Return on Investment for Active Learning

Robbie A. Haertel, Kevin D. Seppi, Eric K. Ringger, James L. Carroll
Department of Computer Science
Brigham Young University
Provo, UT 84602
{rah67,kseppi,ringger}@cs.byu.edu,jlcarroll@gmail.com

Abstract

Active Learning (AL) can be defined as a selectively supervised learning protocol intended to present those data to an oracle for labeling which will be most enlightening for machine learning. While AL traditionally accounts for the value of the information obtained, it often ignores the cost of obtaining the information thus causing it to perform sub-optimally with respect to total cost. We present a framework for AL that accounts for this cost and discuss optimality and tractability in this framework. Using this framework we motivate Return On Investment (ROI), a practical, cost-sensitive heuristic that can be used to convert existing algorithms into cost-conscious active learners. We demonstrate the validity of ROI in a simulated AL part-of-speech tagging task on the Penn Treebank in which ROI achieves as high as a 73% reduction in hourly cost over random selection.

1 Introduction

Labeled data is a pre-requisite for many algorithms in Natural Language Processing (NLP) and machine learning. While large amounts of annotated data are available for well-studied languages in well-studied domains and for well-studied problems such as part-of-speech (POS) tagging, this is not true for less common languages or domains. Unfortunately, obtaining human annotations for linguistic data is labor intensive and typically the costliest part of the acquisition of an annotated corpus; a great deal of funding continues to be devoted to the annotation of data. Hence, there should be substantial interest in reducing annotation costs while preserving quality.

Active Learning (AL) can be employed to reduce the costs of corpus annotation [1, 2, 3]. The primary responsibility of the active learner has been to choose items for which it believes the true annotations will provide the most benefit. However, cost must also be considered [4, 5], and AL should deliver annotations that balance cost and utility.

In this paper, we present AL as an order optimization and decision problem and discuss the optimal approach to AL based on decision theory. We motivate the need for this approach to be based on the "net utility" (NU), which incorporates both the utility and cost of annotating an entire sequence of items. However, due to the intractability of the optimal approach, greedy approaches and heuristics must be used. We present a practical, novel heuristic that combines utility and cost in a measure called Return On Investment (ROI). We apply this cost-sensitive perspective and demonstrate the advantage of this technique over traditional AL methods in a simulated AL part-of-speech tagging task on the Penn Treebank.

The rest of the paper will proceed as follows: Section 2 discusses the role of both benefit and cost in AL. Section 3 provides a framework for AL which we use to discuss previous work and motivate our approach, ROI, which is subsequently introduced in Section 4. We explain our methodology for conducting experiments in Section 5, and results are presented in Section 6. Finally, Section 7 discusses conclusions and indicates our plans for future work.

2 The Role of Cost and Benefit

Pivotal to AL is the observation that not all data are created equal: some data are inherently more beneficial than others. Key to this work is the fact that data also differ in how costly they are to annotate. Although this is true for nearly all annotation tasks, it is particularly obvious when annotating sequences or other structured objects, which are common in NLP and bioinformatics, among other fields. Consider for example the manual POS tagging task. All else being equal, it will clearly take more time, and hence cost more, to annotate longer sentences. On the other hand, sentences containing frequent words or features that provide discriminatory information tend to be more valuable. An active learner should seek sentences that are most beneficial and least costly.

In order for an active learner to determine which data are "most beneficial" and "least costly", benefit (utility) and cost must be clearly defined. The exact determination of cost and utility are typically project-dependent and sometimes difficult to measure. Nevertheless, the performance of an AL algorithm is ultimately determined by both the true cost and utility, and hence, any attempt to characterize the performance of the algorithm should measure both aspects as closely as possible.

