Modeling Annotation Time to Reduce Workload in Comparative Effectiveness Reviews

Byron C. Wallace[†‡], Kevin Small[†], Carla E. Brodley[†]
Joseph Lau[‡], Thomas A. Trikalinos[‡]
[†]Tufts University, Medford, MA
[‡]Tufts University and Tufts Medical Center, Boston, MA
byron.wallace@tufts.edu, kevin.small@tufts.edu, brodley@cs.tufts.edu
jlau1@tuftsmedicalcenter.org, ttrikalinos@tuftsmedicalcenter.org

ABSTRACT

Comparative effectiveness reviews (CERs), a central methodology of comparative effectiveness research, are increasingly used to inform healthcare decisions. During these systematic reviews of the scientific literature, the reviewers (MD-methodologists) must screen several thousands of citations for eligibility according to a pre-specified protocol. While previous research has demonstrated the theoretical potential of machine learning to reduce the workload in CERs, practical obstacles to deploying such a system remain. In this article, we describe work on an end-to-end, interactive machine learning system for assisting reviewers with the tedious task of citation screening for CERs. Specifically, we present ABSTRACKR, our open-source annotation tool. In addition to allowing reviewers to designate citations as 'relevant' or 'irrelevant' to the review at hand, ABSTRACKR facilitates communicating other information useful to the classification model, such as terms that are suggestive of the relevance (or irrelevance) of a citation. The tool also records the time taken to screen citations, over which we conducted a time-series analysis to derive an annotator model. Using this model, we found that both the order in which the citations are screened and the length of each citation affect annotation time. We propose a strategy that integrates labeled terms and timing data into the *Active Learning* (AL) framework, in which an algorithm selects citations for the reviewer to label. We demonstrate empirically that this additional information can improve the performance of the semi-automated citation screening system.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems—*Medicine and science*
General Terms
Algorithms, Human Factors
Keywords
active learning, medical, applications, text classification

IHI'10, November 11–12, 2010, Arlington, Virginia, USA.
Copyright 2010 ACM 978-1-4503-0030-8/10/11 ...$10.00.

1. INTRODUCTION

Comparative effectiveness reviews (a type of *systematic review*) are increasingly used to inform decisions at all levels of health care, from the bedside to the adoption of national policies. The cornerstone of Evidence-based Medicine, comparative effectiveness and other systematic reviews are unbiased and comprehensive analyses of published research on well-formulated questions that rigorously follow a predefined protocol. Part of the process is to identify and analyze all relevant research that fulfills protocol criteria. To this end, reviewers conduct extensive literature searches (e.g., via PubMed), that typically return several thousand citations, of which only a few dozen will eventually meet eligibility criteria. To assess eligibility, reviewers (generally physician-methodologists, and hence expensive) must manually peruse all of the retrieved citations, designating each as either 'eligible' or 'ineligible' for the review at hand. We refer to this process as *citation screening*. Citation screening is laborious; it takes approximately forty hours of uninterrupted work time to manually screen 5,000 citations, which is the size of a typical review. Larger reviews are not uncommon; the Tufts Evidence-based Practice Center once screened over 33,000 abstracts for a systematic review [1, 2, 3].

Mitigating this workload is naturally an appealing proposition. In previous work we re-formulated the problem as a binary classification task and trained a classifier to automatically screen citations [16]. Classifiers are induced, or 'trained', over a set of labeled instances.[1] In our scenario, the training data comprise citations labeled by a reviewer as either 'relevant' or 'irrelevant', and the aim is to induce a discriminative model that can automatically exclude irrelevant citations, while maintaining high sensitivity to relevant articles. False negatives (i.e., excluded citations that in fact meet the inclusion criteria) are costly because they can jeopardize the comprehensiveness and validity of the conducted review [16].

Our strategy involves training a Support Vector Machine (SVM) classifier interactively, using the *Active Learning* (AL) framework. In AL, the classifier requests labels for those instances likely to be most useful in inducing a good classifier [10]; this is in contrast to the typical, *passive* approach to classifier induction, in which a set of training data is selected at random. The intuition is that by selecting a small, infor-

[1] This paper uses the terms 'instance' and 'example' interchangeably. In this work, they refer to individual biomedical citations/documents; more generally they denote individual representations of the objects being classified.

mative set of labeled data with which to train a classifier, one can induce a better predictive model with less effort compared to the passive approach. AL is an attractive paradigm for machine learning applied to biomedical tasks, as annotation costs are often high (specialized experts tend to be expensive and busy). Furthermore, this requisite level of domain expertise rules out low-cost out-sourced annotation solutions, such as Amazon Turk.[2]

The contributions of this work are as follows. First, we describe an open-source annotation tool currently being used at the Tufts EPC for the citation screening task. We argue that such tools are imperative in deploying real-world AL systems. Indeed, despite substantial empirical successes (e.g., [9, 13]), AL has not yet achieved wide adoption in practice – one reason being a lack of available annotation tools [12]. Our tool also provides an interface for the expert to label terms, i.e., indicate words or n-grams useful in predicting a given document's class. Second, we use empirical labeling data (collected with the aforementioned tool) to develop a novel labeling time prediction model. Finally, we incorporate the predicted time it will take to label an instance into the AL query function. We demonstrate that this approach outperforms more traditional 'greedy' AL strategies, which tacitly assume a uniform per-instance labeling cost. In other words, when the (predicted) time to label an instance is factored into the decision of which examples to have the expert label, a better model can be induced in the same amount of time (i.e., at the same cost).

