ANYTIME ACTIVE LEARNING

DISSERTATION

BY

MARIA E. RAMIREZ LOAIZA

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science
in the Graduate College of the
Illinois Institute of Technology

Approved ____________________
Advisor

Chicago, Illinois
May 2016

ACKNOWLEDGMENT

First and foremost, I would like to thank my Ph.D. advisor, Mustafa Bilgic. In acknowledging people who were there during my graduate studies, it is easy to say “I am honored” or “I am fortunate” to have worked with them. When it comes to Mustafa, both of those sentiments are true in every sense of the word. When we started working together, I did not know much about research. I was thankful every day in my tenure at the lab that he decided to take me on as a researcher and tirelessly guided me through the process. He patiently mentored me and taught me about being a scientist and a researcher. He taught me to explore, to look for answers, to ask more questions, and more important, to enjoy the process. I am grateful for his support, encouragement, and mentorship; I feel honored to be one of his students.

I am also greatly thankful to my dissertation defense committee: Aron Culotta, Cynthia Hood, and Sonja Petrovic. They provided valuable feedback and insightful questions, forcing me to think about important issues. Especially, I am thankful to Aron for providing a different angle into the research problems, for all the interesting discussions and collaborations, and for providing valuable feedback to make this research better.

I want to thank the Machine Learning Lab members, Manali Sharma and Caner Komurlu, for all the useful discussions about research and life, for their invaluable friendship, and for the fun working environment.

Personally, I would like to thank my family for their undying support and love, without which I never would have been able to pursue this dream. Especially grateful for my parents, Beatriz and Ramiro, and my siblings, Sandra and Sebastian. I owe many thanks to my dear friends Luz Arabany, Amparo, and Gabriel for their support and faith in me. Last but not least, I am deeply indebted to Theresa for her endless encouragement and patience, always reminding me that I can do anything I set my mind to do. She helped me every step of the way.

I would also like to acknowledge the various funding sources that supported me throughout my graduate career. This work was supported in part by the National Science Foundation CAREER award no. IIS-1350337, by the Fundación para el Futuro de Colombia – COLFUTURO, and by the Ministerio de Tecnologías de la Información y las Comunicaciones de Colombia. Any opinions, conclusions, or recommendations expressed in this thesis are my own and do not necessarily reflect those of the sponsors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Symbol	Definition
$\mathcal{L}$	Labeled data set
$\mathcal{U}$	Unlabeled data set
$\mathcal{V}$	Validation data set
B	Budget, e.g., money or time
$\mathbf{x}_i$	Feature vector calculated from instance i
$\mathbf{x}_i^k$	Feature vector of subinstance k extracted from instance i. Represents interruption at time k.
s_i^k	Feature vector of snippet k extracted from instance i. Represents a condensed version of instance $\mathbf{x}_i$.
y_i	Target label or class label
$P_{\mathcal{L}}$	Probabilistic classifier induced on $\mathcal{L}$
$P(y\|\mathbf{x})$	Posterior probability of y given $\mathbf{x}$ according to classifier P
$P_E(y\|s_i^k)$	Posterior probability of y given snippet s_i^k according to classifier P_E, which model the expert

ABSTRACT

Machine learning is a subfield of artificial intelligence which deals with algorithms that can learn from data. These methods provide computers with the ability to learn from past data and make predictions for new data. A few examples of machine learning applications include automated document categorization, spam detection, speech recognition, face detection and recognition, language translation, and self-driving cars. A common scenario for machine learning is supervised learning where the algorithm analyzes known examples to train a model that can identify a concept. For instance, given example documents that are pre-annotated as personal, work, family, etc., a machine learning algorithm can be trained to automate organizing your documents folder. In order to train a model that makes as few mistakes as possible, the algorithm needs many training examples (e.g., documents and their categories). Obtaining these examples often involves consulting the human user/expert whose time is limited and valuable. Hence, the algorithm needs to utilize the human's time as efficiently as possible by focusing on the most cost-effective and informative examples that would make learning more efficient.

Active learning is a technique where the algorithm selects which examples would be most cost-effective and beneficial for consultation with the human. In a typical active learning setting, the algorithm simply chooses the examples that should be asked to the expert. In this thesis, we take this one step further: we observe that we can make even better use of the expert's time by showing not the full example but only the relevant pieces of it, so that the expert can focus on what is relevant and can provide the answer faster. For example, in document classification, the expert does not need to see the full document to categorize it; if the algorithm can show only the relevant snippet to the expert, the expert should be able to categorize the document much faster. However, automatically finding the relevant snippet is not a trivial task; showing an incorrect snippet can either hinder the expert's ability to provide an answer at all (if the snippet is irrelevant) or even cause the

expert to provide incorrect information (if the snippet is misleading). For this to work, the algorithm needs to find a snippet to show the expert, estimate how much time the expert will spend on that snippet, and predict if the expert will return an answer at all. Further, the algorithm would estimate the likelihood of the expert returning the correct answer. Similar to anytime algorithms that can find better solutions as they are given more time, we call the proposed set of methods *anytime active learning* where the experts are expected to give better answers as they are shown longer snippets.

In this thesis, we focus on three aspects of anytime active learning: i) anytime active learning with document truncation where the algorithm assumes that the first words, sentences, and paragraphs of the document are most informative and it has to decide on the snippet length, i.e., where to truncate the document, ii) given a document, the algorithm optimizes for both snippet location and length, and lastly, iii) the algorithm chooses not only the snippet location and size but also chooses which documents to choose snippets from so that the snippet length, the correctness of the expert's response, and the informativeness of the document are all optimized in a unified framework.

CHAPTER 1

INTRODUCTION

Since the invention of computing machines, we have looked for ways to use them in the automation of tasks such as mathematical calculations, repetitive activities, and complex pattern recognition tasks. *Machine learning* is a subfield of *artificial intelligence* that deals with methods, techniques, and algorithms to learn from data. These methods provide computers with the ability to learn from past data and make predictions for new data. Many applications have been developed based on machine learning methods such as spam detection, web search, speech recognition, and more recently, self-driving cars.

As an example, imagine we would like to automate the classification of documents into categories, and organize them into separate folders based on topics, such as `work` and `school` documents. If we were to enlist the help of an assistant, it is only logical to explain the task by using some example documents from each category. If our assistant has doubts about a particular document, he can ask us, the experts, for help. Similar to training a human assistant, *supervised learning* uses examples and their known labels to train a model or a *classifier*. We can use this classifier to analyze new documents (i.e., unseen document) and assign a category or *label*. The examples used to teach the classifier are also called training data or *labeled* data. These data consist of examples and their corresponding labels used to infer a classification function that, given an input example, will produce a label as an output. We will focus on this kind of classification task in this thesis.

For most supervised machine learning systems, we need as much labeled data as possible to obtain a classifier that makes as few mistakes as possible. The more complex the classification task, the more labeled data is needed to learn the corresponding classification function so as to have an acceptable performance. In some situations, we start with small number of available examples. From our example of document classification, our assistant in training has seen only a few document examples at the beginning, thus he is a novice. Intuitively, he will ask an expert for help if he finds a difficult document to classify, and he will gain new knowledge from the answer of the expert.

Similarly to the assistant training model, we can use an interactive approach to train a classifier under limited labeled data conditions. *Active learning* is a machine learning subfield that seeks to provide solutions for information gathering where labeled data are scarce but unlabeled data are abundant. For example, we can easily gather a number of scientific papers from the web but labeling them with categories requires an expert. In general, an active learning algorithm or *learner* carefully selects unlabeled examples and queries an expert (e.g., human annotator) for their labels, and incorporates the newly acquired labels into its training data. This interaction between learner and expert continues until a stopping criterion is met, typically when a budget in time is exhausted. In our running example of document classification, an active learner will iteratively select documents for annotation based on their potential learning benefit (e.g., how difficult it is to assign a category), and will query an expert for their category label until the expert is no longer available.

An immediate question is how to select the query examples so as to optimize our available budget, e.g., number of questions. There has been copious research on how to select the queries. We will discuss some techniques in Chapter 2. For a comprehensive review of active learning methods, please see Settles [73]. In general, active learning looks for unlabeled examples expected to help the learner the most within the budgetary conditions, and the benefit of an example is defined by the specific active learning method.

In many real-world scenarios we need a domain expert to provide the necessary labels for training (e.g., categorize documents), which can be costly and time consuming. For example, in image diagnostics, where we need a radiologist to identify RMI images [37, 54]; similarly, in citation screening we need a physician to identify relevant literature to a specific topic [94].

Active learning is a framework where labels for training are collected by a learner by asking a limited number of queries to an expert. Implicitly, reducing the number of queries to the expert reduces the overall annotation cost (e.g., time), if the cost of annotating examples is about the same for every query. However, some examples may be more complex than others and it may take longer to be revised and annotated. Furthermore, we have to consider that every query has a different benefit to the classifier, and has a different labeling cost that may depend on many factors, such as difficulty and complexity of the example. *Cost-sensitive active learning* refers to algorithms that reduce annotation cost while training the best classifier possible. Optimizing a learning budget accounting for all these factors is not trivial. Chapter 2 describes formally active learning approaches that address these challenges.

Now we describe in detail three scenarios of active learning where the learner uses the expert's time more efficiently by guiding her to the most relevant pieces of the example to answer faster.

1.1 Summary of Contributions

In this thesis, we describe the research of an active learning framework where the learner algorithm has the ability to affect the annotation cost and further optimize the annotation budget. The basic principle is to show the expert the relevant piece of information, estimate how much time the expert will spend on that piece of information, and estimate how accurate the expert will be on that piece of information. Similar to anytime algorithms, we call the proposed set of methods *anytime active learning*. The research on anytime active learning can be divided into three parts: anytime active learning with document truncation, faster annotation with active snippet selection, and anytime active learning with document-snippet selection.

1.1.1 Active Learning with Document Truncation. Consider the active learning interaction between an active learning algorithm and a human expert; let us assume the interaction occurs one document at a time, and the budget is the expert's available time. Figure 1.1 illustrates the cycle where a learner algorithm selects a document from the unlabeled data, requests the human expert to provide the label, and then incorporates the new answer into its training data, and the loop continues until the available time is finished. Note that the expert peruses the document and takes his time to make a decision on the label without intervention from the algorithm. Furthermore, the expert reads the document and gradually forms his opinion about the label as he reads more and more of the document.

Reading a document can be seen as a stream of words processed by a reader, e.g., the expert. As the expert reads more, he gradually forms an opinion about the label of documents. Furthermore, the expert may have a good degree of certainty about the label before reaching the last sentence, and stop reading further. Even though stopping early may produce some savings, the learner cannot directly control how much the ultimate cost would be per document. If we enhance the learner with the capability to interrupt the expert while he is reading, this may produce additional savings. Ideally, we would like the active learner algorithm to be able to control for how long documents are being analyzed and therefore spend the budget more efficiently and consciously. However, identifying the ideal time to spend per document is not a trivial task. In this research, we study the challenges of interrupting the expert during annotation in Chapter 3.

In many situations, like in our document classification example, we deal with tasks that involve a sequential analysis, such as reading a document. In this study, we will focus on text classification applications of active learning to develop the proposed methods and

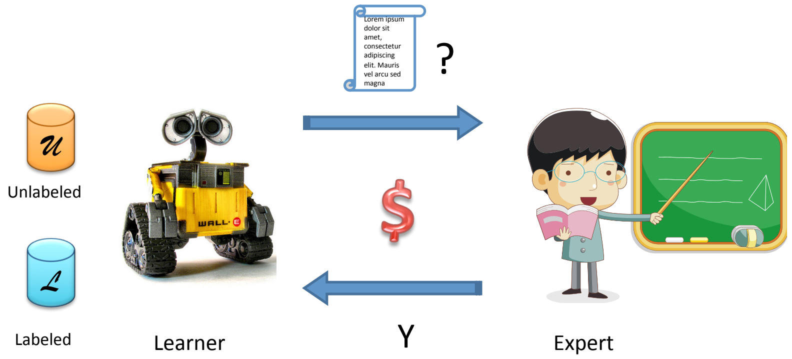

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris vel arcu sed magna
?
𝒰
Unlabeled
ℒ
Labeled
Learner
$
Y
Expert

Figure 1.1. Illustration of an active learning loop to annotate documents

present empirical evidence.

1.1.2 Faster Annotation with Active Snippet Selection. If we look further into to the annotation process, an expert can use a different approach to peruse the document and quickly analyze it. If we consider an experienced expert, he can read the document concentrating only on key sentences or words that indicate the corresponding label. For example, the expert can scan or skim through the document identifying key phrases related to labels. These key phrases are part of the expert's knowledge of the domain. When scanning, the expert is filtering the stream of words he is processing.

Moreover, by skimming through the document the expert is able to read more efficiently and provide labels at a faster rate. However, these savings in labeling are a secondary effect of how the expert behaves. If the active learner is able to learn what pieces of information are key for the annotation, the algorithm can present a snippet of the document that contains the key pieces, and show them to the expert instead to identify the label of the document in a shorter amount of time. However, identifying what to show the expert without inducing errors is challenging. In Chapter 4, we study how to extract key information to form the queries and improve learning efficiency.

1.1.3 Optimal Active Learning with Document-Snippet Selection. Ideally, we would like to build the best classifier possible by collecting the labels of the best examples at the least cost possible (e.g., limited time of expert). Choosing key snippets can improve the annotation time and hence the classifier can learn faster. We can improve this process even further by enabling the classifier to pick not only snippets from a given document, but also choose both an informative document and key snippets from that document simultaneously. We discuss the joint optimization of document-snippet pair selection in Chapter 5.

In this thesis, we will explore the three aforementioned anytime active learning scenarios with combinations of settings with increasing complexity considering document selection, type of interruption, and expert response quality. For document selection, we will consider whether the learner has a choice of instance, i.e., { active document selection, passive document selection}. For type of interruption we consider whether the snippets are from the beginning of the document or in an arbitrary location, i.e., {document truncation, document snippet}. For the expert response quality, we consider a fallible-reluctant expert (with various levels of noise), i.e., {reluctant-noisy expert, reluctant-perfect expert}.

The remainder of this thesis is organized as follows: we will discuss general concepts related to cost-sensitive active learning in Section 2. From these concepts, we will build three anytime active learning formulations with oracle interruption. In Chapter 3, we will discuss several scenarios of active learning with document truncation. In Chapter 4, we will discuss other methods of producing interruptions by generating snippets of documents. In Chapter 5, we formulate an anytime active learning method that is able to simultaneously choose documents and snippets to accelerate learning. Finally, in Chapter 6 we will provide final remarks and possible future research directions.

CHAPTER 2
BACKGROUND AND RELATED WORK

In this chapter, we introduce the common notation used in this document, provide background on active learning, and discuss related work.

Let $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ be a labeled set of input-output pairs instances where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector and $y_i \in \{y^0, y^1\}$ is its target class label.[1] Let $\mathcal{U} = \{\mathbf{x}_i\}_{i=l+1}^{m}$ be a set of unlabeled examples. Let $P_{\mathcal{L}}(y|\mathbf{x})$ be the conditional probability of y given $\mathbf{x}$ according to a classifier trained on the labeled set $\mathcal{L}$.

