Modeling the Annotation Process for Ancient Corpus Creation

James L. Carroll, Robbie Haertel, Peter McClanahan,
Eric Ringger and Kevin Seppi
Brigham Young University

1. Introduction

The ideas in this paper arose from a project to develop an electronic corpus and concordance of ancient Syriac literature. We will use this project to illustrate many of the ideas in this paper. The Syriac project at Brigham Young University involves many individuals from several departments including Linguistics, Computer Science, and the Center for the Preservation of Ancient Religious Texts. The project team also includes scholars from Oxford and Princeton Universities. Syriac texts have been transcribed manually by teams of Maronite, West Syrian and East Syrian Christians and Monks located in Lebanon, Rome, Iraq, Chicago and Oxford. The proximate goal of this project is to produce a corpus tagged with part of speech data for the writings of the fourth century Syriac poet-theologian Ephrem the Syrian (d. 373). This initial corpus is approximately half a million words in size. A further four million words have been added to the corpus in draft format. These texts originate from the third to the thirteenth century. However the majority of the texts are from the fourth to the seventh centuries, the so called Classical period of Syriac literature. It is the long-term aim of the project to build a comprehensive corpus of Syriac literature, working diachronically through the available texts. Much of Syriac literature has already been published, and these published texts are used in the corpus. However, a great deal of Syriac literature is available only in manuscripts. It is impossible to precisely estimate the size of the corpus; however, it is not improbable that the corpus extends to over 30,000,000 words.

We do not have the resources to fully annotate a corpus of this size with morphological tags. We are taking a pragmatic approach to annotating texts for the corpus. The first stage is to prepare a draft transcription with machine annotation. Texts will then be proofread and annotated by hand as scholarly interest is raised to a sufficiently high level to complete the work. Many texts in the corpus may never be fully proofread or annotated. Some text collections, beginning with Ephrem, will, however, be thoroughly proofed and tagged, sufficient to produce a full print concordance. A higher level of accuracy will be required for the print portion of the corpus than for the remainder of the corpus which will be published on the internet.

The production of electronic corpora for ancient languages involves several “annotation” tasks. Transcription, morphological and part of speech tagging, grammatical parsing, and semantic tagging can all be seen as annotation tasks. For example, in transcription the user takes an image and labels (or annotates) the image with transcribed text. In part-of-speech tagging the user takes a transcribed text and annotates the text with parts of speech etc. Thus annotation is central to each step in the creation of a useful electronic corpus. The goal of our part of the Syriac literature project is to reduce human annotation cost as much as possible through the appropriate use of machine learning and active learning techniques. We also seek to achieve lower error rates than could be achieved through human annotation alone and to appropriately balance the value of annotator time on the print corpus with the value of annotator time on the internet corpus.

1.1 Issues in Corpus Creation:

Human annotation can be very expensive, and this expense is often the limiting factor in the creation of electronic corpora. Since ancient languages are generally less well know, their annotation requires more specialized language knowledge, which can make human annotation even more expensive.

One solution to this problem is to use machine learning to automatically annotate the data. Machine learning approaches are available for transcription (OCR or Optical Character Recognition), part of speech tagging, parsing and semantic role labeling. Unfortunately, machine learning approaches to annotation often have higher error rates than human annotation and they often require a large set of previously labeled data in order to “train” the machine learning model. Typically the larger the initial training set the better the algorithm will perform. Often, when dealing with ancient languages this initial training set is either nonexistent or extremely small.

These problems with machine learning can be overcome by combining machine learning with human annotation. The goal of such a combination is to use the expensive but more accurate human annotation in the most beneficial way to lower the error rate of the entire annotated corpus as inexpensively as possible. Typically the computer selects the examples to be annotated by the human that it believes will be the most beneficial. This process is called “active learning.” Active learning is invaluable when there are insufficient resources to use a human annotator over the entire corpus. Even when there are sufficient resources to annotate the entire corpus by hand, many errors likely remain. Active learning can focus human attention on the most problematic sections and can result in higher accuracy than human annotation alone.

Several complex and poorly understood interactions arise when attempting to integrate human annotation with machine learning and active learning in the creation of a large annotated corpus. These interactions arise among the following components of the system:

1. The expense of human annotation
2. The machine learner
3. The active learning technique
4. How annotators are paid
5. The user interface
6. Human annotation error rates
7. Variability in error significance for print vs. internet portions of the corpus

In order to effectively integrate all these elements it is important to understand which elements affect other elements of the system. For example, how we pay our annotators affects the cost of annotating a sentence and can affect the sentence selection of the active learner. User interface design can also affect the cost of annotating a sentence as well as the accuracy of the annotations gained. Since human annotations are not 100% accurate, their accuracy must be modeled in order to determine what we believe and how sure we are about what we believe given a set of human annotations. Human annotations compose the machine learner's training set, therefore the model of human annotation accuracy should affect the behavior of the machine learner.

