MTR120603

MITRE TECHNICAL REPORT

MITRE

Active Learning with a Human In The Loop

Seamus Clancy
Sam Bayer
Robyn Kozierok

November 2012

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Active Learning with a Human in The Loop	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
MITRE Corporation,202 Burlington Road,Bedford,MA,01730-1420	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Text annotation is an expensive pre-requisite for applying data-driven natural language processing techniques to new datasets. Tools that can reliably reduce the time and money required to construct an annotated corpus would be of immediate value to MITRE?s sponsors. To this end, we have explored the possibility of using active learning strategies to aid human annotators in performing a basic named entity annotation task. Our experiments consider example-based active learning algorithms that are widely believed to reduce the number of examples and therefore reduce cost but instead show that once the true costs of human annotation is taken into consideration the savings from using active learning vanishes. Our experiments with human annotators confirm that human annotation times vary greatly and are di cult to predict, a fact that has received relatively little attention in the academic literature on active learning for natural language processing. While our study was far from exhaustive, we found that the literature supporting active learning typically focuses on reducing the number of examples to be annotated while ignoring the costs of manual annotation. To date there is no published work suggesting that active learning actually reduces annotation time or cost for the sequence labeling annotation task we consider. For these reasons combined with the non-trivial costs and constraints imposed by active learning, we have decided to exclude active learning support from our annotation tool suite, and we are unable to recommend active learning in the form we detail in this technical report to our sponsors as a strategy for reducing costs for natural language annotation tasks.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**33**	

MTR120603

MITRE TECHNICAL REPORT

MITRE

Active Learning with a Human In The Loop

Seamus Clancy
Sam Bayer
Robyn Kozierok

November 2012

Sponsor: MITRE Innovation Program
Dept. No.: G063
Contract No.: W15P7T-12-C-F600
Project No.: 0712M750-AA

Abstract

Text annotation is an expensive pre-requisite for applying data-driven natural language processing techniques to new datasets. Tools that can reliably reduce the time and money required to construct an annotated corpus would be of immediate value to MITRE's sponsors. To this end, we have explored the possibility of using active learning strategies to aid human annotators in performing a basic named entity annotation task. Our experiments consider example-based active learning algorithms that are widely believed to reduce the number of examples and therefore reduce cost, but instead show that once the true costs of human annotation is taken into consideration the savings from using active learning vanishes. Our experiments with human annotators confirm that human annotation times vary greatly and are difficult to predict, a fact that has received relatively little attention in the academic literature on active learning for natural language processing. While our study was far from exhaustive, we found that the literature supporting active learning typically focuses on reducing the number of examples to be annotated while ignoring the costs of manual annotation. To date there is no published work suggesting that active learning actually reduces annotation time or cost for the sequence labeling annotation task we consider. For these reasons, combined with the non-trivial costs and constraints imposed by active learning, we have decided to exclude active learning support from our annotation tool suite, and we are unable to recommend active learning in the form we detail in this technical report to our sponsors as a strategy for reducing costs for natural language annotation tasks.

Keywords: Active Learning, Machine Learning, Annotation, Natural Language Processing

This page intentionally left blank.

Table of Contents

This page intentionally left blank.

1 Introduction

Constructing a corpus of annotated examples is an important and expensive prerequisite for building customized information extraction tools. Annotated examples are used to measure the performance of taggers, and in the case of systems trained with examples, annotated examples are used to weight the decisions that taggers make. In the case of trained taggers, and of statistically trained taggers in particular, the larger the corpus of annotated text the better the resulting tagger tends to be. For many text extraction problems, though, attaining human levels of accuracy with a statistically trained tagger requires corpora that consist of hundreds of thousands of labeled words. The task of annotating hundreds of thousands of words is a significant one, and in many cases is a sizable fraction of the total cost of developing a text extraction solution.

The difficulties of annotating text are well-understood in the natural language processing community, and there has been no shortage of creative solutions to obtain text annotations without paying the full costs of annotation. One popular approach is to make the annotator's job easier by pretagging the text to be annotated and asking the annotator to correct the output rather than annotate from scratch. Although pretagging text carries the risk of biasing inexperienced annotators towards the tagger's output, and of wasting the annotator's time if correcting the output is more laborious than annotating, this approach is widely used for information extraction problems such as named entity recognition. In an early description of this work, Day et al. [1997] combines pre-tagging with frequent model re-training and calls this process "mixed initiative annotation" or "tag-a-little-learn-a-little." The authors also provide experimental evidence that the tag-a-little-learn-a-little procedure can reduce the effort required to produce an annotated corpus. Tag-a-little-learn-a-little annotation can also complement a different strategy known as active learning.

Active learning is motivated by the observation that a great deal of natural language data in a corpus of documents is repetitive and probably does not need to be annotated. Therefore, instead of annotating in a random order, active learning algorithms select specific examples that will make *the largest improvements* to the statistical model. In several real-world cases active learning can be shown to significantly reduce the number of labeled examples required to achieve a given level of performance. An important risk, however, is that reducing the number of examples does not necessarily reduce the total cost of annotation. In some settings it is possible for active learning heuristics to select complex examples that are significantly harder and more time-consuming to annotate than randomly chosen examples.

With good reason, active learning has a long history in machine learning. Much of the interest by practitioners of machine learning stems from the widely-held belief that if one could determine the minimal set of required labeled examples, and then acquire labels for only those examples, it would greatly reduce the cost of building a statistical model. Hypothetically, labeling fewer examples would take less time, and the annotator would make fewer total mistakes. Indeed, reducing the number of examples required, usually guided by a least confidence heuristic, is common in the literature. For basic examples of this effect see Duda et al. [2000] or Shen et al. [2004]. When evaluated empirically, active learning strategies for example selection are commonly compared to a baseline strategy in which examples are chosen at random. Figure 1 shows a common example application where a support vector machine classifier is trained on the MNIST dataset of handwritten digits (LeCun et al. [1998]). In the red curve the model is built iteratively: at each iteration the twenty examples with the lowest confidence according the the previous iteration's model are

added to the training set. In the blue curve twenty examples are chosen at random and added to the training set. In both cases the classifier's accuracy is measured on a dataset reserved for evaluation. A comparison of the red and blue training curves indicates an overall diminishing rate of growth in both of the curves, but also demonstrates that the least confidence heuristic achieves a faster growth rate. Examination of the data shows that to achieve an accuracy of 96% on this evaluation set, 2020 labeled examples are needed according to the active learning heuristic, whereas 4020 labeled examples are need if the random selection method is used.

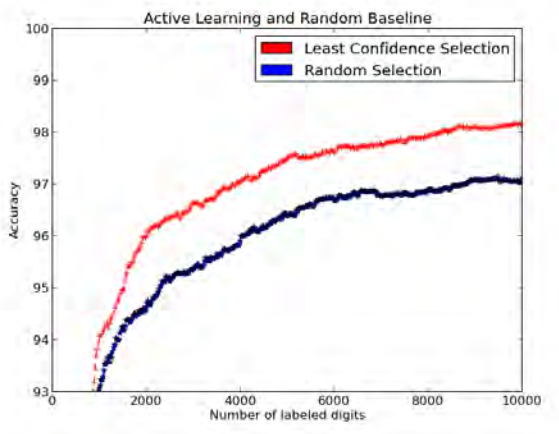


Figure 1: A typical example of active learning outperforming a random baseline. Experiments were performed with MNIST digits data and a libsvm classifier.

With the possibilities of reducing the amount of text to label by nearly half, there have been many attempts to apply active learning approaches to the statistical models commonly used in information extraction systems. Translating the successes of active learning on independent identically distributed classification problems like the MNIST example to real-world information extraction tasks means considering additional facets of the text annotation problem. While information extraction can encompass many different text processing tasks, for simplicity we will restrict the discussion to generic sequence labeling problems, and consider only linear chain conditional random field (CRF) models discussed in Lafferty et al. [2001]. Even starting with a probabilistic model like the CRF there are several choices of selection heuristic. There is also the question of the annotation granularity: should whole documents, sentences, paragraphs or tokens be treated as the atoms for annotation? Furthermore there is the overhead cost of continually retraining and re-tagging the data; unless tools are carefully designed the annotator may be continually interrupted while the active learning processing completes. Additionally pretagging the data is known to improve efficiency, and may account for much of the savings reported for tools using an active learning in conjunction with pretagging. Finally the real cost of human annotation, typically time, must be taken into account instead of relying on the number of labeled examples as an estimate of real annotation cost.