In the following section, we briefly review the AL framework, as well as some related work in applied AL. We also present our AL algorithm, which exploits *labeled terms*, or words and n-grams that have been designated as being indicative of a document being relevant or irrelevant. In Section 3, we present our open-source annotation tool. In Section 4, we discuss our approach to modeling annotator labeling time, and in Section 5 we give our algorithm for incorporating this into the AL framework. We present our experimental setup and our results in Section 6. Finally, we end with conclusions and future work in Section 7.

2. PRELIMINARIES

2.1 Active Learning to Mitigate Workload

Before deployment, a classifier must first be trained with a set of labeled examples. AL is an increasingly popular technique for mitigating the amount of work required to induce a classifier (for a recent survey, see [10]). AL proceeds iteratively; during each round of training, the learning algorithm uses a querying function to select an instance for labeling by the expert. The idea is that by choosing the training data cleverly, rather than at random, a better classifier can be induced with less work.

We have shown in previous work [15, 16] that AL can substantially reduce the burden on reviewers conducting comparative effectiveness reviews in terms of the number of documents that must be manually screened. As mentioned, an important caveat in the citation screening problem is that sensitivity (recall) with respect to the set of relevant articles is more important than specificity; in other words, false negatives are expensive while false positives are relatively cheap. An additional challenge is the severe class imbalance, as there are far fewer relevant than irrelevant articles (typically only around 5-10% of the documents are relevant). We found that uncertainty sampling, in which the model requests a label for the example it is least certain about, tends to induce classifiers with high accuracy but low sensitivity.

To remedy this, we developed a new AL strategy that incorporates *labeled features*, which in our case are terms, i.e., words or n-grams, designated by the reviewer as indicative of a document being either eligible or ineligible; we refer to these as positive and negative terms, respectively. For example, in a systematic review concerning genetic associations with Chronic Obstructive Pulmonary Disease (COPD), the reviewer indicated that 'allele' and 'copd' were positive whereas 'mice' and 'cell lines' were negative (the review included only human studies). We note that our ABSTRACKR tool, presented in Section 3 provides an interface for labeling terms. Denoting the set of positive terms by $\mathcal{P}^F$, the set of negative terms $\mathcal{N}^F$, and the total count of labeled terms in a document as N_d, we can score documents as follows:

$$N_d \cdot \log \left(\frac{\sum_{w^+ \in \mathcal{P}^F} I_d(w^+) + 1}{\sum_{w^- \in \mathcal{N}^F} I_d(w^-) + 1)} \right) \tag{1}$$

where $I_d(w)$ is indicator function which is 1 if w is in d and 0 otherwise. Note that we add 1 to both the negative and positive sums, to avoid taking the log of 0. Intuitively, if the ratio in Equation 1 is large, it should be obvious as to which category this document belongs, because it must contain many labeled terms, and a preponderance of these belong to a particular class. In this article, we use the scores calculated via Equation 1 as a measure of the value, or utility, of acquiring a label for a particular document. We reiterate that any appropriate score could be substituted here (e.g., one could use a measure of uncertainty); but we have found that this score outperforms other baseline strategies, such as uncertainty sampling, for our problem.

Once unlabeled documents are scored, the natural (greedy) approach is to have the expert label the highest scoring citation. However in Section 6 we demonstrate that the strategy of dividing the scores assigned to unlabeled citations by the predicted time it will take to annotate them outperforms this baseline.

2.2 Related Work on Deployed AL

Historically, work in AL has made a number of unrealistic assumptions. It is generally assumed that there is a single, infallible oracle and that labeling instances incurs some constant cost; hence the usual plot in AL research, which scatters some measure of the induced models' performance versus the number of training labels provided, rather than the actual annotation time. Increasingly, however, focus has turned to relaxing these assumptions. For example, Dommez and Carbonell [6] have developed a framework for cost-sensitive AL with multiple, imperfect labelers. Their model also allows for varying cost, and they demonstrated its potential over simulated data.

Recently, researchers have begun to investigate empirical (real-world) annotation times. Arora et al. [4] demonstrated the feasibility of estimating the cost to label instances, even across different annotators, in a movie review classification

[2] http://www.mturk.com

task. As features, they incorporated information such as the word count (i.e., length) of a movie review. Elsewhere, Baldridge and Palmer [5] emphasized the importance of taking annotator cost and expertise into consideration. They demonstrated that the efficacy of AL can be dramatically different depending on what measure of cost is used (e.g., number of labels provided versus the real annotation time), highlighting the need for cost-sensitive AL in real-world systems.

Most similar to our work here, Settles et al. [11] demonstrated that knowing the (true) annotation time can theoretically increase AL performance, though the model they used for predicting annotation times was not sufficient to improve performance – thus resulting in a negative result for their application. They used the same Return-on-Investment (ROI) strategy recently advocated by Haertel et al. [7], in which the utility computed for an unlabeled example (a measure of its informativeness) is scaled by the the predicted time it will take to label it. Haertel et al. demonstrated that factoring in predicted cost can improve AL performance in a Part of Speech (POS) tagging task. They note that the difficult part is estimating cost and utility functions. Here, we present such functions for the citation screening task, achieving substantial improvements over an already strong AL baseline.

3. ABSTRACKR: AN OPEN-SOURCE ANNOTATION TOOL

We now present our open-source annotation tool, which we call ABSTRACKR[3] (the name is an amalgamation of abstract and tracker).

