Appears in *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*. 2008.

Active Learning with Real Annotation Costs

Burr Settles[*†], **Mark Craven**[†*], **and Lewis Friedland**[‡]
[*]Department of Computer Sciences
[†]Department of Biostatistics & Medical Informatics
[‡]School of Journalism & Mass Communication
University of Wisconsin
Madison, WI 53706
{bsettles@cs,craven@biostat,lfriedla@facstaff}.wisc.edu

Abstract

The goal of active learning is to minimize the cost of training an accurate model by allowing the learner to choose which instances are labeled for training. However, most research in active learning to date has assumed that the cost of acquiring labels is the same for all instances. In domains where labeling costs may vary, a reduction in the number of labeled instances does not guarantee a reduction in cost. To better understand the nature of actual labeling costs in such domains, we present a detailed empirical study of active learning with annotation costs in four real-world domains involving human annotators.

1 Introduction

Traditional supervised learning algorithms use whatever labeled data is available to induce a model. An *active learning* algorithm, by contrast, may choose which instances are labeled and added to the training set. Typically, a learner begins with a small set of labeled instances, selects a few informative instances from a pool of unlabeled data, and *query* for labels from an *oracle* (e.g., a human annotator). The goal is to reduce the total labeling cost incurred to train an accurate model.

Most previous work in active learning has assumed a fixed cost for acquiring each label, i.e., all queries are equally expensive for the oracle. However, consider a task that involves classifying or extracting information from text documents; such documents can vary considerably in length and the complexity of language used. These variables most likely affect the amount of work required to label different instances. Also consider that the queries that are most valuable to the learner may be the most difficult or ambiguous cases, and therefore the most expensive for an oracle to label accurately. These issues have serious implications for using active learning in practice. We argue that, in order to truly reduce the labeling cost required to build an accurate model, the notion of annotation cost must be better understood and incorporated into the active learning process.

In some problem domains, the cost required to label an instance is known before the learner makes a query. For example, if labels are acquired by executing a biological experiment, then the cost of a query might be the price of the materials used [9], which is presumably fixed and known to the learner. In this paper, we are concerned with reducing annotation costs that are *not known* in advance. Specifically, we investigate reducing annotation time for tasks involving human annotators.

The vast majority of research in active learning has not considered that instances may vary in labeling cost. Some methods have been developed for the situation in which an *active classifier* may incur a cost to obtain additional feature values at classification time [6]. Our research, in contrast, is focused on settings in which unlabeled instances (and their feature descriptions) are readily available, but the labeling process incurs a cost at training time. One proposed approach for reducing human annotation effort in active learning involves using the current learned model to assist in the labeling

of query instances in structured-output tasks like parsing [1] or named entity recognition [4]. However, these methods do not actually represent or reason about costs. Instead, they attempt to reduce the number of annotation actions required for a query that has *already* been selected.

The prior work most closely related to ours is a group of methods that explicitly account for varying label costs in active learning. One such cost-sensitive query strategy was proposed by Margineantu [13], but differs from ours in that it assumes that labeling costs are known for each instance; the paper also provides no empirical evaluation using real-world data sets. Kapoor et al. [7] have developed an approach that takes into account both labeling costs and estimated misclassification costs. They applied their method in a voicemail classification task, but instead of using real cost information, their experiments make the simplifying assumption that the cost of labeling a message is a linear function of its length (e.g., ten cents per second). King et al. [9] present the only work that, to our knowledge, uses active learning in an attempt to reduce real labeling costs. They describe a "robot scientist" which can execute a series of autonomous biological experiments to discover metabolic pathways, with the objective of minimizing the cost of materials used. In contrast to the tasks we consider, the labeling cost of each instance in this domain is known prior to querying an instance.

In this investigation, we present a detailed analysis of four data sets for which we have measured the actual annotation cost (labeling time, and in some cases labeling actions) incurred by humans annotators. We then attempt to answer several important questions about the role of such annotation costs in real-world active learning.

2 Data Sets and Annotation Methodology

Because most active learning research has not been concerned with reducing real annotation cost, we are not aware of any data sets with real cost information. We contacted the organizers for at least five benchmark efforts involving human annotators to try to obtain cost data for their respective data sets. While some could provide rough estimates about the average annotation time per instance, none of them logged actual annotation times (or any other form of cost) for individual instances. Therefore, we conducted several annotation experiments of our own in which these costs are recorded.

CKB News Corpus. The present work was partially motivated by a collaborative project called Community Knowledge Base (CKB). The goal of this project is to build a software system for local newsrooms that can monitor local and regional news feeds, automatically extract information, and maintain a database of key players in the local community. The system provides access to a structured model of the community's social network for journalists researching news stories.