Under MITRE's TooCAAn project, the authors have been extending the capabilities of the MITRE Annotation Toolkit (MAT) to encompass a range of advanced annotator support facilities. Given its potential to reduce the cost of human annotation, active learning is an obvious candidate for inclusion in this toolkit. This technical report describes our detailed consideration of this possibility, and documents our experiments in applying active learning to a real-world sequence labeling

annotation problem. The discussion will cover the effect of various hyper-parameters on the active learning schedule and our attempts to measure and model the behavior of humans annotators. We will also review relevant natural language processing literature and the existing published experiments on human-in-the-loop annotation. We will conclude with a discussion of why we have decided against incorporating active learning in the TooCAAn toolkit.

2 Simulating Active Learning

Linear chain conditional random field models are a popular approach for sequence labeling problems in natural language processing. We are therefore interested in exploiting CRF models to rank examples for active learning. To review, briefly, the first-order CRF models we consider are fully supervised learners whose training data input is a sequence of observation and label pairs, and whose model assumes that the label at position t depends on the label at position $t-1$, the label at position $t+1$, and the observation at position t. In the language of graphical models, the CRF is an undirected graph consisting of sequence of label vertices in which each vertex is connected to its neighbors and to an observation vertex. In the named entity recognition tasks we consider, observations correspond to text tokens which are represented as a vector of features and labels following the widely-used B-I-O convention described in Ramshaw and Marcus [1995]. In this encoding convention each token is labeled either as "O" for "outside" if it is not part of any span, "B-type" for "begin" if is the first token in a span of a given type, or "I-type" for "inside" if it belongs to a span but is not the first token. In named entity recognition the types of spans are typically "Person", "Location", and "Organization."

For active learning purposes, a convenient property of the linear chain CRF is that it is a probabilistic model for which it is possible to perform efficient inference that will find the most likely label sequence. The model defines a posterior distribution of a labeling, $\mathbf{Y}$, given the data, $\mathbf{X}$, and the model parameters λ as

$$P(\mathbf{Y}|\mathbf{X},\lambda) = \frac{e^{\lambda\cdot\mathbf{F}(\mathbf{Y},\mathbf{X})}}{\mathbf{Z}(\mathbf{X})}$$

where $\mathbf{F}$ denotes a set of feature functions, and $\mathbf{Z}(\mathbf{X})$ is a partition function to normalize over all possible labelings: $\mathbf{Z}(\mathbf{X}) = \sum_{\mathbf{Y}'} e^{\lambda\cdot\mathbf{F}(\mathbf{Y}',\mathbf{X})}$. Since the partition function does not depend on the labels, $\mathbf{Y}$, the Viterbi algorithm can find the most likely labeling without computing it:

$$\arg\max_{\mathbf{Y}} P(\mathrm{Y}|\mathbf{X},\lambda) = \arg\max_{\mathbf{Y}} \frac{e^{\lambda\cdot\mathbf{F}(\mathbf{Y},\mathbf{X})}}{\mathbf{Z}(\mathbf{X})} = \arg\max_{\mathbf{Y}} \frac{e^{\lambda\cdot\mathbf{F}(\mathbf{Y},\mathbf{X})}}{\sum_{\mathbf{Y}'} e^{\lambda\cdot\mathbf{F}(\mathbf{Y}',\mathbf{X})}} = \arg\max_{\mathbf{Y}} e^{\lambda\cdot\mathbf{F}(\mathbf{Y},\mathbf{X})}$$

For active learning we desire a normalized probability for the prediction and therefore both the numerator and the denominator must be computed. Fortunately in linear chain CRF models, the forward-backward algorithm can efficiently compute $\mathbf{Z}(\mathbf{X})$ in time linearly proportional to the length of the sequence. The result is the most likely labeling of the sequence and the model's probability that the most probable labeling is correct. The posterior probability can then be used to to rank sentences in a corpus for active learning by treating the sentences with the lowest posterior probability as the most informative examples since according to the model, [1] they have the smallest

[1] Relying on the model's posterior probability is one strategy of many for automatically identifying errors. In practice other methods are more strongly correlated with prediction mistakes, but a discussion of such methods is beyond the scope of this report.

probability of being correctly labeled.

Posterior probability is not the only ranking strategy, however. Several additional ranking strategies have been proposed for active learning with CRF models, and still more for active learning in structured prediction models in general. A detailed exposition and evaluation of several different methods applicable to linear chain CRFs can be found in Settles and Craven [2008]. Additionally, a description of a fast implementation of a query-by-committee approach can be found in Tomanek and Hahn [2009]. The methods presented in the two works mentioned above capture several different types of strategies for identifying the most helpful sentences to annotate. For example, the entropy-based methods examine the entropy of the predicted labels either on a token-by-token basis or as a complete sequence; the Fisher kernel and expected gradient length capture the relationship between weights in the CRF model and the unlabeled data; and query-by-committee approaches measure model uncertainty by the disagreements produced by different models. We found that the majority of these ranking heuristics — the entropy methods, Fisher kernel, expected gradient length, and query-by-committee methods — were too computationally expensive to run on large corpora without investing considerable effort to first create very efficient implementations. Besides minimum posterior probability only the margin heuristic appeared to be computationally feasible. The margin approach selects examples based on the difference in the posterior probabilities between the best and second best labeling.

Our simulation experiments with active learning on natural language problems were similar to those presented in Settles [2008]. We used three established corpora, MUC6 (Grishman and Sundheim [1995]), CoNLL 2003 NER (Tjong Kim Sang and De Meulder [2003]), and the I2B2 de-identification corpus (Uzuner et al. [2007]). Our experiments used MALLET and Java code available from Settles and Craven [2008] and reproduced the basic experiments in that publication. Exactly duplicating the results was not possible because both the lexicons and the exact sentence segmentations used in the CoNLL corpus were not available to us. Despite these differences we were ultimately satisfied that the basic results of the paper were reproduced. Crucially, these experiments only measured the cost of active learning in terms of the number of labeled sentences, and ignored the true cost in terms of annotator time required to label those sentences. These experiments were aimed at exploring the effectiveness of different confidence heuristics in CRFs, but otherwise were very similar to the MNIST experiment shown in Figure 1. For each confidence method investigated the experiment begins with five randomly chosen seed sentences, and trains a CRF model using only those examples. The model is then used to label each sentence in the corpus, and sentences are then ranked according the confidence method being used. The sentences are then sorted by the confidence method and the five with the lowest confidence are added to the training set along with the correct annotations. In the original published experiments by Settles and Craven [2008] the train-retag-retrain loop is only run for thirty iterations with a batch size of five for a total of 150 sentences. We were interested in the performance of the active learning heuristic considerably beyond 150 sentences so our experiments continue for 500 iterations with a batch size of five sentences for a total of 2500 sentences.

The results of this experiment show that except for Token Entropy shown in Figure 3a and Figure 3b, each of the selection heuristics used has a bias towards longer sentences. This bias is visible in the curves that measure annotation in the number of tokens; in those graphs the active learning curves extend farther along the x-axis than the curves produced with the random baseline. Both curves, however, correspond to 2500 sentences selected. While we did not undertake additional

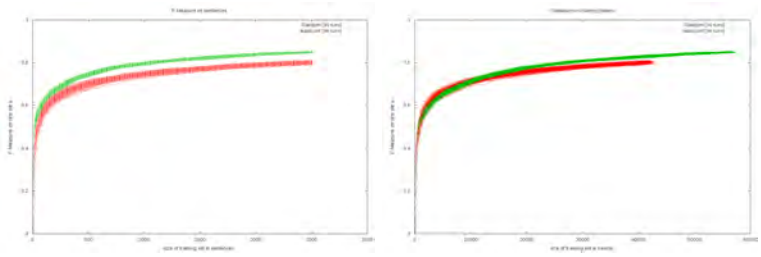
experimentation to explore why this bias occurs; a likely explanation is that sentences that contain multiple named entities and multiple mistakes have very low posterior probabilities, but more tokens than sentences chosen at random.

The graphs in Figures 2a – 3d show the learning curves obtained from repeating the experiments with several active learning heuristics. Each heuristic was tested 100 times by initializing with five random samples and applying the train-tag-sort-retrain loop described above. The randomization helped capture the extent to which random choices in the starting seed could affect the performance of the strategy. Each experiment was also compared with 100 random baselines which have a wide spread in performance. For each heuristic we investigated, we plot the results two different ways: first as f-measure on a held-out test corpus versus the number of sentences in the training set, and the second as f-measure versus the number of tokens in the training set. A curious result is that when measured in terms of sentences the gap between the active learning selection heuristic and the random baseline is consistently large. However when the same experiments are re-plotted and measured in terms of tokens of annotation the results are much less compelling. Additionally we observe that in the graphs where annotation effort is measured in tokens the random and minimum posterior probability curves are not distinguishable at the left-most side of the graph, though the active learning curve eventually overtakes the random baseline after several thousand tokens of labeling.

