# An Analytic and Empirical Evaluation of Return-on-Investment-Based Active Learning

**Robbie Haertel, Eric K. Ringger, Paul Felt, Kevin Seppi**
Department of Computer Science
Brigham Young University
Provo, Utah 84602, USA
robbie.haertel@gmail.com, ringger@cs.byu.edu,
kseppi@byu.edu, paul.lewis.felt@gmail.com

## Abstract

Return-on-Investment (ROI) is a cost-conscious approach to active learning (AL) that considers both estimates of cost and of benefit in active sample selection. We investigate the theoretical conditions for successful cost-conscious AL using ROI by examining the conditions under which ROI would optimize the area under the cost/benefit curve. We then empirically measure the degree to which optimality is jeopardized in practice when the conditions are violated. The reported experiments involve an English part-of-speech annotation task. Our results show that ROI can indeed successfully reduce total annotation costs and should be considered as a viable option for machine-assisted annotation. On the basis of our experiments, we make recommendations for benefit estimators to be employed in ROI. In particular, we find that the more linearly related a benefit estimate is to the true benefit, the better the estimate performs when paired in ROI with an imperfect cost estimate. Lastly, we apply our analysis to help explain the mixed results of previous work on these questions.

## 1 Introduction

In active learning (AL), a sample selection algorithm sequentially chooses instances, or "samples," to be labeled/annotated by an oracle. Each annotated instance results in a measurable benefit, such as an increase in model accuracy, and incurs a specific cost, such as the time needed to obtain the label. Unfortunately some AL research has ignored the fact that instances have varying costs. Decision-theoretic approaches (e.g., Liang et al., 2009) can incorporate per-instance cost but typically ignore it during experimentation, due in part to the difficulty of subtracting cost from benefit when they are measured in different units (Donmez and Carbonell, 2008; Haertel et al., 2008). Return-on-investment (ROI) is a cost-conscious technique that avoids this requirement by selecting the instance $x^*$ having maximum net benefit *per unit cost*, i.e.,

$$x^* = \arg\max_x \frac{benefit(x) - cost(x)}{cost(x)}. \qquad (1)$$

This approach to AL was independently proposed by Donmez and Carbonell (2008), Haertel et al. (2008), and Settles et al. (2008); in addition, Tomanek and Hahn (2010) evaluated the effectiveness of ROI. Unfortunately, the published results regarding the usefulness of ROI are mixed. In addition, despite its intuitive appeal as a practical cost-conscious algorithm, there has been little theoretical justification for the ROI approach to AL.

The purpose of this paper is to provide an initial theoretical analysis of ROI that, in turn, allows us to identify the conditions needed for the successful application of ROI in a practical environment. We also empirically assess the degree to which violated conditions affect the overall performance of ROI and shed some light on the previously published results. The paper is organized as follows: related work is presented in Section 2. Section 3 examines the conditions under which ROI would be optimal. Section 4 discusses the experimental methodology. Section 5 experimentally assesses the extent to which



*Proceedings of LAW IX - The 9th Linguistic Annotation Workshop*, pages 11–20,
Denver, Colorado, June 5, 2015. 

the conditions hold in practice – but outside the context of AL – while Section 5 explores the overall effect on AL. Finally, Section 6 presents our conclusions.

## 2 Related Work

The essence of active learning is to select the next "best" instance to be annotated. Naturally, the question arises: which sample selection function is optimal? Cohn et al. (1996) derive a solution for selecting the instance that minimizes model variance. A related class of solutions based on optimal experimental design uses Fisher information to select the optimal instance (Zhang and Oles, 2000). However, these approaches fail to account for problems in which instances are not equally costly to annotate.

Decision theory offers an elegant framework for (greedily) selecting the next best instance based on the utility of the instance and considering variable query costs. Some examples include Liang et al. (2009), Anderson and Moore (2005), Margineantu (2005), and Kapoor et al. (2007). In this framework, the optimal instance is the one with maximum net utility, that is, utility minus cost. However, this approach requires that utility and cost be measured in the same units. This requirement is particularly problematic when heuristics (such as entropy) are used to approximate expected utility.

Another approach, borrowed from the financial industry, is return-on-investment (ROI) (Donmez and Carbonell, 2008; Haertel et al., 2008; Settles et al., 2008). ROI is related to the decision theoretic approach (Haertel et al., 2008); however, unlike the decision theoretic approach, ROI does not require conversion between units of utility (benefit) and cost. ROI has explicitly been employed with mixed results on a variety of tasks. Donmez and Carbonell (2008) show positive results with ROI on face detection, letter recognition, spam detection, and high revenue detection tasks but do not evaluate ROI using variable instance costs. Settles et al. (2008) evaluate ROI on entity-relation tagging, speculative text classification, and information extraction. They limit themselves to an N-best approximation to entropy for the sequence labeling tasks, but in this study ROI does not outperform basic AL. Haertel et al. (2008) show positive performance of ROI on English part-of-speech tagging. Finally, Tomanek and Hahn (2010) find that ROI slightly outperforms two new cost-conscious algorithms when an appropriate benefit function is used.

