Active Learning for Biomedical Citation Screening

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

ABSTRACT

Active learning (AL) is an increasingly popular strategy for mitigating the amount of labeled data required to train classifiers, thereby reducing annotator effort. We describe a real-world, deployed application of AL to the problem of biomedical citation screening for systematic reviews at the Tufts Evidence-based Practice Center. We propose a novel active learning strategy that exploits *a priori* domain knowledge provided by the expert (specifically, *labeled features*) and extend this model via a Linear Programming algorithm for situations where the expert can provide ranked labeled features. Our methods outperform existing AL strategies on three real-world systematic review datasets. We argue that evaluation must be specific to the scenario under consideration. To this end, we propose a new evaluation framework for *finite-pool* scenarios, wherein the primary aim is to label a fixed set of examples rather than to simply induce a good predictive model. We use a method from medical decision theory for eliciting the relative costs of false positives and false negatives from the domain expert, constructing a utility measure of classification performance that integrates the expert preferences. Our findings suggest that the expert can, and should, provide more information than instance labels alone. In addition to achieving strong empirical results on the citation screening problem, this work outlines many important steps for moving away from simulated active learning and toward deploying AL for real-world applications.

Keywords

active learning, medical, applications, text classification

1. INTRODUCTION

In many real-world scenarios, unlabeled data is cheap and plentiful while obtaining labels is expensive. This observation has motivated the development of *pool-based active learning* [14], in which the learning algorithm is given access to a (typically large) pool of unlabeled examples, $\mathcal{U}$, and is allowed to request labels for those examples in $\mathcal{U}$ which are believed to be the most useful for learning the target concept. The intuition is that by selecting training examples carefully, rather than at random, annotation costs can be reduced. Buoyed by the empirical successes of active learning (AL) on benchmark classification tasks (e.g., [15, 26]), there has been an increased interest in examining issues associated with deploying AL in "real-world" scenarios [25].

The work herein describes obstacles encountered and corresponding solutions for an AL approach to screening biomedical citations for systematic reviews conducted by the Tufts Evidence-based Practice Center (EPC). Citation screening is a task in which reviewers peruse several thousand scientific abstracts to determine whether the corresponding articles are relevant to the systematic review being conducted. A systematic review is an exhaustive assessment of the published medical evidence for a specified drug or treatment. Determining articles suitable for inclusion is a time-intensive process. Moreover, this task is typically conducted by physicians, whose time is expensive. This scenario, further discussed in Section 2, fits well with the pool-based active learning framework [14]. Our collaboration with the Tufts EPC provides an ideal setting to work with domain experts in clinical science, both to illuminate shortcomings in the assumptions of active learning research and to develop new methods for incorporating domain expertise into AL. We are currently using an active learning system for two ongoing systematic reviews; our experiences in this deployed setting have motivated the approaches proposed in this work.

Most existing AL research (see [22] for a survey) emphasizes empirical evaluation of classifier performance resulting from AL simulated over retrospectively assembled datasets. However, these idealized settings tacitly make a number of assumptions that are unwarranted in real-world situations, including: infallible annotators (including disregarding variance in annotation difficulty), time-invariant target concepts, and restricting the expert feedback exclusively to labels (i.e. that no other expert information is available). Furthermore, it is generally assumed that the goal of AL is to derive a good predictive model, but in many document retrieval applications (such as systematic reviews) the goal is to find all of the relevant instances in a finite pool. This alternative aim results in an increased emphasis on accounting for asymmetric misclassification costs (e.g., a false negative is often more costly than a false positive), class imbalance in training, and the application of appropriate evaluation metrics.

Our work deploying an AL system within the Tufts EPC has elucidated many problems associated with these assump-

KDD '10, July 25-28, 2010, Washington, DC, USA

tions. For example, retrospective experiments over existing systematic review datasets showed that uncertainty sampling indeed induced classifiers with higher predictive accuracy using fewer labels than random sampling. However, sensitivity to the minority class ('relevant' citations) suffers dearly for gains in accuracy when uncertainty sampling is employed, particularly when datasets are imbalanced (see Section 3). We also observed that in deployed (prospective) AL, reviewers were frustrated by being restricted to providing only instance labels. For example, during an AL annotation session, a reviewer noted that the model was 'confused' about clinical studies including children and knew that these ought to be excluded. Machinery to explicitly communicate this to the model could save significant expert time. Finally, we have found that concept drift [30] and annotator drift occurs (i.e., the target concept and annotators understanding of it evolves over time).

In the remainder of this article, we make several contributions to address these issues. To accommodate explicit feature annotation, Section 4 describes a novel active learning strategy that exploits labeled features based on the Co-Testing framework [16]. We extend this method using Linear Programming (LP) to constrain the parameter space when labeled features can be ranked. Section 5 describes appropriate evaluation in *finite pool* AL, where the goal is to categorize a fixed set of instances rather than induce a good predictive model. To this end, we utilize a method from medical decision theory [28] that elicits a relative weighting of sensitivity/accuracy from the domain expert(s) to evaluate classifier performance and accounts for imbalanced classes and asymmetric costs. We demonstrate empirically in Section 6 that our methods outperform uncertainty sampling, random sampling and a previous AL strategy for labeled features [20] for the problem of citation screening; showing that *a priori* information is an effective way of circumventing the AL under class imbalance of Section 3. Finally, Section 6.4 presents a practical method for dealing with concept drift that relies on the expert to identify a trustworthy subset of labeled data, with which we achieve promising empirical results.

2. BIOMEDICAL CITATION SCREENING

Systematic reviews are increasingly used to inform all levels of healthcare. To minimize bias, a systematic review develops and follows a protocol of well defined steps, including: formulating answerable research questions, specifying literature review criteria, conducting a comprehensive literature search, screening of the abstracts obtained from the search to select potentially relevant studies, assessing full articles according to specified literature review criteria, extracting data from accepted studies, assessing the quality of included studies, synthesizing results according to the key questions and the pre-specified protocol, performing meta-analyses (when appropriate), and interpreting results [4].