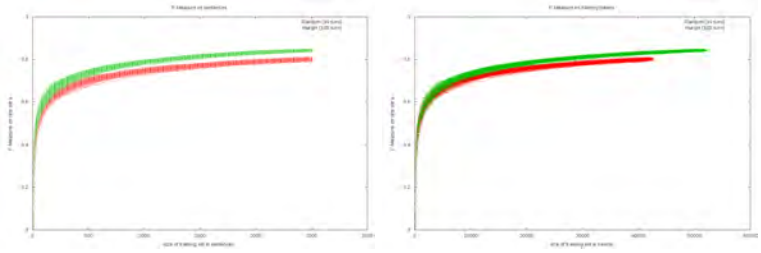
We did experiment with variations in the batch size, and observed that there was no difference in the shape and position of the active learning curves when the batch size was increased from five sentences to twenty sentences. Unless otherwise indicated most of the experiments referenced in this report used batches of size twenty.

We also experimented with the annotation granularity. Many widely-used evaluation datasets for named entity recognition offer annotated documents rather than a collection of annotated sentences or tokens. A question of immediate practical concern was whether the active learning selection strategies described in Settles and Craven [2008] would be efficacious if entire documents were selected rather than sentences. Experiments with the minimum posterior probability heuristic show that documents are not an acceptable substitute for sentences; when measured on a per-token basis there is no evidence that actively querying by documents can outperform a random document baseline. More importantly, when document annotation and sentence annotation are compared on the same axis (see Figure 4) we see that a random ordering of sentences outperforms both document ordering curves. This result suggests that on a per-token basis annotating sentences can be more efficient than annotating whole documents. Such an effect is most likely because a random selection of sentences offers a greater diversity of linguistic phenomena than the sequentially ordered sentences that are encountered in a single document. An important question that this result raises, however, is whether the cost of annotating sentences is greater than the cost of annotating documents.

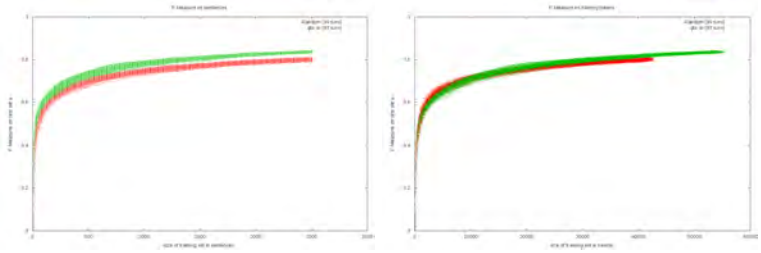
Figure 4 raises the interesting possibility that one can save on the cost of corpus creation simply switching from a document annotation paradigm to a random sentences paradigm. This possibility is appealing because it sidesteps many of the practical concerns of engineering a usable active learning toolkit. Even when the annotator is asked to annotate twenty sentences at a time between re-training and tagging iterations, active learning imposes a significant overhead for the user. In many naive implementations, the user must wait while the system updates the machine learned model with the new annotations and then re-tags and re-ranks the untagged pool of sen-



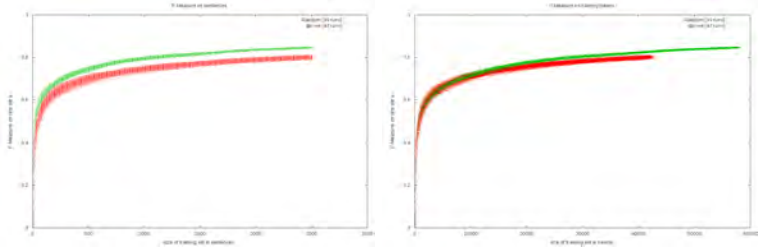
(a) Least Confidence Sent. (b) Least Confidence Tokens



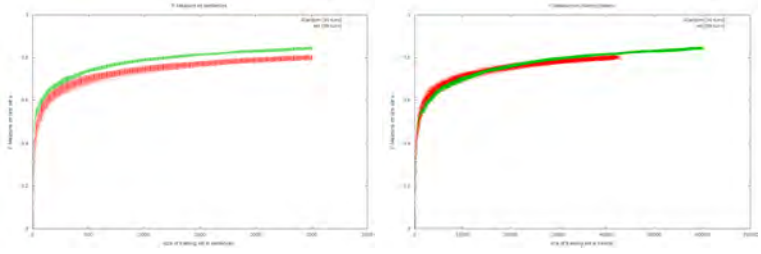
(c) Margin Sentences (d) MarginTokens



(e) Query-By-Committee KL Divergence Sentences (f) Query-By-Committee KL Divergence Tokens

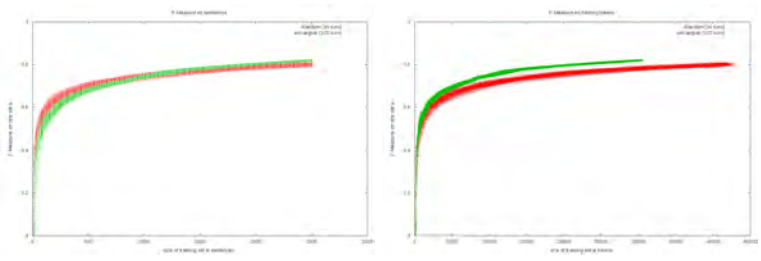


(g) Query-By-Committee Vote Entropy Sentences (h) Query-By-Committee Vote Entropy Tokens



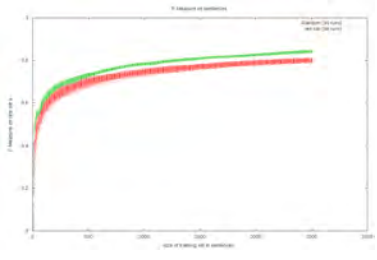
(i) Sequence Entropy Sentences (j) Sequence Entropy Tokens

Figure 2: Simulated active learning experiments with different selection criteria. Red curves correspond to a baseline with random ordering of sentences and green curves correspond to the active learning ordering.

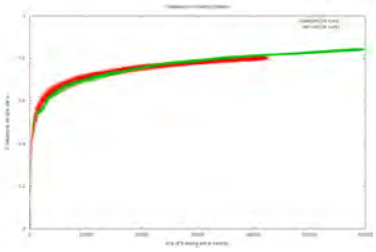


(a) Token Entropy Sentences

(b) Token Entropy Tokens



(c) Total Token Entropy Sentences



(d) Total Token Entropy Tokens

Figure 3: Simulated active learning experiments with additional selection criteria. Red curves correspond to a baseline with random ordering of sentences and green curves correspond to the active learning ordering.

tences. There are notable efforts directed at reducing this overhead cost such as iterative training of conditional random field models and architectures that parallelize the training and tagging across different process, for example Haertel et al. [2010]. Nonetheless a random ordering of sentences that does *not* use active learning is a compellingly simple architecture, and insofar as it can be shown to save on the total costs of annotation it also deserves consideration.

3 Experiments with Human Annotators

To understand the true impact of active learning for text annotation, evaluations must consider a cost function that is well-aligned with real-world costs. Cost itself is not always straightforward; in contractual settings or when using crowd-sourced labor, annotators might be paid by the document or by the sentence. However, in situations where data cannot be released publicly or where annotators are selected based on specialized domain expertise, they are frequently paid by the hour, and the cost we are most interested in is annotator time. Since we are paying annotators by the hour, we begin with pre-tagged documents since we believe that correcting machine output is a more efficient use of annotator time than producing annotations from scratch. The dual goals of our experiments are therefore to understand the characteristics of text that takes annotators' time, and understand if either document-at-a-time or sentence-at-a-time annotation is more efficient on an per-hour basis than the other.

Our experiments leveraged our existing text annotation infrastructure. As part of the TooCAAn project, we had already extended the MITRE Annotation Toolkit (MAT)'s managed document sets (called workspaces) to support document and segment prioritization, and to provide queues of ordered material to review. We had also enhanced our Callisto hand annotation tool (Day et al.