For an initial version of the CKB system, we focused on learning to extract four entities (actor, role, organization, location) and six binary relations among them (actor-role, actor-organization, organization-location, etc.). We began by training a named entity recognition (NER) system using a *conditional random field* (CRF) [10] on the CoNLL-2003 corpus [15], augmented with a small set of articles annotated for the additional role entity (which is not part of that corpus). For this version of the system, we collaborated with the Reynolds Journalism Institute at the University of Missouri. Textual sources consisted of articles published over a year in the *Columbia Missourian*, a working daily newspaper published by the school. After filtering documents for text encoding errors and outliers in length, the final pool consisted of 1,984 articles. The NER model described above was used to automatically pre-annotate this pool of articles.

Articles were then labeled by five University of Missouri journalism students. Figure 1 shows the interactive web-based annotation system used for (i) adding entity annotations or editing automatic pre-annotations and (ii) adding relation annotations. Documents were selected from the pool in a random order, and each was presented to one annotator, thus no two people labeled the same article. The annotation system logged both the time elapsed during the labeling process, and the number of labeling "actions" taken for each article (label an entity, clear an incorrect entity, mark a putative relation, etc.). The resulting corpus consists of 358 labeled articles.

SIVAL Image Repository. In previous work [17], we presented a framework for active learning in *multiple-instance* (MI) problem domains [5]. In an MI learning task, instances are naturally organized into *bags*, and it is the bags, rather than individual instances, that are labeled for training. MI learners assume that every instance in a bag labeled negative is actually negative, whereas only one instance in a bag labeled positive needs to be positive.

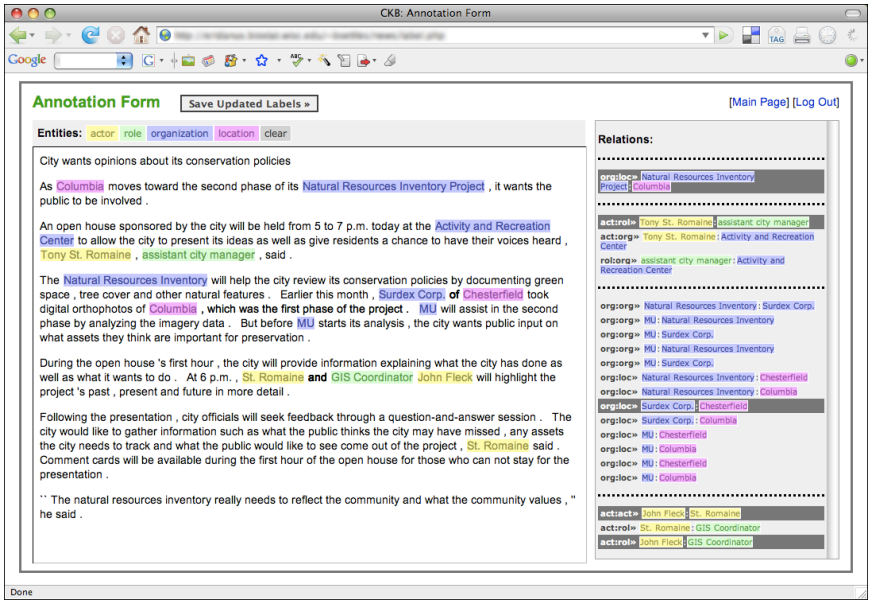

Figure 1: A screenshot of the CKB labeling interface. Article text appears in the window on the left, where annotators can highlight entities and label them with a word-processor style formatting menu. As the entities are labeled, candidate relations among them are dynamically generated in the window to the right, grouped by paragraph. Users then click on these relations to indicate which ones are true (in dark grey).

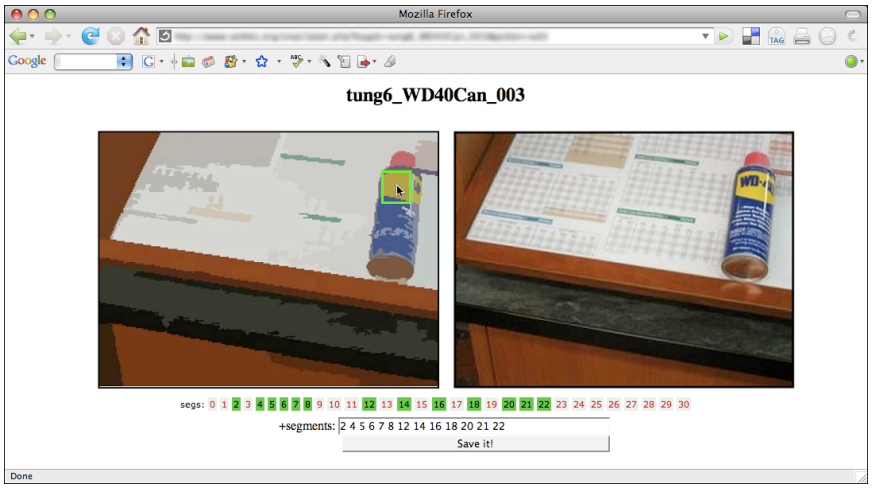

Figure 2: A screenshot of the SIVAL labeling interface. Annotators move the cursor over segments in the processed image on the left, clicking on those belonging to the target object. In this example, the highlighted segment belongs to the product label of a WD40 can. The original reference image is shown on the right.