2.1 Supervised Learning

In supervised learning, a machine learning algorithm learns a function from labeled training examples $\mathcal{L}$. Typically, the training data consists of known input-output pairs $(\mathbf{x}_i, y_i)$ used to induce classifier $P_{\mathcal{L}}$. The goal is to use $P_{\mathcal{L}}$ to predict the label of unseen examples at the lowest generalization error possible. The generalization error is a measure of the accuracy of the classifier at predicting the target class of new examples.

For example, in text classification the input labeled instances are instances documents and their corresponding topic categories. In video classification, the input labeled instances are video sequences and labels of whether they are funny or not funny. In image diagnostics, the input examples are CT-scan images and the labels are whether or not there is a tumor.

Ideally, there are enough labeled examples to learn a good classifier $P_{\mathcal{L}}$. However, in many real-world cases, labeled examples are scarce whereas unlabeled ones are

[1] We assume binary classification for ease of presentation; this is not a fundamental limitation.

abundant, and annotation (e.g., human expert) is necessary to increase $\mathcal{L}$. For example, a video may be available on the web, however, obtaining labels such as `funny` requires a human-annotator. For specialized domains, human-annotation effort is considerably more expensive. For example in the medical domain, obtaining specialized images such as CT-scans and asking a medical professional to review the images and label whether there is a `tumor` requires valuable time and expertise.

2.2 Active Learning

Active learning is a strategy to iteratively build a labeled dataset to learn the best classifier possible with minimal supervision. The intuition is that carefully selecting unlabeled instances and requesting the label from an annotator will produce a better classifier $P_{\mathcal{L}}$ than selecting random instances for requesting the labels. Settles [73] provided detailed discussion on various active learning strategies and serves as an invaluable reference on active learning. Similarly, Fu et al. [33] presented a survey on active learning and discussed time complexity of various active learning approaches.

2.2.1 Active Learning Scenarios. There are three scenarios where active learning is typically applied [73] : i) membership query synthesis, ii) stream-based selective sampling, and iii) pool-based active learning. In membership query synthesis, we assume the learner knows a definition of the input space and it is able to construct hypothetical example instances which are shown to an expert for labeling [3, 4]. In stream-based selective sampling, the learner receives unlabeled instances from a stream of data, in an online fashion, and then decide whether to request the label from an expert [18, 19]. In pool-based active learning, a large collection of unlabeled instances, known as the *pool*, is available at once. In this setting the active learner evaluates the pool of example candidates and determines for which one to query the expert [48].

In general, regardless of the scenario, the idea is to score examples as to how much

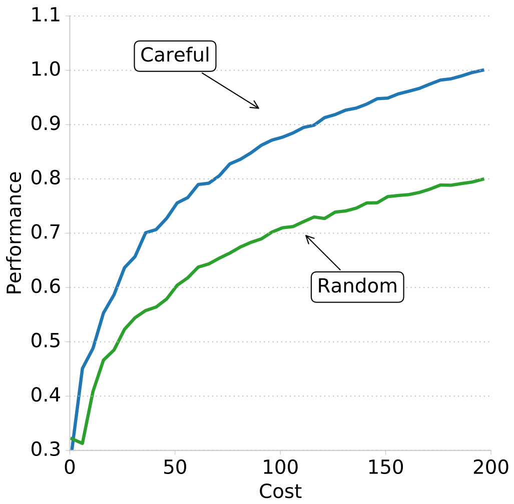


Figure 2.1. Performance of an active learning algorithm illustrated by a learning curve.

benefit they provide to improve the current model. This score is used to make the decision of whether it is worth to request the label for the candidate example or not. How to produce the score for each example depends on the active learning method, and in many cases, because exact computations are not feasible, the method provides heuristics to approximate the scores.

Typically, an active learning method is evaluated on the efficiency of learning with respect to the spent budget to gather information. Common performance measures are area under the ROC curve, accuracy, precision, recall, and F_1 score. The performance of an active learner classifier is visualized using learning curves of performance vs. budget. Figure 2.1 shows an example of a learning curve where the x-axis corresponds to the budget on annotation cost; the y-axis is the performance score of the learner, where the higher the performance the better. We expect that carefully selecting instances for annotation using an active learning approach performs better than simply selecting random instances. In this thesis, we focus on pool-based active learning, which we discuss next in greater detail.

2.2.2 Pool-based Active Learning. Pool-based active learning is inspired by situations where the available data are a small labeled set $\mathcal{L}$, and a large unlabeled set $\mathcal{U}$. In this

Algorithm 1 Pool-Based Active Learning

```
Input: Labeled data $\mathcal{L}$; Unlabeled data $\mathcal{U}$; Budget $B$; Classifier $P(y|\mathbf{x})$;
while $B > 0$ do
    $\mathbf{x}_i \leftarrow$ SelectInstance($\mathcal{U}$)
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\mathbf{x}_i\}$
    $a \leftarrow$ QueryOracle($\mathbf{x}_i$)
    $B \leftarrow B - C(\mathbf{x}_i)$
    $\mathcal{L} \leftarrow \mathcal{L} \cup (\mathbf{x}_i, a)$
    $P(y|\mathbf{x}) \leftarrow$ UpdateClassifier($\mathcal{L}, P$)
```

scenario, the active learning algorithm selects an unlabeled instance $\mathbf{x}_i \in \mathcal{U}$ iteratively and carefully, and obtains the resulting target value y_i by querying a human annotator (*oracle*). The algorithm incorporates the newly labeled example $(\mathbf{x}_i; y_i)$ into its training set $\mathcal{L}$, and retrains the underlying learning classifier, $P_{\mathcal{L}}$ (*learner*). This interactive cycle continues until a stopping criteria is met; usually the cycle stops when a querying budget is exhausted (e.g., time or money for annotation). Algorithm 1 illustrates a general active learning algorithm more formally.

The learner aims to choose the instance that will benefit the learning the most therefore minimizing the number of label queries necessary. We can see that a key component of this algorithm is how to select the query instances (SelectInstance). To this end, the learner tries to minimize the generalization error of the classifier, subject to a budget constraint (i.e. number of question we can ask the oracle). More formally, the optimization objective function of the learner is defined as follows:

$$\mathcal{U}^* \leftarrow \operatorname*{argmin}_{\mathcal{U}_i \subseteq \mathcal{U}} \overbrace{Err(P_{\mathcal{L} \,\cup\, \mathcal{U}_i}(y|\mathbf{x}))}^{\text{Generalization Error}} \text{ s.t. } \sum_{\mathbf{x}_j \in \mathcal{U}_i} C(\mathbf{x}_j) \leq B \tag{2.1}$$

where $C(\cdot)$ is the cost of annotating an example, and $Err(\cdot)$ is the generalization error defined as the expected loss of the classifier trained on $\mathcal{L}$. Note that $Err(\cdot)$ may be replaced by any function that defines utility or importance of an instance, in which case the objective maximizes such function, and the constraints still apply.

Equation 2.1 is typically optimized by greedy algorithms by choosing one or more most informative instances to label. There are numerous studies describing query selection strategies. Some common strategies aim to query instances at the decision boundary [48], query the instance that produces the most disagreement by a committee of classifiers (QBC) [32], query the instance that is expected to reduce the classification error the most [66]. A comprehensive description of active learning strategies can be found in [73].

2.2.3 Batch-mode Active Learning. Typically, these active learning methods (uncertainty sampling, QBC, etc.) use a greedy approach selecting the top scoring instances at each iteration. The issues with this approach is that, in some cases, the top examples are identical, and obtaining the labels increases the cost but not the benefit. An alternative is to construct the most informative batch of queries. Brinker [15] prosed an approach for support vector machines that explicitly incorporates diversity in query batch, and Xu et al. [99] selected examples closest to centroids of clusters near the decision boundary. Guo and Schuurmans [34] proposed a method for logistic regression as an optimization problem directly building the batch through gradient search.

2.3 Cost-sensitive Active Learning

To reach a target performance of the learner, we expect active learning to reduce the annotation effort compared to randomly selecting queries. A typical assumption is that all queries have the same cost, thus reducing the number of queries to the oracle is enough to increase the learning efficiency. However, a more realistic view of the annotation task is to consider variable cost of annotation. For example, labeling a difficult example may be worth money or require more review time. Under a variable cost condition, reducing the number of label queries does not guarantee a reduction in overall cost, and the objective is to select the most cost-effective instances. Several *cost-senstive* approaches address this problem [75, 23, 35, 40, 89].

One common approach is to formulate a decision-theoretic objective to incorporate directly the cost of annotation into the query selection criteria. The objective is to tradeoff the improvement of the classifier and the cost of acquiring a label. Let $Err(P_{\mathcal{L}})$ be the expected loss of the classifier trained on $\mathcal{L}$, the active learner algorithm selects the query $\mathbf{x}_i^*$ that most reduces the expected loss (e.g., how much the classifier improves its performance) of the underlying classifier, updating (2.1):

$$\mathbf{x}^* \leftarrow \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}} \overbrace{\left(\underbrace{Err(P_{\mathcal{L}})}_{CurrentError} - \underbrace{Err(P_{\mathcal{L}\cup(\mathbf{x}_i,y_j)})}_{FutureError} \right)}^{\text{Improvement}} - \lambda C(\mathbf{x}_i) \tag{2.2}$$

where $C(\mathbf{x}_i)$ is the cost of asking the oracle to provide a label for instance $\mathbf{x}_i$, and λ is a conversion parameter between annotation cost and expected error. Equation 2.2 is a look ahead method that compares the current performance with the future performance of the classifier if a label for $\mathbf{x}_i$ is obtained. This kind of formulation works best when $Err(\cdot)$ is synched with the measure used to test classifier $P_{\mathcal{L}}$. For example, if the classification task is better measured by accuracy, then Err may be best defined as accuracy.

2.3.1 Cost-benefit Active Learning Formulation. However, when a conversion parameter is not available, a common approach is to select the query with the highest cost-benefit ratio or *return of investment -ROI*. This then determines the best instance for annotation in terms of improvements per unit cost:

$$\mathbf{x}_i^* \leftarrow \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}} \frac{Err(P_{\mathcal{L}}) - Err(P_{\mathcal{L}\cup(\mathbf{x}_i,y_j)})}{C(\mathbf{x}_i)} \tag{2.3}$$

where $C(\cdot)$ is the cost of annotating an instance. This formulation is expressed in terms of loss; however, the formulation may also be defined in terms of performance. The advantage of cost-benefit ratio is it avoids the use of a conversion parameter, however, extremely small values of $C(\cdot)$ can produce an extremely high utility per unit. This extreme case of small costs may prevent the algorithm to identify the best instances for annotation.

Kapoor et al. [40] accounted for varied annotation per instance. The task was to classify voice messages. This decision-theoretic framework used an annotation cost function that assumes a simplifying linear relation between the length of a message and the cost of annotation. The proposed method trades off the annotation cost (e.g., money) and the cost of assigning a predicted label to an instance (e.g., expected error reduction). A difficulty of this method is to be able to map annotation cost and predicted label cost in the same unit. King et al. [42] used a similar approach with known fixed real costs, where a learning robot performed real lab tests to learn the labels from the results.

Settles et al. [75] proposed an active learning framework that simultaneously learns the real annotation cost (e.g., elapsed time of annotation). This method addressed situations when the annotation cost function is not known beforehand. To learn the annotation cost the authors used a regression cost model using meta-features for each instance, especially designed for this task. The features are domain dependent and defined by the user.

2.4 Common Active Learning Strategies

As mentioned before, a key element of active learning is how to select informative instances; for example, how to define the error function. Typically, $Err(\cdot)$ is defined as a utility value function that represents how much benefit is expected from annotating an instance. A simple approach is to find the decision boundary using a procedure similar to a binary search. For example, initially we can guess where the boundary is and query instances on either side. We can repeat this as we narrow the search with the obtained labels.

2.4.1 Uncertainty Sampling. Uncertainty sampling uses a similar intuition querying instances lying near the decision boundary, where the learner classifier is most uncertain [48]. The intuition is that the decision boundary will be in a region of high uncertainty of the classifier. Furthermore, the instances closer to the decision boundary will provide more information; therefore, those instances will have the most utility for learning. Figure 2.2 shows the degrees of confidence (from blue to red) of a classifier trained on $\mathcal{L}$ and the uncertain region is visible near $X_2 = 0$.

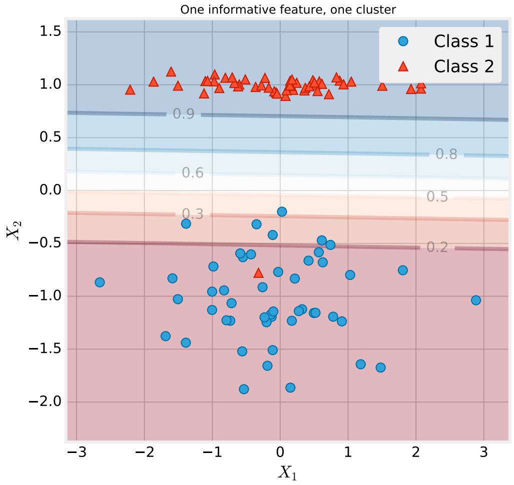


Figure 2.2. Toy example of uncertainty of classifier $P_{\mathcal{L}}$. This is a toy binary dataset, *red* triangles and *blue* circles. The most uncertain region of classifier $P_{\mathcal{L}}(y = blue|\mathbf{x})$ is near the decision boundary

More formally, uncertainty sampling queries the instances whose predicted posterior probability is the least confident, updating Equation 2.3 as follows:

$$\mathbf{x}^* \leftarrow \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}} \frac{1 - \max_{y \in Y} P_{\mathcal{L}}(y|\mathbf{x}_i)}{C(\mathbf{x}_i)} \tag{2.4}$$

Equation 2.4 uses conditional error as a measure of confidence. Another common definition of uncertainty is entropy of the classifer [77]:

$$\mathbf{x}^* \leftarrow \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}} - \frac{\sum_{y \in Y} P_{\mathcal{L}}(y|\mathbf{x}_i) \log(P_{\mathcal{L}}(y|\mathbf{x}_i))}{C(\mathbf{x}_i)} \tag{2.5}$$

Uncertainty sampling uses an intuitive selection criteria that makes active learning easy to implement. Many methods leverage the least confident strategy for information extraction [20, 74]. However, the method has some performance issues in applied settings due to intrinsic active learning bias, complexity of the internal dependencies in the instances, or task complexity [70, 95, 88].

Alternative definitions of uncertainty may be more effective for classification problems [78]. In this work, the uncertainty is divided into two types based on the cause of uncertainty: strong conflicting evidence of a label, and weak inconclusive evidence of a label. This method showed empirically to be more effective than traditional definitions of uncertainty. Furthermore, this type of uncertainty definitions allows the active learning algorithm to provide a reason along with the query on why the example is useful for the current model [79]. This transparent approach allows the learner to gather additional information useful to fine tune the final model.

Overall, uncertainty sampling works well and has been successfully used in several papers and domains, even though it is known to be susceptible to label noise and instance outliers [e.g., 66, 74]. Successful applications include [11, 100, 36, 86, 71, 72, 91, 37, 16, 74, 79, 62, 78], among many others.