A screenshot of the primary tool interface is shown in Figure 1. The interface comprises a main window that displays the current abstract text, and a 'control panel' on the bottom, which includes buttons to facilitate annotation. The buttons on the bottom right of the control panel allow the reviewer to 'accept' or 'reject' the current abstract. To the left of those buttons is a navigational component, which allows the user to iterate over citations or jump to a particular study. Finally, at the bottom left are four buttons that facilitate term annotation; one 'thumbs up' corresponds to a weakly positive term, while two 'thumbs up' indicates a term or n-gram strongly indicative of the positive class (relevant articles). Notice that the tool highlights those terms that have been labeled by the user; negative terms are highlighted red and positive terms are highlighted yellow.

In addition to facilitating annotation, ABSTRACKR records labeling time, which we used to model annotation times (see the following section). In Section 6, we show that this model is sufficient to predict labeling times online during AL, and these predicted times can be used to increase the performance of the classifier. All information (annotations, labeling times, etc.) are stored in a SQLite database, and thus can readily be queried. The tool itself is written in the Python programming language, using the QT Graphic User Interface (GUI) library, and it is therefore cross-platform.[4]

Aside from providing an interface to annotate documents, ABSTRACKR provides a natural method of incorporating labeling assignments and our AL algorithm directly into the software, allowing for easy interaction between the human and the machine learning algorithm (although this is presently done offline). That is, we envision a team of experts using the tool in tandem, talking to a central database; the program will decide which citations to assign to whom at each step in the annotation process, taking into consideration the terms highlighted thus far, as well as the currently labeled documents.

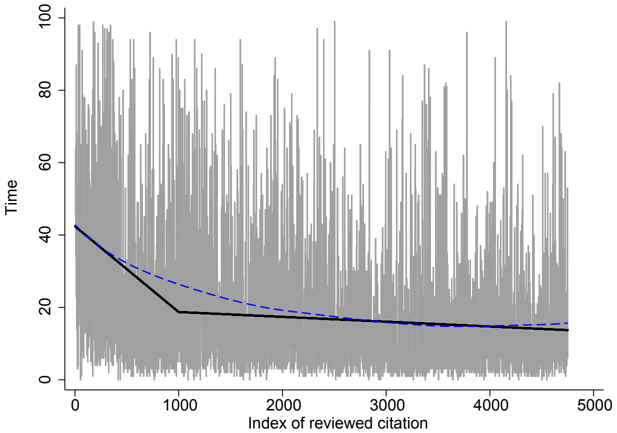


Figure 2: Document labeling time (in seconds) versus the order in which it was labeled.

4. MODELING EXPERTS

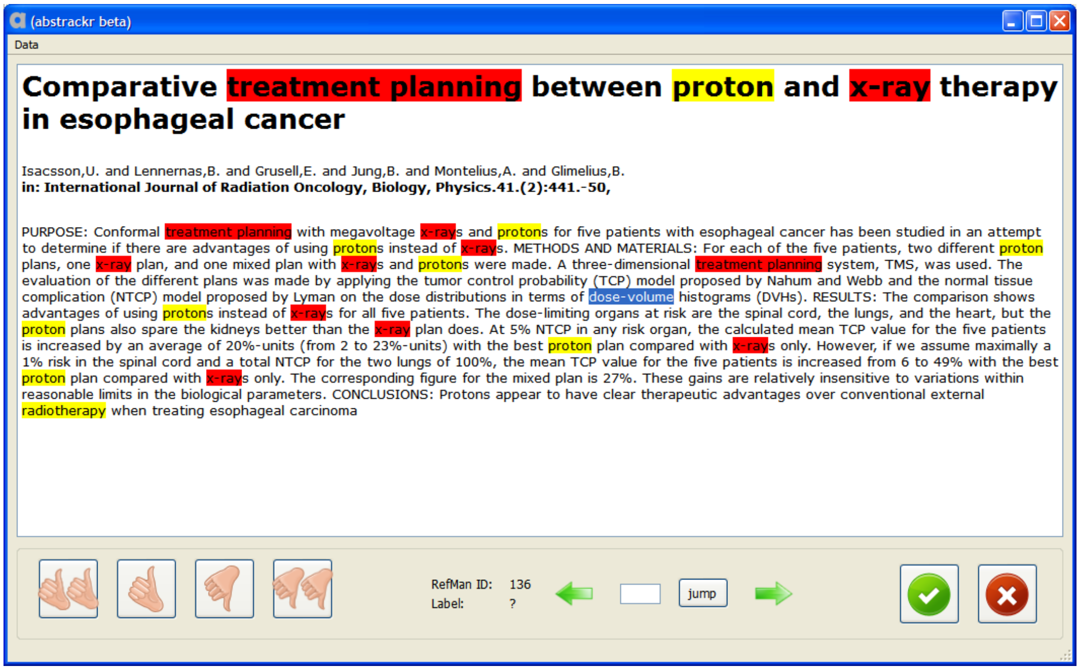
It has been shown that scaling the expected value of attaining a label for a particular instance by the cost (in terms of time) of acquiring said label can improve the performance of AL [11, 7]. However, deriving a statistical model to predict how long it will take to label a given example remains a challenge [4]. Indeed, Settles et al. demonstrated that in certain cases, *if* the true time to annotate were known *then* performance could be improved; however their model was inadequate in predicting labeling times, and thus did not improve performance.