One application for the MI setting is *content-based image retrieval* (CBIR). In this task, images are represented as bags and instances correspond to processed, segmented regions of the image. A bag representing a given image is positive if the image contains some object of interest. The MI paradigm is appropriate here because only a few regions of an image may be part of a particular object, such as the WD40 can shown in Figure 2. An advantage of the MI representation here is that it is significantly easier to label an entire image than it is to label each segment. However, we have demonstrated that an MI learning algorithm can improve significantly if it is allowed to actively query for the labels of specific instances inside a bag. The CBIR task is applicable here because it is possible (though expensive) to acquire these instance-level labels.

Since no MI data sets with instance-level annotations existed, we augmented the SIVAL repository [14] by manually adding instance labels. The data set consists of 1500 images labeled for 25 objects (60 images each) photographed in a variety of positions, orientations, locations, and lighting conditions. Each image (bag) consists of about 30 segments (instances). Figure 2 shows the web-based interface used in the annotation process. Images were labeled by three members of our research group in an arbitrary order, and annotation times for each image were logged. As with the CKB corpus, images were not redundantly labeled by multiple annotators.

Speculative Text Corpus. There has been a growing interest recently in handling subjectivity in natural language tasks. Following work by Light et al. [12], we annotated a corpus of biomedical abstracts for statements that use language that is speculative vs. definite in nature. For our corpus, we selected 100 PubMed[1] abstracts from the GENIA corpus [8] for annotation. Since previous work indicates that many speculative statements appear toward the end of abstracts, we excluded all abstracts that were truncated for length by PubMed. All 850 resulting sentences were labeled by three members of our research group using a simple web-based interface, and annotation times for each sentence were logged. Unlike the previous two data sets, all sentences were redundantly labeled by all three annotators in order to gather data on inter-annotator agreement. Although the ordering was randomized, all annotators saw the sentences in the same sequence.

SigIE Email Corpus. We also created a corpus for the task of extracting contact details from email signature lines [16]. For this, we selected 250 signatures from the Sig+Reply corpus [3] and manually added annotations for twelve address book fields (e.g., name, phone, jobtitle). All annotations were done by the first author, in a random order, using a modified version of the CKB labeling interface. As with CKB, both annotation times and actions were logged.

3 Analysis and Experiments

In this section, we consider six questions that are aimed at understanding how real annotation costs can be learned and exploited by active learning systems.

Are annotation times variable for a given task or domain? If our goal in active learning is to reduce the total time required to train an accurate model, then this first question is critical. If times are approximately constant, our goal can be achieved by simply minimizing the number of instances required, as most work in active learning has done. If these times vary significantly, however, then this variation should be taken into account by the learner.

The answer to this question, however, is complicated. Figure 3 shows histograms characterizing the distribution of annotation times for each domain. For the CKB corpus, the majority of articles took from 56 seconds to just over 16 minutes ($\approx$1000 seconds), but ranged up to 1.73 hours (6275 seconds). SIVAL appears to have two peaks in its distribution, possibly because some objects such as apple are simple (with fewer segments to be labeled), while others like wd40can are more complex (composed of many segments, requiring more time). Most images required less than a minute, but some took as long as 3.4 minutes (204 seconds). The Spec corpus went very quickly, with only 7.6 seconds on average and no sentence taking longer than a minute. The distribution of SigIE is similar to SIVAL, but with a single mode and fewer apparent outliers. For all data sets, the standard deviation is more than half the mean (in the case of CKB, even greater), which demonstrates a fairly high degree of variability. But where does this variance come from? Is it dependent on the annotator, the nature of the task, or is it simply due to random noise? The rest of this section is aimed at better understanding this variance and, more importantly, how it can be utilized by the active learner.

Do times vary from one annotator to the next? Figure 4 provides a more detailed look at the annotation time distributions for each annotator (each identified by a unique ID, e.g., CKB1 and CKB2 are annotators for the CKB corpus). At a glance, we can see that some annotators look quite different from their neighbors. Some are generally faster or slower, some have a larger spread in their annotation times, and some appear more prone to outliers.