## 3 Theoretical Analysis of ROI

The purpose of this section is to provide a bottom-up theoretical explanation of ROI. The analysis also provides a framework within which we can explain why in some previous work ROI has succeeded while in other work it has failed. This section examines Area Under the cost/benefit Curve (AUC) as a suitable objective function and then enumerates a set of conditions that, if true, would lead to ROI maximizing AUC. Within the context of a bottom-up derivation of ROI, the assumptions introduced are somewhat strong, but we dedicate the remainder of the paper to analyzing the degree to which they hold in practice and their effect on practical results.

We begin with a brief set of definitions. AL algorithms sequentially select instances from a set of unlabeled instances $\mathscr{U}$ ("the pool"). As an instance $x \in \mathscr{U}$ is annotated with label $y$, it results in a measurable benefit and also incurs a specific cost. For the purposes of this section, we follow previous work in assuming a single annotator and in recognizing that the benefit and cost of obtaining a particular annotation may depend on previously obtained annotations. Thus, we define *total benefit* and *cumulative cost* to be functions ($b(\cdot)$ and $c(\cdot)$, respectively) of a sequence of labeled data $L = \langle (x_1, y_1), \ldots, (x_n, y_n) \rangle$. For simplicity, we assume that the cost to annotate an instance is independent of its place in the sequence, although it can be shown that this assumption has no bearing on the final analysis. Therefore, $c(L_{1\ldots i}) = \sum_{i'} c(L_{i'})$.

All previous work of which we are aware evaluates AL using cost/benefit curves or some derivation thereof. Cost/benefit curves (a generalization of standard learning curves) parametrically plot $b(L_{1\ldots i})$ against $c(L_{1\ldots i})$ for $i \in \{1, \ldots, |L|\}$. Rather than focusing on a single point, these curves capture the performance of algorithms over a range of costs. AUC represents the expected benefit across the full range of costs and generally speaking algorithms with higher AUC are more desirable. Note that Settles and Craven (2008) and Baldridge and

Osborne (2004) use AUC to evaluate AL algorithms.

We now formally define AUC. Assuming linear interpolation between discrete neighboring points, AUC is the sum of the area of the right trapezoids defined by adjacent points on the curve. Let $a_i(L)$ be the area of the $i^{\text{th}}$ trapezoid:

$$a_i(L) = \frac{1}{2}\left[c\left(L_{1\ldots i}\right) - c\left(L_{1\ldots i-1}\right)\right] \cdot \left[b\left(L_{1\ldots i-1}\right) + b\left(L_{1\ldots i}\right)\right] \quad (2)$$

(where $c(\emptyset) = b(\emptyset) = 0$). Then, the AUC defined by the sequence $L$ is:

$$auc(L) = \sum_{i=1}^{|L|} a_i(L). \quad (3)$$

Maximizing AUC using AL can be seen as a sequential decision problem in which each decision consists of selecting an instance for annotation. The optimal instance to select given previous annotations $L$, will depend on the decision's effect on the next decision, and the effect of the second decision on the third, and so forth, until all decisions have been made. To account for this recursive dependence, we must consider entire sequences of decisions. Note that if we do not allow instances to be selected more than once from $\mathscr{U}$ we will eventually choose every instance and the number of decisions per sequence is $N = |\mathscr{U}|$.[1] Additionally, since the actual annotations that the oracle will provide are unknown, they must be considered in expectation, represented with random variables $Y_i$. Given a sequence of already annotated data $L$, one approach to maximizing AUC in expectation that accounts for this recursive effect of decisions is (see Haertel et al., 2008 for a decision-theoretic variant):

$$x_1^*, \ldots, x_N^* = \arg\max_{x_1,\ldots,x_N} \mathbb{E}_{Y_1\ldots Y_N | x_1\ldots x_N, L}\left[auc\left(L \oplus \langle (x_1, y_1)\ldots(x_N, y_N)\rangle\right)\right] \quad (4)$$

where $\oplus$ represents sequence concatenation. Although finding the optimal sequence in this way accounts for the effects each decision has on successive decisions, in fact, the sequential decision process protocol requires only the first instance in this sequence, viz., $x_1^*$. We then append $x_1^*$ and the oracle's annotation for the instance $y_1$ to $L$. The result is an updated belief reflected in the expectations (via the new $L$) used to select the next instance.

We now derive ROI from equation 4 under the following conditions:

1. The covariance of cost and benefit is zero.
2. The cost and benefit of each instance are independent of the order in which instances are annotated.
3. Each random variable $y_i$ (i.e., label) is conditionally independent of all other $y_{j\neq i}$, given $x_i$ and $L$.
4. Cost and benefit are exact up to a scalar constant.