There are many facets to utility, but in this work we focus on measures of model goodness, in particular, accuracy. However, this may not capture all aspects of true utility. For example, Baldridge and Osborne [6] have suggested that reusability of the annotated data can be important. Similarly, the cost of an AL project depends on many variable and fixed costs [7], but how the annotators will be paid is one of the most significant factors. The usual assumption for AL is unit cost per label, which is unrealistic particularly for tasks involving labeling sequences or other structured data. Becker et al. [4] and Haertel et al. [5] show how the relative performance of different algorithms is determined in part by how cost is measured. Many of the costs employed in other work (e.g., [8, 9]) can be seen as estimating some portion of the hourly cost. The nature of the annotation user interface, the number and efficiency of annotators, etc. are additional components of cost [10]. There is also a human learning curve that inevitably impacts the cost of an AL system.

3 Background and Decision Theoretic Framework for Active Learning

Most work in AL assumes that a single unerring oracle provides annotations (however, this framework can also be applied to cases when the annotator is fallible or there is more than one annotator). AL is an order optimization problem [11]: a rational organization that desires to annotate a particular set of initially unannotated items $\mathcal{U}$, will seek the totally ordered subset $\mathcal{A}^* \subseteq \mathcal{U}$ such that $\mathcal{A}^* = \operatorname{argmax}_{\mathcal{A} \subseteq \mathcal{U}} \text{NU}(\mathcal{A})$ where $\text{NU}\left(\mathcal{A}\right) = \text{utility}\left(\mathcal{A}\right) - \text{cost}\left(\mathcal{A}\right)$, and "utility" and "cost" (discussed in the previous section) are functions of an ordered set that must be on the same scale. Furthermore, no rational organization would undertake a project if $\text{NU}(\mathcal{A}^*) < 0$.

Decision theory provides a statistically optimal approach to finding $\mathcal{A}^*$ with respect to the utility and cost functions. The test selection problem consists of using the Expected Value of Sampling Information (EVSI) [12] to choose which "test", if any, could be performed that would maximize the expected net utility (ENU). In AL, a "test" consists of requesting an annotation from the oracle. Since each query to the oracle potentially affects the NU of future queries, it is necessary to consider the NU of full sequences of tests/queries. Let $\mathcal{A}$ be the set of data with annotations and $\mathcal{U}$ be the unannotated data. The optimal choice of item for annotation is:

$$x^* = \operatorname*{argmax}_{x \in \mathcal{U}} \mathbb{E}_{Y=y|x,\mathcal{A}} \left[R \left(\mathcal{A} \cup \{(x,y)\}, \mathcal{U} - x \right) \right] \tag{1}$$

where Y is a r.v. representing the annotation for instance x and R is the maximum ENU:

$$R(\mathcal{A},\mathcal{U}) = \max \left(\max_{x \in \mathcal{U}} \mathbb{E}_{Y=y|x,\mathcal{A}} \left[R \left(\mathcal{A} \cup \{(x,y)\}, \mathcal{U} - x \right) \right], NU\left(\mathcal{A}\right) \right)$$

which is computed by recursively choosing the next test instance that maximizes the ENU given the updated $\mathcal{A}$. R is at a maximum when future iterations are no longer expected to increase the value of R or there is no data left to consider (in other words, $R(\mathcal{A}, \emptyset) = NU(\mathcal{A})$).

The optimal AL algorithm therefore is to choose an item x^* according to equation 1, query the oracle for the annotation y, add the annotated pair (x, y) to $\mathcal{A}$, and repeat until R is not greater than the current NU. Note that although some have recognized their approach as greedy (e.g. [13]), Carroll et al. [14] and this work are the first to explicate the fully optimal approach, as far as we are aware. Although computing expectations for every possible permutation of unannotated items (performed by the recursive function R) at each stage of AL is clearly intractable, knowing the optimal approach has guided the current work and should help direct future work towards better approximations.

Several authors have used a greedy version of EVSI-based AL in which no recursion is necessary (e.g. [15, 16, 13, 17, 18, 19]). Even this greedy approach can be expensive, since posterior probabilities must be computed for every possible labeling of every possible unannotated item. Some gains in efficiency can be had using models that allow for efficient (albeit often approximate) computation of posterior probabilities (e.g. [15, 19]), and it is also possible to consider a subset of possible tests and use sampling to compute ENU [16].