We conduct two-sided Kolmogorov-Smirnov (KS) significance tests to see if these apparent differences are real. For the CKB corpus, four distribution pairs result in statistically significant differences at the 95% level: CKB1-2, 1-3, 2-4, and 2-5. (After Bonferroni correction, however, only

[1] http://www.ncbi.nlm.nih.gov/pubmed/

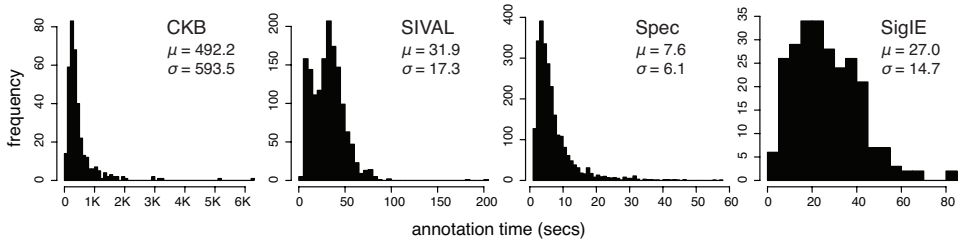

Figure 3: Histograms illustrating the distribution of annotation times for each data set. Mean annotation times (μ) and standard deviations (σ) are also reported.

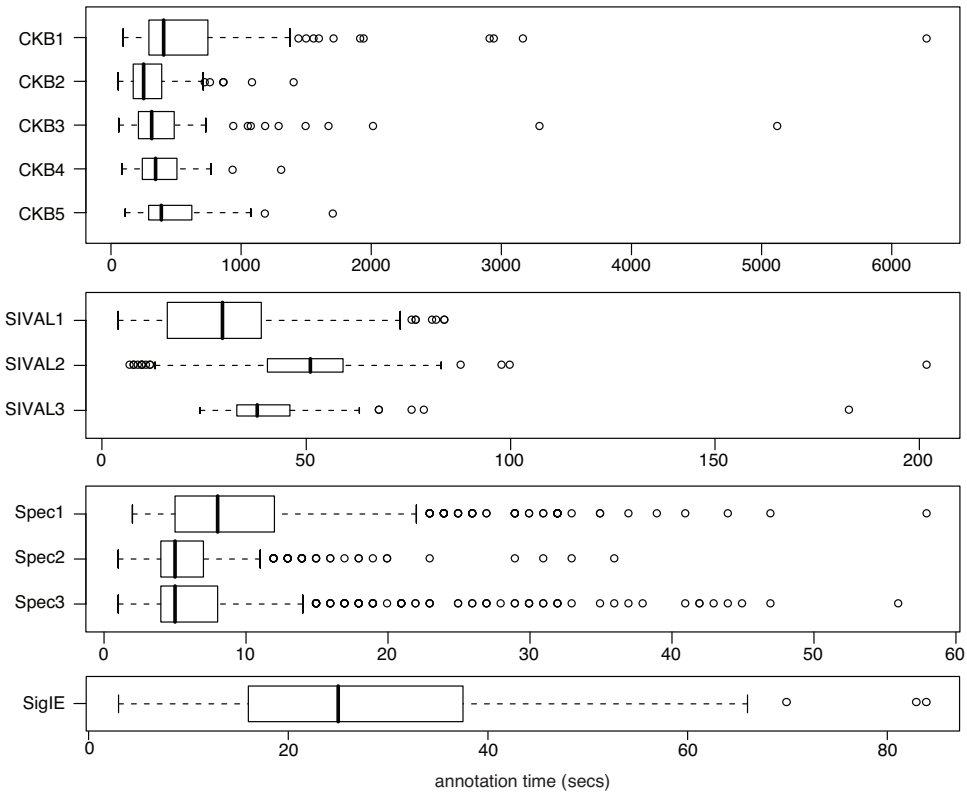

Figure 4: Box plots showing per-annotator labeling time distributions for each data set. A box represents the middle 50% of annotation times, and the median is marked with a thick black line. Box heights are scaled in proportion to the number of instances labeled. Whiskers on either side span the 1st and 4th quartiles of each distribution, up to 1.5 times the inner-quartile range (i.e., box width). Circles indicate possible outliers.

CKB1-2 remains significant.) For the SIVAL and Spec data sets, all differences are significant. We conclude from these results that annotation behavior can vary substantially from one annotator to the next. We argue that, if we wish to leverage annotation cost information into the active learning process, and annotators exhibit these unique trends, then perhaps annotation cost should be modeled on a per-annotator basis (we will return to this idea later on).

Are annotation times stationary? It is possible that annotator behavior can change over time. If this is the case, any modeling of annotation time should be able to account for this variation. Figure 5 plots each annotator's average labeling time per instance as a function of the number of instances labeled thus far (all instances are considered in the order they were actually labeled). As the figure shows, most annotators are able to work somewhat faster as they progress, although the most significant gains seem to be during the first few annotations. Presumably, this is because they are unfamiliar with both the task and the annotation interface early on, but are able to adapt quickly.