We hypothesized that, on average, annotation would take longer in the beginning of the screening, while the reviewer familiarizes him or herself with the topic and screening criteria, and would gradually decrease thereafter. To the best of our knowledge, no previous work on predicting annotation times has considered annotator learning rate. Furthermore, in line with Settles et al. [11], we assume that longer documents would take longer to annotate. These assumptions were borne out by the empirical data collected from a real-world citation screening project.

Figure 2 shows the relationship between mean annotation time and the order in which abstracts were reviewed. This relationship is shown in the smoothed dashed line, obtained from locally weighted linear regression with a sliding window of width 80% of the observations (lowess smoothing). The clear downward trend is intuitively agreeable; the annotator is learning as they label documents, and their speed thus increases as they become more familiar with the task. Moreover, as evidenced by the plot, their learning rate is more pronounced at the start of the task, and tapers off

[3] The code for the ABSTRACKR tool is available via github at: http://github.com/bwallace/citation-screening/, under the UI subdirectory.

[4] However, at the moment we've only tested it on Windows and Mac OS X.



Comparative treatment planning between proton and x-ray therapy in esophageal cancer

Isacsson,U. and Lennernas,B. and Grusell,E. and Jung,B. and Montelius,A. and Glimelius,B.
in: International Journal of Radiation Oncology, Biology, Physics.41.(2):441.-50,

PURPOSE: Conformal treatment planning with megavoltage x-rays and protons for five patients with esophageal cancer has been studied in an attempt to determine if there are advantages of using protons instead of x-rays. METHODS AND MATERIALS: For each of the five patients, two different proton plans, one x-ray plan, and one mixed plan with x-rays and protons were made. A three-dimensional treatment planning system, TMS, was used. The evaluation of the different plans was made by applying the tumor control probability (TCP) model proposed by Nahum and Webb and the normal tissue complication (NTCP) model proposed by Lyman on the dose distributions in terms of dose-volume histograms (DVHs). RESULTS: The comparison shows advantages of using protons instead of x-rays for all five patients. The dose-limiting organs at risk are the spinal cord, the lungs, and the heart, but the proton plans also spare the kidneys better than the x-ray plan does. At 5% NTCP in any risk organ, the calculated mean TCP value for the five patients is increased by an average of 20%-units (from 2 to 23%-units) with the best proton plan compared with x-rays only. However, if we assume maximally a 1% risk in the spinal cord and a total NTCP for the two lungs of 100%, the mean TCP value for the five patients is increased from 6 to 49% with the best proton plan compared with x-rays only. The corresponding figure for the mixed plan is 27%. These gains are relatively insensitive to variations within reasonable limits in the biological parameters. CONCLUSIONS: Protons appear to have clear therapeutic advantages over conventional external radiotherapy when treating esophageal carcinoma

RefMan ID: 136
Label: ?

Figure 1: The abstrackr annotation tool.

toward the end. There is also clear correlation between document length and labeling time, as shown in Figure 3, which plots the association between document length and annotation time. In both plots, we do not show points for 106 documents (out of 4,751) that had associated labeling times longer than 100 seconds. These were considered outliers (it's likely that the reviewer became distracted while the tool was displaying these abstracts), and they made the plots difficult to read.

In addition to order and document length, we also considered the correlation between model uncertainty, i.e., distance from the induced SVMs' hyperplane, and labeling time. It has been conjectured elsewhere that examples that the model is uncertain about may be in some sense difficult and thus take longer to label [6]. To test this, we induced a model over all of the labeled data, and then computed the distance of each document to the separating hyperplane, a proxy for uncertainty (examples near the hyperplane are those the model is uncertain about [13]). As shown in Figure 4, a correlation between model uncertainty and labeling time exists, but is rather weak compared to the observed correlation between, e.g., document length and labeling time. In particular, Spearman's correlation coefficient for the former is -0.05, whereas for the latter it is 0.39 (P-values <0.001 for both). More problematically, the uncertainty will be extremely unstable at the start of AL, as the hyperplane will readjust dramatically as each new labeled example is acquired. For these reasons, we do not include the uncertainty in our annotation time prediction model.

We performed a regression analysis to predict the average time to annotate each abstract based on the order in which it is screened (i.e., first, second, n-th) and its length. We used a linear spline with a single knot at 1,000 abstracts to approximate the nonlinear relationship depicted by the solid line in Figure 2. Using 1,000 documents for the spline

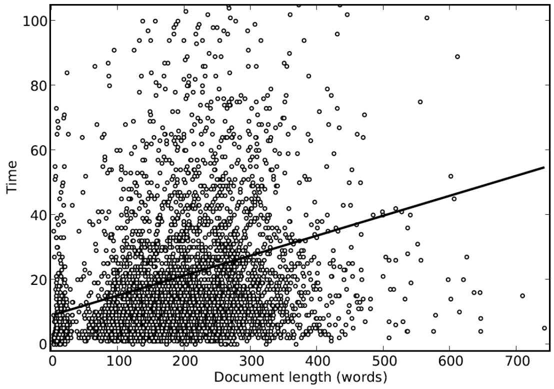


Figure 3: Document labeling time (in seconds) versus length (in words).