Even with such simplifications, the greedy approximation can be computationally costly for many problems and models. Rather than computing posterior probabilities, heuristic methods may be used which typically require training a manageable number of models on the previously annotated data and using these models to rank the remaining items according to some criterion. Most heuristics attempt to score instances in such a way that the scores are proportional to the expected change in utility. The most common heuristics assume that this change is proportional to some type of uncertainty. Let Y be the random variable representing a label assignment to the test instance x. Then we refer to the uncertainty of the distribution of $Y|x$ as *ambiguity*. However, since the true distribution of $Y|x$ is unknown, it is possible for the mode of any estimate to be incorrect, leading to an incorrect classification. The distribution over possible modes of $p\left(Y|x\right)$, i.e. $p\left(Z = \operatorname{argmax}_y p\left(Y = y|x\right)\right)$ (c.f. [20]), represents a type of uncertainty we call *correctness*.

Ambiguity and *correctness* can be quantified in several ways, including entropy. Although entropy takes into account the mass or density across the entire support, the most important question is whether or not the model is correct. A more direct measure is to use the probability that the model is wrong, i.e. $1 - p(mode)$ (One-Minus-Max; OMM) (c.f. [2]). Another interesting measure of uncertainty when using committee-based approaches is K-L divergence from the mean [21]. This method measures the average distance (computed using K-L divergence) of the distribution of each member of a committee to the mean distribution of the committee. This has the benefit that truly ambiguous distributions usually have low variance and hence are avoided. Anderson & Moore [18] prove that using entropy and K-L divergence are in fact the same.

Approaches in which a single model assesses *ambiguity* are referred to as uncertainty sampling [22] (also Query-By-Uncertainty; QBU). Monte Carlo techniques can be used to assess both *ambiguity* and *correctness*; approaches for the latter are known as Query-By-Committee (QBC) [23, 1].

4 Return on Investment

Optimizing ENU directly is difficult, even greedily. First, computing posterior probabilities and therefore ENU can often be prohibitively costly. Assuming annotators are paid while waiting for the active learner, long computation *should* result in lower NU. Lamentably, this has yet to be properly accounted for. A second challenge to using NU is that, in practice, it tends to be difficult to define cost and utility in the same units. Anderson & Moore [18] recognize that this is particularly problematic for entropy-based utility functions. It is also especially true when it is not possible or feasible to calculate the maximum EU of the unannotated data (the most prevalant utility function). The fact that the majority of work on AL assesses performance using graphs of cost vs. utility rather than number of queries vs. NU attests to this difficulty. Most AL work assumes constant cost (often unjustifiably) in which case the graphs are scaled variants of each other. Even those approaches that use more justifiable cost measures frequently measure performance on the basis of cost vs. utility (e.g. [1, 3, 5]).

In the absence of a good conversion between units of cost and utility, the use of plots of cost vs. utility is justified since they still encapsulate useful information. For instance, AL algorithms that produce lines having greater area under them tend to correspond to algorithms with higher maximum NUs. Furthermore, some algorithms tend to exhibit higher utility for nearly all levels of cost when compared to other algorithms, which translates to higher NU in the same regions.

For this reason, it is desirable to employ algorithms that select items that maximize utility per unit of cost, specifically in the case that it is not practical or even possible to compute ENU. A straightforward method of maximizing this quantity is through the use of Return on Investment (ROI). ROI is defined as: $\text{ROI}(x) = \frac{\text{utility}^*(x) - \text{cost}^*(x)}{\text{cost}^*(x)} = \frac{\text{utility}^*(x)}{\text{cost}^*(x)} - 1$. When using ROI as a selection metric, the learner ranks items by the ratio of an estimate of the utility of annotating each item to an estimate of the cost to annotate it; the item that maximizes this quantity is selected for annotation. Note that these estimates can be functions of any form, including non-linear and unbounded functions. Conveniently, it is not necessary to compute the non-trivial conversion ratio required by NU, yet, unlike the traditional heuristics such as QBU and QBC, ROI is clearly cost-sensitive. As desired, ROI (greedily) maximizes utility per unit of cost. ROI selects items whose *estimated* slope on a learning curve of cost vs. utility are at a maximum given the previous points, provided that the cost and utility estimators used in ROI are the same (or similar) to those measured in the graph. This generally leads to learning curves that tend toward the upper-left corner of the graph (lower cost and higher utility), which is precisely what constitutes good AL algorithms.