The reader is reminded that we do not necessarily presume these conditions to hold in practice; we briefly discuss their practicality herein and later empirically examine the degree to which they hold.

First, while it is conceivable that cost and benefit have zero covariance in some annotation problems, there are certainly cases where there may be some correlation. This correlation is especially evident in structured prediction problems, e.g., "larger" instances (e.g., long sentences) will tend to contain more information but be more costly. However, to our knowledge, the amount of correlation in such cases has not been studied previously. Second, although cost may be independent of annotation order (as implicitly assumed by previous work, e.g., Settles et al., 2008), the benefit of an instance will, in fact, usually depend on the order in which it is annotated. Consider, for example, a pool of instances in which there are several similar instances (e.g., the same word in the same context with the same part-of-speech). By annotating one of the instances, the model will likely learn what it needs from this single instance and therefore the benefit of annotating the others is greatly diminished. Third, the conditional independence assumption is similar to the assumption that benefit is independent of the order in which instances are annotated, but applies distributionally and is more mathematically precise. Finally, optimal (exact) benefit estimators are computationally intractable. While some approaches are optimal for the last decision and perform very well (e.g., Roy

[1] This limit is rarely reached in practice due to budgetary constraints, however, such a constraint does not affect the current analysis. One simply performs computation as if they were going to annotate all instances, but then only selects the best instance, repeating the process until the budget is exhausted.

and McCallum, 2001), these approaches are impractical for structured prediction tasks; we will examine the effectiveness of several heuristic benefit estimators in our empirical examination. Similarly, although cost is sometimes knowable *a priori* it often is not. However, Settles et al. (2008) showed that cost can be reliably learned in practice.

While we defer the question of the degree to which these assumptions are violated in practice to our experiments, we proceed with the analysis as if they were true to better understand the theoretical underpinnings of ROI. In the context of maximization, the scalar constants allowed by condition 4 can be ignored. The linearity property of expectations allows us to move the expectation in equation 4 inside of the sum in equation 3. The first condition then allows us to move the expectation further into the area calculation so that equation 2 becomes (omitting expectation indices for brevity):

$$a_i(L) = \frac{1}{2}\left(\mathbb{E}\left[c\left(L_{1\ldots i}\right)\right] - \mathbb{E}\left[c\left(L_{1\ldots i-1}\right)\right]\right) \cdot \left(\mathbb{E}\left[b\left(L_{1\ldots i-1}\right)\right] + \mathbb{E}\left[b\left(L_{1\ldots i}\right)\right]\right). \quad (5)$$

Condition 2 implies that $b\left(L_{1\ldots i}\right) = \sum_{i'=1}^{i} b\left(L_{i'}\right)$; applying linearity, we obtain:

$$\mathbb{E}\left[b\left(L_{1\ldots i}\right)\right] = \sum_{i'=1}^{i} \mathbb{E}_{y_{i'}|x_{1\ldots i'},y_{1\ldots i'-1},L}\left[b\left(L_{i'}\right)\right] \quad (6)$$

(idem. for cost). Finally, condition 3 implies that:

$$\mathbb{E}\left[b\left(L_{1\ldots i}\right)\right] = \sum_{i'=1}^{i} \mathbb{E}_{y_{i'}|x_{i'},L}\left[b\left(L_{i'}\right)\right] \quad (7)$$

(idem. for cost). This result allows us to compute the expected cost and benefit of each instance once per iteration of active learning (as is common outside of decision theoretic frameworks). Because these quantities can be computed independently of one another, we can represent each instance $x_i$ by a line segment with fixed width and height—the expected cost and benefit, respectively, according to the current model—and statically compute the area using these line segments.

It can be proven that, under these conditions, the sequence $x_1^*, \ldots, x_N^*$ that maximizes AUC is the sequence that is in non-strict slope-non-increasing order.[2] This is precisely the ordering provided by ROI (see equation 1). Thus, under these conditions, ROI is optimal. (Recall that typically only the first element $x_1^*$ is annotated, models are updated, then the process repeats).

[2] A detailed proof sketch is provided by Haertel (2013).

## 4 Experimental Methodology

In this section, we describe our methodology for empirically assessing the degree to which the conditions of Section 3 hold in practice and define what we mean by practical contexts. Space constraints limit our experiments to a single task: English part-of-speech (POS) tagging on the POS-tagged Wall Street Journal text in the Penn Treebank version 3 (Marcus et al., 1993).

