workshop on Interactive Data Exploration and Analytics (IDEA)*.

Roy, N., and McCallum, A. 2001. Toward optimal active learning through sampling estimation of error reduction. In *International Conference on Machine Learning*, 441–448.

Settles, B.; Craven, M.; and Friedland, L. 2008. Active learning with real annotation costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, 1–10.

Settles, B. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool.

Tomanek, K., and Hahn, U. 2010. A comparison of models for cost-sensitive active learning. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, 1247–1255.

Wallace, B. C.; Small, K.; Brodley, C. E.; and Trikalinos, T. A. 2011. Who should label what? instance allocation in multiple expert active learning. *In Proc. of the SIAM International Conference on Data Mining (SDM)*.

Yan, Y.; Fung, G. M.; Rosales, R.; and Dy, J. G. 2011. Active learning from crowds. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 1161–1168.