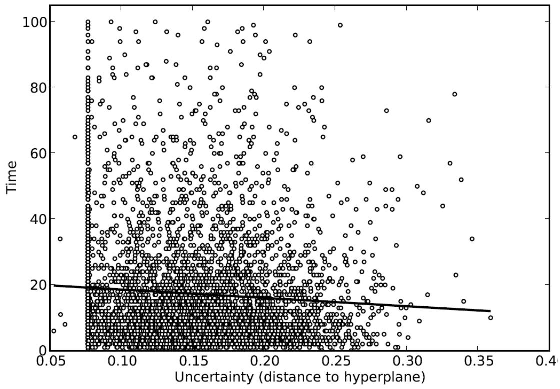


Figure 4: Document labeling time versus its distance to the hyperplane in an SVM induced over the entire dataset.

regression was arbitrary; we just wanted to show that the learning rate increases rapidly at the start of AL and more slowly thereafter. Linear mixed models with autocorrelated errors (to account for similarity of successive abstracts) and with information regarding which abstracts were screened in the same 'session' (to account for 'session'-specific effects) yielded very similar coefficients to those of an ordinary least squares regression, and we therefore used the latter model. Specifically, we model the time to screen a document d as follows:

$$\hat{y}_d(\beta) = \beta_0 + \beta_1 length(d) + \beta_1 n_1 + \beta_2 n_2 \qquad (2)$$

where the n_1 and n_2 variables are functions of the number of documents that have already been labeled, which we will denote by n. Specifically, n_1 is n when fewer than 1,000 documents have been labeled, and fixed at 1,000 thereafter, while n_2 is 0 when fewer when 1,000 documents have been labeled and $n - 1000$ thereafter. This models the desired spline, which reflects the change in the annotator's learning rate.

Of course, while AL is ongoing in practice, β is unknown. We therefore learn an approximation to β, $\hat{\beta}$, online using standard least-squares regression and the annotation times of the documents labeled thus far as target values. We then simply substitute $\hat{\beta}$s for the βs in Equation 2. See Algorithm 1 for more details.

5. ACTIVE LEARNING WITH PREDICTED LABELING TIMES

Our algorithm for AL with predicted labeling times is shown in Algorithm 1. We first use a small sample of labeled data to get an initial estimate of the β coefficients. Additionally, we induce an initial hypothesis with which to begin AL.

At each step in the AL loop, which begins at line 5, we select for labeling the 'best-value' document, i.e., the document with the largest payoff per estimated time unit. This is shown in line 6, where d^* denotes the document selected for labeling by the reviewer. We then have the reviewer label this document, and record the time it took to do so (lines 7 and 8). Next, we re-train our classifier over the newly augmented training set (line 9). Finally, in line 10, we update our estimate of the β coefficients using the document labeling times observed thus far. In this way, we can estimate how long it will take to screen the remaining documents, given their length and the order in which they'll be screened, based on the times taken to screen the documents labeled thus far. This prediction is used as the denominator in line 6.

6. EXPERIMENTAL RESULTS

In this section, we turn our attention to an empirical evaluation of the proposed method. This is meant to demonstrate the advantage of taking into consideration the predicted time-to-label in selecting examples to have annotated in AL, as well as the potential utility of our annotation tool, which facilitates reviewer/computer interaction. In Section 6.1 we discuss our method of evaluation, which is specific to the citation screening problem. Next, in Section 6.2, we outline our experimental setup. Finally, in Section 6.3, we show our empirical results.

Algorithm 1 Active Learning with Labeling Times

```
    Input: Learning algorithm $\mathcal{A}$, scoring function $f$, unlabeled dataset $\mathcal{U}$, labeled data sample $\mathcal{S}_l$, time budget $\mathcal{T}$
 2: $t \leftarrow 0$
    $\hat{\beta} \leftarrow$ least squares estimate using $\mathcal{S}$ {initial estimate of $\beta$ coefficients}
 4: $\hat{h}_t \leftarrow \mathcal{A}(\mathcal{S}_l)$ {learn initial hypothesis}
    while $t < \mathcal{T}$ do
 6:     $d^* \leftarrow \underset{d}{\mathrm{argmax}} \frac{f(d)}{\hat{y}_d(\hat{\beta})}$ over $\mathcal{U}$
        $\mathcal{S}_l \leftarrow \mathcal{S}_l \cup d^*$; $\mathcal{U} \leftarrow \mathcal{U} \backslash d^*$ {label selected point}
 8:     $t \leftarrow t +$ time taken to label $d^*$
        $\hat{h}_t \leftarrow \mathcal{A}(\mathcal{S})$ {rebuild model}
10:     $\hat{\beta} \leftarrow$ least-squares estimate using $\mathcal{S}_l$ {recompute estimate of $\beta$ coefficients using labeled data}
    end while
12: Output: Learned hypothesis $h_t$
```

6.1 Classifier Evaluation

AL algorithms are typically compared by inducing a model over the examples selected by the different strategies and then evaluating these models over a hold-out set of instances. This measures the predictive performance of the induced classifiers. However, the citation screening task is a *finite-pool* scenario, in which we are interested in using a classifier as a means of annotating a fixed set of documents with as little effort as possible. We therefore evaluate AL strategies with respect to this aim.

More specifically, we are interested in two quantities; the burden imposed on reviewers and the number of relevant citations correctly identified. Previously, we have used the number of documents labeled as a measure of the former; here we use the actual labeling time. We refer to the latter as 'yield'; it is the same as sensitivity to the 'relevant' class, except that it takes into account the data with which the model was trained. Thus if an AL querying strategy consistently selects for labeling relevant documents it is 'rewarded' for this behavior. Note that this is not the same as testing on training data; we do not attempt to predict the labels for documents included in the training set. Rather, we are quantifying the fraction of relevant documents correctly identified using a particular strategy, regardless of whether these documents were manually labeled relevant or were correctly predicted to be relevant by the classifier. Without using any machine learning techniques, both burden and sensitivity are 100%, as all relevant citations are identified, at the expense of the reviewers manually perusing all of the citations.