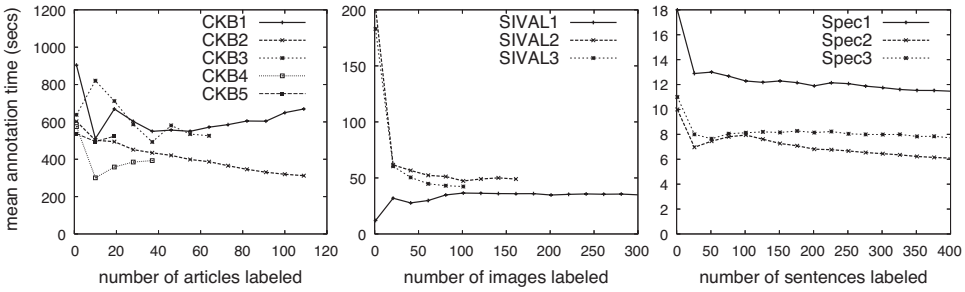

Figure 5: Average annotation time per instance versus the number of instances labeled. (SigIE is omitted for space, but exhibits a similar pattern.)

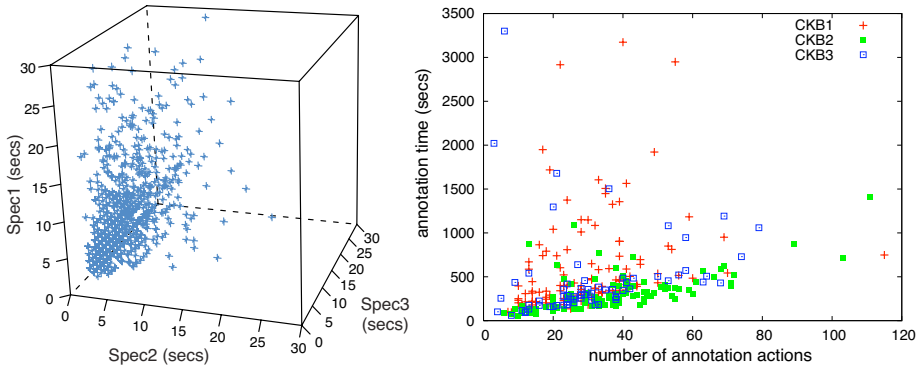

Figure 6: (a) 3D scatterplot of the Spec corpus. Each point represents a sentence, plotted in a 3D space whose axes correspond to labeling times for each annotator. (b) Annotation time vs. actions in the CKB corpus. Each point is an article, and the shape of each point indicates the annotator (only three shown, for clarity).

The notable exception of SIVAL1 slowing down is because the annotator's first few images depicted simple objects that generally took less time than more typical images. CKB1 and CKB4 decelerate slightly as well, although they remain much faster on average than at the beginning. These data indicate that, while most annotators demonstrate a rapid speed-up early during labeling, the "burn-in" period is brief, and annotation times are relatively stationary thereafter.

How stochastic are annotation times? There are two kinds of noise we might encounter when measuring annotation time. The first, which we call *jitter*, is the cumulative effect of small human and/or machine delays, such as momentary fatigue or computer latency. If the same instance were labeled multiple times under similar conditions, we would expect to see minor differences in annotation time due jitter. The second type of noise, which we call *pause*, arises from unexpected interruptions such as a phone call or taking a lunch break. Labeling times subject to pause should be faster under normal circumstances.

We consider two analyses to try to determine the extent to which jitter and pause factor into two of our data sets. First, since each instance in the Spec corpus was labeled by all three annotators, we can assess how well their labeling times correlate with one another. We might think of these redundant labelings as a surrogate for instances being labeled multiple times under similar conditions. Figure 6(a) shows a scatterplot of annotation times in this corpus for the three annotators. Although there is a positive correlation among annotators, the correlation is not strong (pairwise correlation coefficients are between 0.258 and 0.328). This result suggests that jitter has a fairly large effect on labeling times for this data set. This is probably because the labeling times are short in this domain, leaving more room for such stochastic effects.

Our second analysis considers the relationship between annotation time and the number of "actions" for each query in the CKB corpus, as shown in Figure 6(b). If we assume that the time required

to annotate an article is proportional the the number of actions taken, we would expect a strong correlation between them. Indeed, there is a fairly linear relationship. However, a few articles took much longer than the number of actions would imply, suggesting that the annotator was somehow distracted for an extended period. We argue that these large departures from the expected annotation time are indicative of pause. Note that some annotators seem more prone to pause than others. For example, CKB1 and CKB3 annotated all of the extreme points in the above figure (as well as the extreme outliers for CKB in Figure 4). The correlation coefficient between actions and time for both of these annotators is only 0.108, whereas the other annotators are all between 0.624 and 0.744.

In practice, there is little that can be done about jitter, and we conjecture that its effect on annotation time is minimal anyway. However, if we wish to reason about annotation time and utilize this information in active learning, our methods should be able to detect and remain robust to pause.