2.4.2 Expected Error Reduction. In general, an optimal active learning algorithm will request the label for an instance that is expected to improve the current model the most [66]. In the absence of ground truth, the learning algorithm computes an expectation of the classifier performance should the label of instance $\mathbf{x} \in \mathcal{U}$ be added to training data. The following objective allows the learning algorithm to select the instance that is expected to reduce the generalization error the most:

$$\mathbf{x}^* \leftarrow \underset{\mathbf{x}_i \in \mathcal{U}}{\operatorname{argmax}} Err(P_{\mathcal{L}}) - \sum_{y_j} P_{\mathcal{L}}(y_j|x_i^k) Err(P_{\mathcal{L} \cup (x_i, y_j)}) \tag{2.6}$$

where $Err(P_{\mathcal{L}})$ is the expected loss of a classifier trained on $\mathcal{L}$. This function is also known as the generalization error, profit function, utility function, etc. Equation 2.6 is also known as *value of information*. This method is expected to perform well when the loss function Err is well aligned with the performance function used to evaluate the classifier. Note that this method computes the expected future loss for every candidate in the unlabeled data $\mathcal{U}$ for every possible label y.

2.4.3 Query by Committee. Query by committee (QBC) is an active learning strategy that queries the label of instances for which a committee of classifier disagrees the most [76]. Committee classifiers often perform better than single classifiers [22]. Typically, the committee is formed by sampling the training data (bagging) and inducing the classifiers [1, 14] or using an *AdaBoost* algorithm [31, 30]. QBC has shown to reduce that prediction error exponentially on the number of queries asked [76, 32, 31]. However, this method can be computationally expensive in order to build the committee of classifiers. There are various measures to determine the disagreement of the committee. Two common approaches are margin of disagreement, i.e. the difference between number of votes for the top most popular labels [58], and vote entropy [21].Vote entropy is defined as follows:

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathcal{U}}{\operatorname{argmax}} - \sum_{y \in Y} \frac{V(y)}{C} \log \frac{V(y)}{C} \tag{2.7}$$

where y ranges over all possible labels in Y, $V(y)$ is the number of votes that a label receives from the committee members, and C is the committee size. When the target class y is binary, both margin and vote entropy approaches rank instances in the same order.

2.4.4 Other Active Learning Approaches and Applications. Other active learning approaches aim to improve the underlying model by requesting missing features in the data. Melville et al. [59] proposed a method that acquires values for missing features when dealing with a classification task. This method tries to improve the general classification performance by improving the quality of the training data without overspending. The method selects the feature values that are expected produce the highest improvement of accuracy. Active feature-value acquisition has many challenges and the topic has been studied extensively [67, 81].

Applying techniques such as dimensionality reduction within an active learning algorithm should improve learning efficiency but it is not a trivial task. Bilgic [9] recently proposed an adaptive dimensionality reduction technique that determines the number of dimensions at every cycle of interaction. The method uses labeled and unlabeled data to learn more accurate models.

There are a number of extensive active learning empirical studies that focus on specific tasks or improvements over typical active learning. Settles and Craven [74] compared several variations of uncertainty sampling and query-by-committee for sequence labeling. The study concentrated on $F1$ score as the performance measure and used conditional random fields [46] as the underlying model. Schein and Ungar [69] evaluated a number of uncertainty-based approaches and query-by-committee for logistic regression, and focused only on accuracy. In other cases, the studies of active learning used for domain-specific tasks, such as natural language processing [90, 5], splog detection [41], text segmentation [68], image retrieval [91], sequence labeling [74], outlier detection [91, 2], class imbalance problem [107, 27], etc.

A related area is active inference. In active inference, unlike active learning, the labels are collected not during training time but rather at inference/prediction time. The objective is to use the labels of few instances to do a better job at predicting the remaining ones. For this to work, the labels of the instances need to be correlated. For example, [64, 10] looked at classification of nodes in a network, [43] looked at battery optimization in sensor networks by selectively choosing which sensors should communicate, and [44] looked at data collection for tissue engineering experiments.

2.5 Reluctant and Fallible Experts

Typically, the active learner interacts with an experienced domain expert. The assumption is that when an experienced oracle provides an answer to a query, his answer is likely to be correct with a high probability. Otherwise, the expert may refuse to answer given his own uncertainty. However, this may not be always the case in a real scenario. For example, long and continuous periods of annotation may be tiring for a human oracle, and cause the oracle to make mistakes; the oracle may not be infallible and provide incorrect answers to some queries; the oracle may be a novice and be less confident in his answers. Under these realistic conditions of variable annotation cost, and imperfect oracle, optimizing the information gathering is not a trivial task.

Donmez and Carbonell [23] proposed a method called *proactive learning* that relaxes typical active learning constraints related to oracle properties such cardinality, reliability and responsiveness. This work uses a decision theoretic approach under three main scenarios each focusing on a single characteristic of the oracle. However, the main objective in all cases is to select the right unlabeled instance for labeling and the right oracle. Whenever necessary, the learning process is a two-part strategy where a prior exploratory phase is used to learn the reluctancy or accuracy of the oracles. In the first scenario, the active learner has two available oracles with known annotation cost, one reluctant and one reliable. The active learner incorporates into its objective the probability of receiving an answer from the reluctant oracle. In the second scenario the two oracles are reliable, however one of them is fallible because it may provide an incorrect answer. Finally, the authors explore an active learning configuration with two reliable oracles, however, with different variable cost of annotation. The authors provide a framework that allows to consider reliability and responsiveness of the oracles with an active learning setting relaxing traditional assumptions regarding the oracle conditions. This work is usually treated as a baseline for other multi-oracle active learning frameworks.

Zheng et al. [106] developed an active learning method for crowdsourcing scenarios (i.e., multiple labelers) under conditions of varied costs and accuracy of the oracles. The proposed method works as a two part approach: first, the method ranks the labelers and selects the top ones to serve as a subset of optimal cost and high accuracy. This subset is used to cast a majority vote for labeling; second, the learner exploits the pre-selected set of labelers to query uncertain instances.

Wallace et al. [96] studied active learning settings where there are several oracles available with different levels of expertise. In this work, the query assignments are determined by the oracle known expertise. In addition, the learner balances oracles workload to guarantee maximum utilization of resources. There are two main aspects of this work: modeling the oracles workload distribution, so that an oracle is not overworked; and defining a cost-effective query strategy that relies on a novice oracle to identify difficult queries (i.e., meta-cognitive ability). This method is tested on two tasks: sentiment analysis with reliable oracles to model the expert, and biomedical citation where the expert is a reluctant oracle. The proposed method is compared against a proactive approach.

Fang et al. [29] selected queries that are useful for learning but that will be likely known to the oracle. This work provides a way to model the concepts known to the oracle by keeping track of the answer (including rejected queries) from the oracle. The queries are selected by maximizing the entropy of the learner model with respect to the unlabeled instances and the knowledge (or lack there of) of the oracle.

Du and Ling [25] proposed a two part framework where the oracle is noisy, and the noise depends on the instance. An exploratory phase of the method uses uncertainty sampling to select query instances, penalizing the sampling as the learner acquires more labels. The premise is that at early iteration of active learning the classifier acts as a novice and the uncertain instances are likely to be answer by the oracle. However, as the learner reaches better performance levels, the uncertain instances are increasingly more difficult to label even for the oracle. An exploitative phase of the method decides what instances from the labeled data to re-query in order to clean up the set as a validation step. Both phases are traded off with a probability α predetermined by the user depending on what is valued the most. This method does not consider the difference in annotation cost per instance.

Other applications of active learning use variable cost in their formulation based on the specific domain. In computer vision, Vijayanarasimhan and Grauman [93] presented a method where the learning algorithm determines the 'net cost' of labeling and balances the gain associated with requesting said label. The annotation cost is computed considering image complexity, and the time necessary to produce the segmentation of the image. Similarly, Liu et al. [51] formulated an active learning framework on spatial data where label acquisition costs are proportional to distance traveled.

2.6 Document Summarization

For faster annotation through snippets, our method searches for the snippet that represents best the label of a document, similar to document summarization where the methods synthesize text. Automatic document summarization is divided into two groups: single-document summarization and multiple-document summarization. In this work, the most relevant research refers to single-document summarization; thus we will concentrate on discussing this topic.

Typically, a summarization algorithm uses sentences as units, although larger passages can be used. For each unit, the algorithm extracts a set of features typically normalized and produces a score value. The units are sorted by their score and the highest ranking ones are used to compose a summary or extract. The key to this methods is the scoring function that determines the relevance of the sentence to the extract of the document. In early literature, methods used position in text [e.g. 7], word and phrase frequency [e.g., 52], and key phrases [e.g., 26]. Other methods were developed using machine learning techniques based on bayesian classifiers [e.g., 45, 47], and methods that use more complex approaches based on natural language processing [e.g., 6], among other methods.

Some studies use summarized documents for training classification models without risk of losing performance [80], or for reducing the effort of producing reference summaries by increasing agreement among multiple human annotators [105]. In sentiment classification, a combination of meta-features is defined to extract sentences that summarize the sentiment of reviews [8], or aspect-based sentiment [87].

To the best of our knowledge, the current related work does not consider added capabilities of the learning algorithm to control annotation time. That is, the learning algorithms do not determine how much time to spend on annotation for each query.

2.7 Active Document-Snippet Pair Selection

We first introduced the idea of actively selecting document and snippets of documents to accelerate annotation in [61]. To enhance an active learning model with snippet selection capabilities, we should consider repeating labeling for cleaning training data, noisy labels, snippet building techniques, and short-text classification.

Relabeling allows the learning algorithm to request labels of instances already in the training data, and through a voting heuristic determines the true label of the re-labeled examples. In anytime active learning, we expect the learner to make better selection of documents and snippets as it acquires more labels and improves its performance. However, document-snippet pairs selected early in the active learning loop may be revised through re-labeling to correct possible mistakes and improve the current classifier.

Sheng et al. [82] proposed repeating labeling of the training set using various methods: round robing with general cost, and selective relabeling based on uncertainty of the label. Du and Ling [25] used a trade-off parameter between exploration and relabeling phases of the learning algorithm. In a recent study, Lin et al. [49] argued that relabeling is not an effective strategy in many cases, but provides evidence of cases where relabeling is beneficial. The main considerations for relabeling are type of classifier (expressive classifier with weak inductive bias, linear classifier with large number of features), the accuracy of the labelers (moderate accuracy), and labeling budget (relabeling is more effective with small budgets).

Furthermore, in realistic active learning setting the experts are fallible and may return noisy labels. Moreover, the quality of annotation depends on the difficulty of the query and the annotator's level of expertise. The gold standard for the training data is estimated by combining annotation quality and difficulty [39]; by probabilistic inference of labels for image annotation [97]; by modeling expertise and label dependencies [101].

For specific domains such as sentiment analysis, some studies model the relationships between the examples and labels through specific representations. For example, Mcdonald et al. [57] used a structured model on a sentiment analysis task. The model learns and infers the sentiment at different levels of granularity in the document. This model learns and infers sentence-level sentiments and document-level overall sentiment, where there is propagation from one level to the other. The dependencies between observed sentence-level labels and the document label are included in the model. Wilson et al. [98] presented a phrase-level sentiment analysis that first determines the neutrality of an expression, and determines the polarity of the non-neutral expressions. The neutral classifier model used 28 features for classification. To determine the sentiment of a non-neutral expression, this work presented a 10-feature classifier to disambiguate the sentiment. Both classification models used a classifier with word token features, and work token and prior (from sentiment lexicon) as baselines.

A number of research efforts use engineered domain specific features to improve short-text classification performance [83]. In contrast, Bobicev and Sokolova [13] used a prediction by partial matching method that does not require feature engineering. This method considered text as a sequence of characters instead of words.

Other short-text classification methods are based on semantic analysis; latent semantic analysis (LSA) is used to extract the potential semantic structure of the text. Chen et al. [17] proposed a method that extracts topics at multiple granularities, modeling short-text better. Among short-text classification method LDA is also used, as well as semi-supervised approaches especially to provide background context to short text [104]. Similarly, Sun [84] used important words as query to retrieved documents from a labeled set, and then uses the top k document labels to predict the label of the query (e.g., top five document labels and majority vote). Other methods use shortened text to request labels and build models specifically to collect labels. However, those models are not used for classification of unseen documents but rather to guide the active selection [61, 62].

In general, these research studies address active learning scenarios where fallible reluctant oracles are available for labeling. The methods select the most cost-effective instances for learning and the best oracle for annotation. However, the cost of annotation is determined by the oracle and depends on the instance. In this research, we propose a method to further increase the annotation efficiency by interrupting the oracle during annotation. In contrast to related work, and to the best of our knowledge, our proposed method is the first to allow the active learner to determine how much to spend on each query. Table 2.1 shows a summary of main challenges addressed by previous research compared to our proposed method *anytime active learning*.

Table 2.1. Comparing anytime time active learning to previous research considering annotation cost (known/unknown), oracle response (fallible/reluctant), and query shown to the oracle (truncated/snippet).

Research Work	Cost		Oracle		Query	
	Kn.	Unkn.	Fallible	Reluctant	Trunc.	Snippet
Kapoor et al. [40]	✓					
Settles et al. [75]		✓				
Donmez and Carbonell [23]	✓		✓	✓		
Wallace et al. [96]	✓		✓	✓		
Zheng et al. [106]	✓		✓			
Fang et al. [29]			✓			
Du and Ling [25]			✓			
Anytime active learning	✓		✓	✓	✓(Ch.3)	✓(Ch.4,5)

CHAPTER 3
ANYTIME ACTIVE LEARNING THROUGH TRUNCATION

In this chapter, we will describe an *anytime active learning* (AAL) framework when the learner is able to interrupt the oracle by truncating instances. We will discuss research results on the performance of the proposed methods and the corresponding baselines. This research is divided into two approaches of increasing complexity: static and dynamics AAL methods. The AAL concepts discussed here are applicable to a number of domains where the oracle analysis is incremental, i.e., the more time the annotator spends reviewing an instance the more quality is expected from his answer. Examples of such domains are text classification and video annotation to name two. In this chapter, we will use a text classification task for ease of explaining and representing our methods.

3.1 Introduction

As we discussed in Section 2, active learning seeks to maximize classifier accuracy while minimizing the effort of human annotators [73]. This is typically done by prioritizing example annotation according to the utility to the classifier (see Section 2.2.2). In this chapter, we begin with the simple observation that in many domains human annotators form an opinion about the label of an example incrementally as they review the instance. For example, while reading a document, an annotator makes a more informed decision about the topic assignment as each word is read. Similarly, in video classification the annotator becomes more certain of the class label the longer she watches the video.

The question we ask is whether we can train a classifier more efficiently by interrupting the annotator to ask for a label, rather than waiting until the annotator has completed her inspection fully. For example, in document classification the active learner may request the label after the annotator has read the first 50 words of the document. For video classification, the active learner may decide to show only a short clip. We refer to this approach as *anytime active learning* (AAL), by analogy to anytime algorithms, whose execution may be interrupted at any time to provide an answer.