To identify all eligible reports, reviewers begin by conducting broad searches of the literature (e.g. PubMed) and manually screen titles and then abstracts to obtain a corpus of possibly pertinent citations (typically between 3% to 15% of the broad search). All potentially eligible citations are then retrieved and reviewed in full text to select those that are ultimately included in the systematic review. Citation screening is a laborious and time-consuming, yet critical, step in conducting systematic reviews where failure to identify eligible research reports threatens the validity of the review. Reviewers typically screen between 2,000 and 5,000 citations for a given review, of which approximately 200 to 1,000 are deemed relevant and are reviewed in full text where at most a few dozen are ultimately included in the systematic review. Much larger projects are not uncommon. For example, a project that involved evidence reports conducted for the United States Social Security Administration on the association of low birth weight, failure to thrive, and short stature in children with disabilities, the Tufts EPC screened over 33,000 abstracts.

An experienced reviewer (usually a physician) can screen an average of two abstracts per minute, thus a project with 5,000 abstracts requires up to five person days (forty hours) of uninterrupted work. Abstracts for difficult topics may take several minutes each to evaluate, multiplying the total screening time several fold [29]. Furthermore, this review effort will only continue to grow due to the exponential growth of biomedical literature [10], motivating methods to semi-automate this process.

Two interesting, interrelated properties of the citation screening problem are the profound class imbalance and the asymmetric misclassification costs inherent in the task. The prevalence of the minority class ("relevant" citations) is usually around 10%. Moreover, incorrectly classifying a relevant article as "irrelevant" may sacrifice the integrity of the entire review, whereas incorrectly labeling as "relevant" an irrelevant article will only incur the additional cost of a reviewer manually perusing the document. Hence the former type of error is considerably more expensive than the latter type, i.e., false negatives are costlier than false positives. A corollary of this observation is that a "successful" machine learning approach to semi-automating citation screening needn't achieve particularly good accuracy, but rather must maintain high sensitivity while eliminating at least some of the irrelevant citations (i.e., achieving some degree of specificity).

3. THE PROBLEM OF HASTY GENERALIZATION

This need for partial automation of citation screening naturally fits within the pool-based active learning paradigm, in which the model requests labels for the unlabeled examples likely to be most helpful in learning the target concept. We initially experimented with active learning over a few citation corpora from previously conducted systematic reviews. For these reviews, we have the set of abstracts initially retrieved from the searches, and the subset thereof that was deemed "relevant" after manual screening of the whole corpus. We used Support Vector Machines (SVMs) as the base classifier [27] due to their good empirical performance over text data [12] and the Simple [26] method for uncertainty sampling, which selects for labeling those instances closest to the current separating hyperplane.

These experiments indicated that uncertainty sampling rapidly produces models with high accuracy but *lower* sensitivity compared to models trained on randomly selected data. This is undesirable given the cost asymmetry present in citation screening. We therefore set out to answer two questions. First, why might uncertainty sampling induce models with poorer sensitivity? Second, how can we mitigate this effect, so as to make the best use of our expert via AL in imbalanced, asymmetric cost scenarios? The remain-

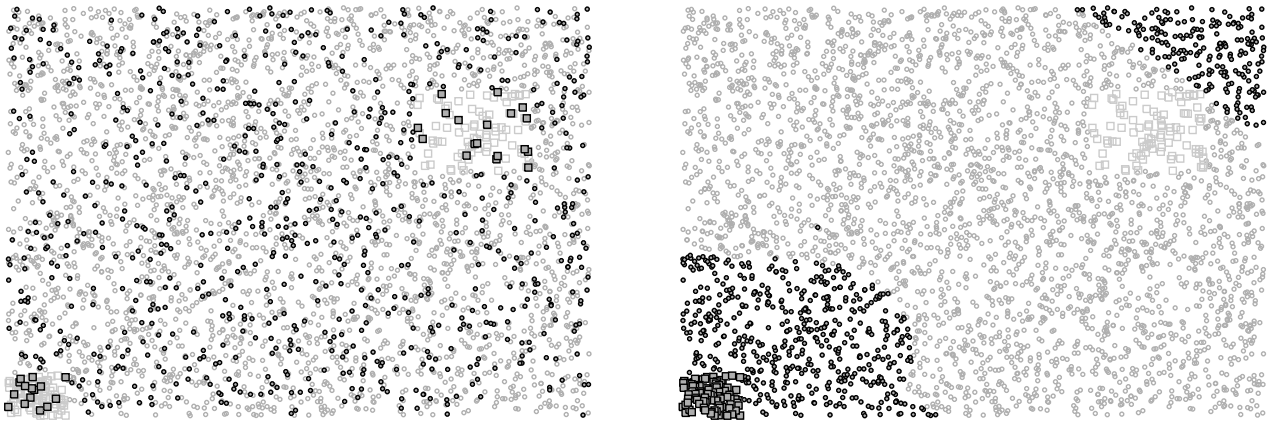

Figure 1: Figures 1a (left) and 1b (right) show the examples for which the passive (random) and Simple strategies requested labels, respectively. In both plots the entire pool of examples ($\mathcal{U}$, at the start of active learning) is shown; examples that are darkened are those for which a label was requested by the corresponding learning algorithm.

der of this section addresses the former question; we address the latter question in Section 4.

Uncertainty sampling methods focus on *refining* the current decision boundary [18]. This is done by first establishing a rough approximation to the ideal decision boundary and then sequentially requesting labels for examples nearest this boundary. Intuitively, this strategy exploits the labeler by ignoring examples whose labels are unlikely to move the decision boundary, thus expediting the training process. Indeed, uncertainty sampling has been shown to work well in a variety of contexts [15, 26, 22]. However, this strategy implicitly assumes that the initial approximation to the decision boundary is reasonable in the sense that as the learner continues requesting labels, the learned boundary will approach the optimal boundary. This assumption is violated in the case of XOR-like or multiple input distribution concept clusters [1, 21], as uncertainty sampling may continue to request labels along the initially discovered boundary, ignoring as-yet undiscovered partitions.