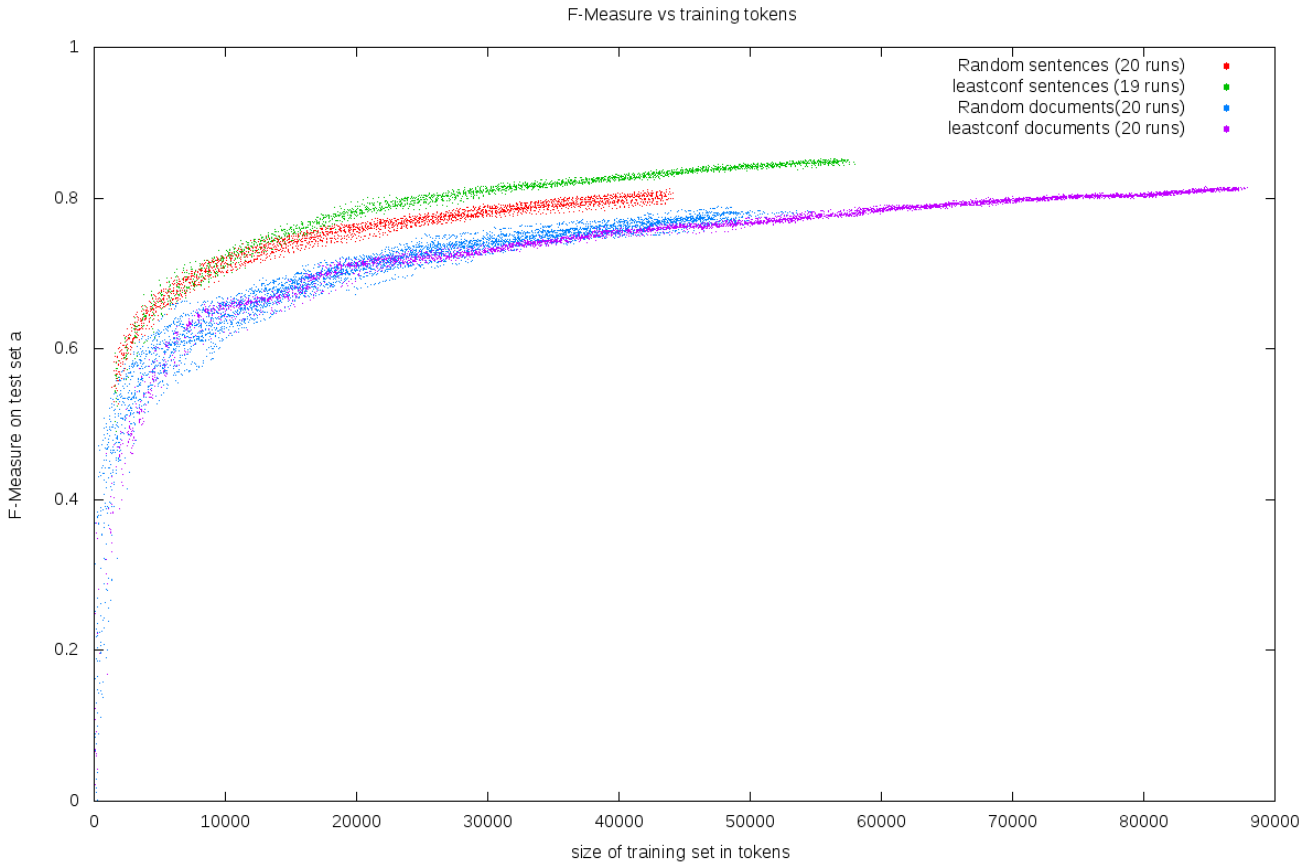
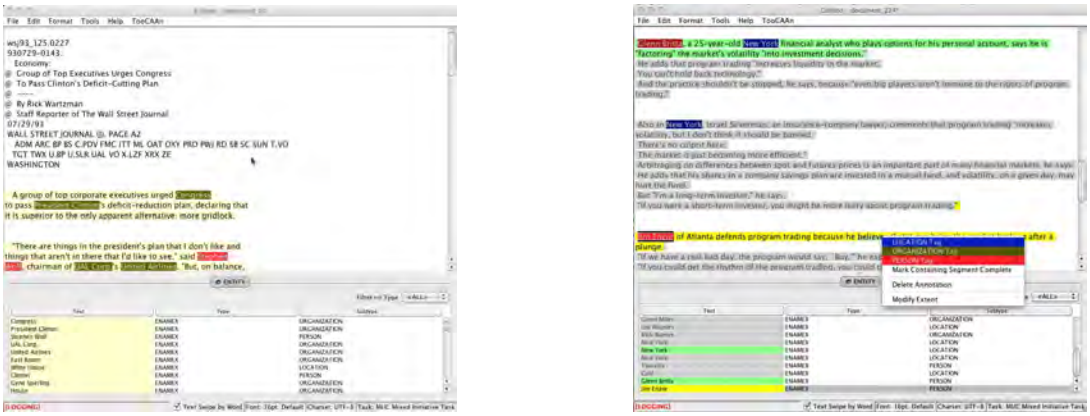


Figure 4: An experiment with example granularity. We see that active learning with document-level selection does not outperform a random baseline with document-level selection, and that both conditions using sentence-level granularity outperform document-level approaches.



(a) (b)

Figure 5: Annotation user interfaces. (a) User interface for document-level annotation. (b) User interface for segment-level annotation.

[2004]) to communicate with the MAT service API, and to be able to navigate through MAT's workspaces.

To examine the effect of segment-by-segment annotation, we added a segment-based annotation mode to Callisto, in which the annotator's activities would be restricted to particular chosen segments of a document. We also enhanced the original document annotation mode with additional feedback about the different document regions.

We illustrate the document mode in Figure 5a. Here, the regions highlighted in pale yellow are the regions of the document which are eligible for annotation, and the regions with a white background are to be ignored. The annotation table in the lower pane echoes the document region highlighting in its "Text" column.

The segment mode is illustrated in Figure 5b. Here, the regions of the document are either white (regions which will never be annotated); gray (potentially annotatable regions which the annotator should ignore in this round); bright yellow (regions the annotator should review); or green (regions that have already been reviewed and corrected). As in document mode, the annotation in the lower pane echoes the document region highlighting. Once the user corrects a yellow segment and marks it complete, the tool automatically advances to the next yellow segment, until all relevant segments have been reviewed, at which point the tool prompts the user to save and move on to the next document in the queue.

We chose the MUC named entity annotation task as the task for this experiment. First, we created a condensed version of the MUC annotation guidelines, following Grishman and Sundheim [1995]. These guidelines were 2 pages long, with numerous examples, enough to provide a reasonable foundation to an annotator unfamiliar with the task. Our condensed guidelines are included as Appendix A.

Next, we created two experiment workspaces, one featuring document-oriented annotation and the other featuring segment-oriented annotation. The text in each of these workspaces was pre-tagged with the output of an imperfect CRF model, and the human subjects were asked to correct the tagging. We randomly selected 15 gold-standard documents from the MUC-6 corpus and used

them to seed each workspace; we chose this number because it established a poor, but not dreadful, initial F-measure of 47% for each workspace which allowed annotators to make noticeable improvements to the corpus in a small amount of time. We established a queue in each workspace. We also prepared two training workspaces with comparable configurations.

We also prepared a set of written instructions, to ensure consistency across trials. We conducted two dry runs with experienced MITRE annotators, and made improvements to the tool and materials as a result.

For the actual experiment, we recruited 8 subjects from among our colleagues at MITRE. These subjects differed widely in the levels of annotation experience; some were experts, and some were neophytes. Each subject performed both document-oriented and segment-oriented annotation; half were selected to begin with the document condition, the rest with the segment condition.

We conducted the experiments using a wide-screen monitor connected to a IBM (X60) laptop. Each trial was conducted one-on-one; the same proctor (Bayer) administered each trial. The proctor guided each subject through a 30-minute training session, during which the proctor read through the written instructions with the subject, and helped the subject work with the Callisto tool in each of the two conditions, using the training workspaces. The subject then had the option of taking a short break. The subject then spent 35 minutes annotating in the first condition, took a break, then spent 35 minutes annotating in the second condition. If the 35 minutes expired while the subject was in the middle of a document, the subject was instructed to finish the document. Finally, the subject filled out a questionnaire about her experience.

Because some of the annotators were neophytes, and even the relatively simple MUC task features some complex details, we provided the proctor with an answer key for each workspace, and during the experiment, the subjects were encouraged to ask the proctor for guidance if they could not decide how to annotate a particular phrase. We judged this option to be better than requiring them to make their own best guess at an answer, because we were far more interested in the differential performance of the subjects among conditions, rather than the absolute level of accuracy achieved by a given annotator in a given condition. While we did not record how often each subject asked for guidance, the proctor's subjective observation is that most, if not all, subjects used this option sparingly, and none of them appeared to resort to it differentially between conditions.

We logged all aspects of the subjects' interactions. The Callisto user interface reported timing information about what annotation gestures the subjects performed, and the MAT workspaces logged every change in the state of the workspace. Once the experiments were completed additional statistics were compiled. Models were built from each annotator's marked-up text and those models were evaluated against the MUC-6 test set; the annotators' work was also scored against the ground-truth marking. The resulting compiled dataset associates model performance with human annotation time and numerous summary statistics about the quantity and classes of annotations performed. This data is the basis for our conclusions about human annotation performance.