5 Methodology

In the remainder of this work, we consider several AL selection algorithms that leverage the idea of ROI for the POS tagging task. As previously defined, these algorithms consist of two parts: a utility estimator and a cost estimator. By combining different cost and utility estimators in the ROI model introduced above, we create novel selection algorithms and then evaluate their performance.

5.1 Utility Estimation

Computation of uncertainty proceeds differently for *ambiguity* and *correctness*. In order to compute the *ambiguity* of a sentence by way of entropy, we approximate the per-sentence tag sequence entropy by summing over an estimate of the conditional per-word tag entropy where the previous tags are taken from the Viterbi-best sequence of tags [5]. For OMM, it is sufficient to subtract the probability of the Viterbi-best tag sequence from one.

Similar to Engelson & Dagan [1], for *correctness*, each member of a committee produces the most probable tag sequence for a sentence and a vote histogram is created for each word; each member votes once for the tag it chose. The same measures of uncertainty can be computed for this vote distribution as above.

5.2 Cost Estimation

There are three cost estimates that we can use in the absence of the truth during the annotation process. First, we can assume constant cost per sentence. Second, we can assume that cost is proportional to the number of words in the sentence. Engelson and Dagan [1] as well as Tomanek et al. [3] do the latter and thereby implicitly assume the "length" cost function without explicitly acknowledging the question of cost. Lastly, Haertel et al. [5] argue that the most important cost is hourly cost. For the purposes of the experiments reported in this work, we estimate the length of time required to annotate any given sentence with POS tags using the "hourly cost model" of time required for annotation that Ringger et al. [2] developed based on a user study:

$$h = (3.80 \cdot l + 5.39 \cdot c + 12.57) / 3600 \tag{2}$$

where h is the time in hours spent on the sentence, l is the number of tokens in the sentence, and c is the number of tags in the sentence that need correction. The expected value of this cost function can easily be estimated in ROI. Since l is known for each sentence, we need only estimate $\hat{c}$, the expected number of tags to be changed in a sentence $\underline{\text{w}}$ with the most-likely tags $\underline{\text{t}}$ selected by the tagger. In our model (described in the following section), this is $\hat{c} = \sum_{i=1}^{|\underline{\text{W}}|} \left[1 - p\left(t_i | \underline{\text{w}}, t_{i-2}, t_{i-1}, \boldsymbol{\phi}\left(\underline{\text{w}}, i, t_{i-2}, t_{i-1}\right)\right)\right]$.

5.3 Experimental Setup

We employ an order-two Maximum Entropy Markov Model (MEMM) for all models. The features used in this work are mostly based on work by Toutanova and Manning [24]; the tagger is near the state-of-the-art, achieving 96.90% in tag accuracy once AL reaches completion.

			Cost Estimator		
	Distr. Type	Uncertainty Calc.	Constant	Num Words	Exp. Hourly Cost
Utility Est.	ambiguity	entropy	QBUE*	QBUE/N*	QBUE/EHC
		one-minus-max	QBUOMM	QBUOMM/N	QBUOMM/EHC
	correctness	entropy	QBCE	QBCE/N	QBCE/EHC
		one-minus-max	QBCOMM*	QBCOMM/N*	QBCOMM/EHC
		K-L divergence	QBCKL*	QBCKL/N*	QBCKL/EHC

Table 1: Naming conventions for the various combinations of utility and cost estimators used in ROI for the experiments; asterisks indicate combinations not investigated in this work.