Can annotation times be accurately predicted? To reduce the total annotation cost in active learning, we argue that query costs should be taken into account by the learner. Unlike the forms of cost previously considered by others [7, 9], knowledge about the labeling time for each instance in our domains is not available to the learner before querying. Therefore, we consider whether or not these unknown annotation times can be accurately predicted.

We approach this problem as a regression learning task, where each query candidate is described by a few simple numerical features. For the CKB corpus, we use seven features, such as the number of words, entities, candidate relations, paragraphs, etc. Note that some of these features depend on quantities that are unknown at query time, such as the number of entities or relations in an article. To handle this, we use the current task-model predictions to estimate these quantities (details for task-models, which are trained alongside the cost-model, are given in the next section). For SIVAL, we use five features: the min, max, mean, and standard deviation of the image segment sizes (in pixels), plus the task-model's predicted number of positive segments. For Spec, we use four features: the number of ASCII characters, words, and unique features used by the classification task-model (we use a "bag-of-words" representation, subject to stop-word removal and stemming), plus this task-model's uncertainty (i.e., entropy) about the class label. For SigIE, we use four features: the number of entities, lines, and characters, plus the percentage of characters that are non-alphanumeric. We emphasize that, for all domains, we made little effort to "engineer" these features. Predicting annotation times could valuable if it can be done with few training instances, and with a minimum of human effort. Therefore, we run these experiments using our initial intuitions about easy-to-compute, fairly domain-independent features.

We have experimented with several regression learning algorithms using these representations, and found the SMO algorithm [18] for support vector regression to be the most accurate. To evaluate the accuracy of the cost-model's prediction p against the true annotation time t, we use the correlation coefficient[2] $r = \frac{\sum_i (p_i - \mu_p)(t_i - \mu_t)}{(n-1)\sigma_p \sigma_t}$, where μ_t and σ_t are the mean and standard deviation of t, respectively. We plot learning curves averaged using ten-fold cross-validation (five-fold for CKB). Following previous work on SIVAL [17], experiments in that domain are done on a binary per-class basis and averaged over 20 independent runs. Half the positive images are used for the training pool and the other half are held aside for evaluation.

Results for the annotation-time prediction experiments are shown in Figure 7. These plots show that, in general, annotation times appear to be fairly learnable. We emphasize that where this is the case, they can be learned from only a few instances. One exception is the SIVAL data set, which seems generally unlearnable using our representation (for most image objects we tested, correlation is nearly zero). On the other hand, the CKB2 cost-model reaches a correlation of 0.626, the Spec(combined) model achieves 0.587, and the SigIE model reaches 0.852 (the most accurate by far). To show that these learned cost-models are better than simple linear functions of length (as considered by Kapoor et al. [7]), we report correlation coefficients between annotation time and the document length (in characters) for the following: CKB2 = 0.291, Spec(combined) = 0.572, and SigIE = 0.455. Other simple estimators we tried (e.g., number of words, lines, or sentences) produce similar results. This demonstrates that at least the CKB2 and SigIE cost-models produce significantly more accurate estimates of cost than a simple heuristic approach.

[2]We have also evaluated these cost-models using relative absolute error, but omit these results here due to space and because they are consistent with the correlation coefficient results.

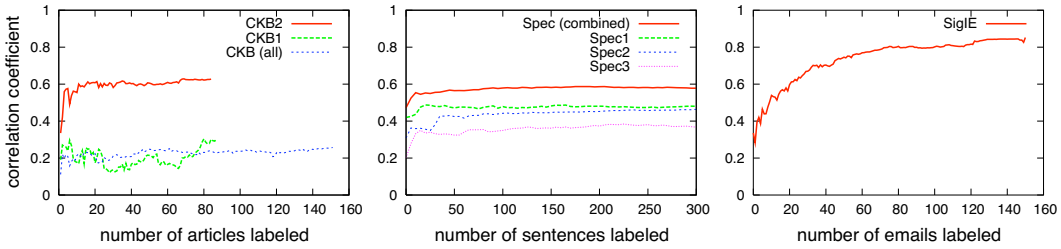

Figure 7: Learning curves for predicting annotation time in terms of correlation coefficient. CKB plots show three curves: annotator-specific models for CKB1 and CKB2, plus all annotators pooled together. Spec plots include a model that predicts the combined time for all annotators, compared to each annotator-specific model.

We also see that the cost-model predicting CKB2-specific labeling times is more accurate than the one predicting times for CKB1, or for all annotations pooled together. We know from Figure 4 that labeling behavior varies greatly from one annotator to the next. We conclude that these differences can have an acute impact on how learnable labeling behavior is. We noted earlier that CKB1 was prone to a type of noise called pause, while CKB2 is not. This probably widens the gap in accuracy between their two models: since the CKB1-specific learner does not detect and handle pause, it may be prone to learn this noise. We see some variation in learnability among Spec annotators as well, though it is far less profound. Interestingly, the cost-model that aims to predict the combined annotation time of all Spec annotators is the most accurate. We hypothesize that this is because the combined annotation time factors out some of the effects of jitter, the other type of noise.