4 Related Research

The idea of monitoring human annotators with the aim of understanding the true costs for active learning is not a new one. There are several prior efforts that have attempted to measure human

annotation times on natural language processing tasks.

Arora et al. [2009] study annotation times on a movie review task, and attempt to estimate the time it will take to annotate new examples. Their study includes twenty annotators and is notable in that they observed that modeling attributes of individual annotators, in addition to attributes of the text to be annotated, improves the performance of their predictors. While this finding is likely unsurprising to practitioners who have experience annotating text, it is noteworthy in that it establishes a quantifiable relationship between annotation costs and the individual performing the annotation.

Haertel et al. [2008] collects human annotation times from annotators correcting automatically generated part-of-speech tags. This study is similar to ours in that the annotators are shown the pretagging, and the task involves sequence labeling. From the data captured, a linear function of sentence length, number of corrections, and bias is computed. In their linear function the authors find that the number of corrections is a more important factor than the length of the sentence. Further attempts are made to derive cost-sensitive active learning selection algorithms by utilizing their learned cost measure. In this work annotators are acknowledged to have differing skill levels, but those differences are not explicitly captured in their regression model.

Baldridge and Osborne [2004] performed a similar study where annotators select the correct parse tree by interacting with a set of candidate parse trees. This paper introduces the *discriminant cost* which measures the number discriminating decisions the annotator needs to make before the space of possible parse trees is reduced to the correct answer. Like the previous two papers this work avoids simplistic assumptions about annotation costs, but unlike the previous two it does not attempt to model human annotation times explicitly, nor to establish that the discriminant cost correlates with real costs.

Most similar to our task, Settles et al. [2008] collects timing data from annotators in several NLP tasks including named entity recognition. Like the work of Haertel et al. [2008], these annotator timings are collected in order to construct a new cost-aware selection heuristic. Along the way the authors observe the connection between the total number of actions required and the total time required for annotation. Beyond that observation, this work stands out in that it is the only known work to complete the active learning loop. That is, they use the the machine-learned cost predictions to re-weight the least confidence predictions, and select example sentences that will make the largest improvement in the model *per unit time*. The new cost-sensitive selector is then evaluated using human annotators and wall-clock timings. Ultimately the new predictor is not successful at saving annotation time; the authors attribute this failure to difficulties in estimating the time required to annotate new material.

As to the relative costs of document-level annotation versus sentence-level annotation, the prior work is mostly silent. The experiments in Settles et al. [2008] include annotations on whole documents in the "Community Knowledge Base" (CKB) corpus, and also sentence-level annotations in the "SigIE" corpus. The timings on these different corpora are not comparable because the annotation tasks are different (CKB is a named entity and relation task while SigIE is a straight sequence labeling task). In Baldridge and Palmer [2009] experiments are conducted for creating inter-gloss text for the Uspanteko language; these offer some insight into the merits of document-based annotation and segment-based annotation. The timings cover many hours of work for two annotators, one expert and one non-expert, and the results consider the true costs of annotation to be the time

spent annotating. The document-level annotation strategy (called "sequential") is never the most helpful, though it is competitive with the sentence-level active learning for the non-expert annotator. Despite the large amount of annotation collected, the observation that the relative strengths of selection and presentation strategies are not consistent across annotators calls into question the wider applicability of these findings.

By similarity to independent and identically distributed (i.i.d.) machine learning tasks, we have used basic least confidence ranking methods for our active learning simulations in our sequence labeling task. As indicated in section 2, the minimum posterior probability is not the only ranking strategy that might work for structured prediction tasks. Indeed there is a rich literature on active learning for general structured output spaces which considers the structured learning problem as a set of interrelated decision problems rather than as single prediction. For example, Mejer and Crammer [2010] and Roth and Small [2006], each treats active learning in a structured setting as a ranking problem spanning many machine-learned decisions. While neither of the papers mentioned considers the human factors involved involved in implementing active learning, they raise the possibility that many attempts to integrate annotation for natural language processing and active learning might be sub-optimal because selection strategies are not sufficiently adapted for the complex machine learning taking place.

Sequence tagging tasks also allow for the possibility of integrating machine learning with the annotation process beyond pre-tagging with the model's output. This idea appears in Baldridge and Palmer [2009] where annotators are asked to reduce the space of possible parse trees, but is also applied to sequence tagging in Culotta and McCallum [2005] in the form of constrained Viterbi correction. In constrained Viterbi correction, the model dynamically propagates changes made to a token's label to all of the other tokens in the sequence. Updating the annotations dynamically can propagate corrections more efficiently when the model can correctly infer other tokens' labels once a correction has been made. The authors also measure the effort to correct a sequence as a function of the number of user operations required to correct the sentence, rather than the number of annotations that are ultimately updated. The authors quantify the *difficulty* of correcting a sequence in the raw number of operations required, but not the *cost* in human effort to effect those changes.

5 Our Findings

5.1 Document-level annotation bests segment-level annotation

Timed annotation experiments showed that annotating documents was slightly more efficient than annotating segments. Each of the annotators spent approximately the same number of minutes annotating in document mode as in segment mode and the resulting data enables a comparison between the two conditions. Ideally we would like our data to address the question of which annotation condition should be used if the annotation could stop at any moment.

Figure 6a shows segment-level annotations overlaid with document-level annotations for one of our subjects. To build a comparable dataset two pre-processing steps were required. First, since test subjects were allowed to continue the document they were working on at the end of the 35 minute session, the last point in the document condition sometimes extends beyond the last measured point in the segment condition. To select only paired measurements, we considered only the points

up to the 1824 second mark, instead of the full 2287 seconds of observation that we collected in the document condition. Second, our timing data did not come paired. For example a user might have finished annotating a segment at 1553 seconds in the segment condition, but in the document condition the closest observation came at 1530 seconds. To cope with this and create a paired dataset we picked points on the document curve and used linear interpolation to add a corresponding point on the segment curve. The resulting dataset consists of a set of paired points, one for each point on our document curve, indicating what model F-measure a single annotator could achieve by annotating in document mode and in segment mode. A plot of these paired points can be seen in Figure 7.

The data in Figure 7 indicate more points above the line $y = x$ than below it. If performance were equal between the two conditions one would expect an equal number of points below the line as above it. Further analysis with the Wilcoxon Signed Rank test reveals that this difference is statistically significant at the $p = 0.05$ level. This finding came as a surprise to the authors, who having reviewed a version of Figure 6a for each annotator had suspected that neither condition would prove superior at the $p = 0.05$ level.

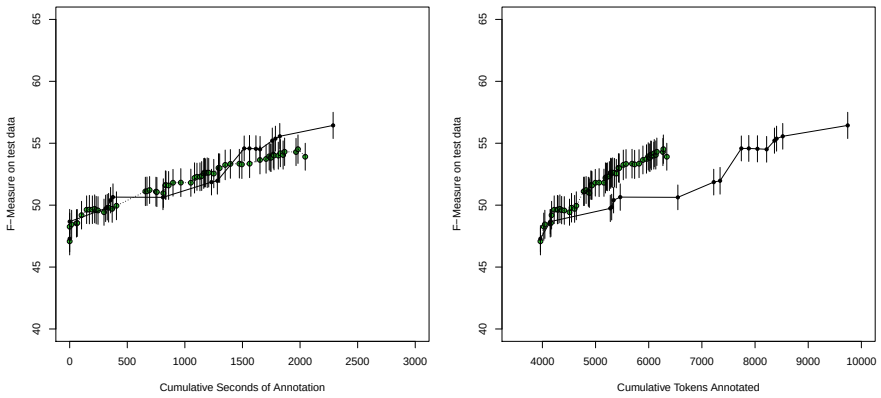
We caution against reading too much into this result. While we have tried to work our data into a framework appropriate for the Wilcoxon test, the F-measures in Figure 7 are not truly independent and identically distributed which is a pre-condition for the Wilcoxon test. In essence the points at 1500 seconds are dependent on the points at 1000 seconds. To further emphasize this point, consider an alternative formulation where we had one point for every segment condition observation (instead of one for every document condition observation) and interpolated the point's pair from the document curve. In such a formulation we would have many times more points in Figure 7 and the Wilcoxon test would yield a far lower p-value, but those additional points would be little more than noisy duplicates of the ones already tested. [2] It is therefore very possible that the statistically significant p-value we found is artificially low because of a similar effect.