The most relevant existing work with respect to addressing hasty generalization is that of Schütze et al. [21], in which they discuss practical issues in active learning for text classification. Consistent with our observations, they found this problem, which they call the *missed-cluster effect*, to be problematic in real world active learning for text classification, particularly when there is class imbalance (and many real world datasets are imbalanced). Other work [18, 1] has also addressed this problem more generally as a tradeoff between exploration (random sampling) and exploitation (uncertainty sampling). The problem with these approaches in the more specific case of imbalanced data is that they are greedy in that they explore (i.e., with random sampling or the Kernel Farthest-First heuristic [1]) with probability proportional to how successful exploration has been thus far. These methods therefore tend to regress to "standard" active learning, because exploration will only rarely be fruitful when there is class imbalance; namely, on rare occasions when it selects a minority example.

The problem of hasty generalization is perhaps easiest understood with a toy example. Consider the two-dimensional target concept depicted in Figure 1. Here the instances represented by squares comprise the minority class, of which there are two clusters (one in the lower left-hand corner, the other in the upper-right quadrant). We simulated AL over this data using a Support Vector Machine (SVM) [27] with an RBF kernel and two different learning strategies: passive, which randomly selects examples from $\mathcal{U}$ for the expert to label, and Simple [26]. The examples selected for labeling by these two algorithms are darkened in the two sub-plots, Figures 1a and 1b, which correspond to random sampling and Simple, respectively. We allowed the learners to request labels for 25% of the total data.

Figure 1a shows the examples that were selected using the passive (random) strategy. In this case, the learner was trained on a representative, i.i.d. sample of the data, and discovered examples from each of the two minority clusters. However, random sampling was clearly inefficient, in the sense that it queried for the labels of many irrelevant examples, thus wasting our simulated expert's time. To expedite the training process, and to induce a more accurate model, one might appeal to active learning here. However, hasty generalization is a potential pitfall in this approach. This is illustrated in Figure 1b, which shows the examples for which Simple requested labels. The training examples selected via Simple are visibly biased, clustering around the initial approximation to the decision boundary in the lower left quadrant. The learner completely misses the upper-right cluster of squares. The active learner hastily generalized from the examples it initially encountered, and will subsequently misclassify squares in the missed cluster as circles.

The question, then, is: how can we exploit the expert via AL when we have an imbalanced class distribution and asymmetric costs? In the following section, we propose using labeled features to achieve this aim. In particular, labeled features (n-grams, in our case) that are known to the expert at the outset of AL can be used to circumvent the problem of

hasty generalization by combining *a priori* knowledge with the model induced over the current set of labeled instances. Indeed, Shütze et al. [21] explicitly suggested that using domain knowledge may be a fruitful way of avoiding the missed-cluster effect.

4. EXPLOITING LABELED FEATURES

Most existing work in active learning restricts the model, or "learner", to requesting only instance labels from the oracle, likely due in part to the constraints of available benchmark datasets. However, there has been some recent work in exploiting *labeled features*[1] in addition to labeled instances [7, 20, 31]. Druck et al. [7] propose an active learning framework in which the expert labels features rather than instances and uses the generalized expectation (GE) criteria to build a predictive model from labeled features [6]. In addition, they present a method for selecting unlabeled features for the expert to label, analogous to active learning methods selecting unlabeled instances to be labeled.

Raghavan et al. [20, 19] present a method for interleaving feature and instance labels. They augment the standard pool-based AL scenario by incorporating feature feedback during the learning process. They also propose a method for selecting features to have the expert label. This information is then incorporated into the classification algorithm somehow; either through feature scaling or adding labeled pseudo-instances to the dataset. Most relevant to our work here, they propose an active learning strategy that uses the labeled features directly by performing uncertainty sampling in the reduced space of the labeled features [19]. We include comparisons of our proposed approach to this method in Section 6.

In other related work, Zaidan et al.'s *annotator rationale* approach [31] elicits from the expert an 'explanation' for their labels. These rationales are then used to construct *contrast examples*, in which features associated with the provided rationale are removed. The intuition is that the model should be less confident about its prediction for these contrast examples; this is encoded in the SVM constraint function. Elsewhere, Sindhwani et al. [24] presented a method for querying an oracle for labels on both features and instances.

What distinguishes our work primarily is that we are not interested in actively querying the user to label features. Instead, we assume that sets of terms that are indicative of class membership are known *a priori* to the expert. This is a realistic assumption, particularly in the citation screening scenario, wherein the physicians bring a wealth of domain knowledge to the task. Indeed, the PubMed search strings used to find the initial corpus of potentially relevant citations is itself constructed using such keywords. The physicians undertaking systematic reviews start with a well-formulated question, and their domain expertise allows them to enumerate n-grams pertinent to this question. More generally, it is not unreasonable to assume that users would know some discriminative terms upfront in other text classification domains. For example, consider an active learning task to classify a set of newspaper articles into 'sports' and 'world news' categories; surely someone training such a classifier could provide terms indicative of the former rather than the latter ('golf' and 'baseball', for example). An additional difference in our work is that we are not interested in building a predictive model using the labeled features directly. Instead, we want to exploit these features to improve AL performance. To this end, we adopt the Co-Testing approach introduced by Muslea [16].

[1]A labeled feature is a feature that has been designated as being indicative of membership in a particular class.

4.1 Labeled Features for AL via Co-Testing

One way of looking at labeled features is as a distinct *view* of the data. A view is a particular feature space used to represent a given dataset. Blum and Mitchell [2] demonstrated that multiple, redundant views can be exploited in supervised learning through the *co-training* paradigm. Muslea et al. [16] extended this method for active learning via their *Co-Testing* strategy, which works as follows. Suppose we have two views, V_1 and V_2. Learn two hypotheses H_1 and H_2 over these views, respectively. Now define *contention points* as those unlabeled examples about whose labels H_1 and H_2 disagree and request the label for one of these points. This approach is appealing because if these two models disagree on a particular example x, then by definition the label for x must be informative, as at least one of the two models is currently incorrect. Note that Co-Testing is a specific case of Query by Committee [8].