Understanding these interactions is important in order to answer several important questions. How often should we employ a second annotator and where would that annotator's work be most effective? Is it better for the second annotator to annotate a completely unseen example, or to verify the work of an annotator whose answer disagrees with the machine learner's annotations? Should we attempt to learn the abilities of each annotator separately? Should we give the annotators sentences where we already know the answer in order to determine their abilities? If so, how many should we give them? How should we effectively deal with the fact that errors in one portion of the corpus are more important than errors in another portion?

1.2 Modeling the Corpus Annotation Process:

In this paper we propose a Bayesian, decision-theoretic, model of the corpus creation process. The model helps to answer the above questions and clarifies the above interactions. Given the model of the process we can construct the theoretically optimal techniques for answering many of the above questions. The optimal solution to these questions can be computationally infeasible; however, the model provides a clear way of thinking about the problem. With the optimal solution in mind better heuristics can be developed which approximate the optimal solution. This approach (developing an optimal model and then approximating it to achieve a computationally tractable solution) has guided several advances in the annotation process in the Syriac language project as well as in many other machine learning projects.

In section 2, we present a Bayesian, decision-theoretic, model of the machine learning process itself and describe how that model can be extended to deal with the sequential data of natural language. In section 3 we show how this model also accommodates active learning. In section 4 we describe how utility in the model can be used to deal with situations where errors in one part of the corpus are more important than errors in another part (as is the case in our Syriac project). In section 5 we use the model to illuminate how annotation costs affect the active learning technique. In section 6 we discuss the interaction between the user interface, and the annotation cost, and illustrate this with a cost model obtained from a user study. In section 7 we extend the model to incorporate human annotation error rates. In section 8 we conclude by providing recommendations for the corpus creation projects.

2. Modeling the Machine Learning Process:

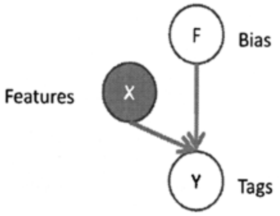

Figure 1: A simple graphical model for the classification problem in machine learning

We believe that the best way to think about the machine learning (ML) problem is as a graphical model (or Bayesian Network) (See Figure 1). (Carroll and Seppi, 2007; Carroll, et al., 2007; Buntine 1992). We will first discuss the implications of this simple model and then explore the additions that must be made to the model to represent the more complex sequential problems often encountered in NLP.

In machine learning there are features x, classes y (in the case of our project the part of speech tags), and an unknown function, f, that maps x to y. This network models the standard classification machine learning problem. The F node represents a distribution over possible functions f, and incorporates the ideas of a bias. This node can be represented in many different ways. For example, the distribution over possible functions could be represented by placing a distribution over the weights of a neural network (Freitas et al., 1998). If a distribution is placed over the parameters of most traditional machine learning (ML) representations then the result is often a distribution over possible functions and can be used to represent f (Caroll, et al., 2007). Bayes' law optimally dictates how the parameters should be updated in the presence of data (a set of labeled features and classes). An actual distribution over possible classes y can be produced by integrating over the parameters of F.

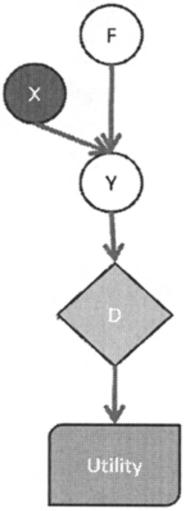

Figure 2: A decision network for classification D is a decision which results in an outcome with a given utility.

This formulation clarifies issues dealing with "No Free Lunch" (Wolpert, 2001; Carroll and Seppi, 2007). When this procedure is computationally intense, heuristics can often be used. Many current parameter update techniques can be seen as approximations to this network with specific distributional and simplifying assumptions; for example, backpropogation can be thought of as a maximum likelihood estimation of the more correct Bayesian approach to updating the weights (Freitas et al., 1998). Thus we can think of many machine learning techniques as heuristics to the more mathematically correct formulation. Thinking of traditional techniques in this way can often help us to better understand their behavior.

Perhaps the most important reason to think about machine learning in terms of a Bayesian network is the connection between Bayesian statistics and Utility Theory. The principles of decision/utility theory provide a technique for maximizing expected utility using probabilities, but these techniques are only guaranteed to be optimal when probabilities are determined by the laws of Bayesian Statistics (See Figure 2). All machine learning involves decisions. If utility is not included explicitly then hidden implicit utility assumptions are being made that may or may not correspond to reality. For example, since Artificial Neural Networks approximate maximum likelihood solutions for the weights, they therefore correspond to a utility function based on misclassification error on the weights. This is not always correct. The misclassification error utility assumption will fail whenever one type of error is more important than another, or when precision and recall are not exactly balanced. For example, in our case we care more about errors in one part of the corpus than we do in another.

By thinking about the problem in terms of a Bayesian Decision Network, utility is handled explicitly and we can see the theoretically optimal solution to computing the probability of a class p(y) that will maximize the expected utility (Carroll and Seppi, 2007). Then, if this optimal solution is too computationally intense to compute, we can at least design heuristics in a more principled way.