For this task, we employ Maximum Entropy Markov Models (MEMMs) to model the distribution of tags given words, $p(\mathbf{t}|\mathbf{w})$. The model choice is motivated primarily by the speed of retraining. AL typically begins with a small set of randomly selected instances: we use 100 instances annotated "from scratch" (i.e., without AL). However, we do account for the cost incurred by annotating the seed set using the cost simulation described below. Each experiment is run 5 times with a different random seed. For TVE (a committee-based approach; see below), we use a committee size of 5 and train all members in parallel. We additionally score instances in parallel, using 4 threads; the remaining processors are used for training the cost model, evaluating benefit, and garbage collection. For non-committee methods, we found that extra scoring threads do not improve results. All simulations are run on dual hex-core Intel Westmere 2.67 GHz CPUs equipped with 24 GB of RAM.

### 4.1 Active Learning Simulation

We are interested in empirically testing ROI outside of the clean mathematical environment implied by Section 2. However, the number of experiments we performed necessitated running AL in simulation. Nevertheless, we employ various techniques to keep the simulation as true-to-life as possible.

Most importantly, each time we select an instance for an annotator to annotate, we simulate the length of time the annotator will need to annotate the instance (i.e., the cost) using Ringger et al.'s (2008) linear cost model derived from user study data. This

model assumes that instances are pre-annotated using an automatic annotation model, and the task of the annotator is to correct the errors from the predictive model. The length of time required to annotate a sequence $\mathbf{w}$, pre-annotated with hypothesis tags $\mathbf{t}$ and true tags $\mathbf{y}$, is:

$$cost(\mathbf{w},\mathbf{y},\mathbf{t}) = \alpha + \beta \cdot |\mathbf{w}| + \gamma \cdot \sum_{i=1}^{|\mathbf{y}|} \mathbb{1}(y_i \neq t_i) \quad (8)$$

The sum represents the number of tags from the pre-annotation that the annotator changed. We estimate the parameters of the linear model ($\alpha = 50.534, \beta = 2.638, \gamma = 4.440$) using the user-study data from Ringger et al. (2008). To add noise to the simulated cost, we generate a random deviate from a shifted Gamma distribution having mean equal to the time predicted by the model, a variance of 5063.35 (the empirical variance of the user-study data), and a shift of 10.0 (near the minimum time). We chose a (shifted) Gamma distribution because the data from the user study appear to be Gamma distributed; as an added benefit, the generated values are guaranteed to always be positive.

In our experiments, we simulate the scenario in which annotators request instances to annotate on demand, e.g., by requesting work on a crowdsourcing service; we call this annotator-initiated AL. This AL contrasts to the alternative in which the algorithm spends time determining the next instance to be annotated and then sends the instance to an annotator to perform the work. We call this latter paradigm learner-initiated AL. The usual implicit assumption in learner-initiated AL is that no cost is incurred between the time the machine sends a request to the annotator and the time the annotator actually starts the work. This assumption is unrealistic, despite being the approach to AL simulation in previous work; real annotation projects are annotator-initiated (e.g., crowd-sourcing). The "Parallel No-Wait" active learning framework introduced by Haertel et al. (2010) follows the more true-to-life annotator-initiated paradigm and provides the guarantee that annotators never need to wait for an instance. We further extend the framework by scoring instances, training the cost model, and training the tagging model in parallel.

Realistic annotation environments also often involve multiple annotators (cf. Donmez and Carbonell, 2008). We take an incremental step towards allowing multiple annotators by assuming that all annotators are infallible and have the same distribution over the amount of time to annotate any given instance. Under these circumstances, each instance needs to be annotated only once, and annotators are interchangeable. We simulate in real time 20 tireless oracles who continuously and simultaneously annotate instances for 50 hours each. In contrast to learner-initiated AL, this represents the *worst* possible case for the no-wait framework since models are maximally out-of-date. Thus, this simulation provides an empirical lower bound on the AUC.

### 4.2 Cost Estimation

The denominator in ROI is an estimate of the cost to obtain a label for the instance being scored. This estimate is not to be confused with the simulation of annotation times for selected instances, as described in the previous section. The cost estimate (as used in ROI) is computed over many instances to help select an informative instance when the annotator requests one. Once the instance has been selected, we then (noisily) simulate what it would cost for the annotator to annotate it, as described above. For algorithms that estimate cost as the time to annotate an instance, we learn a linear model of the same form as equation 8. The coefficients are learned using the data obtained during AL (ultimately obtained from the noisy simulation). However, since we do not know which of the automatically pre-annotated tags are incorrect during estimation, we must compute the expected number of incorrect tags in place of the sum in equation 8.

The results of our experiments are potentially better than in practice since our cost estimate has exactly the same form as the simulated true cost. However, the results are still useful because (1) the gamma-distributed noise in the true cost has high variance and (2) the estimate is computed in expectation (using the learned model).