Our data consists of English prose from the POS-tagged Wall Street Journal text in the Penn Treebank (PTB) version 3. Sections 2-21 serve as initially unannotated data and section 24 is the set on which tag accuracy is computed at the end of every round of AL. By using the true labels and comparing them to the output of a tagger trained on the sentences annotated up through the end of the current round, it was possible to use equation 2 to estimate the amount of time that would have been required to annotate each sentence. Ideally, we would evaluate the effectiveness of the various algorithms by measuring the actual time of human annotators. Unfortunately, this would require that we pay annotators to tag an entire data set many times to achieve a statistically significant comparison of the algorithms. Clearly this is not feasible, although Hachey et al. [25] do this (once) for three algorithms on a set of 100 sentences. Our experiments show that the largest cost reductions over baseline come long after the first 100 sentences. In a real AL environment, it would be possible to estimate annotation cost during the annotation process, an idea we wish to pursue in future work.

In order to minimize the amount of time that the human waits for the computer to retrain the model and choose the next sentence to annotate, it is sometimes desirable to process sentences in batches, even though this can slightly decrease the efficiency of AL. Sentence selection and model training is relatively cheap in the early stages of AL and accuracy rises very quickly, whereas the opposite is true at the later stages. Based on initial tests, we found that a batch size that increases exponentially as a function of the number of previously annotated sentences had little perceivable effects on the accuracy achieved for any (human) hourly cost. In a real annotation task, batch size can be self-determined by the amount of time it takes the algorithm to select a sentence in the previous round. Following Engelson & Dagan [1], rather than scoring every unannotated sentence, we instead score only a sample of the sentences. We found that maintaining a candidate set size 500 times the size of the batch worked well. For committee-based approaches, the k committee members were trained using separate bootstrap samples from the available annotated data. Following Tomanek et al. [3], we use a committee of size three as a reasonable balance between computational burden and model diversity. Results are averaged over a minimum of 15 runs from different random initial sentences.

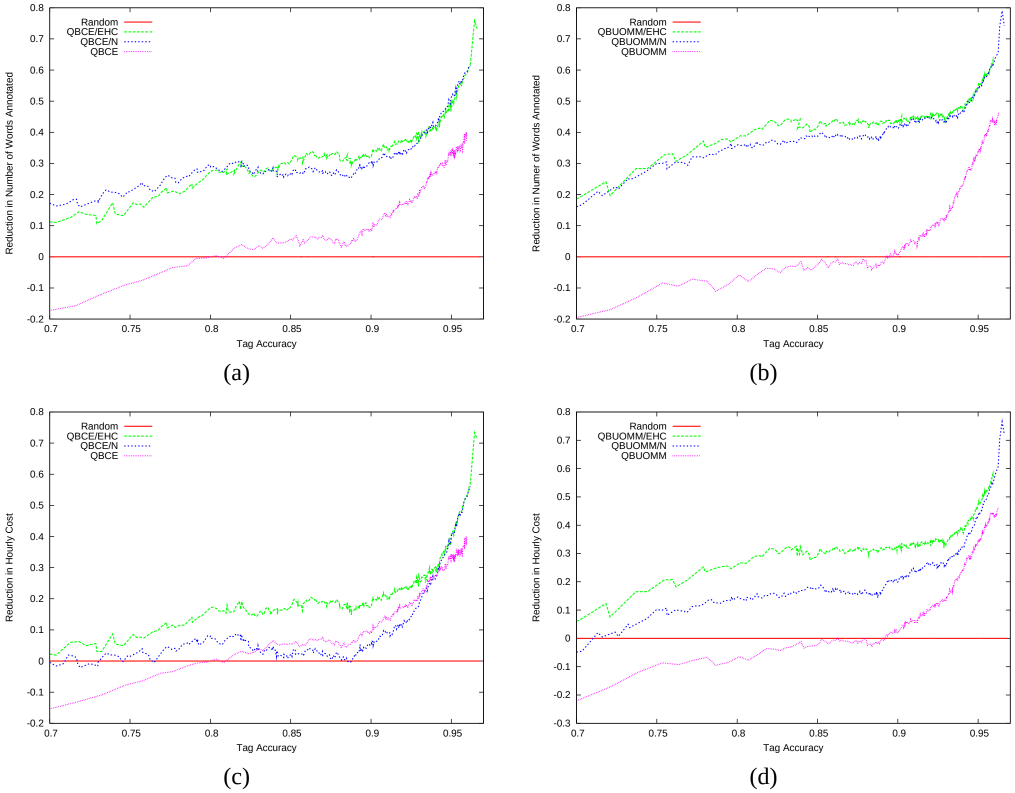
To assess the performance of each algorithm, we follow Haertel et al. [5] in their use of "cost reduction" plots. In these plots, the accuracy (utility) is the independent variable. The percent reduction in cost over the cost of the random baseline for the same accuracy a is then computed as $r(a) = \left(cost_{rnd}\left(a\right) - cost\left(a\right)\right) / cost_{rnd}\left(a\right)$. Consequently, the random baseline has no reduction compared to itself and represents the trajectory $r = 0.0$. For a specific (cost, accuracy) point on a learning curve, the corresponding value of the cost on the baseline learning curve is estimated by linear interpolation between neighboring points on the baseline. In these graphs, a higher reduction represents a greater advantage.