2.1 NLP Extensions to the ML Model

In many natural language processing (NLP) problems the network is much more complex than in the above examples. Primarily this is because data in NLP is inherently sequential. In machine learning there are many x and y pairs that have been observed (the training set) and a set of x and y pairs where the y node is unobserved (the test set). In the sequential tagging problem there are sequences of words (sentences) in the training and test sets. This means that the label at one time step is actually part of the feature set for the next time step (see Figure 3). Common approximations for computing the probability of a class/tag in this network include techniques such as Maximum Entropy Markov Models (MEMMs). A Maximum Entropy (MaxEnt) model is a log-linear model whose parameters are those that create the distribution of maximum entropy that still satisfies the constraints imposed by the evidence found in the training data (Ratnaparkhi, 1996; Toutanova & Manning, 2000; Toutanova et al., 2003). A gradient descent optimization procedure, such as LBFGS, is used to find the parameters during training.

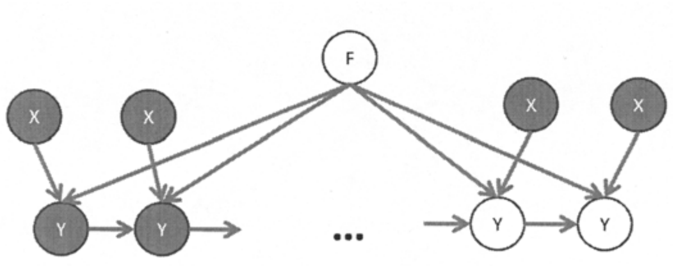

Figure 3: A sequential classification network.

An MEMM is a Conditional Markov Model (CMM) in which a Maximum Entropy (MaxEnt) classifier is employed to estimate the distribution

$$p(y_i|\underline{w},\underline{y}_{1..i-1}) \approx p_{\mathrm{ME}}(y_i|w_i,\underline{v}_i,y_{i-1},y_{i-2})$$

over possible labels y_i for each element in the sequence—in this case, for each word w_i in a sentence $\underline{w}$. The model also has access to any predefined attributes (represented here by the collection v_i) of the entire word sequence and to the labels of previous words $\underline{y}_{1..i-1}$ trained from labeled data. Our implementation employs an order-two Markov assumption so the classifier has access only to the two previous tags y_{i-1}, y_{i-2}. We refer to the features (w_i, f_i, t_{i-1},t_{i-2}) from which the classifier predicts the distribution over tags as "the local trigram context" (Ringger et al., 2007). State-of-the-art Part-of-Speech tagging results have been achieved with MEMMs (Ratnaparkhi, 1996; Toutanova & Manning, 2000; Toutanova et al., 2003). Part of the success of MEMMs can be attributed to the absence of independence assumptions among predictive features and the resulting ease of feature engineering.

3. Active Learning:

The sequential classification network can also be used to model active learning. The active learning task involves selecting an unobserved node (known as the test) and observing it. In the sequential classification network, this is done by selecting an un-annotated sentence which is then "observed" by having a human annotate it. The goal is to select a test that will provide the greatest improvement to the computer's estimates over the rest of the unlabeled classes. To select the test that will achieve the greatest improvement the sequential classification network is expanded into a decision network by adding decision nodes and utility measures (see Figure 4) as was done previously in Figure 2. The decision to be made is what part of speech tag should be applied to each word. Since the objective of the annotation process is to correctly annotate the corpus. We use accuracy as our measure of the utility of a choice.

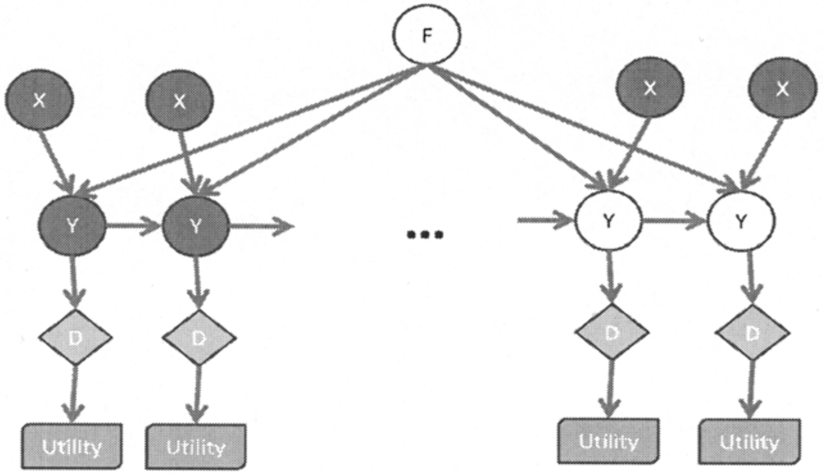

Figure 4: Utility in sequential learning for active learning.