### 4.3 Benefit Estimation

ROI's numerator is an estimate of the benefit of obtaining a label for a given instance. As previously mentioned, optimal benefit estimators are impractical for structured learning problems; uncertainty-based heuristics are typically employed instead. Let $\mathbf{t}$ represent a sequence of tag assignments for sen-

tence $\mathbf{w}$. Drawing mostly from previous studies, we consider the following:

**Constant (CONST)** assumes all instances have equal benefit.

**Approximate Token Entropy (ATE)** (Settles and Craven, 2008) approximates the true sequence entropy as the sum of the entropy of the individual marginal distributions $p(t_i|\mathbf{w})$. The marginal distributions can in turn be approximated as $p(t_i|\mathbf{w}) \approx p(t_i|t^*_{i-1}, \mathbf{w})$ where $t^*_{i-i}$ is the $(i-1)th$ tag in the Viterbi best sequence $\mathbf{t}^*$; a beam search can significantly reduce computation.

**Monte Carlo Entropy (MCE)** uses a Monte Carlo approximation to compute the entropy, i.e., $\mathbb{E}[-\log p(\mathbf{t}|\mathbf{w})]$, using samples taken from $\mathbf{t}|\mathbf{w}$ (the trained MEMM).

**N-best Sequence Entropy (NSE)** (Settles and Craven, 2008) approximates sequence entropy by computing the entropy of the top-$n$ sequences, where the probabilities are re-normalized to sum to unity.

**Least Confidence (LC)** (Culotta and McCallum, 2005), in contrast to entropy, is not concerned with the distribution over the entire support, but rather focuses on the best option and its complement (the rest of the support). It is the probability of being wrong, i.e., $1 - \max_{\mathbf{t}} p(\mathbf{t}|\mathbf{w})$.

**Negative Max Log Probability (NMLP)** (Haertel et al., 2010) is defined as $-\max_{\mathbf{t}} \log p(\mathbf{t}|\mathbf{w})$; it ranks instances the same as LC but with different scores under the assumption that the relationship between probabilities and change in accuracy is logarithmic rather than linear.

**Token Vote Entropy (TVE)** (Engelson and Dagan, 1996) uses a committee of classifiers trained from bootstrapped samples of the annotated data. For each word, each committee member votes for the tag it predicts for its word; the entropy of the distribution over votes is summed over each word in the sentence.

## 5 From Theory to Practice: To What Degree Are the Conditions Met?

In this section, we empirically test some of the conditions from the preceding analysis in practical contexts. For the purposes of this work, we are mostly interested in examining conditions 1 and 4. While condition 3 (conditional independence) is assumed in most previous work, we leave quantification of the effects of violating this condition and the related condition 2 to future work.

For these experiments, it is necessary to estimate true benefit and cost. Due to the complexity of so doing, we compute the various metrics along a *passive* learning curve (i.e., without AL). We compute the true cost of each instance as described in Section 4.1. In order to estimate the true benefit of a particular instance at a particular point on the learning curve, we assume that the true benefit of an instance is the change in held-out accuracy that would result from incorporating the instance with its annotation into the training data; we ignore the effects of future choices. We compute the change in accuracy (benefit) by adding the instance and its true label to the training data, retraining the model, and then computing the model's new accuracy on the held-out set. The process is repeated to compute the true benefit of at least 1,000 instances and the statistics noted below are averaged over 5 random initial training sets. We use one standard error as a simple measure of statistical significance.

**Is the covariance of cost and benefit zero?** Using the aforementioned methodology, we compute Pearson's correlation coefficient (a normalized form of covariance) between benefit and cost. As seen in figure 1a, true benefit and cost have virtually no correlation when model quality is high, and is only weakly correlated in the early stages. Thus, condition 1 roughly holds.

**To what extent is the cost estimate a scalar multiple of true cost?** Using the technique mentioned above, we produce pairs of true cost and estimated cost at various locations along the learning curve and compute $R^2$ values of a linear model estimated with the $y$-intercept fixed at zero. Pearson's correlation coefficient is inappropriate since it would allow for the cost estimate to be shifted in addition to being scaled. An $R^2$ of 1.0 would indicate that the cost estimate was an exact scalar multiple of the true cost, while a zero would indicate no scalar relationship. We repeat this test with differing amounts of variance in the simulated cost, which allows us to assess the effect of poor cost models (good models will account for most of the variance). The results are shown in Figure 1b. The exponential decay