We propose building a simple, intuitive model over the labeled n-grams in tandem with a linear-kernel Support Vector Machine [27] over a standard bag-of-words (BOW) representation of the corpus. For the former, we use an 'odds-ratio' based on term counts, i.e., the ratio of positive to negative terms in a document. In particular, suppose we have a set of positive features (i.e., n-grams indicative of relevance), $\mathcal{P}^F$, and a set of negative features $\mathcal{N}^F$. Then, given a document d to classify, we can compute the likelihood of d being a relevant as:

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

Where $I_d(w)$ is indicator function which is 1 if w is in d and 0 otherwise. Note that we add pseudo-counts to both the negative and positive sums, to avoid division by zero. Then the direction of this ratio gives a class prediction and the magnitude of the ratio gives a confidence.[2] For example, if d contains ten times as many positive terms as it does negative terms, the class prediction is + and a proxy for our confidence is ten.

We can now use this model for Co-Testing as follows. First, generate the set of contention points, i.e., those unlabeled examples about whose class membership the SVM model induced over the BOW representation disagrees with the labeled feature classifier defined above. Of these, select for labeling the example x with the largest ratio. In this case the SVM model predicts that x belongs to one class, but the labeled features present in x strongly suggest that it belongs to the other. The hope is that such examples will be informative to the model, given the disparity between the shallow "semantic" classifier that uses labeled features

[2]In order to ensure that the magnitude is symmetric in the respective directions, one may either flip the ratio so that the numerator is always larger than the denominator, or one may take the log of the ratio.

and the more nuanced "black-box" SVM method, induced on the instances labeled thus far. This strategy should not be subject to the problem of hasty generalization because it relies on *a priori* information external to the current SVM model. Our empirical results, presented in Section 6, confirm that this method - which we call *CoFeature* - improves classifier performance (with respect to the metric of interest) compared to passive learning, AL via SIMPLE and SIMPLE performed over the pruned labeled feature space as proposed by Raghavan et al.[20].

4.2 Exploiting Ranked Labeled Features with Linear Programming

In the preceding section, we assumed that the expert provided a list of features with binary labels (either indicative of relevance or indicative of irrelevance). However, in many cases the expert may also be able to provide a ranking, indicating which features are more or less representative of class membership, relative to one another. For example, in the proton beam systematic review the doctor indicated that *hadrontherapy* is more indicative of a relevant abstract than *proton ion*, and conversely that *electron beam* is more indicative of an irrelevant abstract than *photon beam*. Encoding such domain information is an attractive proposition because it exploits domain knowledge provided by the expert to (hopefully) induce a better generalized model.

Here we present our Linear Programming (LP) formulation for learning a linear classifier with the ability to explicitly encode parameter constraints based on ranked features as provided by the expert. Similar to existing LP methods [17], we begin by assuming that we have a set of positive instances (relevant citations) $\mathcal{P}$ and a set of negative instances $\mathcal{N}$. We define our objective function as:

$$\min \; c_1 \frac{\sum_{\mathcal{P}} p_i}{|\mathcal{P}|} + (1 - c_1) \frac{\sum_{\mathcal{N}} n_j}{|\mathcal{N}|} - c_2\rho - c_3\gamma \qquad (2)$$

$$0 \leq c_1 \leq 1; \quad 0 \leq c_2, c_3$$

In line with intuition, this penalizes false positives and false negatives in the training set (note that the relative costs of these mistakes is governed by the c_1 term). This formulation also encourages a large gap between the least negative and least positive terms, i.e., the negative and positive n-grams nearest one another (γ), as well as between the terms within the respective classes (ρ). The relative emphasis on these two latter terms is defined by c_2 and c_3, respectively; these are user defined constants that represent the tradeoff between expert knowledge and optimizing the parameter vector using available data. Next, we write down constraints for false positives and false negatives [17]:

$$p_i \geq -\mathbf{w} \cdot \mathbf{x} + b + 1 \quad \forall \mathbf{x} \in \mathcal{P}; i = 1, \ldots, |\mathcal{P}| \qquad (3)$$
$$n_j \geq \mathbf{w} \cdot \mathbf{x} - b + 1 \quad \forall \mathbf{x} \in \mathcal{N}; j = 1, \ldots, |\mathcal{N}| \qquad (4)$$
$$0 \leq p_i, n_j \qquad i = 1, \ldots, |\mathcal{P}|; j = 1, \ldots, |\mathcal{N}|$$

Note that the p_is and n_is denote the magnitude of the error for false negatives and false positives, respectively (this is a function of their distance from the learned hyperplane). Thus for each positive instance, the constraint specified by Equation 3 is added such that $p_i > 0$ iff the optimal weight vector as defined by the utility function of Equation 2 will result in a false negative for the particular instance. Correspondingly, Equation 4 specifies the constraints added for negative instances such that $n_j > 0$ iff the instance will be classified as a false positive. If the data is linearly separable, $p_i = 0$ and $n_j = 0$ for all i, j.

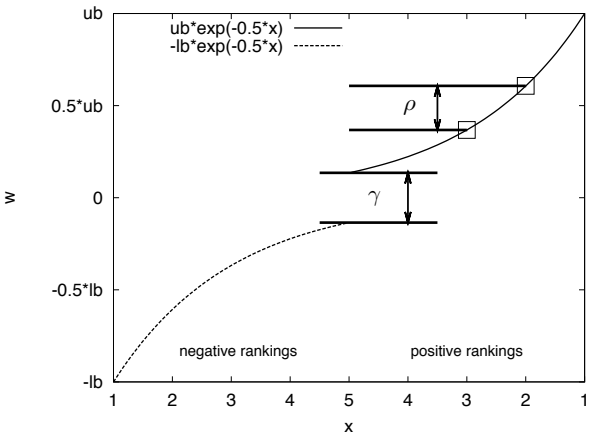

Figure 2: Parameter space function enforced by LP.