In this decision theoretic context, active learning is selecting the un-annotated data to be annotated which maximizes the expected gain in utility. This value is known as EVSI, or the Expected Value of Sample Information (Raiffa and Schlaiffer, 1967). EVSIi involves the computation of the expected improvement in utility that would result from revealing the value in a particular hidden node (in our case a specific yi) in a Bayesian network and is computed as follows:

$$EVSI_i = \sum_{y_c \in D_y} P(y_c|x_i) \sum_{x_k \in D_x} p(x_k) \max_{a_j \in D_a} \sum_{y_l \in D_y} P(y_i|x_k, x_i\, y_c)\ U(a_j, y_l, x_k)$$
$$- \sum_{y_c \in D_y} P(x_k) \max_{a_c \in D_y} \sum_{y_l \in D_y} p(y_l|x_k)\ U(a_j, y_l, x_k)$$

where D represents the domain of various variables, a is an action, U is the utility, yc is the annotation of the test i and yl is the annotation in location k. Intuitively we are taking the expectation over every possible result of the test yc and computing the expected utility of making the decision if we had that information minus the expected utility of making the decision without any extra information. The net expected value of a node in the network is: $ENET_i$ = $EVSI_i$ - $ECSI_i$ where $ECSI_i$ (The Expected Cost of Sample Information) is the expected cost of gathering the information about node *i*.

Unfortunately the computation of ENET is computationally difficult and awkward. If, for example, the utility for EVSI is computed in terms of expected improvement in classification accuracy, while ECSI is measured in hours spent by the human annotator, then the difference ENET= EVSI-ECS only makes sense when both EVSI and ECSI are converted into the same scale (units). Since it is awkward to convert the number of hours of annotation to an increase in model accuracy, and it is likewise awkward to convert accuracy to hours, we could convert to both to some other measure. In other decision theory problems it is possible to convert all measures to some common measure, like money. In annotation, such a conversion would be very much a function of the particular project.

Note also that the computation of a single step of ENET alone is insufficient in annotation. If the active learner could only perform a single test then the optimal policy would be to sample the test with the highest ENET value. However, if the learner can perform multiple tests it is possible for two tests taken together to have a higher net value than any one single test alone. EVSI could be performed over every possible combination of tests, but this would make an already intractable computation worse (Carvalho and Puterman, 2003).

An intuitive heuristic can be used to solve both the scaling and the combination of tests problem. Tests can be selected greedily based upon the quotient EVSI/ECSI. This can be thought of as selecting the test with the most "Expected Bang per Buck" (EBPB) rather than the test with the highest ENET value. Geometrically this can be thought of as selecting line segments with the highest slope rather than line segments with the highest endpoint. Several shorter line segments with higher slope can eventually lead to a higher endpoint with less cost.

The relative ordering of $EVSI_i/ECSI_i$ will be the same as $\alpha EVSI_i/ECSI_i$ for all positive α. If EVSI and ECSI can be placed in the same units by a linear transformation then the "expected bang per buck" technique does not force us to define EVSI and ECSI in the same units. For our Syriac problem, cost/hour is linear and it is reasonable to assume that the usefulness of the corpus is a linear function of its accuracy.

Unfortunately computing the EVSI (and thus EBPB) of a node even in this oversimplified network is far too computationally intense. Things only get worse when we begin to add the additional complexity imposed by the sequential nature of NLP problems. Therefore we need an approximation to EVSI.

Several common techniques for performing active learning include Query by Uncertainty (QBU) and Query by Committee (QBC). QBU is a technique that selects the next sample as the node with the maximum uncertainty concerning its value. QBC trains multiple learners and selects the node with maximum disagreement between the learners. It can be shown that these techniques actually approximate EVSI given some simplifying assumptions. Under those assumptions EVSI of a node will be proportional to the uncertainty in that node, or to the disagreement between learners for that node. The existence of these heuristics can make the approximation of EVSI tractable inasmuch as the simplifying assumptions are met.

Traditionally these techniques are used directly to perform the active learning selection. For our purposes however, they will be used as a proportional approximation to EVSI, which will allow the EBPB calculation. If ECSI is uniform then this will be equivalent to the standard active learning techniques. For example, if we are paying annotators by the word, we could compute EBPB=EVSI/ECSI≈QBU/N where N is the number of words in the sentence to be annotated. We call this approximation of EBPB "NQBU" (or Normalized QBU).

4. Variability in Error Importance:

For our Syriac corpus we have part of the corpus that will be published to print and on the internet and part that will be published only on the internet. We will refer to these two portions of the corpus as the print and internet portions, respectively. Errors in the print portion of the corpus are more significant than errors in the internet portion.

A simple, if typical, solution to this problem is to have two annotators annotate the print corpus, with a third annotating whenever they disagree, and then to spend any remaining money on the internet portion. However, given our utility model, this approach is sub-optimal. Do we really want to spend the cost of a human annotator (especially of the second human annotator) on portions of the print corpus even though they have an extremely high degree of certainty? The implicit utility implication of demanding the second annotator regardless of how certain we are is to assume that there is infinite utility in the print corpus with finite or zero utility in the internet corpus. Furthermore, do we really want to be spending money on the internet corpus when there are portions of the print corpus that have low certainty even when the two human annotators agree? This can happen when the computer disagrees with both human annotators. In this case would it not be worthwhile to use a third annotator in such locations? Thus the typical policy assumes either that we are certain of the correct annotation after two annotators (which we are not) or that the utility of errors goes down after two annotators (which it does not). This combination of behavior is therefore irrational for all utility models.