6 Results

To show the efficacy of the ROI approach, we separately investigate the effects of different cost and utility estimators. The naming conventions for the various combinations are summarized in Table 1.

6.1 Cost Estimators

In Figure 1, we plot the reduction of hourly cost using each of the three different cost estimates for the ROI algorithm (constant per-sentence cost, length of sentence, and estimated hourly cost) and two different utility estimators (QBCE and QBCOMM). Each of these algorithms is evaluated on the



(a) (b)

(c) (d)

Figure 1: Comparison of three cost measures (EHC, N, and constant) for QBCE ((a),(c)) and QBUOMM ((b),(d)) when paying annotators per tagged word ((a),(b)) or per hour ((c),(d)).

basis of two different costs: pay-by-the-word, and pay-by-the-hour. We chose to use QBCE since it is the one employed in [1] for POS tagging (the actual algorithm is our QBCE/N variant). Haertel et al. [5] have previously shown QBCOMM/N to be superior to QBCE/N when paying annotators by the hour on this task for this dataset.

The cost-conscious algorithms (i.e. the N- and EHC-variants) outperform those that assume constant cost. The similarity of these variants is due in part to the role of sentence length in the hourly cost model and the fact that after only a few dozen iterations, the tagger has reached high enough accuracy that the number of words needing correction is low. Interestingly, when paying annotators per word annotated (Figures 1(a)-(b)), the variants that use estimated hourly cost tend to (barely) out-perform those that directly estimate the per-word-cost. We note that the models are error-prone in the early stages of AL. We hypothesize that since the same error-prone model is used in the computation of both the numerator and the denominator in the EHC-variants, errors effectively “cancel out” whereas for the N-variants, the model is used only in the computation of the numerator.

When evaluating the performance of the algorithms based on paying annotators by the hour (Figures 1(c)-(d)), QBCE/EHC reaches an advantage over baseline of around 73%; even greater advantages are reached by the cost-conscious QBUOMM variants. Importantly, the EHC-variants enjoy a greater reduction than the other cost estimators at all accuracy levels, providing empirical evidence that ROI typically maximizes utility per cost when the estimators match those being measured. However, the advantage of the EHC-variants compared to the N-variants diminishes at high levels of accuracy because EHC is approximately proportional to the length of the sentence for high levels of accuracy. Both variants exhibit a growing advantage over the non-cost-conscious QBCE and QBUOMM at these higher levels of accuracy. Interestingly, even these constant cost estimators have a clear advantage over the baseline (nearly a 40% reduction in cost for QBCE and 46% for QBUOMM). Finally, QBCE/N has little advantage over baseline when paying annotators by the hour until it reaches around 89% accuracy (Figure 1(c)).