We next extend this model to account for term rankings using explicit parameter constraints:

$$w_a - w_b \geq z_{ab} \left[\rho \cdot f\left(r(a), r(b)\right)\right] \quad \forall a, b : a \succ b \qquad (5)$$
$$w_c - w_d \geq \gamma \quad \forall c, d : c \succ d \qquad (6)$$

This formulation encodes a *parameter gap* associated with each rank position as shown in Equation 5. Namely, given two sets of terms $\mathcal{A}$ and $\mathcal{B}$ such that $\mathcal{A}$ and $\mathcal{B}$ are adjacent rankings and all of the members $a \in \mathcal{A}$ are ranked higher than the members $b \in \mathcal{B}$, denoted as $a \succ b$. $z_{ab} \in \{-1, 1\}$ denotes if $\mathcal{A}, \mathcal{B}$ represent irrelevant or relevant terms respectively and $f(x, y) \rightarrow \mathbb{R}$ denotes a function used to scale the relative parameter gap between each pair of rankings. Additionally, we encode a *boundary gap* between the lowest ranking positive terms $c \in \mathcal{C}$ and the lowest ranking negative terms $d \in \mathcal{D}$; this is a method for maximizing the classifier margin. We again note that the objective function attempts to maximize these gaps to effectively encode the available domain knowledge.

The final element of this formulation is an appropriate function to model the relative parameter values based on the ranked term information. We note that one could use any function deemed appropriate for the domain and features. For the proton beam dataset and associated n-grams, we assume that the magnitude of the parameters grows exponentially with rank, displayed graphically in Figure 2. The intuition behind an exponentially growing function is that the highest ranked terms are significantly more indicative of relevance/irrelevance than lower ranking terms. This assumption is made in part due to informal discussions with the our expert regarding the relative importance (in his view) of certain terms versus others. Formally, we have:

$$f(x, y) = e^{-\kappa x} - e^{-\kappa y} \qquad (7)$$

Using this LP formulation, we can use one of several solvers[3] to learn the weight vector directly. As in our CoFeature method, this classifier is used as a view to select contention points with the SVM model.

5. CLASSIFIER EVALUATION FOR THE CITATION SCREENING PROBLEM

In this section we first propose two metrics appropriate for evaluating classifiers in situations wherein the primary aim is annotating a fixed dataset. We next propose a method for eliciting from the user a relative weighting on the cost of false positive versus the cost of a false negative. Indeed, without knowing the tradeoffs involved, it is impossible to assess how a classifier is performing. If sensitivity is twice as important as specificity, then the relative performances of two classifiers will potentially be quite different than if the reverse holds. Thus classifiers must be evaluated with respect to the task to which they are to be applied.

5.1 Finite Pool Active Learning

Active learning methods are typically compared using a hold-out set. This evaluates the predictive performance of the classifier induced with a given AL strategy, with respect to some metric (e.g., accuracy or F-measure). However, there is an important distinction to be made between the goal of constructing a good predictive model and the transductive task of categorizing a finite set of instances into their respective ranked classes. We are interested in the latter for the biomedical citation screening problem. We are not primarily concerned with building a good discriminative model, but rather we are attempting to designate all of the documents in a database of citations as "relevant" or "irrelevant"; aside from the ability to derive this annotated database, the predictive performance of the induced classifier is inconsequential. We are thus viewing the classifier as tool to reduce labor in annotation as opposed to an end in itself.

To formalize the above intuition, we define two metrics we have proposed elsewhere [29]: yield and burden. Recall that we are concerned with the following two outcomes: the fraction of truly relevant citations in $\mathcal{U}$ correctly identified, and the amount of reviewer effort expended, compared to manually screening all of the citations. Let tp^T and tn^T denote the positive ("relevant") and negative ("irrelevant") citations labeled by the reviewer during the training process. Further, let tp^U, fp^U, tn^U, and fn^U denote the number of true positives, false positives, true negatives and false negatives over the remaining, unlabeled abstracts in the pool, $\mathcal{U}$, as generated by the classifier. Finally, let N denote the total number of citations. Then we can calculate these two metrics as shown in Equations 8 and 9.

$$yield = \frac{tp^T + tp^U}{tp^T + fn^U + tp^U} \tag{8}$$

$$burden = \frac{tp^T + tn^T + tp^U + fp^U}{N} \tag{9}$$

We note that yield and burden are roughly equivalent to sensitivity and specificity, except that they also take into account those examples for which a learner has requested labels. (Also, burden is a cost measure, and therefore should be minimized rather than maximized). Thus if a learner is somehow good at querying for the labels of difficult examples, and therefore does not have to predict labels for these instances, it is rewarded.

[3]We use GLPK (http://www.gnu.org/software/glpk/).

5.2 Eliciting Relative Weights from the Expert

Most work in information retrieval on metrics for the evaluation of text classifiers has focused on the weighted F-measure [13], i.e., the weighted harmonic mean of sensitivity and precision. This weighting is defined by β, which appropriately encodes the tradeoffs inherent in the scenario under consideration. We follow in this tradition here, save for the caveat that rather then sensitivity and precision, we use the above proposed metrics of yield and burden. We assume that $cost(fp) = \beta \cdot cost(fn)$ for some β, implying that maximizing yield is β times as important as minimizing burden.