In order to correctly balance annotation costs on the print and internet portions of the corpus it is necessary to be explicit in the cost of an error in the print and internet corpora. A reasonable approximation might be to assume that $EVSI_i \propto UNC_i \times U_i$ where U_i is inversely proportional to the cost of a mistake in that part of the corpus where i is found. Of course this is not strictly the case, since sample information in one part of the corpus could aid in the other part of the corpus. However, this could be a reasonable approximation if we assume that the data provides the most information in the section where it is found. Other more accurate (but more computationally intense) approximations could be imagined; however, any approximation will require an explicit measure of utility. Specifically we must be explicit about the importance of errors in each portion of the corpus in order to make reasonable decisions about the allocation of annotator effort.

5. Annotation Cost:

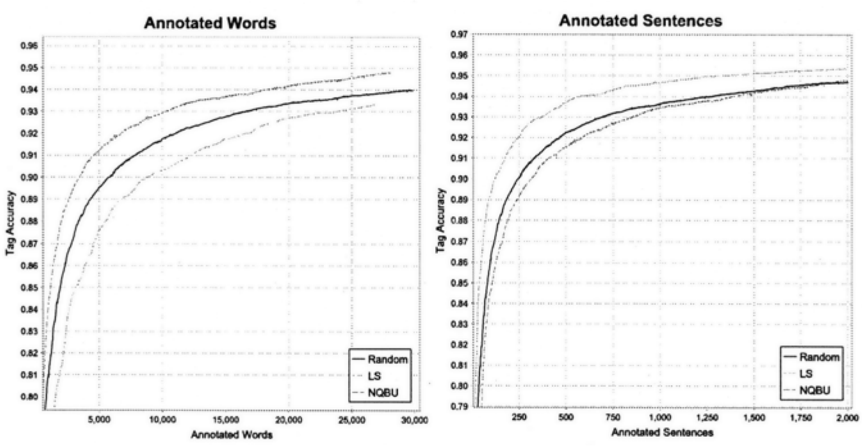

Figure 5: A comparison of three possible active learning techniques, random, longest sentence and NQBU (the uncertainty divided by the sentence length). Notice that if you are paying your annotators by the word then NQBU is the best approach, but if you are paying by the sentence then longest sentence is the best active learning approach.

Many active learning techniques (including QBU and QBC) ignore ECSI, yet it plays an important part in active learning: $NET_i = EVSI_i - ECSI_i$, and "expected bang per buck" $EBPB = EVSI_i/ECSI_i$. For either technique ECSI is central to the calculation. QBU and QBC both approximate EVSI but completely ignore ECSI. In corpus annotation it is often the case that some samples will cost more than other samples depending on the user interface involved. This can make a huge difference for active learning. For example, three reasonable ways that annotators could be paid are by the word, by the sentence, or by the hour. Different active learning techniques perform better depending on which cost metric is applied. Figure 5 indicates that when annotators are paid by the sentence a rather simple active learning technique (select the longest sentence) performs well, while NQBU performs worse than random. On the other hand, if our annotators are paid by the word, then longest sentence performs worse than random, while NQBU performs well. Not only did the method of payment affect the results, its influence was dramatic.

It is reasonable to assume that $EVSI_i \approx \alpha\ LengthSent_i$, because longer sentences tend to have more information in them. If $ECSI_i$ is constant for sentences of any length, then $EBPB_i \approx \alpha\ LengthSent_i$ and selecting the longest sentence is a reasonable policy if you pay by the sentence. On the other hand, if your annotator charges by the word, then $ECSI_i$ *is also* $\approx \alpha\ LengthSent_i$. Thus EVSI will tend to be larger with larger sentences, but so will ECSI, and a measure of the uncertainty per word (NQBU) is now the preferred measure.

When using active learning and paying by the hour it is important to know approximately how long you expect your annotators to take to annotate a given sentence, and use this expected time and their hourly rate to approximate ECSI. Then using an approximation to EVSI you can compute $EBPB = EVSI_i/ECSI_i$. This will appropriately penalize longer sentences because it will take your annotator longer to annotate them. Thus, without some model of how long it will take an annotator to annotate a sentence it is impossible to correctly determine whether that sentence should be selected by active learning. This observation motivates the user study which we will present in the next section.

6. User Interface and Modeling ECSI

We have seen that ECSI is an essential component of active learning. After this paper was presented but before the time of this writing, we performed a user study motivated by the above ideas to determine the expected time for annotating the part of speech of a sentence in English with the Penn Treebank tagset (Ringger et al., 2008). The user interface made suggestions based on the machine learner's current model and the user only had to change those words that were annotated incorrectly. Using the data from this study we developed a linear model for part of speech annotation cost suitable for use as the expected annotation cost in the context of Active learning algorithms. The final rational cost model is: $ECSI_i = 3.795\ l_i + 5.387c + 12.57$, where l is the length of the sentence and c is the number of words the user had to change. The resulting model has an appealing intuitive interpretation: the annotator reads each word and decides whether or not it needs to be corrected (3.795 seconds per word); correcting a word takes (5.387 seconds per correction); finally, there is 12.57 seconds of overhead per sentence.