Even if the statistical significance is an artifact of the data preparation stage, these results are a strong rebuttal to the conjecture that segment mode annotation is more efficient (Figure 4). We had initially expected tokens and time to track fairly closely, leaving segment mode the clear winner. This did not happen. In fact, when we re-plot the data in Figure 6a so that the x-axis is measured in tokens we recover the anticipated result shown in Figure 6b. In the re-plotted graph we find segment mode annotations appear to outperform document mode annotations, but this effect is merely an illusion based on the faulty assumption that tokens annotated is a reasonable proxy for time. When we use real annotation times, segment mode is not more efficient.

5.2 Annotation times vary widely

Our data reveal that annotation times vary widely by segment and by annotator. In other words, the assumptions that all segments take the same amount of time to annotate, or that the relative performance of the annotators would be consistent across the data, do not appear to be true. The first assumption is made implicitly when plotting active learning times against the number of segments annotated. Constant or approximately constant annotation times would justify the unit cost assumptions. The second assumption is useful for predicting the amount of time it will take to

[2] Thanks to Warren Greiff for pointing out this methodological weakness.



(a) Cumulative annotation *time* (b) Cumulative *tokens* annotated

Figure 6: Model F-Measure vs. human annotation time for one annotator. Solid line with black dots indicates document mode annotations, and dashed line with green dots indicates segment-mode. Points are plotted at our estimate of the model's true F-measure and error bars mark one standard deviation of spread. (Best viewed in color)

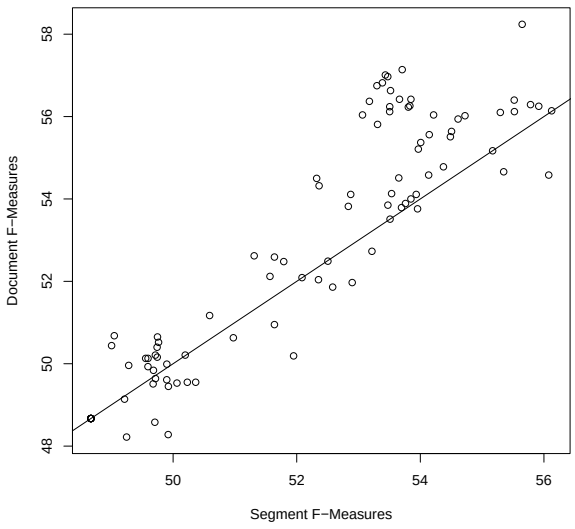


Figure 7: F-Measure scores for document and segment conditions, holding annotator and time constant.

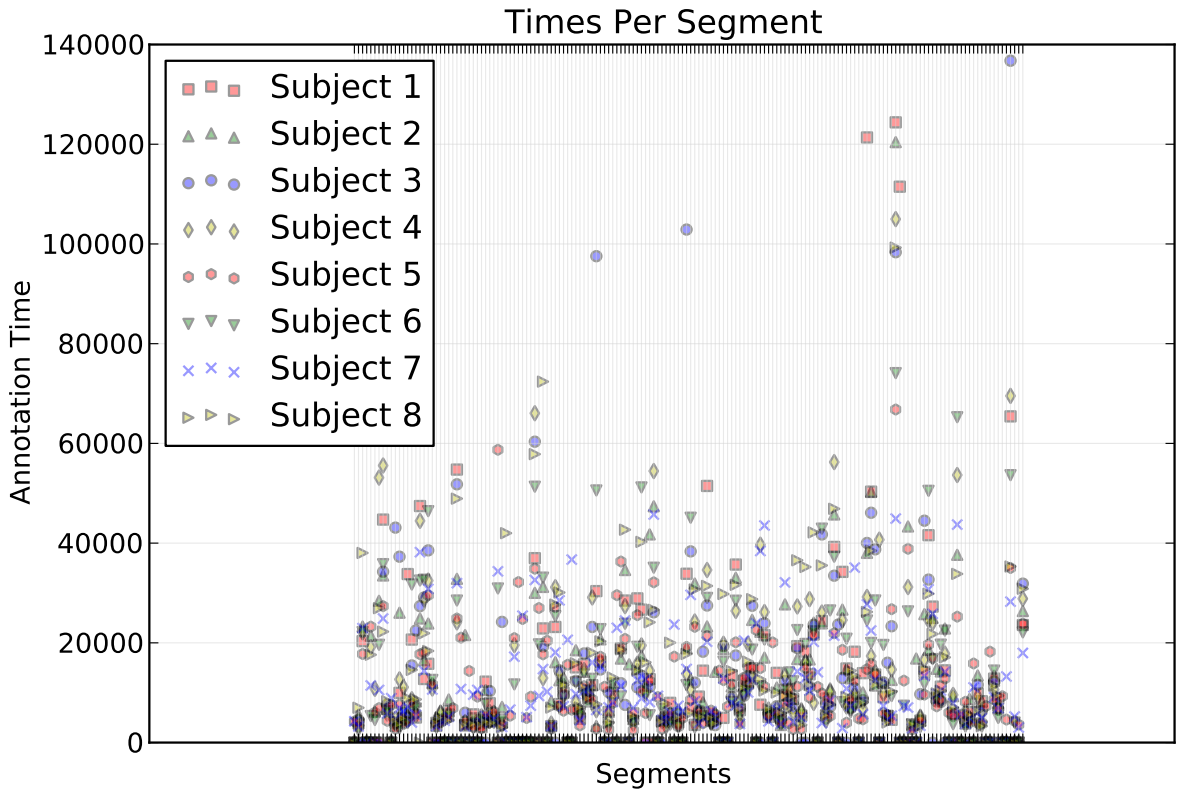


Figure 8: Wall-clock time required to correct individual segments in our dataset.

annotate a segment. A necessary but not sufficient condition for predicting annotation times is that the annotators' relative completion times are the same across all of the segments. Figure 8 offers a view the number of milliseconds it took each annotator to complete each segment. Segments that were time-consuming for some annotators were not necessarily time-consuming for the others, and moreover the outliers appear to be quite large. Such outliers complicate the task of prediction.

It is important to note that while times vary widely there are possibly other factors at work that we did not analyze completely. Chief among these factors is how well individual annotators performed when annotating the sentences. The annotators did make mistakes and the neophyte annotators made more mistakes than the experienced annotators. Thus it is possible that the most time consuming segments by the expert annotators were instances where the non-experts made mistakes that the experts did not. Another possibility is that the Callisto annotation tool was responsible for the outliers; one user reported that random misclicking with the mouse could create or alter spans in a way that made sorting out the problem more difficult than annotating from scratch.

5.3 Predicting annotation time well is difficult

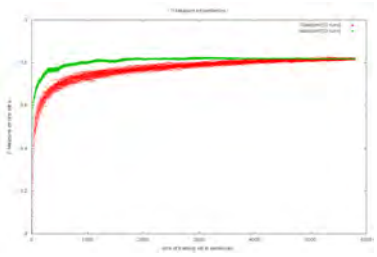
Predicting human annotation time leaves a lot of room for improvement. We used a support vector machine regression model to estimate the correction times shown in Figure 8 based some of the attributes we logged. The attributes used in the regression model included the number of spans the user created, the number of segments deleted, the number of label changes, the number of extent changes, the total number of actions performed, the total number of tokens in the segment at the time it was saved, the total number of characters in the segment at the time it was saved, the number of annotations in the ground truth, the number of annotations provided by the pretagging at the start, and the number of annotations provided by the pretagging that remained at the end. Some of these features can only be computed *after* the subjects annotated the sentences, and for this reason our results should be taken as an upper bound for predicting annotation time. Following Settles et al. [2008] the performance was measured by way of the correlation coefficient for which we achieved a score of 0.75 on our data. By way of comparison, on SigIE data Settles et al. [2008] reported a correlation of 0.85; however, that result was achieved using features that could be computed after pretagging but before human annotation. Unfortunately in Settles et al. [2008] even predictions at the 0.85 level were not sufficient to realize a working cost-sensitive active learning algorithm, and so we did not attempt this task in this project. Our machine-learned predictions are, however, *better estimates* of human time than the unit cost or token length proxies commonly presented in the literature.

5.4 The number of edits is a decent predictor of annotation time

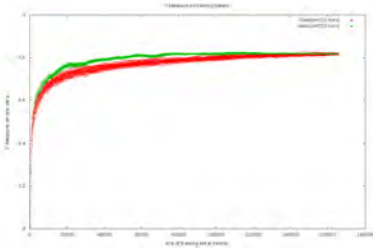
Our regression experiments uncovered that the number of edits performed was a reasonable predictor of total time. Our analysis indicates that token count is a better estimator than the unit cost, and that number of changes the annotator made is an even better estimate than token count. The predictions obtained from the support vector machine regression model described above have the advantage of combining all of these factors and as a result performed even better, though they still fall short of predicting the true wall-clock time.