We are left with the question of how to elicit from the domain expert this β. We borrow a method from medical decision making developed for diagnostic test assessment to infer this weight by means of a thought experiment [28]. Suppose that a predictive model, or an oracle, provides the probability that a given citation is irrelevant. If this probability is sufficiently low, a rational reviewer will want to peruse the abstract in full to ascertain if it should be included or not. On the other hand, if the probability is high enough, a rational reviewer will not bother to read the abstract. There is some threshold probability p_t at which the reviewer forgoes reading the abstract. In other words, they are at this point *indifferent* to whether or not they read the abstract because the expected value of reading it at this point is equal to the expected value of not reading it. Suppose that we elicit this p_t from the expert. Further, let $\mathcal{V}(tp)$, $\mathcal{V}(fp)$, $\mathcal{V}(fn)$, and $\mathcal{V}(tn)$ denote the value of a true positive, false positive, false negative and true negative, respectively. We have:

$$p_t \cdot \mathcal{V}(tp) + (1-p_t) \cdot \mathcal{V}(fp) = p_t \cdot \mathcal{V}(fn) + (1-p_t) \cdot \mathcal{V}(tn) \tag{10}$$

The LHS of Equation 10 is the expected value of reading the abstract; the RHS is the expected value of not reading the abstract. This implies:

$$\frac{\mathcal{V}(tp) - \mathcal{V}(fn)}{\mathcal{V}(tn) - \mathcal{V}(fp)} = \frac{1-p_t}{p_t} = \beta \tag{11}$$

Then $\mathcal{V}(tp) - \mathcal{V}(fn)$ is the penalty of not reading a relevant abstract, and $\mathcal{V}(tn) - \mathcal{V}(fp)$ is the cost associated with reading an irrelevant abstract. Thus $\frac{1-p_t}{p_t}$ is the ratio of the cost of a false negative to the cost of a false positive; this is our desired β. We propose using this β directly in an evaluation metric. We define our metric, which we call $Utility_\beta$, as follows:

$$\frac{\beta \cdot yield + (1 - burden)}{\beta + 1} \tag{12}$$

For evaluation purposes, we elicited this weighting from the project lead on one of the ongoing systematic reviews here at the Tufts EPC. We asked him at what probability of a document being irrelevant would he exclude it without reading the abstract. We asked the same question, increasing the number of citations that needed to be screened for the hypothetical project. In line with our expectations, p_t decreased slightly when the set of citations that needed to

be screened became large. Specifically, for N <= 10,000 abstracts, the threshold p_t given was 95%, which translates to a β of 19. When N is > 10,000, he changed p_t to 90%, giving a β of 9. We use β=19 in our experimental evaluations, because most systematic reviews conducted here comprise 10,000 or fewer citations.

6. EXPERIMENTAL RESULTS

We first present experimental results using our feature 'odds ratio' Co-Testing algorithm (referred to as *CoFeature*) over three systematic reviews for which we were given labeled terms by the reviewers. We compare our approach to random sampling, uncertainty sampling via SIMPLE, and SIMPLE in the labeled-features space [20]. We then present results using our Linear Programming method over the Proton Beam dataset, which is the only dataset for which we have ranked labeled terms. Finally, we present promising results on mitigating the effects of concept drift in a deployed active learning setting by incorporating expert feedback.

Evaluation is carried out *with respect to the metric of interest*, i.e., U_{19}, following our above results. This disproportionately emphasizes sensitivity to the minority class ("relevant" citations), as is pertinent for our scenario. We note that SIMPLE outperforms our method on all datasets with respect to accuracy; this again illustrates the necessity of using the correct metric for the situation for evaluation.

6.1 Experimental Setup

All classification is performed using Support Vector Machines (SVM) [27] with linear kernels as they have been shown to perform well with high dimensional data [9]. We use a modified version of LibSVM [3] and its Python interface. All SVMs are induced over a feature space comprising a binary bag-of-words encoding of concatenated citation title and abstract text, with the exception of SIMPLE in the pruned space, which operates in the labeled terms space only. Prior to evaluation over the as-yet unlabeled examples, the C parameter is tuned via grid search[4] over the training data acquired during AL. We also set the class penalty ratio to 100:1, i.e., we set the cost of a false negative to 100x that of a false positive. Our experimental setup is as follows. We instantiate the four learners and give each of them labels for the same two 'seed' citations; one "relevant" and one "irrelevant". We then allow each learner to request 5 labels per round of active learning. Every 25 labels, we evaluate the learners as described above and report results. Due to our severe class imbalance, we under-sample the majority class (at random) so that the class distribution is equal prior to building the classifier used in evaluation; this strategy has been shown effective in mitigating the effects of class imbalance [11]. All results reported are averages over ten independent runs.

6.2 Feature Co-Testing Results

Results over the COPD dataset are shown in Figure 3. Note that the COPD is a smaller dataset than proton beam, comprising 1,601 citations, 196 of which are "relevant". We show performance for up to 800 labeled examples. We were given 22 labeled n-grams; 15 positive and 7 negative. Our CoFeature method maintains higher Utility$_{19}$ until about

[4]When performing grid search, we keep the C that maximizes a weighted metric, i.e., $\beta \cdot sensitivity + specificity$.

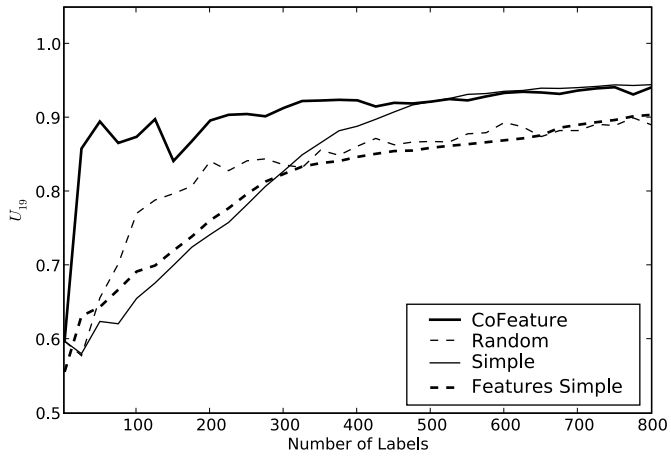

Figure 3: Utility$_{19}$ over the *copd* dataset. Our CoFeature approach outperforms all baseline methods.