The model uses only a small subset of the raw statistics we collected. There are two reasons for this: first, some of the statistics which we collected (for example, "Self Evaluation of Tagging Proficiency") were not included in the model because we explicitly wish to assume that tagging will be conducted by a mix of people with tagging skills similar to the mix of skills tested in the user study. Second, some variables fail to have a statistically meaningful effect on the resultant model. We employed linear regression and the Bayesian Information Criterion (as implemented in the LEAPS package in R) to assess which variables should be included in the model.

We intend to repeat this study for Syriac and would expect the results to be different depending on the language, the tag set, and the user interface. The results of the user study provide an expected approximation to the cost of sample information and will be an essential element to any effective active learning technique for language annotation. Furthermore, the amount of overhead (for English approximately 12.57 seconds) can have significant impact on the best way to present data to the annotators. It may be better to give the annotators a single most uncertain word in a sentence and ask them to only correct that one word so that they do not waste time on other words in the sentence which the computer may already have a good model for; or, it may be better to let them annotate the rest of the sentence since they have already paid the overhead. A user study is imperative for making such decisions.

Notice that the cost model implies that there is some overhead in reading each sentence. This could imply that the best user interface would ask annotators to annotate all the words in a sentence while they already have the context in mind. On the other hand, if fixing a single word in the sentence can drastically lower the uncertainty in the rest of the sentence such that the rest of the sentence need not be annotated by a human, then it will be better to annotate a word at a time.

Notice also that, for our user interface, the cost model is directly related to the accuracy of the classification machine learning model. The better the machine learning model, the fewer corrections the user has to make. For our user interface, these corrections accounted for a large proportion of the cost. This means that improvements in the machine learning model allow us to collect more human annotated data because that human annotated data is now cheaper.

Any other technique for speeding up human annotation will clearly result in being able to afford more human annotation. Thus, any active learning project for corpus annotation should involve the development of the best user interface possible. Time spent designing and evaluating user interfaces can have significant benefits later on. The best interface will likely differ from language to language. Changes in the user interface can affect more than just the speed with which an annotator annotates data. Different user interfaces can lead the same annotator to different levels of accuracy. Multiple user interfaces should be proposed and then evaluated for both speed and accuracy.

7. Dealing with Human Annotation Error:

Until now we have assumed that human annotation reveals the true tag y. Unfortunately, human annotators are not one hundred percent accurate. Therefore a class/tag/annotation is never actually directly observed. Rather an annotator's opinion concerning the correct class is observed. An example graphical model that takes this into account for three different annotators is shown in Figure 6.

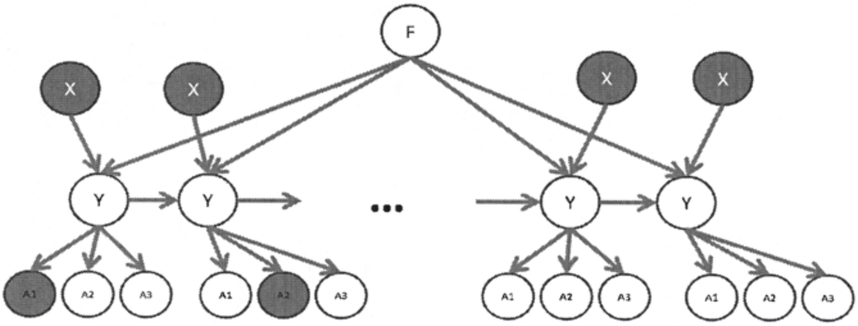

Figure 6: The annotator's annotations are observed.

These changes to the model will have implications for active learning. We are no longer interested in selecting the most important y node, but in selecting the most important annotation node $A_{a,i}$ (where a is the annotator involved and i is the instance to be annotated). For example, if annotator number 1 is willing to annotate another example then the goal of active learning would be to select from the $A_{1,i}$ nodes the node that provides the maximum improvement in expected utility.

The optimal solution will come from directly solving the EVSI equations on the annotation nodes of the model, but this is again too computationally intense and a heuristic is needed. Luckily, if we are using uncertainty to model the EVSI of sampling a given annotation, and if we assume that our uncertainty about an annotator's annotation is proportional to our model's uncertainty about the class y, then we can fall back to QBU on the y nodes. This means that the active learning technique can remain relatively unchanged and still accurately model the situation.

Although the changes to the model did not significantly change the active learning approach, they do significantly affect the machine learning technique. The effect of an annotator's annotation on the machine learner's belief about the class y should be directly modeled $p(A_a|y_i)$. Typically human annotations are used as training examples for the machine learner. This approach ignores the fact that there could be errors in the human annotations and relies on the machine learning algorithm's robustness to noise to compensate for this shortcoming. This can be problematic, especially when combined with active learning. This is because both the actual EVSI calculation and the QBU approximation are both dependent on the uncertainty of a given word after it has been annotated by a human annotator. Although the new model didn't change the active learning technique, it will change the values that the active learner will use to make decisions through changes in the machine learner. The uncertainty of an annotation after a human has already annotated it is an important piece of information, especially for determining if a second annotator should be used to validate the results of the first. Unless we can compute the probability of an error after an annotator has annotated a word, then the active learner cannot appropriately make this decision. This means that machine learning techniques that incorporate probabilistically annotated training data are necessary in order to take full advantage of active learning with multiple annotators.