5.5 Evaluation with an alternative proxy for cost shows no benefit from active learning

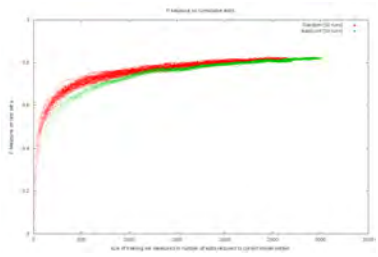
The benefits from active learning disappear when we replace the *sentences annotated* cost proxy or the *tokens annotated* cost proxy with the *cumulative edits* proxy. Our experiments with regression showed that the number of edits the subject actually performed was a better predictor of correction time than either total tokens or the unit cost assumption, thus it seems reasonable that if one is going to use a proxy for annotation cost then using one derived from real experiments is a better plan. The challenge is that the true number of edits performed is only accessible in hindsight. What is known at simulation time is the number of edits required to transform the pretagged segment into the ground truth segment. We therefore compute a new proxy metric, "number of edits" by taking the cumulative number of changes required to transform from pretagged to ground truth. The changes we consider include changes in span extent, changes of span label, deletions of spans, and insertions of spans. We did not attempt to measure specific types of changes such as changing a



(a) Minimum probability ranking, measured in sentences

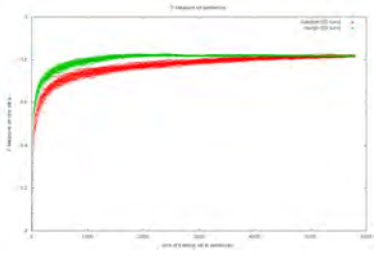


(b) Minimum probability ranking, measured in tokens of annotation

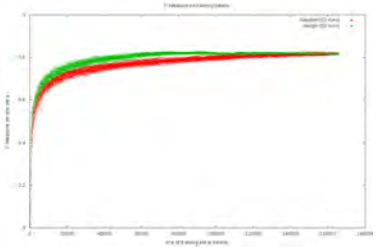


(c) Minimum probability ranking, measured in edits required

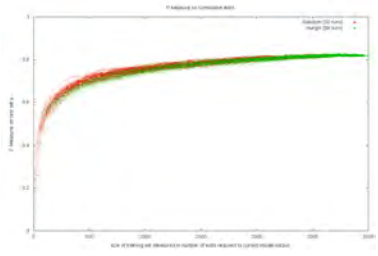
Figure 9: Active Learning curves using least confidence measured in on three different scales



(a) Minimum margin ranking, measured in sentences



(b) Minimum margin ranking, measured in tokens of annotation



(c) Minimum margin ranking, measured in edits required

Figure 10: Active Learning curves using least confidence measured in on three different scales

tag's label from a location to an organization. It is possible that the edit path the subject followed was not the same as the number of edits we estimate. The subjects, after all, did not always arrive at the correct ground truth annotations after they corrected the pretagging, nor did they necessarily follow the optimal path when the did. While we propose using the "number of edits" proxy, we should caution that this metric is similar to but different from the metric used in our regression experiment which was the number of edits actually performed. Unfortunately we realized only after collecting our data that it would not be possible to include the "number of edits" feature that is available at simulation time in the regression experiments. Nevertheless, the feature we did record and the feature we can compute share the important similarity in that they both capture low-level annotation actions.

Using the saved pre-tagged output, we can re-plot existing active learning experiments with a new cost proxy and see that the benefit of active learning vanishes. Figures 9a, 9b, and 9c illustrate this effect on the minimum probability ranking function. Figures 10a, 10b, and 10c also illustrate this effect with the minimum margin ranking heuristic, although this method appears indistinguishable from random rather than worse than random.

5.6 Cost-aware active learning is unlikely

Could cost-sensitive predictions improve active learning? A natural extension of attempts to learn a good regression model for human annotation costs is to apply that cost model to select the sentences

that give the most help to the model per unit of cost. Unfortunately the experiments we conducted were not designed to produce data that could test that hypothesis. We would need to know the true cost of annotating every sentence in the MUC-6 corpus, and compare that to the cost predicted from a regression model. Re-annotating all of MUC-6 was not feasible. To our knowledge the only work to explore this scenario is Settles et al. [2008]. In the work presented, machine learned annotation costs under-perform the random baseline on the sequence labeling task the authors consider. Only when the selection heuristic has access to the true annotation cost – a scenario akin to having a flawless regression model of annotation time – does the active learned model outperform the random baseline.

5.7 Recognized Limitations

With the exception of the final observation about the difficulties of realizing a cost-sensitive active learning strategy, which is not an empirical conclusion from our experiments but rather an under-appreciated negative result in the literature, our conclusions are only as good as the data we collected. What follows is a partial list of recognized shortcoming in our experiment methodology which could reasonably limit the generality of the conclusion we have drawn.

1. *The Callisto segment annotation UI may not be efficient.* One might argue that the segment-oriented interface we developed for Callisto isn't an optimal interface for segment review, since it presents the entire document context, which might often be irrelevant. One might imagine an alternative UI in which the document context is hidden, or even a UI which presents a list of segments outside of the document context and allows the user to click through to the context if needed.

   These alternatives illustrate that when evaluating human-in-the-loop annotation, UI affordances have a potentially big impact on user performance.

2. *The complexity of this task may not be representative.* In fact, we can strengthen this statement: the complexity of this task is not representative. In fact, no task is. MUC annotation is relatively simple and context-insensitive, and this has consequences for the value of particular UI design choices, as well as the load on the user of examining the document context on a document basis vs. from the point of view of an individual segment. But tasks vary widely along this continuum. The most we can say about MUC annotation is that it's simple enough that other tasks are likely to impose a heavier load on the user for context review.

3. *The subjects were not representative.* We used both experts and neophytes in our experiment. Perhaps the results would have been different if we'd used only annotators with MUC annotation experience, or annotators with Callisto experience, etc.

4. *We may have started at the wrong place on the F-measure curve.* We chose a point on the F-measure curve which was likely to show us observable F-measure changes in a time frame compatible with an experiment like this. But it's possible that this was the wrong thing to do. The value of active learning differs considerably depending on where you are on the F-measure curve, and one can imagine a situation where segment-oriented annotation is deployed only at a particular point in the curve, different than the one we examined.

5. *We didn't consider the effects of extended annotation.* 35 minutes is about as long as we could justify asking subjects to annotate in each condition - but it's not enough to evaluate the effects of stamina. For instance, it might be that the value of segment-oriented annotation is partially found in its continued novelty; all our subjects reported that they preferred that mode, and many commented that it was more interesting.

6. *We didn't consider the effect of the order of material.* If we want to apply these results to active learning, we can only do so under an assumption that annotation mode, but not the order of material, has an effect on the annotator. It's possible that this assumption is wrong; maybe active-learning-ordered segments are experienced differently by the annotator than randomly-ordered segments.

7. *Segment-oriented review is frequently inapplicable.* You can only annotate segment-by-segment if you don't have to make non-local judgments. So non-span tasks like co-reference annotation, where the user is asked to annotate co-reference chains, for instance, might not be amenable to segment-by-segment annotation.

6 Discussion

Our experiment, initially designed as a boat-race between segment-based annotation and document-based annotation, has highlighted the immense space of variables that can affect annotation costs and the efficacy of active learning at reducing those costs. In summary we found that it is straightforward to use active learning heuristics to reduce the number of annotated sentences required for a particular f-measure target compared to a random ordering of sentences, but that reducing the time costs of annotation is a much harder problem. There is no simple transition from that well-known offline result shown above to a production-ready software environment for annotating text.

To our surprise, the popular least confidence heuristic performs worse than the random baseline when it is rescaled and plotted with an improved cost estimate, namely cumulative edits. This result strongly suggests that the least confidence heuristic is biased towards selecting text regions that require lots of correction, and are therefore more expensive to annotate. On an estimated cost per unit of f-measure basis, there is no evidence to support using raw least confidence heuristics for bootstrapping a named entity recognizer. While it is possible that a new cost-aware selection heuristic could be created by fusing a traditional method like least confidence with a highly accurate predictive model of the user's annotation cost, demonstrating this feat remains an open problem. Our efforts to model human annotation times with a regression model have achieved comparable performance to previous efforts, but failed to make significant improvements on those results which would justify additional investigation of cost-aware fusion algorithms.