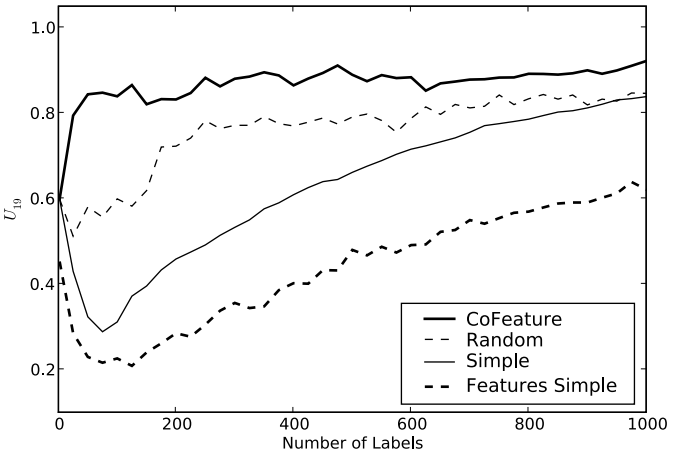

Figure 4: Utility$_{19}$ over the *micro nutrients* dataset. Our CoFeature approach outperforms all baseline methods.

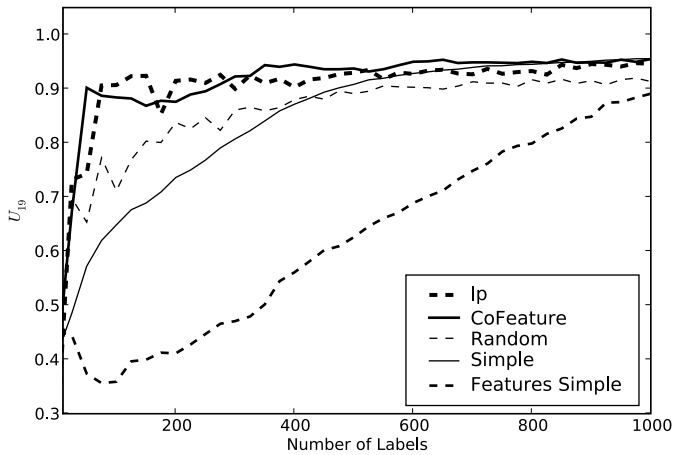

Figure 5: Utility$_{19}$ over the *proton beam* dataset. Our Linear Program (lp) and CoFeature approaches outperform all baseline methods.

the 500 label mark, at which point SIMPLE performs comparably.

Figure 4 displays results over the micro nutrients dataset. There are 4,010 citations in this dataset, 258 of which were found to be "relevant". This is an interesting dataset because there is a preponderance of positive n-grams; 47 versus 2 negatives. In this case, our feature Co-Testing strategy clearly dominates the other methods.

6.3 Ranked Labeled Features Results

Figure 5 shows results over the proton beam dataset for the four methods. There are 4,751 documents in this dataset, of which 243 were deemed "relevant", where we follow the experimental procedure delineated above. The reviewer provided us with 43 ranked positive features and 26 ranked negatives. There were 5 discrete groups of ranked positive terms (the terms in the most positive group were thus five times as indicative of a "relevant" citation as those in the least positive group) and 3 groups of ranked negative terms provided by the reviewer. We show results for up to 1,000 labels, at which point the performance of the classifiers remains relatively constant. The first significant observation is that both our Co-Feature and LP approaches clearly dominate the other baseline methods until $\sim$ 600 queries, at which point SIMPLE catches up. The second important observation is that the LP method is able to exploit ranked features in early active learning rounds to outperform Co-Feature.

For this particular experiment $\kappa = 0.1, c_1 = 0.75, c_2 = 0.1, c_3 = 0.2$. As each weight parameter was bounded between the range $-100 \leq w_i \leq 100$ to cover 7 rankings, we selected user defined values which balanced expected gap sizes with empirical error without significant parameter tuning due our limited data setting. With these settings, the LP method slightly outperforms Co-Feature from 50-150 and 200-300 queries. Our results with other parameter settings show that if we set the parameters to bias the LP to favor domain knowledge, we then see large gains during early rounds, but performance plateaus more slowly. If we set the parameters to bias the LP toward empirical error, then we observe a less pronounced early jump with a steadier performance increase. We thus conclude that the best use of rankings is to begin with a bias toward the ranking and decrease this importance as labeling proceeds. How this should be done precisely remains a problem for future work.

6.4 Dealing With Concept Drift

We have presented our core technical contribution, and now briefly turn our attention to work done on an ongoing systematic review regarding sleep apnea, in which the doctors are using our system. Specifically, we investigate how one might use the expert to help identify the presence of, and mitigate the effects due to, concept drift.

It is generally assumed that the target concept being learned in AL is fixed and immutable over time; indeed, simulating AL retrospectively would not be possible without such an assumption. However, in practice this is rarely the case. In our application we have found that the target concept changes over time, i.e., undergoes *concept drift*. We can utilize some established ideas in addressing this problem. Notably, Widmer and Kubat [30] suggest building a classifier over a window of the W most recently labeled instances.

While intuitively appealing, the obvious problem with this strategy is the W parameter; how can we know which labels are reliable? In our case, we found that experts themselves are capable of identifying when during the training process the labels likely became reliable. For example, in our deployed work with a systematic review pertaining to sleep apnea, the criteria for inclusion of abstracts changed a number of times at the start of screening. In particular, the criteria was tightened, meaning more citations were included than should have been. We asked the reviewer when during labeling this tightening occurred, and used all subsequent examples as our training set (i.e., our window W) to build a classifier c_W. We then applied c_W to the noisy labels, i.e., those preceding W. Because the criteria was tightened, we asked the expert to review those articles that were previously labeled as relevant, but that c_W designated as irrelevant. The reviewer ended up flipping the label from 'relevant' to 'irrelevant' for 21 out of the 45 (or about 47%) we showed him. This was valuable, because in addition to training a new classifier with the corrected labels, we saved labor in that those 21 citations needn't be pulled and reviewed in full text, which is a time-intensive endeavor compared to just rereading the abstract. The fact that nearly half the labels of the selected examples were flipped indicates the extent to which concept drift can occur in the initial stages of citation screening; this approach of active, iterative re-labeling is a potential way around this problem.