If the decision of when to interrupt the annotator is made optimally, we can expect to reduce total annotation effort by eliminating unnecessary inspection time that does not affect the returned label. However, the annotator may not be able to provide a label or may return an incorrect label if interrupted too early — e.g., the annotator will not know how to label a document or will be unsure of the answer after seeing only the first word. AAL strategies, then, must balance two competing objectives: (1) the time spent annotating an instance (*annotation cost*); (2) the likelihood that the annotator will be able to produce a non-neutral label, and further, a correct label when noisy (*annotation response rate*). In this chapter, we propose and evaluate a number of anytime active learning strategies applied to the domain of document classification. In this domain, it is natural to implement this approach by revealing only the first k words to the annotator, which we refer to as a *subinstance*.

We build on our earlier work where we performed experiments with a simulated oracle on two document classification tasks [62], comparing two classes of anytime active learning strategies: (1) *static* strategies select subinstances of a fixed size; (2) *dynamic* strategies select subinstances of varying sizes, optimizing cost and response rate simultaneously. We tested these active learning strategies under a *reluctant-perfect* scenario where the annotator can refuse to answer when he is in doubt; but when he answers, the labels are guaranteed to be correct. Our additional contributions in this chapter include a set of experiments investigating how anytime active learning performs under a *reluctant-noisy* scenario, in which the annotator might return incorrect labels (in addition to refusing to answer a query). We provide a variant of our original objective function for this scenario, and provide insights into how the method may be adapted to different levels of label noise.

Throughout this chapter, we address the following questions and provide answers:

RQ1. How does subinstance size affect human annotation time and response rate? We conducted a user study in which each user labeled 480 documents from two domains under different interruption conditions (e.g., seeing only the first k words). We find that as subinstance sizes increase, both response rates and annotation times increase (non-linearly), and that the rate of increase varies by dataset.

RQ2. How do static AAL strategies compare with traditional active learning? We find that simple static strategies result in significantly more efficient learning, even with few words shown per document. For example, with an annotation budget of one hour, labeling only the first 25 words of each document reduces classification error by 17% compared with labeling the first 100 words of each document.

RQ3. How do dynamic AAL strategies compare with static strategies when there is a reluctant-perfect oracle (reliable)? The drawback of the static strategy is that we must select a subinstance size ahead of time; however, we find that the optimal size varies by dataset. Instead, we formulate a *dynamic* AAL algorithm to minimize cost while maximizing response rate. We find that this dynamic approach performs as well or better than the best static strategy, without the need for additional tuning.

RQ4. How do dynamic AAL strategies compare with static strategies when there is a reluctant-noisy oracle (unreliable)? When the oracle is prone to provide incorrect labels, one can reduce annotation noise by presenting longer subinstances. We find that the optimal size varies not only by dataset but also by oracle. We add an additional parameter to our formulation to allow one to fine-tune the tradeoff between annotation quality and cost. The resulting AAL-α algorithm performs as well as or better than the best static strategy when there are adequate estimates of the oracle accuracy.

The remainder of this chapter is organized as follows: we first formalize the anytime active learning problem, then propose static and dynamic solutions. Next, we describe our user studies and how they inform our simulation experiments. Finally, we present the empirical results and discuss their implications with perfect and noisy oracles.

3.2 Anytime Active Learning (AAL)

In this section, we formulate our proposed anytime active learning as an extension of standard active learning, and describe the concept of oracle interruption.

3.2.1 Oracle Interruption. We propose an alternative formulation of the active learning problem in which the student has the added capability of *interrupting* the human oracle to request a label while the annotation of $\mathbf{x}_i$ is being performed. For example, in video classification, the student may request a label after the oracle has spent only one minute watching the video. Similarly, in document classification, the student may request a label after the oracle has read only the first ten words of a document.

Let $\mathbf{x}_i^k$ indicate this abbreviated instance, which we call a *subinstance*. The nature of subinstances will vary by domain. For example, k could indicate the time allotted to inspect the instance. In this document, we focus on document classification, where it is natural to let $\mathbf{x}_i^k$ be the feature vector derived from the first k words of document $\mathbf{x}_i$.

The potential savings from this approach arises from the assumption that $C(\mathbf{x}_i^k) < C(\mathbf{x}_i)$; that is, subinstances are less costly to label than full instances. While the magnitude of these savings are data-dependent, our user studies below show substantial savings for document classification.

The immediate problem with this approach is that $\mathbf{x}_i^k$ may be considerably more difficult for the oracle to label. We therefore must account for imperfect oracles [23, 102]. There are at least two scenarios to consider — (1) a *reluctant-perfect* oracle may decide not to produce a label for some examples, but labels that are produced are assumed to be correct; (2) a *reluctant-noisy* oracle may decide not to produce a label for some examples, and labels produced may also be incorrect. In this chapter, first we concentrate our attention on the reluctant-perfect oracle, and we formulate the problem and algorithms under this assumption in Section 3.2, Section 3.3, and Section 3.4; then we introduce the reluctant-noisy oracle and adapt the formulation to reflect this noisy condition in Section 3.5.

In each interaction between the student and oracle, the student presents a subinstance $\mathbf{x}_i^k$ to the oracle, and the oracle returns an answer $a \in \{y^0, y^1, n\}$, where the answer can be either label y or neutral, n, which represents an "I don't know" answer. If the oracle returns a non-neutral answer a for $\mathbf{x}_i^k$, the student adds $\mathbf{x}_i$ and the returned label $(y^0$ or $y^1)$ to its training data, and finally updates its classifier. If n is returned, the labeled data is unchanged. In either case, the annotation cost $C(\mathbf{x}_i^k)$ is deducted from the student's budget because the oracle spends time inspecting $\mathbf{x}_i^k$ even if she returns a neutral label. To choose the optimal subinstance, the student must consider both the cost of the subinstance as well as the likelihood that a non-neutral or incorrect label will be returned. Below, we propose two AAL strategies.

3.2.2 Static AAL Strategies. We first consider a simple, static approach to AAL that decides *a priori* on a fixed subinstance size k. For example, the student fixes $k = 10$ and presents the oracle subinstances $\mathbf{x}_i^{10}$, which is the feature vector derived from the first 10 words in document i (please see Algorithm 2).

Let $\mathcal{U}^k = \{x_i^k\}_{i=l+1}^m$ be the set of all unlabeled subinstances of fixed size k. In SELECTSUBINSTANCE (line 3), the student picks $\mathbf{x}_i^{k*}$ as follows:

$$\mathbf{x}_i^{k*} \leftarrow \operatorname*{argmax}_{\mathbf{x}_i^k \in \mathcal{U}^k} \frac{U(\mathbf{x}_i)}{C(\mathbf{x}_i^k)} \tag{3.1}$$

Note that the utility is computed from the full instance $\mathbf{x}_i$, not the subinstance, since even though the oracle inspects only $\mathbf{x}_i^k$, the oracle is asked to label $\mathbf{x}_i$ and therefore $\mathbf{x}_i$ will be added to the labeled set $\mathcal{L}$ (line 8). In our experiments, we consider two utility functions: uncertainty (STATIC-K-UNC), which sets $U(\mathbf{x}_i^k) = 1 - \max_y P_{\mathcal{L}}(y|\mathbf{x}_i)$, and

Algorithm 2 Static Anytime Active Learning

```
1: Input: Labeled data $\mathcal{L}$; Unlabeled data $\mathcal{U}$; Budget $B$; Classifier $P(y|\mathbf{x})$; Subinstance size $k$
2: while $B > 0$ do
3:     $\mathbf{x}_i^k \leftarrow$ SelectSubinstance($\mathcal{U}$, $k$)
4:     $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\mathbf{x}_i\}$
5:     $a \leftarrow$ QueryOracle($\mathbf{x}_i^k$)
6:     $B \leftarrow B - C(\mathbf{x}_i^k)$
7:     if $a \neq n$ then    // Non-neutral response
8:         $\mathcal{L} \leftarrow \mathcal{L} \cup (\mathbf{x}_i, a)$
9:         $P(y|\mathbf{x}) \leftarrow$ UpdateClassifier($\mathcal{L}$, $P$)
```

constant (STATIC-K-CONST), which sets the utility of each subinstance to one, $U(\mathbf{x}_i^k) = 1$. We use STATIC-K-CONST as a baseline for other AAL methods because it is an anytime version of random sampling.

3.2.3 Dynamic AAL Strategies. The static strategy ignores the impact that k has on the likelihood of obtaining a neutral label from the oracle. In this section, we propose a dynamic strategy that models this probability directly and uses it to guide subinstance selection (see Algorithm 3).

Let $Q(z|\mathbf{x}_i^k)$ be the probability distribution that models whether the oracle will return an "I don't know" answer (i.e. a neutral label) for the subinstance $\mathbf{x}_i^k$, where $z \in \{n, \neg n\}$. The objective of SelectSubinstance (line 3) is to select the subinstance that maximizes utility and the probability of obtaining a non-neutral label, $\neg n$, while minimizing cost:

$$\mathbf{x}_i^{k*} \leftarrow \operatorname*{argmax}_{\mathbf{x}_i^k \in \mathcal{U}^k \in \mathcal{S}} \frac{U(\mathbf{x}_i)Q(z = \neg n|\mathbf{x}_i^k)}{C(\mathbf{x}_i^k)} \tag{3.2}$$

In contrast to the static approach, where the algorithm searches over a predetermined set

Algorithm 3 Dynamic Anytime Active Learning

```
Input: Labeled data $\mathcal{L}$; Unlabeled data $\mathcal{U}$; Budget $B$; Classifier $P(y|\mathbf{x})$; Neutrality classifier $Q(z|\mathbf{x}_i^k)$; Neutrality labeled data $\mathcal{L}^z \leftarrow \emptyset$.
while $B > 0$ do
    $\mathbf{x}_i^k \leftarrow$ SelectSubinstance($\mathcal{U}$)
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\mathbf{x}_i\}$
    $a \leftarrow$ QueryOracle($\mathbf{x}_i^k$)
    $B = B - C(\mathbf{x}_i^k)$
    if $a \neq n$ then    // Non-neutral response
        $\mathcal{L} \leftarrow \mathcal{L} \cup (\mathbf{x}_i, a)$
        $P(y|\mathbf{x}) \leftarrow$ UpdateClassifier($\mathcal{L}, P$)
    $\mathcal{L}^z \leftarrow \mathcal{L}^z \cup (\mathbf{x}_i^k,$ IsNeutral$(a))$
    $Q(z|\mathbf{x}_i^k) \leftarrow$ UpdateClassifier($\mathcal{L}^z, Q$)
```

of substances of size k (e.g., $\mathcal{U}^k$), the dynamic algorithm searches over an expanded set $\mathcal{S}$ of size p, that contains p different subinstance sizes: $\mathcal{S} = \{\mathcal{U}^{k_1} \ldots \mathcal{U}^{k_p}\}$. We can illustrate the search space use by static and dynamic methods as a matrix where each row i allocates document x_i and all its derived subinstances x_i^k in ascending order of size. For example, if each subinstance is built with increments of 10 words, the diagram illustrates the idea:

$$\left[\begin{array}{ccccccc}
 & & & \text{Fixed} & & & \text{Full} \\
x_1^{10} & x_1^{20} & \dots & x_1^{k} & \dots & x_1^{100} & x_1 \\
x_2^{10} & x_2^{20} & \dots & x_2^{k} & \dots & x_2^{100} & x_2 \\
 & & \vdots & & \vdots & & \\
x_i^{10} & x_i^{20} & \dots & x_i^{k} & \dots & x_i^{100} & x_i \\
 & & \vdots & & \vdots & & \\
x_n^{10} & x_n^{20} & \dots & x_n^{k} & \dots & x_n^{100} & x_n
\end{array}\right]$$

where x_i^k truncates document $\mathbf{x}_i$ after the k-th word.

The immediate question is how to estimate $Q(z|\mathbf{x}_k^i)$. We propose a supervised learning approach using the previous interactions with the oracle as labeled examples. That is, we maintain an auxiliary binary labeled dataset $\mathcal{L}^z$ containing $(\mathbf{x}_i^k, z_i)$ pairs (line 10), indicating whether subinstance $\mathbf{x}_i^k$ received a neutral label or not.[2] This dataset is used to train the neutrality classifier $Q(z|\mathbf{x}_k^i)$ (line 11). Algorithm 3 outlines this approach, where ISNEUTRAL maps the oracle answer to z (i.e., n or $\neg n$). As in the static strategy, we consider two settings of the utility function: uncertainty (DYNAMIC-UNC) and constant (DYNAMIC-CONST). While both DYNAMIC-UNC and DYNAMIC-CONST consider the cost of annotation, DYNAMIC-UNC balances utility with the chance of receiving a non-neutral label, while DYNAMIC-CONST simply maximizes the chance of a non-neutral label.

Even though the formulation in Equation 3.2 uses predetermined subinstances sizes, it does not lose generality. For example, consider the option of estimating the next size to be tested instead of picking a predefined size. Estimating the next subinstance size is equivalent to considering all possible sizes, i.e., considering all possible values of k in $\mathbf{x}_i^k \in \mathcal{U}^k \in \mathcal{S}$. For example in a text classification task, we can define k as a variable in

[2] Mazzoni et al. [55] use a similar approach to identify "irrelevant" examples.

the discrete range $[1, K]$, thus the algorithm will search over all possible sizes from 1 to K. However, in practice predetermined values based on domain knowledge allows for a smaller search space and fewer computations.

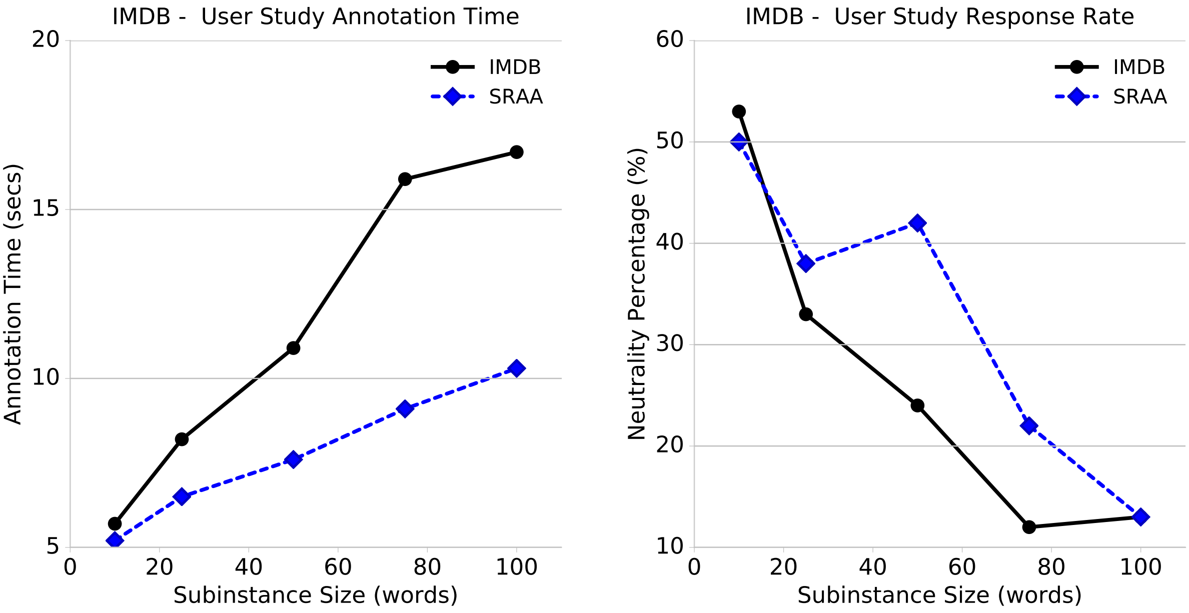
Furthermore, with enough domain knowledge, $\mathcal{S}$ may be built as a flexible set of subinstances equivalent to a set of subinstances by a predetermined criterion other than size. For example, for a document one may consider interrupting the oracle after the appearance of a significant word (e.g., highly weighted feature by the underlying classifier). However, similar to size, one should determine how many significant words to include per interruption. We opted for a simple interruption representation given by the number of words included in the text, also allowing to measure cost of annotation.