To get a single measure of performance (and thus be able to compare strategies), these metrics must be combined. However, as mentioned, maximizing yield is more important than minimizing burden, and therefore taking a mean is not appropriate, because this tacitly assigns equal weights to both. Rather, we want a weighted mean that incorporates the relative importance of identifying all relevant citations versus reducing the workload. We call this measure U_λ, where λ is a scalar that encodes this tradeoff. Formally, we have:

$$U_\lambda = \frac{\lambda \cdot yield + (1 - workload)}{\lambda + 1} \qquad (3)$$

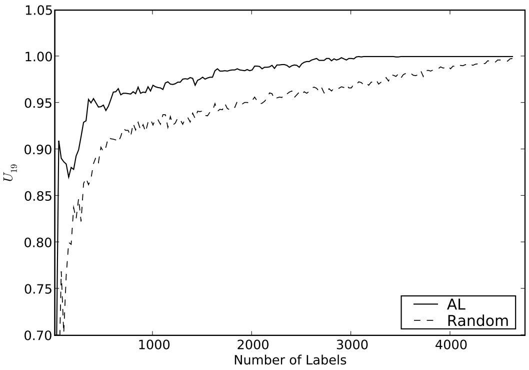


Figure 5: Classifier performance of AL and passive learning.

Where both yield and the measure of the workload are assumed to be normalized to fall in the range [0, 1]. As mentioned, here we use the annotation time spent by the reviewer as a measure of the workload.

The question now becomes what value to assign to λ, i.e., how to quantify the aforementioned tradeoff. Asking the expert (reviewer) directly to provide λ does not work, because it's difficult to intuit this parameter. In previous work we proposed using a method from Medical Decision Theory to quantify the analogous costs for diagnostic tests [14] to elicit this parameter from the expert in a natural way by means of a thought experiment and some algebra. For details on this method, see [15]. In our case, this resulted in a λ of 19; i.e., maximizing yield is 19 times as important as minimizing the workload, according to the reviewer.

6.2 Experimental Setup

We compare the strategy of taking into account the predicted time it will take label a document when selecting examples with a strong baseline strategy that we have previously shown to outperform random sampling [15]. Figure 5 plots U_{19}, the measure of performance described above, against the number of instances labeled, i.e., the size of the training set used to induce the classifier. In this case, we quantify workload by the number of documents that must be screened by a reviewer. This includes the number of labeled documents and the number of documents predicted to be 'relevant' by the induced model, as these will need to be screened (whereas those documents that are designated 'irrelevant' by the classifier needn't be screened). Given this result, we use our previously developed AL strategy, rather than random, as our baseline.

To measure workload, we would like to use the actual time spent screening citations, rather than the raw number of documents screened. This is a bit tricky, however, because the time it will take to screen a particular citation is at least partially a function of the order in which it is screened (see Figure 2). Thus we cannot simply use the raw observed screening times in our experiments, because those times make sense only when the documents are labeled in the order in which the reviewer originally screened them. Therefore, to calculate the time spent labeling a citation for our experiments (line 8 in Algorithm 1) we use an order-adjusted time.

Denoting the raw observed time taken to label a document d by t_d, we have: $r_d = \hat{y}_d(\beta) - t_d$, where here we use the original order in which d was labeled for n (see Equation 2). Then r_d is the residual time taken to screen a citation, unaccounted for by our model. We then recompute $\hat{y}_d(\beta)$, setting n equal to the number of documents labeled thus far in the ongoing experiment, and substract from this the residual, r_d.

There is one additional factor that complicates our evaluation; in addition to totaling the time spent labeling, we must take into account the amount of time it will take to label the documents that were predicted to be relevant. However, the 'true' annotation times for these documents (computed as described above) will be partially contingent on the order in which they are screened. To eliminate this issue, we first sort all of the documents classified as 'relevant' by the model in descending order of length, and then simulate labeling them in this order. Finally, we compute a normalization constant for workload as follows (recall that it is expected to fall between 0 and 1): sort all of the documents in descending order of document length, and sum the (simulated) time taken to label them in this order.

To recapitulate, we quantify performance using the U_λ metric, which is a weighted mean of the two quantities of interest: yield and burden. The former is the fraction of relevant citations correctly identified, the latter is a measure the total reviewer workload. In this case, we quantify workload by the total labeling time, which includes the time taken to label the training set, as well as the time taken to screen the citations categorized as 'relevant' by the classifier. The λ in this case was elicited from a reviewer, as we have described elsewhere [15]. A final note on evaluation: because we have extreme 'class imbalance', i.e., there are far fewer relevant than irrelevant citations, we under-sample the majority class of irrelevant citations before training our classifiers for evaluation. In other words, we remove irrelevant citations from the training set at random until there are an equal number of irrelevant and relevant citations. This strategy has been shown to be effective in mitigating the effects of class imbalance [8] (if this is not done, the induced model tends to have high accuracy but low sensitivity, even when a higher cost is assigned to false negatives in training the SVM).

6.3 Results

We compare three AL strategies, described as follows:

- **greedy** This strategy greedily selects for labeling the most promising document, based on labeled terms (see Section 2.1).

- **predicted time** This method divides document scores (computed using the labeled terms, via Equation 1) by the predicted time it is going to take to screen them, based on the regression model described in Section 4 and the current estimate of β, $\hat{\beta}$. This is the strategy we're proposing be used in practice.