7. CONCLUSIONS AND FUTURE WORK

Our work on a deployed active learning system at the Tufts Evidence-based Practice Center (EPC) has provided us a unique opportunity to collaborate with domain experts in clinical science to extend the applicability and utility of the AL framework. We have focused on making the most of our domain experts by incorporating their *a priori* domain knowledge into the active learning process, as opposed to exploiting only instance labels. This was achieved with a novel algorithm for active learning with labeled features based on Co-Testing, which empirically outperformed existing active learning methods on three real-world systematic review datasets. We extended this approach for the special case when the expert is able to provide ranked labeled features via a novel Linear Programming algorithm.

We proposed a new framework for evaluating active learning methods of domains like citation screening that emphasizes correctly categorizing all examples in a finite pool of instances, rather than focusing on the predictive performance of induced classification models. Furthermore, making use of existing work in medical decision theory [28], we elicit from the domain expert the relative costs of incorrectly categorizing positive and negative examples as 'negative' and 'positive', respectively. We maintain that such context-specific costs must be taken into account when evaluating classification systems; such systems will not be deployed in a vacuum.

Additionally, our collaboration with the EPC has highlighted other problems with the assumptions in the standard pool-based AL framework. For example, concept drift is tacitly assumed not to occur in the oracle-model, but is a reality nonetheless. We proposed a simple method with good empirical results for addressing this issue that relies on the expert to identify a set of trustworthy labels.

In future work, we plan on applying the decision-theoretic *ProActive Learning* framework developed by Donmez and Carbonell [5] to the citation screening problem. This framework relaxes some of the unrealistic postulates in AL and

addresses how best to make use of multiple experts (in this work, we eschewed the issue of multiple reviewers even though often 1-3 doctors will participate in the citation screening task of a systematic review). We plan on extending their approach for the case in which the costs of each expert are known up front, as they are in our application. Continuing in this vein, it may be fruitful to consider under what conditions we should request redundant labels from multiple reviewers for a given citation. Sheng et al. [23] addressed this 'crowd-sourced' scenario, i.e., a situation wherein you have access to multiple, noisy labelers. Their strategies may be adopted in the future to the case of multiple physicians with different reviewing abilities.

8. ACKNOWLEDGMENTS

Byron Wallace and Tom Trikalinos were supported in NIH grant R01HS018494-01. Kevin Small was supported by NIH grant 3UL1RR025752-02S2 and Carla Brodley was supported by NSF grant IIS-0803409.

9. REFERENCES

[1] Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. *J. Mach. Learn. Res.*, 5:255–291, 2004.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *CLT*, pages 92–100, 1998.

[3] Chih-Chung and C.-J. Lin. *LIBSVM: A library for support vector machines*, 2001.

[4] C. Counsell. Formulating questions and locating primary studies for inclusion in systematic reviews. *Ann. Intern. Med.*, 127:380–387, Sep 1997.

[5] P. Donmez and J. G. Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *CIKM*, pages 619–628, 2008.

[6] G. Druck, G. S. Mann, and A. McCallum. Learning from labeled features using generalized expectation criteria. In *SIGIR*, pages 595–602, 2009.

[7] G. Druck, B. Settles, and A. McCallum. Active learning by labeling features. In *EMNLP*, pages 81–90. ACL Press, 2009.

[8] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. In *Machine Learning*, pages 133–168, 1997.

[9] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification, 2003.

[10] L. Hunter and K. B. Cohen. Biomedical language processing: What's beyond pubmed? *Mol Cell*, 21(5):589–594, March 2006.

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

[12] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML-98*, pages 137–142, 1998.

[13] D. Lewis. Evaluating and optimizing autonomous text classification systems. In *SIGIR*, pages 246–254, 1995.

[14] D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *SIGIR*, pages 3–12, New York, NY, USA, 1994.

[15] A. Mccallum and K. Nigam. Employing EM and pool-based active learning for text classification. In *ICML*, pages 350–358, San Francisco, CA, USA, 1998.

[16] I. Muslea, S. Minton, and C. A. Knoblock. Active learning with multiple views. *Journal Artificial Intelligence Research (JAIR)*, 27:203–233, 2006.

[17] W. N. S. O. L. Mangasarian and W. W. Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, (43):570–577, 1995.

[18] T. Osugi, D. Kun, and S. Scott. Balancing exploration and exploitation: A new algorithm for active machine learning. In *ICDM*, pages 330–337, Washington, DC, USA, 2005.

[19] H. Raghavan and J. Allan. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *SIGIR*, pages 79–86, 2007.

[20] H. Raghavan, O. Madani, and R. Jones. Active learning with feedback on features and instances. *J. Mach. Learn. Res.*, 7:1655–1686, 2006.

[21] V. E. Schütze, H. and J. Pedersen. Performance thresholding in practical text classification. In *CIKM*, pages 662–671, New York, NY, USA, 2006.

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

[23] V. S. Sheng, F. Provost, and P. G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, 2008.

[24] V. Sindhwani, P. Melville, and R. D. Lawrence. Uncertainty sampling and transductive experimental design for active dual supervision. In *ICML*, pages 120–128, 2009.

[25] 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, Boulder, Colorado, June 2009.

[26] 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.

[27] V. N. Vapnik. *The Nature of Statistical Learning Theory*. 1995.

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

[29] 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.

[30] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. In *Journal of Machine Learning*, pages 69–101, 1996.

[31] O. F. Zaidan, J. Eisner, and C. Piatko. Machine learning with annotator rationales to reduce annotation cost. In *NIPS Workshop on Cost Sensitive Learning*, December 2008.