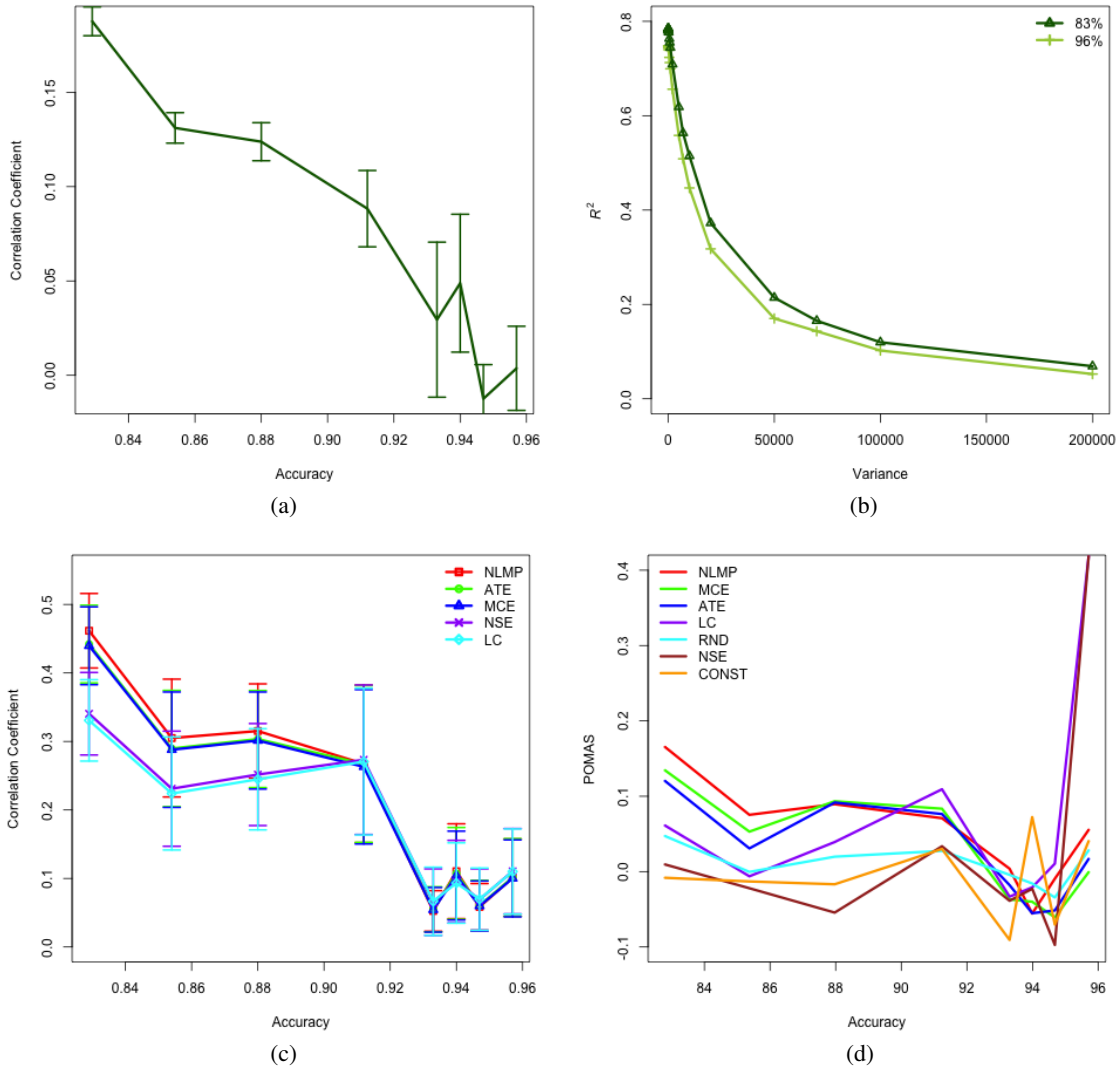


Figure 1: At various points on the learning curve: (a) correlation between true cost and true benefit (b) $R^2$ values representing the degree to which the cost estimate is a scalar multiple of the true cost, for varying amounts of variance in the noise model at two points on the learning curve (83% and 96%). (c) $R^2$ values representing the degree to which various benefit estimators are scalar multiples of true benefit (d) POMAS of the top-20 instances. Error bars represent one standard error.

as variance increases underscores the importance of accounting for as much variance as possible in the cost model. We found the $R^2$ values to be around 0.745 and 0.785 when the variance was equal to that of the aforementioned user study (and the one used through the remainder of the experiments). We note that these numbers may be overly optimistic given the similarity between the model used to simulate annotation times and that used to estimate cost. As a point of reference, Settles et al. (2008) and Arora and Nyberg (2011) report $R^2$ values for cost models for *different* tasks on the order of 0.3–0.4. Even these values indicate some scalar relationship between true and estimated cost as per condition 4.

**To what extent are various benefits estimators scalar multiples of true benefit?** We repeat the experiment described for cost, but reporting the $R^2$ values for the fit between true benefit and several benefit estimators; Figure 1c depicts the results. Although the $R^2$ values are much worse than for the

cost estimate, they are still reasonable. Most of the separation of algorithms (where it exists statistically) occurs during the beginning stages of learning. NMLP has a slight (though not statistically significant) advantage over ATE and MCE while all three are more linearly related to true benefit than NSE and LC. Once the model achieves 91% accuracy, there is no separation. The results suggest that condition 4 holds weakly for benefit estimators.