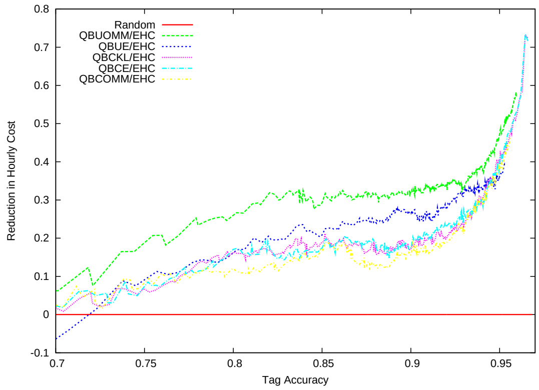


Figure 2: Comparison of utility estimators.

6.2 Utility Estimators

Since the EHC estimate of cost outperforms the others, we fix this estimate of cost to evaluate the different utility (accuracy) estimators. Figure 2 depicts the relative advantage of five accuracy estimators: two single model *ambiguity*-based estimates, QBUE/EHC and QBUOMM/EHC; two committee-based *correctness* estimates, QBCE/EHC and QBCOMM/EHC; and KL-divergence to the mean, QBCKL/EHC.

All algorithms enjoy a relatively large reduction in cost over baseline, with QBCE/EHC and QBCKL/EHC achieving nearly a 73% reduction in cost. We were unable to collect as much data for QBUOMM/EHC, but it would appear to enjoy an even greater advantage. The OMM approach outperforms the entropy-based method for *ambiguity* estimators, although no such advantage exists for *correctness* estimators. All of the committee-based variants performed nearly equally well. We suspect that one limiting factor is the committee size and possibly our use of bootstrap sampling.

We were somewhat surprised to see that the *ambiguity*-based estimates of accuracy outperformed the correctness-based estimates. The *ambiguity*-based estimates are known to struggle with cases that are truly ambiguous and hence *correctness*-based approaches were thought to perform better in this context. Small committee size could be a factor, but perhaps entire tag sequences are less ambiguous than individual tags, due to potentially disambiguating interactions between words.

7 Conclusions and Future Work

We began this paper by discussing the role and importance of utility and cost in AL. By explicating a framework for AL, we hope that better cost-conscious approximations can be built. To this end we introduced a new heuristic, ROI. One advantage to ROI is that any existing algorithm can become cost-conscious by simply normalizing existing utility estimators by an estimate of cost.

In our experiments, we focused on POS tagging of the Penn treebank, comparing the accuracy of different combinations of cost and utility in ROI. We showed that, not surprisingly, one should usually employ that estimate of cost in the ROI algorithm which matches the one to be measured. We also found that algorithms based on uncertainty due to ambiguity worked better than the correctness-based algorithms for POS tagging on the Penn Treebank.

To the best of our knowledge, whereas some prior work (e.g., [19]) acknowledges query cost and other prior work (e.g., [1]) estimates query cost in an implicit fashion, this work is the first to present results of AL algorithms that explicitly incorporate a realistic query cost based on a predictive model of human annotation time. The ROI framework also generalizes those earlier efforts. Furthermore, our results suggest that, despite its simplicity, ROI is a valuable first step towards developing cost-conscious AL algorithms. We note that although ROI is a simple concept, the difficult part consists in developing appropriate estimates of cost and utility, and every project will need to develop their own estimates. Nevertheless, this work suggests that doing so is worthwhile since ROI can improve the efficiency of AL, making effective cost-conscious AL feasible.

It should be possible to allow the human to annotate while the computer selects a sentence. In future work, we intend to explore how to take better advantage of overlapping the work of the human and the computer to reduce overall cost. Finally, although our experiments dealt with POS tagging on the Penn Treebank, we have explained why ROI performs well and hypothesize that ROI performs equally well on many other AL tasks, especially other sequence-labeling problems. Testing this hypothesis is also the subject of future work.

References

[1] S. Engelson and I. Dagan, “Minimizing manual annotation cost in supervised training from corpora,” in *Proc. of ACL*, pp. 319–326, 1996.