Can we improve active learning by utilizing cost information? So far, we have presented an extensive analysis of the nature and learnability of annotation time as a labeling cost. We now consider whether or not predicted annotation times can benefit an active learner by reducing the amount of time required to achieve a certain level of accuracy.

A common approach to active learning is *uncertainty sampling* [11], in which the learner queries the instances whose labelings are least certain. In this study, we use probabilistic models for which uncertainty can be estimated using the entropy over the label posteriors. Let the uncertainty ϕ of an instance x be: $\phi(x) = -\sum_i P(y_i|x;\theta) \log P(y_i|x;\theta)$, where each y_i is a possible labeling of x according to the task-model θ. The best query, then, is the instance with the greatest value of $\phi(x)$.

For the CKB corpus, we use a CRF trained with a typical NER feature set [15] to extract entities. The CRF parses one sentence at a time, and we dynamically generate a set of candidate relations based on the predicted entities. These relations are described with contextual features (e.g., entity labels and the bag-of-words between them), and then classified with a *maximum entropy* model [2]. We treat relation extraction as a seven-label classification task: the six legal relations plus no-relation. Unlike the typical active learning setting, a query in the CKB domain is an *article*, which in fact is a *set* of instances: an entity sequence (plus several candidate relations) per sentence. For simplicity, we treat each instance in article $\mathcal{X}$ as independent, thus the article uncertainty is given by $\phi(\mathcal{X}) = \sum_{x \in \mathcal{X}} \phi(x)$, the sum of all its instance entropies. Following previous work on active learning for sequence labeling [16], we approximate $\phi(\mathbf{x})$ for the input sequence $\mathbf{x}$ using only the N-best parses, rather than enumerating all possibly label sequences $\mathbf{y}$. We also generate candidate relations at query time based on the most likely parse of each entity sequence. For Spec, we train a maximum entropy model using bag-of-words features subject to stop-word filtering and stemming. Since a query in this domain corresponds to a single instance x, estimating $\phi(x)$ is straightforward. For SigIE, we use the same feature set as in CKB (minus syntactic features). A query for this task is a single sequence $\mathbf{x}$, thus estimating $\phi(\mathbf{x})$ is straightforward with the N-best approximation.

We compare standard entropy-based query selection with a simple "bang for your buck" cost-sensitive approach, where entropy is divided by the predicted labeling time for an instance. We employ our cost-models from before, which are trained alongside these task-models (using only the labeled instances) to predict the labeling time. We also compare against two baselines: the cost-sensitive method using *known* annotation times, and random sampling. The task-models are evaluated with the F_1 measure and averaged using ten-fold cross-validation (five-fold for CKB).

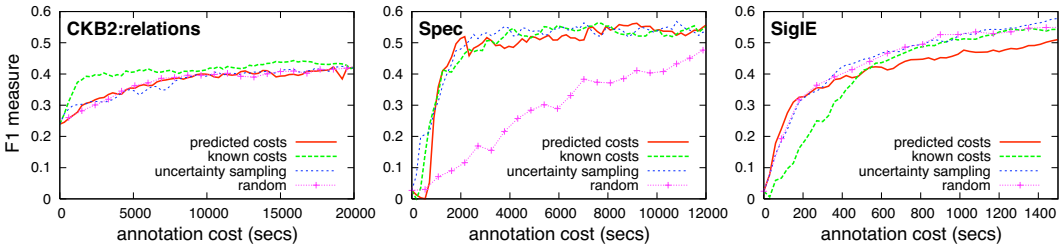

Figure 8: Active learning curves. Cost-sensitive variants using predicted and known annotation costs are compared to standard entropy-based uncertainty sampling and a random sampling baseline. Note that the horizontal axis represents actual annotation cost (in seconds).

Results for these active learning experiments are shown in Figure 8. For the CKB and SigIE corpora, the standard entropy-based strategy does not reduce the time required to achieve the same accuracy as random sampling. However, it is important to note that when these curves are instead plotted as a function the number of queries (not shown here), entropy does produce better learning curves. This indicates that naïve uncertainty sampling is prone to select informative, but time-consuming queries for these problem domains, resulting in no net reduction in cost.