**Are instances with the highest slopes being selected?** The success of ROI depends on its ability to select the instance with the highest slope. Using the aforementioned setup, we compute the largest slope of the candidate instances on the basis of estimated benefit and cost and divide it by the largest slope according to the true values; we call this value the Percentage of Maximum Attainable Slope (POMAS). Since multiple instances can be selected using the same model in the no-wait framework, we repeat this procedure for the second highest slopes, etc., for the top-20 slopes and average them. The results are in Figure 1d. The separation between algorithms at the beginning mirror those of Figure 1c. We note that there is ample room for improvement even amongst the best algorithms we tried.

## 6 Active Learning Results and Discussion

The previous experiments were conducted outside of the context of AL in order to gain insight into how well the conditions of section 2 are met in practice. However, the most direct evaluation is the comparison of the actual quantity of interest, AUC, in the type of practical AL defined above. We compare normalized AUC (expected benefit) for several benefit estimators and two cost estimates and discuss the results in light of the previous section and the theory from Section 2.

Although not predicted by the theory per se, we would expect AUC to decrease with degradations in the cost and/or benefit estimates. First, we compare the AUC when using the true cost in the ROI calculations (thus satisfying one half of condition 4) and compare the results to using estimated cost learned during AL. The results are displayed in Figure 2. Interestingly, when cost is exactly known (perfectly predictable), all estimators–even CONST–readily outperform the random baseline. Furthermore, the difference between most of the estimators (except perhaps CONST) is not statistically significant, which suggests that a good cost estimate may be capable of overcoming deficiencies in even very poor benefit estimators like CONST. Not surprisingly, all algorithms perform worse when using the learned estimate of cost during AL (indicated by the downward arrows), even though the MSE of the learned cost models was high–on the order of the variance in the simulated times.

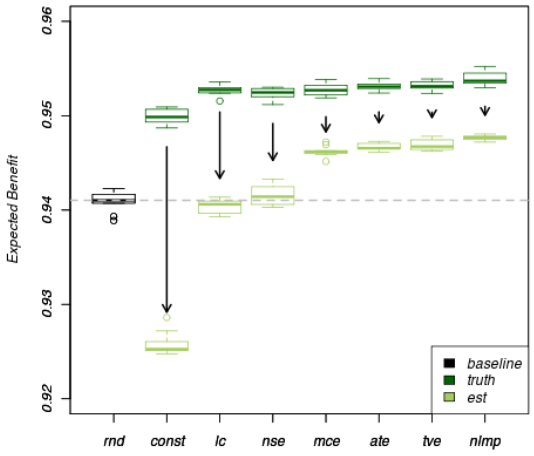


Figure 2: Expected benefit (normalized AUC) for various benefit estimators with true and estimated costs. The median baseline performance (rnd) is depicted as a dashed line and is the same for both experiments. Estimated cost affects the benefit estimates to different degrees.

Further support that AUC depends on the quality of estimates comes from the fact that the performance of the various algorithms exactly follows the quality of the corresponding benefit estimate (see Figures 1c and 1d). In fact, LC and NSE do no better than random and CONST does much worse. Upon further examination, we found a common property between these benefit estimators, namely, that most scores fall within an extremely narrow range; i.e., there was very little difference between the (benefit) scores of most instances (CONST being the extreme case). NSE differs from the other entropy estimates primarily in the re-normalization. Due to their nature, structured prediction problems have very large supports which tend to have long tails. Therefore the top-$n$ probabilities grossly underrepresent these

distributions and renormalization makes scores very similar to each other—even for instances of differing lengths. Similarly, although LC and NMLP would rank instances the same (before dividing by cost; dividing by cost alters the rankings), the *log* in NLMP produces greater spread in the score. Since the cost estimates are better dispersed, they tend to dominate ROI for these "low-spread" benefit estimators. To illustrate, consider the extreme case of CONST by substituting an arbitrary constant for benefit in equation 1: instances are selected lowest-expected-cost first. On our particular task, this scenario is particularly undesirable as the shortest sentences are nearly always the cheapest but disproportionately information poor (a contributing factor to the non-zero correlation). In more general terms, as the spread in the benefit estimates approaches zero (as in CONST), the cost estimates increasingly become the discriminating factor. While this behavior is correct for perfect benefit and cost estimates, it is problematic when condition 4 is violated.

The results also highlight the fact that expensive scoring algorithms are naturally penalized in annotator-initiated AL. The relatively expensive sampling in MCE leads to slightly lower performance than cheaper entropy estimates (ATE); the relatively cheap NMLP outperforms TVE, which incurs the expense of multiple models.