Faced with the very real possibility that active learning may be doing worse than random, we were hesitant to include it in the TooCAAn toolbox. When we further considered that active learning imposes non-trivial constraints on the annotation process, we decided to remove support altogether. Ultimately the constraint that the annotator work exclusively in sentence-mode and pay the overhead cost of waiting for the models to re-train, re-tag, and re-rank the corpus proved to be too great a cost.

The decision to remove support for active learning from the TooCAAn toolkit should not be taken to mean that active learning can never work for text annotation problems, but rather results from the

failure of our experiment to uncover sufficient evidence that it would improve annotator efficiency. What is more, our study is far from exhaustive. We have found negative results using minimum posterior probability and minimum margin ranking heuristics in linear chain conditional random field models for a named entity recognition task. Although our natural language processing task is an important one with many practical applications, it is but one of many applications of conditional random field models, and, more broadly, it is one of many applications of machine learning to natural language processing. It is entirely possible that active learning could be found to be successful on some different combination of datasets, tasks, machine learning models, ranking strategies, and annotation interfaces.

Acknowledgments: The authors are grateful to Ben Wellner for many helpful technical discussions as well as for providing insightful comments on earlier drafts of this report.

7 References

S. Arora, E. Nyberg, and C. P. Rosé. Estimating Annotation Cost for Active Learning in a Multi-Annotator Environment. In *Proceedings of Active Learning for NLP workshop at NAACL-HLT 2009*, ALNLP '09, pages 18–26, 2009.

J. Baldridge and M. Osborne. Active Learning and the Total Cost of Annotation. In *EMNLP*, pages 9–16, 2004.

J. Baldridge and A. Palmer. How well does active learning *actually* work? Time-based evaluation of cost-reduction strategies for language documentation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 296–305. ACL, 2009.

A. Culotta and A. McCallum. Reducing Labeling Effort for Structured Prediction Tasks. In *AAAI*, pages 746–751, 2005.

D. Day, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson, and M. Vilain. Mixed-initiative development of language processing systems. In *Proceedings of the fifth conference on Applied Natural Language Processing*, pages 348–355, 1997.

D. Day, C. McHenry, R. Kozierok, and L. Riek. Callisto: A Configurable Annotation Workbench. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation*, LREC '04, pages 2073–2076, 2004.

R. Duda, P. Hart, and D. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, October 2000.

R. Grishman and B. Sundheim. Design of the MUC-6 evaluation. In *Proceedings of the 6th conference on Message understanding*, MUC6 '95, pages 1–11, 1995.

R. Haertel, E. Ringger, K. Seppi, J. Carroll, and P. McClanahan. Assessing the Costs of Sampling Methods in Active Learning for Annotation. In *Proceedings of the ACL-08*, pages 65–68, 2008.

R. Haertel, P. Felt, E. Ringger, and K. Seppi. Parallel active learning: eliminating wait time with minimal staleness. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, ALNLP '10, pages 33–41, 2010.

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

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

A. Mejer and K. Crammer. Confidence in Structured-Prediction Using Confidence-Weighted Models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 971–981, Cambridge, MA, October 2010. Association for Computational Linguistics.

L. Ramshaw and M. Marcus. Text Chunking using Transformation-Based Learning. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94, June 1995.

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 *Proceedings of the Linguistic Annotation Workshop*, LAW '07, pages 101–108, 2007.

D. Roth and K. Small. Active Learning with Perceptron for Structured Output. In *ICML Workshop on Learning in Structured Output Spaces*, June 2006.

B. Settles. *Curious Machines: Active Learning with Structured Instances*. PhD thesis, University of Wisconsin–Madison, 2008.

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, 2008.

B. Settles, M. Craven, and L. Friedland. Active Learning with Real Annotation Costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, 2008.

D. Shen, J. Zhang, J. Su, G. Zhou, and C.-L. Tan. Multi-Criteria-based Active Learning for Named Entity Recognition. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 589–596, Barcelona, Spain, July 2004.

E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, 2003.

K. Tomanek and U. Hahn. Semi-Supervised Active Learning for Sequence Labeling. In *ACL/AFNLP*, pages 1039–1047, 2009.

Ö. Uzuner, Y. Luo, and P. Szolovits. Viewpoint Paper: Evaluating the State-of-the-Art in Automatic De-identification. *JAMIA*, 14(5):550–563, 2007.

Appendix A Appendix: Condensed MUC6 Named Entity Instructions

Summary

You will be annotating three elements, all of which are proper names. We'll indicate the boundaries of the annotations with XML-like brackets in this document; in your experiment, the different labels will be represented by contrasting background colors.

Definitions

ORGANIZATION

An ORGANIZATION label marks a named corporate, governmental, or other organizational entity. These will be delimited in this document with the <0> tag. Examples: <0>IBM</0>, <0>Merrill Lynch</0>, <0>Microsoft</0>, <0>Department of Justice</0>, <0>Warsaw Pact</0>, <0>New York Stock Exchange</0>.

Names of sports teams, stock exchanges, newspapers, multinational organizations, political parties, orchestras, unions, governmental bodies at any level of importance, all count as ORGANIZATION. Names of facilities (e.g., factories, hotels, universities, airports, hospitals, churches, ballparks, bridges) count as ORGANIZATION only if they refer to the organization and not the actual structure or place. Other mentions of facilities are not tagged The name of a city or country counts as an ORGANIZATION rather than a LOCATION if it refers to a team or other organization, rather than the place.

PERSON

A PERSON label marks a named person or family. These will be delimited in this document with the <P> tag. Examples: <P>Bill</P>, <P>Sara Smith</P>, the <P>Kennedys</P>.

LOCATION

A LOCATION label marks the name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains, etc.). These will be delimited in this document with the <L> tag. Examples: <L>Crater Lake</L>, <L>Tunisia</L>, <L>Lubbock</L>, <L>Texas</L>, <L>West Coast</L>, <L>Northeast</L>, <L>Midwestern Corn Belt</L>, <L>the West</L>.

Things not to mark

- Brand names and product names (e.g., Macintosh, MacBook)

Right	**Wrong**
<L>Lubbock</L>, <L>Texas</L>.	<L>Lubbock, Texas.</L>
<L>California</L>'s	<L>California's</L>
<0>Microtest Inc.</0>	<0>Microtest Inc</0>.
the <0>McDonald's</0> chain	the <0>McDonald</0>'s chain
<L>Phila-delphia</L>	<L>Phila</L>-<L>delphia</L>

- Name uses that are generic and refer to individuals who don't have that name (e.g., "the Intels of the world")
- Street addresses, street names, and adjectival forms of location names (e.g., "African").

General rules

The following rules cover the majority of the cases you'll encounter. If you have a specific question about a particular phrase during the experiment, feel free to ask.

Most of the time, periods, commas, apostrophes, etc. should not appear within annotations. The primary exception is if the punctuation is part of the name itself, either as a matter of convention or because of, e.g., word hyphenation.

In general, you should mark the largest name that you can (including final elements like "Jr." or "Co."), but don't mark annotations inside other annotations, or include extraneous words like conjunctions or titles which aren't part of an official name:

You should mark the name everywhere you find it, as long as it's clear that it's being used in one of the appropriate ways. You should also mark nicknames and acronyms for known names:

Right	**Wrong**
<0>Bridgestone Sports Co.</0>	<0>Bridgestone Sports</0> Co.
<P>John Doe, Jr.</P>	<P>John Doe</P>, Jr.
<L>North</L> and <L>South America</L>	<L>North and South America</L>
<0>U.S. Fish and Wildlife Service</0>	<0>U.S. Fish</0> and <0>Wildlife Service</0>
<0>U.S. Fish and Wildlife Service</0>	<0><L>U.S.</L> Fish and Wildlife Service</0>
<0>Boston Chicken Corp.</0>	<0><L>Boston</L> Chicken Corp.</0>
<0>Boston Chicken Corp.</0>	<L>Boston</L> <0>Chicken Corp.</0>
<0>Temple University</0>'s <0>Graduate School of Business</0>	<0>Temple University's Graduate School of Business</0>
Mr. <P>Harry Schearer</P>	<P>Mr. Harry Schearer</P>
Secretary <P>Robert Mosbacher</P>	<P>Secretary Robert Mosbacher</P>
<0>FEDERAL HOME LOAN MORTGAGE CORP.</0> (<0>Freddie Mac</0>)	<0>FEDERAL HOME LOAN MORTGAGE CORP. (Freddie Mac)<0>

Right	**Wrong**
<0>Intel</0> Vice President <P>John Hime</P>	Intel Vice President <P>John Hime</P>
the <P>Kennedy</P> family	the Kennedy family
the <P>Kennedys</P>	the Kennedys
<0>IBM</0> [alias for International Business Machines Corp.]	IBM
<L>the Big Apple</L>	the Big Apple

This page intentionally left blank.