While our cost-sensitive approach using predicted annotation times does not outperform the random baseline for CKB, we do see significant gains when the annotation time is known. This indicates that better learning curves can be achieved if labeling time can be predicted accurately and utilized appropriately. We note that, while we only report the CKB2 relation subtask here, results for CKB1 and both their entity subtasks are nearly identical. For the Spec corpus, standard entropy-based active learning does produce better curves, and in this case our cost-sensitive variants are roughly equivalent. We surmise that this is because annotation times are indeed approximately constant (with observed variations being due to jitter), thus cost information is of little value. Curves for the SigIE task are interesting because cost-sensitive active learning with known costs actually performs worse than random. We suspect that the greedy approach we use here may not properly utilize cost information for this task. Another explanation is that shorter instances may actually contain less valuable information. For example, a brief email signature might only contain name and email fields and is therefore quick to annotate, but lacks important rare fields such as jobtitle or phone. This result underscores the importance of understanding the relationship between the annotation cost of an instance and its overall value to the learner.

4 Conclusions and Future Work

To date, most work in active learning has assumed that the cost of acquiring a label is the same for all instances. Some recent work has considered cases where labeling costs are variable, but these have either assumed that the cost is known for each instance [9, 13] or can be approximated by a simple estimator [7]. In this paper, we have presented an extensive empirical study of annotation costs in four real-world text and image domains. To our knowledge, this is the first empirical investigation of annotation costs in a real setting. Our analysis provides several conclusions that have implications for active learning in domains where labeling is done by human annotators:

- In most of the problem domains we consider, annotation costs are not (approximately) constant across instances, and can instead vary considerably.
- Consequently, active learning approaches which ignore cost information may perform no better than random instance labeling. However, improved learning curves are achievable if an active learner can take these variable costs into account appropriately.
- In some domains, the cost for annotating an instance may not be intrinsic, but instead vary according to the person doing the annotation.
- In some domains, the measured cost for an annotation may include a stochastic component. The effects of this seem to depend, in part, on the typical time required to label an instance, and the proficiency of the annotators.
- In some domains, we can accurately learn to predict annotation costs, even after seeing only a few training examples.

This last result suggests that, even when annotation costs are not known before querying, an active learner may be able to profitably reason about them. We propose exploiting this by training a cost-model to predict annotation costs while simultaneously training the actual task-model. However, the simple "bang for your buck" cost-sensitive approach considered here does not appear to capture the necessary aspects of the problem. A main focus in future work will be to investigate cost-sensitive active learning strategies that are more robust when given approximate, predicted annotation costs.

Acknowledgments

We would like to thank Tom Warhover, Reuben Stern, and Brian Hamman of the *Columbia Missourian* for help coordinating annotation efforts for the CKB corpus. Thanks also to Soumya Ray and David Andrzejewski for assistance in labeling the SIVAL and Spec data sets, respectively. This work is supported by NIH grants T15-LM07359 and R01-LM07050, and the University of Wisconsin School of Journalism & Mass Communication.

References

[1] J. Baldridge and M. Osborne. Active learning and the total cost of annotation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9–16. ACL Press, 2004.

[2] A.L. Berger, V.J. Della Pietra, and S.A. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[3] V.R. Carvalho and W. Cohen. Learning to extract signature and reply lines from email. In *Proceedings of the Conference on Email and Anti-Spam (CEAS)*, 2004.

[4] A. Culotta and A. McCallum. Reducing labeling effort for stuctured prediction tasks. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 746–751. AAAI Press, 2005.

[5] T. Dietterich, R. Lathrop, and T. Lozano-Perez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 1997.

[6] R. Greiner, A. Grove, and D. Roth. Learning cost-sensitive active classifiers. *Artificial Intelligence*, 139:137–174, 2002.

[7] A. Kapoor, E. Horvitz, and S. Basu. Selective supervision: Guiding supervised learning with decision-theoretic active learning,. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 877–882, 2007.

[8] J. Kim, T. Ohta, Y. Teteisi, and J. Tsujii. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl. 1):i180–i182, 2003.

[9] R.D. King, K.E. Whelan, F.M. Jones, P.G. Reiser, C.H. Bryant, S.H. Muggleton, D.B. Kell, and S.G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–52, 2004.

[10] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 282–289. Morgan Kaufmann, 2001.

[11] D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 148–156. Morgan Kaufmann, 1994.

[12] M. Light, X.Y. Qiu, and P. Srinivasan. The language of bioscience: Facts, speculations, and statements in between. In *Proceedings of the ISMB BioLINK*, pages 17–24. ACM Press, 2004.

[13] D. Margineantu. Active cost-sensitive learning. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1622–1623, 2005.

[14] R. Rahmani and S.A. Goldman. MISSL: Multiple-instance semi-supervised learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 705–712. ACM Press, 2006.

[15] E.F.T.K. Sang and F. DeMeulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Conference on Natural Language Learning (CoNLL)*, pages 142–147, 2003.

[16] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1069–1078. ACL Press, 2008.

[17] B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, pages 1289–1296. MIT Press, 2008.

[18] A.J. Smola and B. Schölkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, NueroCOLT2 Technical Report Series, 1998.