- **true time** This is the same strategy **predicted time**, except that it uses the true coefficients, β, as learned over the entire time series. This approach is therefore 'cheating', because it uses coefficients learned over data that wouldn't be available during AL. The idea is to see how this compares to using the predicted time approach, which uses an estimate of β.

Note that all three strategies essentially follow Algorithm 1. The key difference is line 6; the **greedy** strategy does not normalize by anything, the **predicted time** strategy uses $\hat{\beta}$, as shown in the algorithm, while the **true time** variant uses β in the denominator.

The dataset we use for experimentation is the proton beam systematic review dataset. This dataset comprises 4,751 citations, of which 457 the reviewer labeled as relevant (i.e., screened in).[5] The reviewer provided 43 terms suggestive of 'relevance' and 26 indicative of 'irrelevance'. Unfortunately, this is the only dataset for which we currently have recorded screening times, and thus is the only dataset we run experiments over.

Our experiments were conducted as follows. We allotted six hours for (simulated) labeling, and evaluated performance every hour. We take the most recently reported performance at each check-in point (i.e., on the hour). All results are averaged over ten independent runs in this way. This experimental framework matches our scenario: we are assuming that we have a fixed amount of time to annotate a corpus, and want to evaluate our performance with respect to categorizing this set of documents under the time (equivalently, budget) constraints.

Figure 6a plots the average cumulative number of examples that were labeled using each of the three strategies at the end of each hour. The error bars for the **predicted time** strategy show the standard deviations at each time point; the other two querying strategies are deterministic. It is reassuring that both strategies that take time into consideration are indeed able to have the reviewer label more citations in the same amount of time, compared to the **greedy** strategy. Interestingly, using the **predicted time** approach often results in acquiring more labels than when the **true time** strategy is used. We suspect that this is because the time prediction model learned online is 'pessimistic', in that it tends to predict that documents will take longer to label than they actually do. This is likely because of bias in the documents for which labels are requested during AL (over which the time prediction model is subsequently induced); these tend to be difficult, and thus the 'true' labeling time is higher than it would be if an i.i.d. sample were used.

The average performances of the respective strategies at each time point are shown in Figure 6b. The error bars are standard deviations. Note that even the deterministic querying strategies have standard deviations because we have to under-sample the majority class (irrelevant citations) to mitigate the effects of the severe class imbalance, as described above (this introduces a stochastic element). The first thing to note is that both strategies that take time into account outperform the **greedy** strategy at all points after the first hour. It is intuitive that taking the 'long-view' strategy should only pay off after some sufficient amount of time has passed. The **greedy** strategy (almost by definition) will rapidly achieve good performance, but will quickly exhaust its budget. On the other hand, time-sensitive strategies pay off by being prudent in their example selection; the aggregate benefit of this strategy takes some time to manifest.

It is also encouraging that our **predicted time** strategy, which learns to predict how long it's going to take to label citations online (i.e., during AL), performs comparably to the **true time** strategy, which uses the 'true' model coefficients β, as learned over the entire labeled dataset. This is in contrast to previous work [11] in which the predictive model was not sufficiently accurate to achieve the same performance as when the true times were used. It is possible that our incorporation of the annotator learning rate, i.e., the number of documents labeled prior to the document for which labeling time is to be predicted, accounts for the success of our approach.

7. CONCLUSIONS & FUTURE WORK

We have presented ABSTRACKR, an open-source annotation tool. This tool provides an interface for reviewers to screen citations for systematic reviews. It also facilitates AL, i.e., the interactive training of a classification model to automatically categorize remaining citations as 'relevant' or 'irrelevant', thereby reducing workload. Moreover, ABSTRACKR allows the user to communicate additional information to the model; namely, labeled terms, which are words or n-grams whose presence indicates that a document is more or less likely to be 'relevant' to the review. Finally, our annotation tool records labeling times, which we empirically investigated.

We defined an AL scoring function that exploits terms provided via our interface. Typically, the approach is to sort the unlabeled examples (documents) by their scores and then greedily have the expert label the highest scoring instance. However, we demonstrated that normalizing these scores by the predicted time it will take to label the corresponding document results in a better performing system. Moreover, we presented a simple spline regression that incorporates document length and the order in which a document is labeled as predictive variables. The spline serves as a simple model for the annotator's learning rate. The coefficients for this model can be learned online, as AL is ongoing. We showed that using this 'return on investiment' approach results in better performance in the same amount of time, compared with the **greedy** strategy.

In future work, we plan on conducting more experiments to test the strategy outlined here on additional datasets. We also plan to address the problem of optimizing labeling assignments when there are multiple reviewers participating in a systematic review. Finally, we are working to incorporate the labeled terms directly into the SVM optimization function, rather than only using them during AL.