3.3 Experimental Evaluation

We used two datasets: (1) **IMDB:** A collection of 50K movie reviews from IMDB.com labeled with positive or negative sentiment [53]; (2) **SRAA:** A collection of 73K Usenet articles labeled as related to aviation or auto documents [60].

3.3.1 User Studies. To estimate the real-world relationships among subinstance size, annotation time, and response rate, we first performed several user studies in which subjects were shown document subinstances of varying sizes and asked to provide a correct label or an “I don’t know” answer, i.e., a reluctant-perfect scenario.

Each user performed six classification tasks per dataset, labeling document subinstances of sizes $\{10, 25, 50, 75, 100, \text{All}\}$. For example, to create the 50-word task, we truncated the documents to the first 50 words. For each classification task (IMDB and SRAA), the users were asked to annotate 20 randomly-chosen documents from each class, resulting in 40 annotations per task. The documents were presented to the users in random order. For every subinstance, we recorded the annotation time, the number of words seen, and the label. We used the average over five users on the IMDB and three users on the



(a) Annotation time

(b) Response rate

Figure 3.1. User study results reporting average annotation time in seconds and percent of neutral labels by subinstance size.

SRAA data.

Figure 3.1 shows the average annotation time (in seconds) and average percentage of neutral labels returned for each subinstance size. We find that the annotation time varies by subinstance size and dataset. For instance, in IMDB annotation time of subinstances of size 50 is 25% greater than for subinstances of size 25. These responses are influenced by the user experience and domain knowledge familiarity, among other factors.

Intuitively, the annotator will be more likely to provide a non-neutral label when he can see a larger part of document. This intuition was confirmed by our user studies. Figure 3.1 shows that the percentage of neutral labels decreases as subinstance size increases. However, the rate at which the neutral answer decreases differs by dataset. For example, there was an approximately 50% neutral rate on both datasets for subinstances with 10 words; yet for 75 words the neutral responses were 12% on IMDB and 22% on SRAA. We speculate that the SRAA dataset is a more specialized domain, whereas classifying movie reviews (IMDB) is easier for non-expert human annotators.

3.3.2 Simulations. We use the results of the user study to inform our large-scale studies on the two datasets.

3.3.2.1 Oracle. In order to compare many AAL strategies at scale, it is necessary to simulate the actions of the human annotators. Specifically, we must simulate for which examples the annotator will return a neutral label. We wanted to better reflect the fact that the lexical content of each subinstance influences the reluctance of the oracle — e.g., if a subinstance has strong sentiment words it is not likely to be labeled neutral. To accomplish this, we trained two oracles (one per dataset) that mimic the human annotators. We simulated the oracle with a classifier trained on held-out data; a neutral label is returned when the class posterior probability for a subinstance $\mathbf{x}_i^k$ is below a specified threshold. We tune this classifier so that the pattern of neutral labels matches that observed in the user study.

At the start of each experiment we fit a logistic regression classifier on a held-out labeled dataset (25K examples for IMDB; 36K for SRAA). We use L_1 regularization controlled by penalty C to encourage sparsity. When the oracle is asked to label a subinstance $\mathbf{x}_i^k$, we compute the posterior probability with respect to this classifier and compute oracle's uncertainty on $\mathbf{x}_i^k$ as $1 - \max_y P(y|\mathbf{x}_i^k)$. If the uncertainty is greater than a specified threshold T, then the oracle returns a neutral label. Otherwise, the true label is returned (a reluctant-perfect oracle scenario).

For each of the datasets, we set C and T so that the distribution of neutral labels by subinstance size most closely matches the results of the user study. We searched values $C \in [0.001, 3]$ with 0.001 step and $T \in [0.3, 0.45]$ with 0.05 step, selecting $C = 0.3, T = 0.4$ for IMDB and $C = 0.01, T = 0.3$ for SRAA. Figures 3.1a and 3.1b show the simulated distribution of neutral labels by subinstance size over the same documents from the user study, indicating a close match with human behavior.

To simulate the cost of each annotation, we used a fixed cost equal to the average annotation time from the user study for subinstances of that size — e.g., for all subinstances of size 10 for IMDB, the cost is the average annotation time for all subinstances of size 10 in the IMDB user study. In future work, we will consider modeling annotation time as a function of the lexical content of the subinstance.

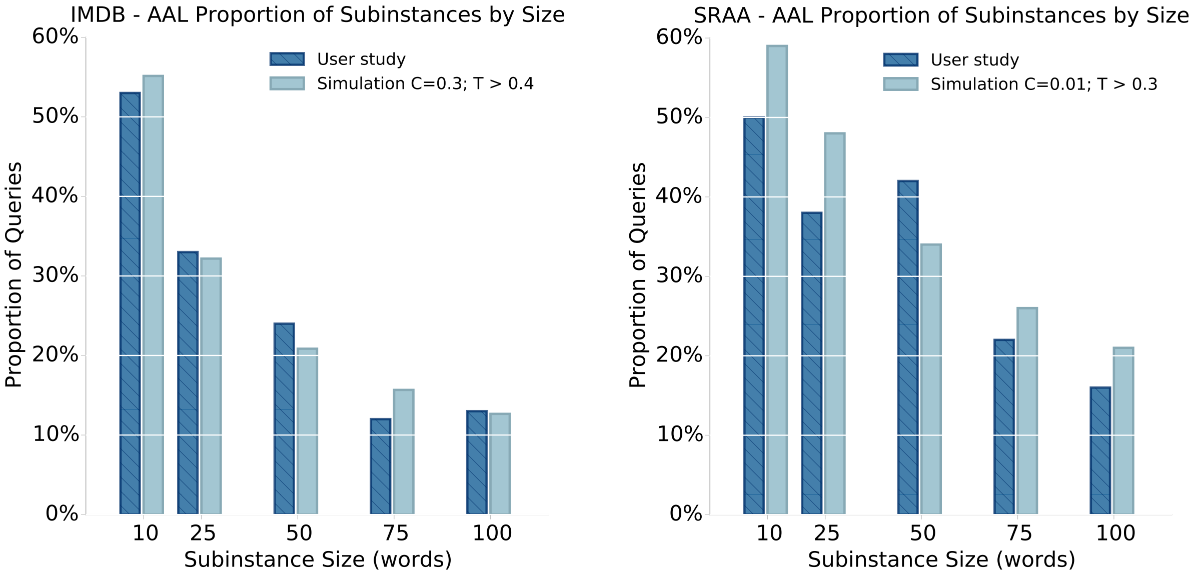
3.3.2.2 Student. For the student, we use a logistic regression classifier with L_1 regularization using the default parameter $C = 1$, seeded with a labeled set of two examples. At each round of active learning, a subsample of 250 examples is selected uniformly from the unlabeled set $\mathcal{U}$. Following the user study, subinstances of sizes $\{10, 25, 50, 75, 100\}$ are created for each example in the subsample and scored according to the appropriate strategy (Equation 3.1 for static; Equation 3.2 for dynamic). We reserve half of the data for testing, and use the remaining to simulate active learning. For all methods, we report the average result of 10 trials.

For both datasets, we use documents that contain at least 100 words. We created binary feature representations of the documents, using stemmed n-grams (sizes one to three), pruning n-grams appearing in fewer than five documents. In SRAA, we filtered header information, preserving only the subject line and body of the messages.

3.4 Results and Discussion

With an oracle simulation and annotation cost in place, we explored the performance of several AAL strategies. We examined learning curves for accuracy and area under the ROC curve (AUC) and observed the same trends and behaviors for each; therefore we include only AUC results here.

In this section, we discuss results of experiments performed under the scenario where the labels obtained by the student are correct; however, the oracle is allowed to reject a query (reluctant-perfect). In the subsequent section, we additionally consider an



(a) IMDB

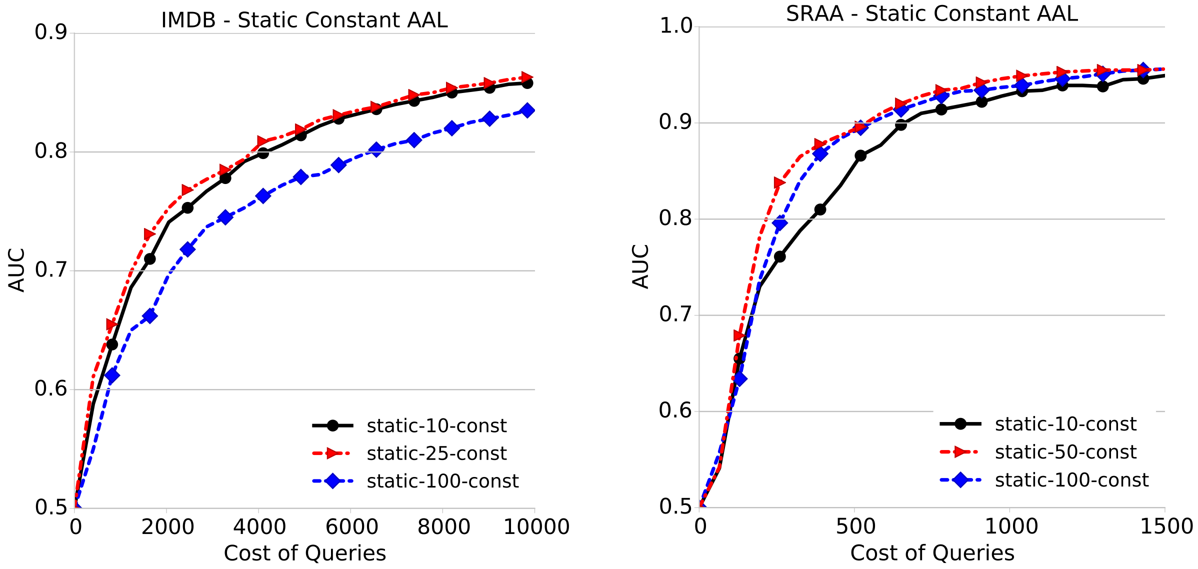
(b) SRAA

Figure 3.2. A comparison of the proportion of neutral responses by subinstance size for the user studies and the simulated oracles.

oracle that can return incorrect labels (reluctant-noisy).

3.4.1 Smaller subinstances generally outperform larger subinstances. Figure 3.3 shows the performance of STATIC-K-CONST for IMDB and SRAA datasets. These results consistently show that savings can be achieved by selecting smaller subinstances. For example, after an hour of annotation (3600 seconds) on IMDB, inspecting the first 100 words of each document results in an AUC of 0.75; whereas inspecting only the first 25 words results in an AUC of 0.79. This suggests that while a smaller k results in a high neutral percentage, the time saved by reading shorter documents more than makes up for the losses. The results for STATIC-K-UNC are similar but are omitted to avoid redundancy.

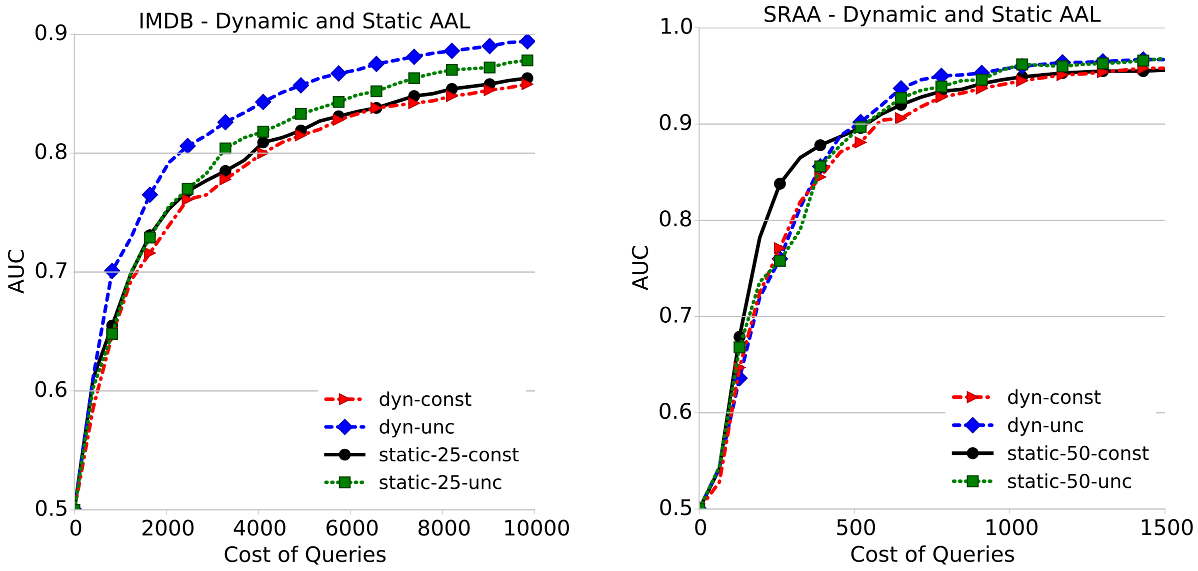
3.4.2 The optimal subinstance size varies by dataset. Comparing Figure 3.3a to Figure 3.3b indicates that the optimal k^* varies by dataset ($k^* = 25$ for IMDB, $k^* = 50$ for SRAA). This follows from the observed differences between these datasets in the user study (Figure 3.1); i.e., the annotation cost rises more slowly with subinstance size in SRAA.