The mixed results of previous work are explainable based on our analysis. While condition 4 *requires* that cost and benefit estimators be scalar multiples of the true values, our empirical results suggest that better estimates yield higher AUC. We have explained why NSE has poor mathematical properties for structured learning tasks and is therefore expected to produce relatively low AUC, hence the negative results on the structured prediction tasks of Settles et al. (2008). In contrast, the authors report positive results on a standard classification problem using exact entropy calculations, coinciding with our results in which the good (i.e., non-NSE) entropy estimators are good estimators. We have also explained the poor properties of LC for structured prediction; the results of Tomanek and Hahn (2010) present further empirical evidence. Interestingly, they find that exponentiating LC leads to positive results. Mathematically, $exp(\beta(1-p))$ behaves similarly to $-log(p)$ (NLMP) in that they both separate scores that are close together—the former much more so than the latter, especially for probabilities of the very low magnitudes seen in structured prediction problems. This separation gives the benefit estimate more influence relative to cost as compared to LC. In sum, the negative results of previous work are due to poor benefit estimators, in particular LC and NSE; in contrast, positive results are due to better benefit estimators.

## 7 Conclusions and Future Work

ROI-based AL successfully reduces annotation costs in practice by maximizing the area under the cost/benefit curve. We have provided an initial theoretical justification for ROI-based AL in a bottom-up fashion. We have shown empirically that, for our task, true benefit and cost have little-to-no correlation when model quality is high; cost estimates have a scalar relationship to true cost; similarly for benefit estimates, though to a lesser degree; and the estimators that demonstrated the most scalar relationships to the truth resulted in higher AUC.

Although we focused our empirical analysis on a single task, other studies have applied ROI to several tasks and problem types, and their results are consistent with our analysis. As a result of this work, we recommend that practitioners carefully select their benefit and cost estimators, ensuring that they are "good" estimators for their task as described above. Particular attention should be paid to the cost estimator: even trivial benefit estimators out-performed random with a perfect cost estimator. Also note that estimators (e.g. NSE and LC) that produce scores with relatively little "spread" should be avoided. Future work could consider using a small set of annotated data to estimate how scalar the relationship of the estimators are to true benefit and cost before annotation begins.

Our empirical results suggest that deficiencies in even the best benefit estimators lead to the selection of suboptimal instances. Future work could focus on directly and tractably estimating true cost and benefit for structured prediction problems, and automatically tuning heuristic estimators to match true benefit during AL. Future work may also benefit from investigating a different set of conditions for simplifying equation 4.

## References


Brigham Anderson and Andrew Moore. 2005. Active learning for hidden Markov models: Objective functions and algorithms. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 9–16.

Shilpa Arora and Eric Nyberg. 2011. Assessing benefit from feature feedback in active learning for text classification. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 106–114. Association for Computational Linguistics.

Jason Baldridge and Miles Osborne. 2004. Active learning and the total cost of annotation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. 1996. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145.

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, page 746.

Pinar Donmez and Jaime G. Carbonell. 2008. Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. In *Proceeding of the 17th ACM Conference on Information and Knowledge Management*, pages 619–628. ACM.

Sean P. Engelson and Ido Dagan. 1996. Minimizing manual annotation cost in supervised training from corpora. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, pages 319–326.

Robbie A. Haertel, Kevin D. Seppi, Eric K. Ringger, and James L. Carroll. 2008. Return on investment for active learning. In *Proceedings of the Neural Information Processing Systems Workshop on Cost Sensitive Learning*.

Robbie Haertel, Paul Felt, Eric Ringger, and Kevin Seppi. 2010. Parallel active learning: Eliminating wait time with minimal staleness. In *Proceedings of the HLT-NAACL 2010 Workshop on Active Learning for Natural Language Processing*, pages 33–41. Association for Computational Linguistics.

Robbie A. Haertel. 2013. *Practical Cost-Conscious Active Learning for Data Annotation in Annotator-Initiated Environments*. dissertation, Brigham Young University.

Ashish Kapoor, Eric Horvitz, and Sumit Basu. 2007. Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *Proceedings of the International Joint Conferences on Artificial Intelligence*.

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning from measurements in exponential families. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 641–648. ACM.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Dragos D. Margineantu. 2005. Active cost-sensitive learning. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, volume 19, page 1622.

Eric Ringger, Marc Carmen, Robbie Haertel, Kevin Seppi, Deryle Londsale, Peter McClanahan, James Carroll, and Noel Ellison. 2008. Assessing the costs of machine-assisted corpus annotation through a user study. In *Proceedings of the International Conference on Language Resources and Evaluation*.

Nicholas Roy and Andrew McCallum. 2001. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 441–448.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079. Association for Computational Linguistics.

Burr Settles, Mark Craven, and Lewis Friedland. 2008. Active learning with real annotation costs. In *Proceedings of the Neural Information Processing Systems Workshop on Cost-Sensitive Learning*, pages 1069–1078.

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

Tong Zhang and Frank J. Oles. 2000. The value of unlabeled data for classification problems. In *Proceedings of the Seventeenth International Conference on Machine Learning, (Langley, P., ed.)*, pages 1191–1198.