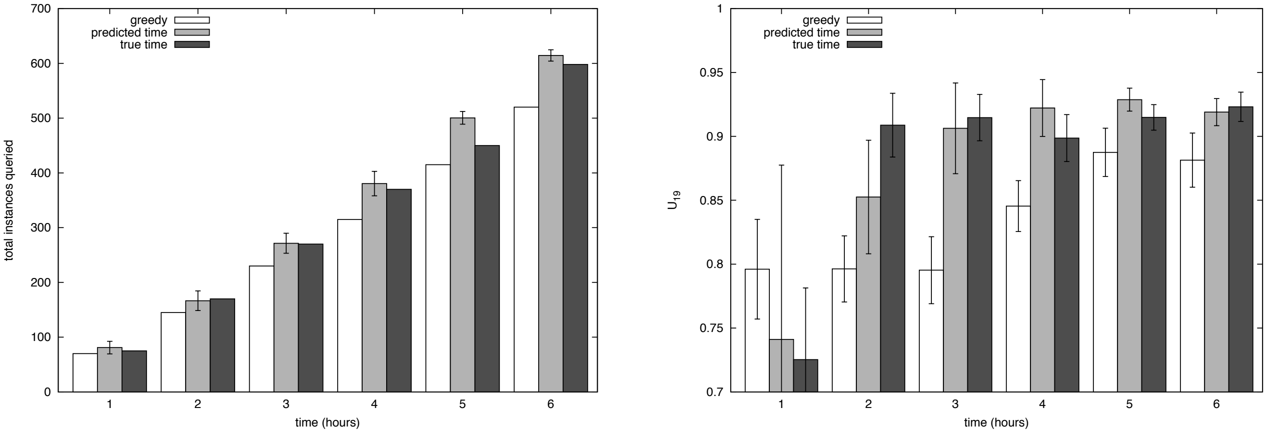
8. ACKNOWLEDGMENTS

We thank James Gaylor for screening the proton beam dataset, and Gary Wallace for technical help with our cluster. Support for this work came from R01HS018494 (BW, TT, JL, CB), CTSI 5UL1 RR025752 (BW, TT, JL, CB), 3UL1 RR025752-02S2 (KS) and NSF grant IIS-0803409 (CB).

9. REFERENCES

[1] P. Wheeler, E. Balk, K. Bresnahan, B. Shephard, J. Lau, D. DeVine, M. Chung, and K. Miller. Criteria for determining disability in infants and children: short stature. *Evidence Report/Technology Assessment No. 73. Prepared by New England Medical Center*

[5] We have previously used this dataset with labels from a different reviewer, who screened this data before the ABSTRACKR tool was developed. We had a colleague re-screen them in order to test our tool; the class distribution breakdown is thus slightly different in this case than in our previous work.



(a) Time versus the number of documents screened.

(b) Time versus U_{19}

Figure 6: Empirical results. In both plots, the white bar corresponds to the greedy strategy, the light grey bar to the predicted time strategy, which normalizes by the predicted time-to-label, and the dark grey bar to the true time strategy, which also normalizes by the predicted time-to-label, but uses the 'true' β coefficients in doing so (see text).

Evidence-based Practice Center under Contract No. 290-97-001, Mar 2003.

[2] C. Cole, G. Binney, P. Casey, J. Fiascone, J. Hagadorn, C. Kim, C. Wang, D. Devine, K. Miller, and J. Lau. Criteria for determining disability in infants and children: Low birth weight. *Evidence Report/Technology Assessment No. 70. Prepared by New England Medical Center Evidence-based Practice Center under Contract No. 290-97-0019*, 2002.

[3] E. Perrin, C. Cole, D. Frank, S. Glicken, N. Guerina, K. Petit, R. Sege, M. Volpe, P. Chew, C. MeFadden, D. Devine, K. Miller, and J. Lau. Criteria for determining disability in infants and children: failure to thrive. *Evidence Report/Technology Assessment No. 72. Prepared by New England Medical Center Evidence-based Practice Center under Contract No. 290-97-0019*, Mar 2003.

[4] S. Arora, E. Nyberg, and C. Rosé. Estimating annotation cost for active learning in a multi-annotator environment. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL) Workshop on Active Learning for Natural Language Processing*, pages 18–26. Association for Computational Linguistics, 2009.

[5] J. Baldridge and A. Palmer. How well does active learning actually work?: Time-based evaluation of cost-reduction strategies for language documentation. In *Empirical Methods on Natural Language Processing (EMNLP)*, pages 296–305. Association for Computational Linguistics, 2009.

[6] P. Donmez and J. G. Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Conference on Information and Knowledge Management (CIKM)*, pages 619–628, 2008.

[7] R. Haertel, K. Seppi, E. Ringger, and J. Carroll. Return on investment for active learning. In *NIPS Workshop on Cost Sensitive Learning*, 2009.

[8] N. Japkowicz. Learning from imbalanced data sets: A comparison of various strategies. *AAAI Workshop on Learning from Imbalanced Data Sets*, 2000.

[9] A. Mccallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *International Conference on Machine Learning (ICML)*, pages 350–358, San Francisco, CA, USA, 1998.

[10] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[11] B. Settles, M. Craven, and L. Friedland. Active learning with real annotation costs. In *Proceedings of the Neural Information Processing Systems (NIPS) Workshop on Cost-Sensitive Learning*, pages 1069–1078. Citeseer, 2008.

[12] K. Tomanek and F. Olsson. A web survey on the use of active learning to support annotation of text data. In *NAACL Workshop on AL for NLP*, pages 45–48, June 2009.

[13] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Journal of Machine Learning Research*, pages 999–1006, 2000.

[14] A. J. Vickers and E. B. Elkin. Decision curve analysis: A novel method for evaluating prediction models. *Medical Decision Making*, 26:565–574, 2006.

[15] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos. Active learning for biomedical citation screening. In *Knowledge Discovery and Data mining (KDD)*, 2010.

[16] B. C. Wallace, T. A. Trikalinos, J. Lau, C. E. Brodley, and C. H. Schmid. Semi-automated screening of biomedical citations for systematic reviews. *BMC Bioinformatics*, 11, 2010.