(a) IMDB

(b) SRAA

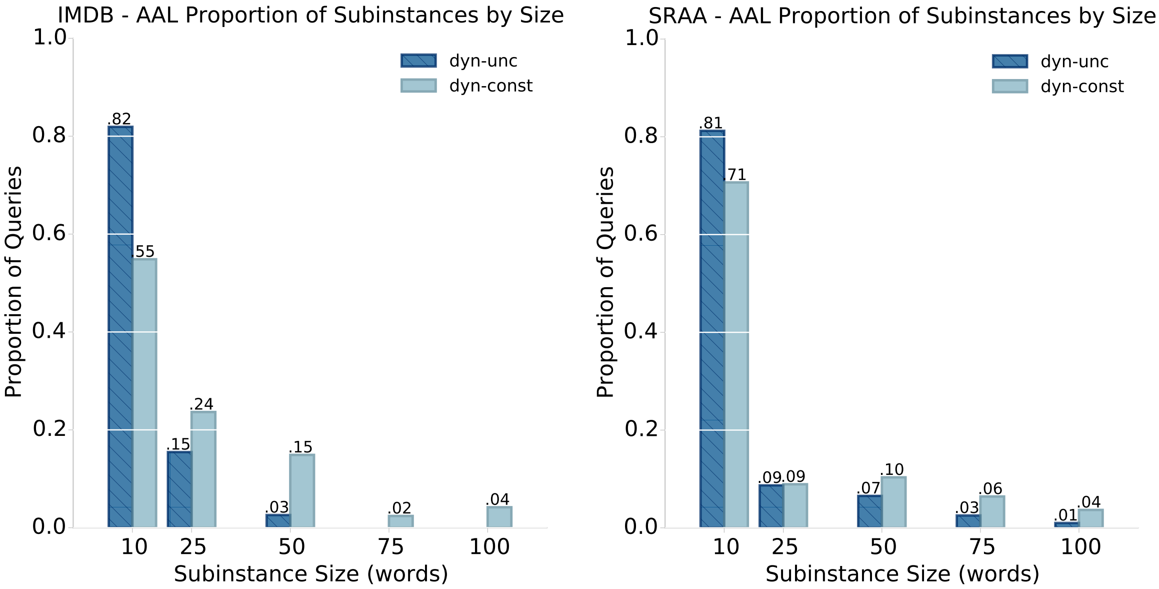
Figure 3.3. AUC learning curves for STATIC-K-CONST. The trends for STATIC-K-UNC are similar. The optimal k^* depends on the domain. For IMDB, $k^* = 25$ while for SRAA, $k^* = 50$.



(a) IMDB

(b) SRAA

Figure 3.4. Comparing dynamic AAL to the best of the static AAL approaches. DYNAMIC-UNC outperforms all methods for IMDB; whereas it is comparable to the best static approaches for SRAA.



(a) IMDB

(b) SRAA

Figure 3.5. Proportion of subinstance sizes selected by dynamic AAL.

Table 3.1. The percentage of observed neutral labels for DYNAMIC-UNC and DYNAMIC-CONST, compared with what is expected for subinstances of the observed sizes.

		Observed	**Expected**
IMDB	CONST	38%	48%
	UNC	36%	45%
SRAA	CONST	36%	50%
	UNC	39%	45%

Thus, somewhat bigger subinstances are worth the small additional cost to reduce the likelihood of a neutral label.

3.4.3 DYNAMIC-UNC does better than, or equal to, the best static AAL algorithm.

Given the fact that the optimal subinstance size varies by dataset, we examine how the dynamic approach compares to the static approach. Figure 3.4 compares the dynamic approach with uncertainty and constant utility (DYNAMIC-UNC, DYNAMIC-CONST) with the best static methods. We find that the DYNAMIC-UNC outperforms the best static method for the IMDB dataset (Figure 3.4a). For the SRAA datasets, in Figure 3.4b, the static and dynamic approaches are comparable; however, the advantage of DYNAMIC-UNC is that there is no need to specify k ahead of time.

3.4.4 Dynamic approaches tend to pick a mixture of subinstance sizes. To better understand the behavior of the dynamic approaches, Figure 3.5 plots the distribution of subinstance sizes selected by both approaches. As we can see, the dynamic approaches select a mixture of subinstance sizes, but heavily favor smaller sizes. Combining this observation with the results that DYNAMIC-UNC is either able to outperform or perform comparable to static approaches, this suggests that the dynamic approach is able to pick small subinstances that receive non-neutral labels.

Table 3.1 further investigates this by comparing the proportion of neutral labels observed with the expected proportion based on the user study. That is, by combining the data from Figure 3.1 and Figure 3.5, we compute the proportion of neutral labels we expect to see for the observed distribution of subinstance sizes. We can see that the neutrality classifier $Q(z|\mathbf{x}_i^k)$ enables the dynamic approach to select small subinstances while limiting the impact of neutral labels.

3.5 Reluctant-Noisy Oracle

The results in the previous section assume that all non-neutral labels returned by the oracle are correct. In this section we consider an alternate setting in which the oracle may provide incorrect labels. Thus, the *reluctant-noisy oracle* may produce one of three responses: neutral (no label), a correct label, or an incorrect label. This setting may better reflect cases in which non-expert annotators are used, perhaps to reduce cost.

The reluctant-noisy oracle setting clearly poses additional challenges to the student, as the noisy labels will likely decrease classification accuracy. Thus, in this setting the tradeoff between annotation cost and quality becomes even more critical for anytime active learning. Whereas a neutral label incurs cost with no value, an incorrect label can both incur cost and degrade the model.

To accommodate this setting, we make a simple modification to our Dynamic AAL objective function (Equation 3.2): we introduce a real-valued meta-parameter (α) that modulates the relative importance of annotation quality and query cost. The resulting objective function becomes:

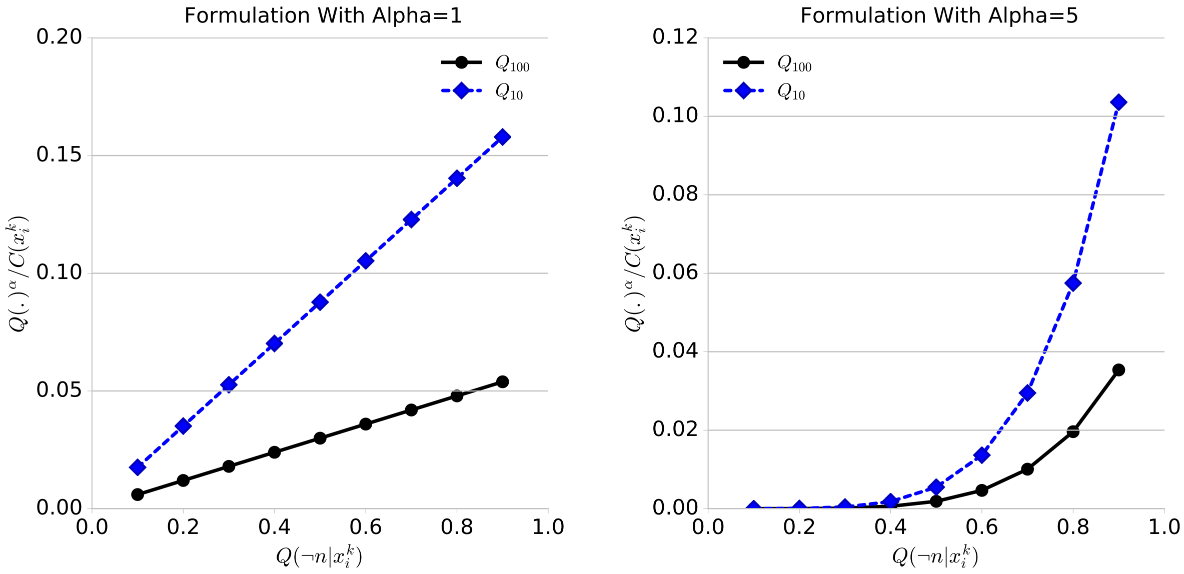
$$\mathbf{x}_i^{k*} \leftarrow \underset{\mathbf{x}_i^k \in \mathcal{U}^k \in \mathcal{S}}{\operatorname{argmax}} \frac{\big(U(\mathbf{x}_i) \times Q(z = \neg n|\mathbf{x}_i^k)\big)^{\alpha}}{C(\mathbf{x}_i^k)} \tag{3.3}$$

where $U(\cdot)$, Q, $\mathbf{x}_i^k$, z, and $C(\cdot)$ are the same as they were defined in earlier sections. In this new formulation, the student implicitly makes the natural assumption that the higher the student's estimate of oracle's confidence, i.e., the higher the $Q(z = \neg n|\mathbf{x}_i^k)$, the less likely the oracle is to make a mistake in the returned label, whether it be positive or negative. We call the set of anytime active learning methods that use this formulation AAL-α.

A noisy oracle poses additional challenges for the student because the noisy labels will likely decrease the student performance. We propose a simple solution in Equation 3.3 to address the noisy labels. In a real-world setting, when the oracle is presented with a subinstance of 10 words, he is more likely to provide an incorrect label than when presented with a 100-word subinstance of the same document; however, because of the annotation cost, the student is more likely to pick a short 10-word subinstance. For example, consider two subinstances of 10 and 100 words identified as $\mathbf{x}_i^{10}$ and $\mathbf{x}_i^{100}$ respectively. To pick $\mathbf{x}_i^{100}$, the student Q model has to be highly confident on the non-neutrality of the 100-word subinstance, following this inequality $Q(\neg n|\mathbf{x}_i^{100}) > \frac{C(\mathbf{x}_i^{100}) \times Q(\neg n|\mathbf{x}_i^{10})}{C(\mathbf{x}_i^{10})}$. For example on the IMDB data, $Q(\neg n|\mathbf{x}_i^{100})$ should be at least $3 \times Q(\neg n|\mathbf{x}_i^{10})$ so that the student picks the longer subinstance. However, it is easy to see that if the student is slightly confident about $\mathbf{x}_i^{10}$, the student will not pick a 100-word subsintace even with full confidence of obtaining a label. Figure 3.6 shows the effect of α in modulating the utility, Q_k refers to $Q(\neg n|\mathbf{x}_i^k)$. Note that in Figure 3.6b Q_{100} and Q_{10} are comparable at lower values of Q_{10}, thus giving chance of longer subinstances to be picked. In contrast, in Figure 3.6a only high values for

Q_{100} are comparable to Q_{10} lower values.

We tested various functions to modulate the importance of annotation. In particular, when considering two subinstances $\mathbf{x}^{10}$ and $\mathbf{x}^{100}$, considering a large subinstance should be more likely when confidence Q is low and even small improvements are valuable, as compared to considering a large instance when the confidence on a label is already high for a subinstance with only 10 words. An exponential function showed this desirable, although, other functions may also have a similar shape.



(a) $\alpha = 1$

(b) $\alpha = 5$

Figure 3.6. Example of effect of α parameter in modulating Q probabilities. α allows larger subinstances to be consider along smaller less expensive subinstances.

We perform experiments under two configurations: one in which the student has access to the true Q function, which we denote as Q^*, and the usual setting where the student has to learn Q from the data. We fit Q using the same approach as in the reluctant-perfect setting: the student keeps an auxiliary training set $\mathcal{L}^z$ containing (x_i^k, z_i) pairs indicating whether subinstance x_i^k received an answer or not. We assume that queries that the oracle refuses to answer would be incorrect if a label were provided. In general, the formulation is flexible and one may use other domain knowledge to replace Q with any model of oracle accuracy.

3.5.1 Simulating a Reluctant-Noisy Oracle. We extend the oracle simulation described in Section 3.3 to accommodate noisy labels. To better reflect human oracles, we assume that the probability of the oracle returning a correct label is proportional to its confidence in the label. This assumption is supported by recent empirical evidence that human oracles are aware when they are likely to make a mistake [96]. The choice of whether a query receives a correct or incorrect label is determined by the outcome of a Bernoulli trial in which the probability of a correct label is equal to the oracle's confidence. More formally, for a subinstance query $\mathbf{x}_i^k$ and neutrality threshold T, a response is sampled as follows:

- Compute the oracle's *confidence* $c = \max_y P(y|\mathbf{x}_i^k)$.
- **if** $1 - c > T$:
    - **return** a *neutral* label
- **else**: // Return a non-neutral label
    - $p \sim$ Bernoulli(c)
    - **if** $p == 1$, return a correct label. **Else**, return an incorrect label

To ensure a fair comparison between methods, the oracle's response for a particular subinstance is fixed across all baselines. As before, the oracle classifier is trained on held-out data, and its regularization parameter C is set to match the user studies, as described in Section 3.3.

In the experiments below, we additionally investigate how the student performs under varying levels of oracle noise. To vary oracle noise, we reduce confidence c by a constant factor in the sampling procedure above to reduce the probability of success in the Bernoulli trial. We consider settings of this factor that vary the overall percentage of incorrect labels between 0% and 20%.

3.5.2 Experimental Results. We performed active learning experiments with a reluctant-noisy oracle considering randomly sampled documents, where each method sees the same sequence of documents, and each document has a constant utility. We tested values of $\alpha \in \{0.5, 1, 3, 5, 10, 25, 50, 100\}$. Methods that use the estimated Q are referred to as DYN-q-CONST-α, and methods that use the true function Q are referred to as DYN-q^*-CONST-α. Furthermore, we will use α^* to denote methods that use the empirically optimal value of α for a given dataset (where performance is measured by area under the learning curve). We compare the AAL-α strategies with the best performing static strategies for each dataset (STATIC-K-CONST). The main conclusions of these experiments are as follows:

3.5.2.1 When Q* is Known, AAL-α Outperforms the Best Performing STATIC-K-CONST. Figure 3.7 shows the performance of DYN-q^*-CONST-α with a known model Q^* and STATIC-K-CONST. On the noise levels tested, DYN-q^*-CONST-α was able to outperform the best performing STATIC-K-CONST, except for SRAA where at 20% noise the performances are comparable. Compared to students with Q, Q^* students perform much better mainly because the estimates of the oracle correctness are better, and thus, the quality of the labels used. In general, the performance of the student highly depends on the quality of the estimates of Q^* available to the student.