Recent projects like Wikipedia and YouTube have illustrated the promise of user generated content on the web. Opening the corpus creation process to user involvement can be beneficial since it could increase the number of annotators available. Unfortunately, the quality of annotations obtained in this way can vary widely. In such situations average annotator accuracy is insufficient, and it is important to model each annotator's abilities separately. If we could spot a bad annotator, and appropriately adapt both our machine learner and active learning technique to his level of ability, then we could allow anyone to provide annotations with no fear that they could lower our overall accuracy.

The model in Figure 6 is only correct if we assume that all annotators are equally accurate in their annotation abilities, a simplifying assumption that may be reasonable for some applications. However, in order to model each annotator's abilities separately we need to model $p(A_{a,i}|y_i,c_a)$, as shown in Figure 7, where c_a represents the annotation abilities of annotator a.

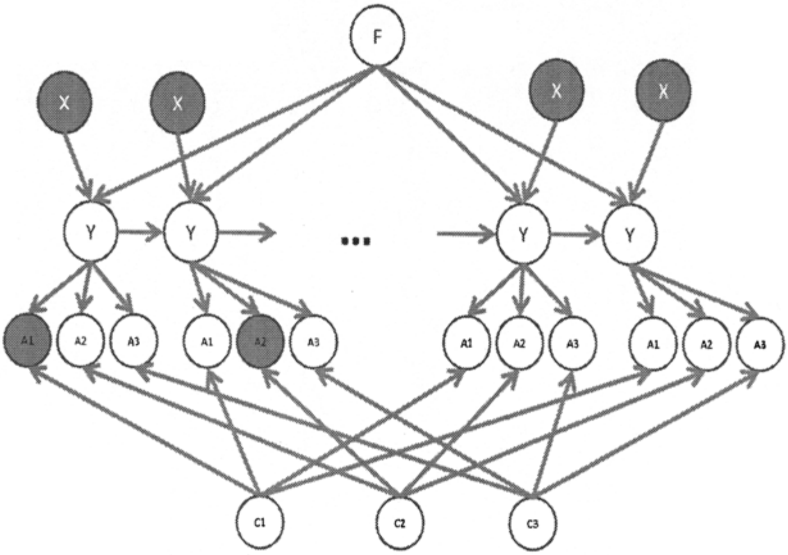

Figure 7: Modeling the accuracy of the annotators.

This more complex model also has implications for active learning. Now there are two possible reasons why a sample location could have value. Each annotation provides information about the true value of the class (and thus about the model f). Each annotation also provides information about c_a the annotation abilities of annotator a. Again the optimal solution could be found through the EVSI equations. Interestingly enough, this optimal solution could involve giving an annotator a problem with a well known solution because it teaches us about the annotator's annotation abilities. An active learning algorithm must balance the need for information about the annotator's abilities with the need for information about the class. These desires can sometimes be mutually exclusive since querying in locations where y is known with some high degree of certainty often gives more information about the quality of the annotator, while sampling in locations where y is unknown often gives more information about y and f. The principles of decision theory and EVSI will automatically balance these issues.

If EVSI in the simple network was intractable then this is far worse. Therefore a heuristic is again required. Luckily, it is possible to learn about both f and c simultaneously, and they are not always mutually exclusive. Even when we sample a location with high uncertainty we will still learn something about c. There are at least two reasons for this. First, the quality annotators will produce annotations that lower the entropy of the model. The model f imposes some belief about y, and if an annotator is consistently proposing annotations that are unlikely given the model f, then the most likely reason is that the annotator is making mistakes. This is especially true if all the other annotators are proposing annotations that are consistent with the model f. Secondly, sample locations that are unknown now will become more certain after more data is collected at which time they can provide more information about c.

This means that we can often fulfill both goals at once. Several possible approximate policies could be tried. For example, a subjective prior could be chosen to estimate the initial probability of annotator error. Then we could begin collecting annotations. Assuming that our prior is correct we could build a machine learning model (compute p(f)). Then assuming that the model is correct we could recompute the annotator's accuracy c. This process could be repeated in an EM like fashion. Once sufficient data is available for each annotator, then we could begin to model the abilities of each annotator separately. Infinitely more complex models could also be imagined. Some annotators could be better at certain tags and worse at others. Whether or not to take such factors into consideration will depend on the complexity of the model created and on the amount of available training data.

8. Consequences, Conclusions, and Future Work:

The purposes of this research has been to analyze the machine learning sequence annotation problem from the perspective of Bayesian utility and decision theory and to make suggestions based upon these observations that can be used to improve accuracy and decrease cost in the upcoming creation of an annotated Syriac corpus. It is hoped that many of these suggestions will be widely relevant in other annotated corpus creation projects. We will now present several suggestions based upon the observations made above concerning: how to deal with different parts of the corpus having different requirements of quality; building a user interface to minimize *ECSI*; performing a user study to assess/approximate *ECSI*; estimating the quality of an annotator; necessary modifications and enhancements of the machine learning algorithm itself; and selecting examples for active learning.