[2] E. Ringger, P. McClanahan, R. Haertel, G. Busby, M. Carmen, J. Carroll, K. Seppi, and D. Lonsdale, “Active learning for part-of-speech tagging: Accelerating corpus annotation,” in *Proc. of Linguistic Annotation Workshop*, pp. 101–108, 2007.

[3] K. Tomanek, J. Wermter, and U. Hahn, “An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data,” *Proc. of EMNLP-CoNLL*, pp. 486–495, 2007.

[4] M. Becker, B. Hachey, B. Alex, and C. Grover, “Optimising selective sampling for bootstrapping named entity recognition,” in *Proc. of the ICML Workshop on Learning with Multiple Views*, 2005.

[5] R. Haertel, E. Ringger, K. Seppi, J. Carroll, and P. McClanahan, “Assessing the costs of sampling methods in active learning for annotation,” in *Proc. of ACL*, (Columbus, OH, USA), June 2008.

[6] J. Baldridge and M. Osborne, “Active learning and the total cost of annotation,” *Proc. of EMNLP*, 2004.

[7] G. Ngai and D. Yarowsky, “Rule writing or annotation: cost-efficient resource usage for base noun phrase chunking,” in *Proc. of ACL*, pp. 117–125, 2000.

[8] R. Hwa, “Sample selection for statistical parsing,” *Computational Linguistics*, vol. 30, pp. 253–276, 2004.

[9] M. Osborne and J. Baldridge, “Ensemble-based active learning for parse selection,” in *Proc. of HLT-NAACL*, pp. 89–96, 2004.

[10] J. L. Carroll, “Modeling the annotation process for ancient corpus creation,” in *Chatreššar 2007: Proceedings of the International Conference of Electronic Corpora of Ancient Languages* (P. Zemánek, J. Gippert, H.-C. Luschützky, and P. Vavroušek, eds.), pp. 25–42, Charles University, 2007.

[11] A. Culotta and A. McCallum, “Reducing labeling effort for structured prediction tasks,” in *Proc. of AAAI*, 2005.

[12] H. Raiffa and R. Schlaifer, *Applied Statistical Decision Theory*. Harvard University Press, 1961.

[13] G. Schohn and D. Cohn, “Less is more: Active learning with support vector machines,” in *Proc. of ICML*, pp. 839–846, 2000.

[14] J. L. Carroll, N. Toronto, K. D. Seppi, and R. A. Haertel, “Explicit utility in supervised learning.” Poster at NIPS 2008 Workshop on Cost Sensitive Learning, 2008.

[15] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, “Active learning with statistical models,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[16] N. Roy and A. McCallum, “Toward optimal active learning through sampling estimation of error reduction,” in *Proc. of ICML*, pp. 441–448, 2001.

[17] D. D. Margineantu, “Active cost-sensitive learning,” in *Proc. of IJCAI*, 2005.

[18] B. Anderson and A. Moore, “Active learning for hidden markov models: objective functions and algorithms,” in *Proc. of ICML*, pp. 9–16, 2005.

[19] A. Kapoor, E. Horvitz, and S. Basu, “Selective supervision: Guiding supervised learning with decision-theoretic active learning,” in *Proc. of IJCAI*, 2007.

[20] I. Dagan and S. P. Engelson, “Committee-based sampling for training probabilistic classifiers,” in *Proc. of ICML*, p. 150–157, 1995.

[21] A. K. McCallum and K. Nigam, “Employing EM and pool-based active learning for text classification,” in *Proc. of ICML*, pp. 350–358, 1998.

[22] S. Thrun and K. Moeller, “Active exploration in dynamic environments,” in *NIPS*, vol. 4, pp. 531–538, 1992.

[23] H. S. Seung, M. Opper, and H. Sompolinsky, “Query by committee,” in *Proc. of CoLT*, 1992.

[24] K. Toutanova and C. Manning, “Enriching the knowledge sources used in a maximum entropy part-of-speech tagger,” in *Proc. of EMNLP*, pp. 63–70, 2000.

[25] B. Hachey, B. Alex, and M. Becker, “Investigating the effects of selective sampling on the annotation task,” in *Proc. of CoNLL*, 2005.