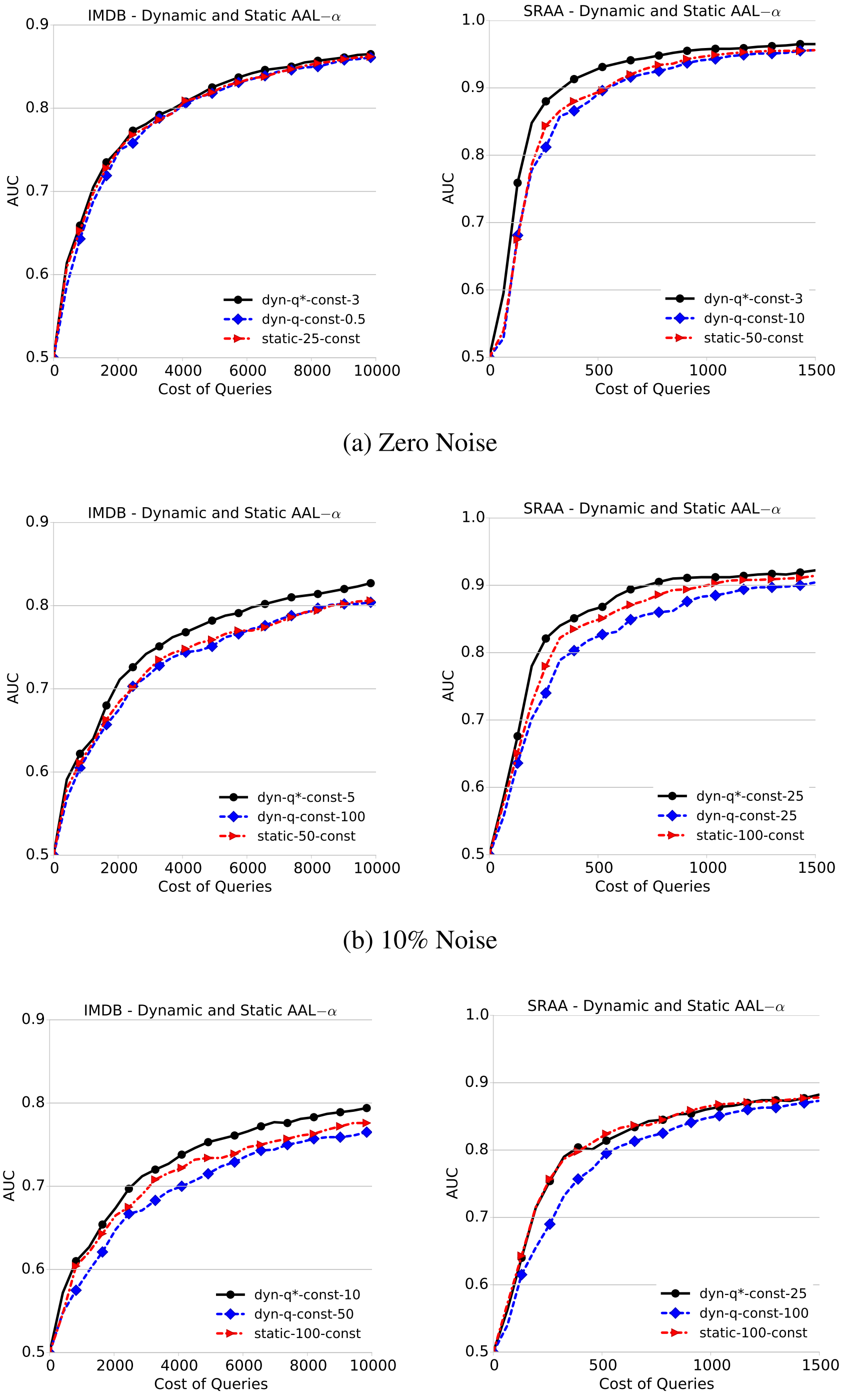
There are several supervised and unsupervised methods available in the literature where Q^* is estimated for use in active learning algorithms. Du and Ling [25] used the underlying model class posterior as a proxy for Q^* and penalize the estimates proportional to the size of $\mathcal{L}$, considering that as the student improves, so do the estimates of Q. Donmez et al. [24] used small subsample of unlabeled queries to request labels from various oracles, infer the true labels of the queries, and compute the estimated oracle accuracy. Other strategies use domain knowledge, such the method used by [96] where they used salary information of the oracles as an indication of their expertise, thus their reliability.

3.5.2.2 When Q is Learned, AAL-α Performance is Comparable to the Best Performing STATIC-K-CONST under Low Noise Settings. Figure 3.7 shows the performance of DYN-q-CONST-α with a learned model Q and and the best performing static approach, STATIC-K-CONST*. When the label noise is low, for example less than 10%, the performance of DYN-q-CONST-α is comparable to STATIC-K-CONST. For instance, for IMDB the area under the curve of DYN-q-CONST-α is 0.75 and of STATIC-K-CONST is 0.76 when the noise is 5%, and the area under the curve of DYN-q-CONST-α is 0.73 and STATIC-K-CONST is 0.74 when the noise is 10%. We observe the same trend on SRAA results. This is particularly true at larger budget levels. This can be explained considering that as the student gets better so do its estimates of Q, thus obtaining better labels. Furthermore, when the level of label noise is higher, for example 20%, DYN-q-CONST-α is worse than the best STATIC-K-CONST, as shown in Figure 3.7c.

3.5.3 How to Pick Best α. We have discussed the results of the best performing AAL-α methods. In this section we will discuss the effect of α on the student performance and provide some guidance on how to best select this parameter in practice.

3.5.3.1 Best α^* Value Increases as the Noise Level Increases. Intuitively, as the label noise increases, the quality of the annotations should have more importance. Table 3.2 shows the respective best α^* and best STATIC-K-CONST* used in Figure 3.7. In these results, we confirm that the quality of labels is increasingly important as the noise is higher. This is true both when the student knows the true model Q^* and when the student learns the model Q. The α parameter can improve the student tradeoff between the quality and cost of AAL-α.

Figure 3.8 illustrates an example of how AAL-α is affected by α when Q is learned by the student. Figure 3.8a and Figure 3.8b show how α affects the learning efficiency



(a) Zero Noise

(b) 10% Noise

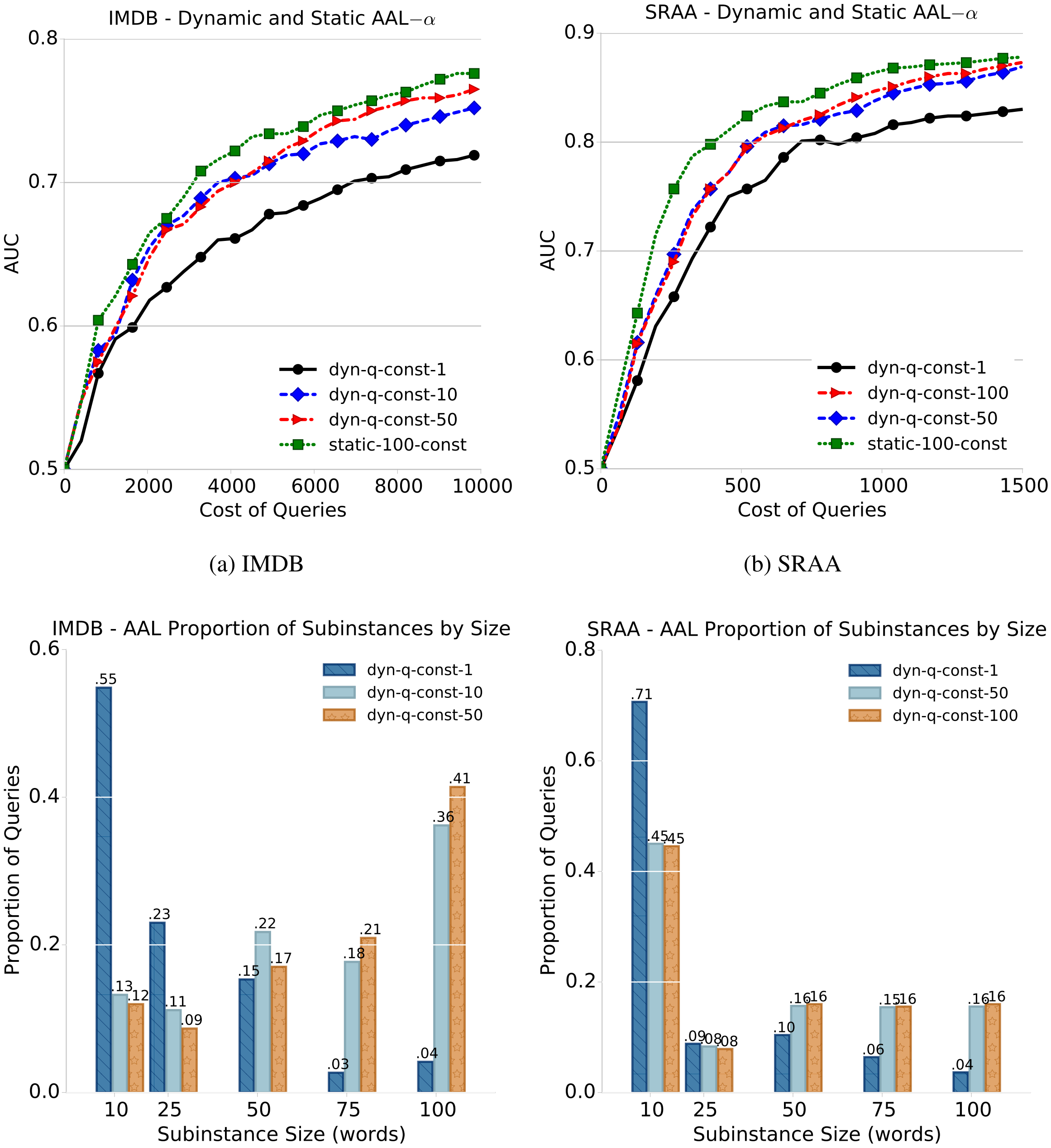
(c) 20% Noise

Figure 3.7. Best performing static and dynamic AAL-α methods under various levels of label noise. k^* is the best performing STATIC-K-CONST, and α^* is the best performing AAL-α methods. Noise levels are based on the average error of the simulated oracle.

by affecting the annotation cost. Figure 3.8c and Figure 3.8d show that as α increases the student prefers longer documents, thus affecting the use of the budget and the quality of annotation. In general, we observed a similar trend on the AAL-α method when the perfect Q^* is known so we omit the graphs to avoid redundancy.

3.5.3.2 The Quality of Q Estimates Affect the Best α Value. For AAL-α methods, let α_Q be the alpha resulting in the best performance using a learned Q model, and let α_{Q^*} be the α resulting in the best performance using a known Q^* model. We observed that best α_Q is much larger than α_{Q^*}, especially with higher levels of noise, making the quality of the label more relevant than the cost. In general, a high value of α means that the student will value a small improvement in quality even at a high cost, thus performing better in cases of high noise. However, when the small improvements happen at a high confidence level, the student may prefer a cheaper label. For example, Figure 3.8d shows that $\alpha = 100$ allows the student to prefer longer subinstances, which improves the learning efficiency of the student. Note that there is only a small difference in the size distribution of $\alpha = 100$ and $\alpha = 50$, as well as a small difference between both students. We conjecture that this is due to the student estimate of Q, which produces extreme values (e.g., ≈ 1.0). In which case, the student prefers a cheaper subinstance over a more confident one because the student is already extremely confident.

Further analysis of the effect of α shows that with a big enough α the student will prefer subinstances the oracle is most confident about, regardless of the annotation cost. Let $q_{max} = \max_y Q(y|\mathbf{x}_i^k)$ be the maximum confidence of Q for subinstances $\mathbf{x}_i^k$ in document $\mathbf{x}_i$, and let $q_j = \max_y Q(y|\mathbf{x}_i^j)$ be the confidence of the oracle for subinstance $\mathbf{x}_i^j$ where $\mathbf{x}_i^j$ is picked by AAL-α algorithms (according to Equation 3.3). Our experiments show that a big value of α encourages the AAL-α student pick the most confident subinstance of a document all the time, i.e., $q_{max} = q_j$. However, when the confidences of q_{max} and q_j are close to 1, AAL-α student may prefer a less confident subinstance at a lower cost,



(a) IMDB

(b) SRAA

(c) IMDB

(d) SRAA

Figure 3.8. Example of the effect of α on the student performance and subinstance size distributions. The average oracle error is 20%. As α increases the student performance increases, and the number of larger subinstances also increases.

Table 3.2. The best setting of α for each noise level, as seen in Figure 3.7. Noise level is based on the average oracle error. k^* correspond to best STATIC-K-CONST, α_{Q^*} is the best DYNAMIC-CONST-α with known Q^*, and α_Q is the best DYNAMIC-CONST-α with learned Q.

	IMDB			SRAA		
Noise Level	k^*	α_{Q^*}	α_Q	k^*	α_{Q^*}	α_Q
0%	25	3	0.5	50	3	10
5%	25	3	1	100	5	50
10%	50	5	100	100	25	25
15%	100	10	100	100	25	50
20%	100	10	50	100	25	100

where $q_{max} > q_j$. Usually, q_j is within an ϵ margin from q_{max}. Table 3.3 shows the percentage of subinstances picked by AAL-α that match the most confident subinstance, where $q_{max} \leq q_j + \epsilon$. This suggests that under extremely high label noise, it may be best to choose the most confident subinstances for labeling.

Table 3.3 suggests that as the value of α increases, AAL-α increasingly selects more most confident subinstances. Furthermore, a big value of α only picks most confident subinstances, within ϵ margin.

3.6 Chapter Conclusions

We have shown that more efficient active learning is possible when the student is allowed to interrupt the oracle. This new approach to active learning requires us to relax traditional assumptions of uniform annotation cost, reliability, and perfect answers from the oracle. When the oracle is reluctant, we present an anytime active learning framework in which the student is allowed to interrupt the oracle to save annotation time. We build on our previous work that conducted user studies to quantify the relationship between subinstance size, annotation time, and response rate. These were used to inform a large-scale simulated study on two document classification tasks, which showed that although inter-

Table 3.3. Percentage of subinstances picked by AAL-α that are most confident according to Q^* within an ϵ margin. **Bold** face numbers show the minimum α to reach complete agreement per ϵ between AAL-α formulation and maximum confidence.

	IMDB - ϵ Margin				SRAA - ϵ Margin			
α	**0.0**	**0.001**	**0.01**	**0.05**	**0.0**	**0.001**	**0.01**	**0.05**
0.5	3%	3%	3%	8%	11%	21%	31%	51%
1	4%	4%	5%	10%	13%	27%	40%	62%
3	32%	32%	36%	52%	28%	47%	63%	89%
5	53%	54%	59%	79%	33%	54%	71%	96%
10	74%	75%	81%	98%	38%	62%	81%	**100%**
25	89%	90%	96%	**100%**	46%	75%	94%	100%
50	94%	96%	**100%**	100%	53%	84%	**100%**	100%
75	96%	97%	100%	100%	57%	88%	100%	100%
100	97%	98%	100%	100%	59%	91%	100%	100%

ruption can cause the oracle to return neutral labels, interrupting at the right time can lead to significantly more efficient learning. We found that optimal interruption time depends on the domain and proposed a dynamic AAL strategy that is better than or comparable to the best static strategy that uses a fixed interruption time.

In scenarios where the oracle is noisy in addition to being reluctant, we further adapted our original formulation by adding a parameter α that controls the importance of annotation quality versus annotation cost. We conducted experiments with various levels of average oracle noise to study the effect on the anytime active learning algorithms. We found that, for low to moderate levels of noise, dynamic anytime methods perform at least as well as the best performing fixed interruption method. Additionally, good estimates of label quality allow dynamic methods to improve over the fixed interruption approaches. Furthermore, we provide a deeper analysis of α and offer guidelines on how to select the best parameter value. We found a positive correlation between the α value and the level of noise, and we recommend using estimated noise levels to guide the selection of α.

CHAPTER 4
ACTIVE SNIPPET SELECTION FOR FIXED DOCUMENTS

In this chapter, we will discuss research results on a second scenario of anytime active learning based on the idea that a human annotator scans an instance during annotation to find evidence of a label. In contrast to our previous scenario, this setting does not assume a sequential analysis of an instance.

Active learning aims to reduce annotation effort by carefully selecting the queries posed to a human annotator. However, traditional approaches leave to the annotator the task of finding the relevant piece of information to answer the query. In this chapter, we propose a snippet-based active annotation framework that aims to speed-up the annotation process by dynamically selecting the relevant piece of information (a snippet) to show to the annotator. Experiments on text classification datasets show that carefully choosing the snippets consistently outperforms baselines.

4.1 Introduction

Typically, active learning assumes uniform cost but subsequent work improved efficiency by recognizing that cost varies by example, e.g., longer documents may take more time for a human to classify than shorter ones [40, 75, 92].