We propose the following technique for performing selection for active learning. ECSI should be first minimized using various user studies on several possible user interfaces. QBU or QBC can be used to approximate EVSI, and then ECSI can be approximated appropriately depending on the method used for paying annotators. If annotators are paid by the sentence or by the word, then ECSI can be computed directly. If they are paid by the hour, then this value can be approximated through user studies. There is some utility in giving an annotator a problem with a known solution; however, with enough data we will eventually learn the abilities of any annotator so specifically sampling for this purpose is less important. We propose that a good prior for the abilities of annotators can be selected subjectively. A few set questions with a known solution could be initially asked each annotator to refine this prior, but the number of such questions need not be large. Samples should be taken in locations with the highest *EBPB*. The abilities of each annotator can then be refined in an EM like fashion as detailed above.

In order to take advantage of these suggestions and observations the machine learning algorithms used will need to have several properties. Obviously simply solving the Bayesian network gives the theoretically optimal solution but will be computationally intense. Algorithms used as an approximation to this network will need to be able to report its uncertainty, preferably over the possible annotator's responses, but at least over the possible output tags y. This measure of uncertainty is essential to QBU. Next, an approximation will need to be able to deal with probabilistic training data. In other words it will need to be able to deal with training data that comes from an annotator with a known error rate and to be able to deal with different annotators each with different error rates. Our current MEMM machine learners report their uncertainty but currently do not deal with uncertain training data. Creating a learner that can handle probabilistic training data is an important part of our future work.

This work has been primarily theoretical and motivational. Its purpose has been to motivate the research that will follow. Many of the unsubstantiated claims of this paper will be validated in our upcoming publications. These future publications include: a publication on the details of the user study (in submission); experimental exploration of EBPB active learning (in submission); mathematical and experimental exploration of the connections between EVSI, QBU, and QBC (in preparation); and the eventual publication of the Syriac concordance and corpus itself. We refer the interested reader to these forthcoming publications for further experimental verification of these ideas.

References:

Buntine, Wray L. 1992. *A Theory of Learning Classification Rules.* A Dissertation Submitted to the School of Computing Science in the University of Technology, Sydney.

Carroll, James L. and Kevin D. Seppi. 2007. No-Free-Lunch and Bayesian Optimality. At: *Meta-Learning IJCNN Workshop.* `http://james.jlcarroll.net/publications/Bayesian%20NFL.pdf` (3.2.2008)

Carroll, James L. — Christopher K. Monson and Kevin D. Seppi. 2007. A Bayesian CMAC for High Assurance Supervised Learning. *Applications of Neural Networks in High-Assurance Systems, IJCNN Workshop.* `http://james.jlcarroll.net/publications/CMACHighAssurance.pdf` (3.2.2008)

de Freitas, Nando — Mahesan Niranjan — Andrew Gee and Arnaud Doucet. 1998. *Sequential monte carlo methods for optimisation of neural network models.* Technical Report TR-328, Cambridge University Engineering Department, Cambridge, England. `http://mi.eng.cam.ac.uk/reports/svr-ftp/auto-pdf/freitas_tr328.pdf` (3.2.2008)

Carvalho, Alexandre X. and Martin L. Puterman. 2003. *Dynamic Pricing and Learning Over Short Time Horizons.* Working Paper. University of British Columbia, Vancouver, BC, Canada.

Raiffa, H. and R. Schlaiffer. 1967. *Applied Statistical Decision Theory.* New York: Wiley-Interscience.

Ratnaparkhi, Adwait. 1996. A Maximum Entropy Model for Part-Of-Speech Tagging. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 133-142.

Ringger, Eric — Marc Carmen — Robbie Haertel — Noel Ellison — Kevin Seppi — Deryle Lonsdale — Peter McClanahan and James Carroll. 2008. Assessing the Costs of Machine-Assisted Corpus Annotation through a User Study. In submission, *Language Resources and Evaluation Conference, Marrakech.*

Ringger, Eric — Peter McClanahan — Robbie Haertel — George Busby — Marc Carmen — James Carroll — Kevin Seppi and Deryle Lonsdale. 2007. Active Learning for

Part-of-Speech Tagging: Accelerating Corpus Annotation. *The ACL 2007 Linguistic Annotation Workshop,* pp. 101-108.

Toutanova, Kristina — Dan Klein — Chris Manning and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In: *Proceedings of HLT-NAACL.* pp. 252-259. http://nlp.stanford.edu/pubs/tagging.pdf (3.2.2008)

Toutanova, K. and Manning, C. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In: *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pp. 63-70. http://nlp.stanford.edu/pubs/emnlp2000.psn (3.2.2008)

Wolpert, D.H. 2001. *The supervised learning no-free-lunch theorems.* Technical Report 269-1, NASA Ames Research Center. http://www.denizyuret.com/ref/wolpert/papers/37.pdf (3.2.2008).