In the previous chapter, we further refined active learning by letting the learner directly influence the cost of annotating an example by interrupting the oracle. This interruption was in the form of truncated documents. In this work, we allowed the learner to truncate documents to their first k words when presenting them to the human for annotation. This was found to significantly increase annotation efficiency, even accounting for the fact that the annotators might return more “I do not know” answers (*neutral*) for the labels of the truncated documents. By allowing the learner to select a different length k for each document, the learning system can navigate the trade-off between the utility, cost, and quality of an annotation.

In many cases, authors use the lead sentence to reveal the topic of the text, thus truncating the document to its first k words is intuitive. However, in this paper, we generalize and extend this prior work by allowing the learner to intelligently select an arbitrary portion of an instance to reveal to the annotator (as opposed to simply selecting the first k words). For example, consider the task of classifying movie reviews by sentiment. Reviews may include many sentences irrelevant to the class label such as "I watched this movie with friends." However, there is often a key sentence that reveals the sentiment of the review, such as "I really enjoyed this film" or "It was a waste of time." Our goal is to enable the learner to identify such key sentences and show them to the annotator rather than showing the full document, so that the annotator can return the label much faster. We refer to this approach as *snippet-guided active learning*.

By directing attention to the important parts of each instance, snippet-guided active learning can reduce overall annotation time and improve learning efficiency. However, poorly chosen snippets can result in missing or incorrect annotations. For example, in review classification, if the learner chooses a snippet with neutral sentiment, the annotator may be unable to determine the sentiment of a review. Conversely, the learner might pick a sentence that does not capture the overall sentiment, and hence the annotator might mislabel the full review. Therefore, the learner has to balance between likelihood of selecting the right snippet and its annotation cost. We propose a model-based criterion that selects snippets with low entropy class posteriors according to the learner. We perform experiments on two document classification tasks to investigate the following research questions:

RQ1. Does snippet selection outperform document truncation? We find that considering snippets from any location in a document often results in more efficient learning than only considering the initial sentences of the document. For example, in one experiment, one-sentence snippet selection method was able to reach 80% learners accuracy while being 24% more efficient than a method that always selects the first sentence.

RQ2. How does oracle noise affect snippet selection? Using a simulated oracle, we find that as the error rate of the oracle increases, the optimal snippet size grows as well. For a movie review classification task for example, with no label noise[3], selecting snippets containing a single sentence is optimal; but with 15% noise, five sentence snippets perform best. We attribute this in part to the fact that longer snippets result in fewer oracle errors, and in part to the fact that oracle errors reduce the accuracy of the learner, in turn hindering its ability to identify suitable snippets.

RQ3. How does adjusting the snippet size dynamically fare against a fixed size approach? Given the effect of noise on optimal snippet size, we consider a dynamic approach that searches over snippets of many possible sizes (e.g., one to five sentences in length). We find that under low to moderate noise levels, this dynamic approach meets or exceeds the effectiveness of the best fixed-size method.

RQ4. How does the budget size affect learning efficiency? We find that when the annotation budget is low, one should choose smaller snippets and explore as many documents as possible, even at the expense of collecting many neutral answers and incorrect responses, whereas when the budget is high, one should increase the snippet size to increase the overall non-neutral response rate and annotation quality.

The rest of the chapter is organized as follows: we first discuss the problem formulation and details of the proposed methods in Section 4.2. In Section 4.3 we present the experimental evaluation, followed by the discussion of the results in Section 4.4. We

[3] No noise on the non-neutral answers only; the oracle is still allowed to return an "I do not know" answer if the snippet does not reveal the label.

discuss related work in Section 2.6 and present our future research directions and conclude in Section 4.5.

4.2 Methodology

We propose a framework where the active learner is able to decide which snippets to show the oracle instead of showing full-length instances, thus reducing the overall annotation time and accelerating learning. For example, in video classification the active learner decides which short representative segment to show the oracle for annotation instead of full-length videos. Similarly, in document classification the active learner decides which snippet of the document to show the oracle. In this section we first review anytime active learning and then formalize our approach to snippet-guided active learning.

4.2.1 Anytime Active Learning. Anytime active learning (AAL), as described in Chapter 3, is an active learning framework where a learner presents only the first k words of a document to an oracle to obtain training labels. This uses the first k words as key pieces of documents and redefines $\mathcal{U}$ as the set of all possible fragments of size $k \in \{10, 25, 50, 75, 100\}$. At each iteration of active learning, AAL selects the document fragment that maximizes the following objective function:

$$\mathbf{x}_i^{k*} \leftarrow \underset{\mathbf{x}_i^k \in \mathcal{U}^k \in \mathcal{S}}{\operatorname{argmax}} \frac{U(\mathbf{x}_i)Q(z = \neg n|\mathbf{x}_i^k)}{C(\mathbf{x}_i^k)}$$

where $\mathbf{x}_i^{k*}$ is the fragment of first k words that optimizes the utility-cost and the probability of obtaining a non-neutral answer. As we will discuss further in this chapter, one drawback of this approach is that in order to find a key piece of information within a document, it has to increase the value of k and thus the cost of annotation, whether k is defined statically or dynamically. For example, consider a movie review with the plot description followed by the sentiment statement. AAL has to use a k large enough so that the sentiment of the document is included in the piece shown to the oracle. In contrast, we address this drawback allowing for flexibility in the search space while reducing the cost of using the

Algorithm 4 SNIPPET-GUIDED Active Learning

```
Input: Labeled data $\mathcal{L}$; Unlabeled data $\mathcal{U}$; Budget $B$; Classifier $P_{\mathcal{L}}(y|\mathbf{x})$
while $B > 0$ do
    $\mathbf{x}_i^* \leftarrow$ SELECTBESTINSTANCE($\mathcal{U}$)
    $s_i^* \leftarrow$ SELECTBESTSNIPPET($\mathbf{x}_i^*$)
    $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\mathbf{x}_i^*\}$
    $y_i \leftarrow$ QUERYORACLE($s_i^*$)
    $B \leftarrow B - C(s_i^*)$
    $\mathcal{L} \leftarrow \mathcal{L} \cup (\mathbf{x}_i^*, y_i)$
    $P_{\mathcal{L}}(y|\mathbf{x}) \leftarrow$ UPDATECLASSIFIER($\mathcal{L}, P$)
```

key sentences. We next describe our proposed method.

4.2.2 Snippet-guided Active Learning. We propose snippet-guided active learning as a method by which the learner may alter the queries presented to the oracle in order to accelerate the annotation process. We assume that each instance object can be represented as a sequence of elements $\mathbf{x}_i = \langle e_i^1, e_i^2 \ldots, e_i^K \rangle$. Let $\mathcal{S}_i = \{s_i^k \subseteq \mathbf{x}_i\}_{i=1}^{2K}$ be the set of all possible subsequences (*snippets*) of instance $\mathbf{x}_i$. In text classification $\mathbf{x}_i$ is a sequence of words, and $\mathcal{S}_i$ is the set of all subsequences of words. Similarly, in video classification, every frame or group of consecutive frames are the elements that compose the videos.

At each iteration of active learning, the learner selects the best instance for learning $\mathbf{x}_i^* \in \mathcal{U}$. To request the label of $\mathbf{x}_i^*$, the learner searches for the best snippet $s_i^* \in \mathcal{S}_i$ and reveals s_i^* to the oracle. The cost of annotation, $C(s_i^*)$, which is the time it takes to inspect the snippet, is deducted from the budget B, and the answer y_i returned by the oracle is added to the current training set, $\mathcal{L} \cup (\mathbf{x}_i^*, y_i)$, to retrain the learner. This cycle continues until the budget is exhausted. Algorithm 4 describes this process more formally.

Exhaustive search through all elements in $\mathcal{S}_i$ is clearly intractable for most problems as $\mathcal{S}_i$ contains all possible subsequences; in practice, however, we can use domain knowledge to restrict the search space. For example, in text, rather than searching over all subsequences of individual words, one can search over subsequences of phrases/sentences/paragraphs. In video, rather than searching over all subsequences of individual frames, one can search over subsequences of short clips.

The key component of the snippet-guided learning algorithm is how to select the best snippet s_i^* for instance $\mathbf{x}_i$. The intuition is to find a key snippet in the instance with a strong signal of its label. If an inappropriate snippet is chosen, it can cause the oracle to return an incorrect label or an "I do not know" answer, which we refer to as the neutral label. Consider the task of sentiment analysis of movie reviews. If the chosen snippet contains a plot summary and no positive/negative sentiment, the oracle might not be able to judge the overall sentiment of the review.

Following this intuition, our approach is to fit a classification model to snippets and to use this model to select snippets with the highest posterior class probabilities. Let $Q_{\mathcal{L}}(y|s_i^k)$ be the conditional probability distribution that models whether a snippet $s_i^k \in \mathcal{S}_i$ belongs to class y. We use $Q_{\mathcal{L}}$ to scan through snippets in $\mathcal{S}_i$ and identify the best snippet for annotation. We assume that a snippet is more likely to reveal the label of the document when $Q_{\mathcal{L}}$ is more confident about its label. Since we do not know the true label of a snippet, we take the maximum posterior for each possible assignment. More formally, we define the best snippet as follows:

$$s_i^* = \underset{s_i^k \in \mathcal{S}_i}{\operatorname{argmax}} \frac{\max_y Q_{\mathcal{L}}(y|s_i^k)}{C(s_i^k)} \tag{4.1}$$

where $Q_{\mathcal{L}}$ is the posterior probability of s_i^k according to the snippet classifier, s_i^k is a snippet in $\mathcal{S}_i$, y is a possible label of s_i^k, and $C(\cdot)$ is the cost of annotation. Assuming s_i^* is representative of $\mathbf{x}_i$'s label, we show s_i^* to the oracle and use the oracle's response to update $\mathcal{L} \cup (\mathbf{x}_i, y_i)$.

In the experiments below, we fit $Q_{\mathcal{L}}$ using all snippets from the examples the learner collects during the annotation cycle ($\mathcal{L}$). To do so, we use the "document as a sentence" approach [85], which assumes all the sentences have the label of their document for training purposes.

4.3 Experimental Evaluation

In this section, we describe the datasets, the evaluation methodology, and the baselines. To allow for comparison with our previous work in Chapter 3, we use the same settings wherever applicable (i.e., pre-processing, classifiers, parameters, etc.) and datasets. The following selections provide details of the settings used for our experiments.

4.3.1 Budget. We set the budget to the time required by the weakest baseline to reach within 3% of maximum achievable accuracy. Note that this provides an ample budget because stronger methods would reach the target performance even faster. The baseline required 5 hours of annotation time for IMDB and 2 hours for SRAA, where annotation times were estimated using the user study from Chapter 3.

4.3.2 Evaluation Methodology. We use a train-test split and report results of accuracy as average over 10 trials. Statistical significance is measured with one-tailed paired t-tests among the average performance per trial, and significance level $p < 0.05$. We replicate the evaluation methodology of our work [62] and use the user studies to inform our simulations, as follows:

4.3.2.1 Simulated Oracle. To enable large scale experiments we simulated an oracle using a classifier that is fit on held-out data. However, whereas in prior work we assumed the oracle would give either a correct or neutral response, in this work, to better reflect human oracles, we assume that the oracle can return an incorrect answer. To simulate this, we assume that the probability of the oracle returning a correct label is proportional to its confidence in the label. This assumption is supported by recent empirical evidence

Algorithm 5 Simulated Oracle

```
Input: Query $s_i^k$, confidence threshold $T$
$c \leftarrow \max_y P(y|s_i^k)$.                ▷ Compute the oracle's confidence
if $1 - c > T$ then
    return neutral label
else                                              ▷ Return a non-neutral label
    $p \sim$ Bernoulli($c$)
    if $p == 1$ then
        return a correct label.
    else
        return an incorrect label
```

that human oracles are aware when they are likely to make a mistake [96]. The choice of whether a query receives a correct or incorrect label is determined by the outcome of a Bernoulli trial in which the probability of a correct label is equal to the oracle's confidence. More formally, for a query s_i^k and neutrality threshold T, a response is sampled as shown in Algorithm 5. For oracle noise, we experimented with various noise levels up to 20% in 5% increments. To ensure a fair comparison between methods, the oracle's response for a particular snippet is fixed across all baselines, i.e., the oracle's responses are fixed at the beginning of the experiments so that it will return the same answer for the same snippet no matter which method asks for an answer.

The oracle classifier is an L_1-regularized logistic regression implementation of LibLinear [28]. For IMDB, we simulate the oracle using a $C = 0.3$ parameter, and uncertainty threshold to return a neutral label of $T = 0.4$. For SRAA, we use $C = 0.01$ and $T = 0.3$ as parameters.

4.3.2.2 Active Learner. For the learner, we also use a logistic regression classifier with L_1 regularization. We set its complexity parameter C to its default value of 1. We start every experiment with a labeled set of 50 documents selected at random. $P_{\mathcal{L}}$ is bootstrapped with the sentences of labeled set $\mathcal{L}$. At each round of active learning, we select 10 documents at a time for labeling.

4.3.2.3 Cost Function. For the cost function, we re-use the annotation cost reported the previous chapter (Figure 3.1 Chapter 3) for 10, 25, 50, 75, and 100 words, and use linear interpolation for intermediate values. Furthermore, for these two datasets, we assume that the minimum annotation time is that of 10 words, and the maximum annotation time is worth 100 words. The reason for these assumptions is that i) even if the learner shows a very small snippet, the oracle still has to spend time to make a decision, which is roughly 5 seconds on both datasets, and ii) the annotation cost levels off as the snippet size gets large enough (e.g., a five second increase for 50 to 75 words but only a one second increase for 75 to 100 words on the IMDB dataset).

4.3.3 Implementation Details and Baselines. Unlike our previous approach where we assumed that the snippets were first 10, 25, 50, 75, and 100 words of a document, in this chapter, we assume the individual elements e_i in documents are sentences and hence the student is searching over subsets of sentences. The sentences are obtained using a pre-trained sentence tokenizer [12].

To select the snippets from the learner's selected documents, we implemented static and dynamic approaches. Static approaches use snippets of a fixed number of sentences. For example, all snippets are composed with k sentences. The following are static approaches:

- **FIRST-K**, a baseline, the learner selects the first k sentences of each document as a snippet to query the oracle.
- **SS-K**, active snippet selection method, where the learner selects the best snippet using Equation 4.1. Unlike our baseline, this method is allowed to search for the best

snippet of k sentences anywhere in the document.

In contrast, dynamic approaches look through snippets of various sizes. We studied the following methods:

- **FIRST-1-TO-5**, a baseline, as an adaptation of our work in [62], that looks for the ideal length of the snippet. This method selects from a candidate set of snippets limited to only first j sentences of each document where the method is allowed to determine the best j per document $j \in \{1, 2, \ldots, 5\}$.
- **SS-1-TO-5**, snippet selection, is the proposed method where the learner selects the best snippet using Equation 4.1 and optimizes for both the snippet size and snippet location per document, where snippets are allowed to be size j for $j \in \{1, 2, \ldots, 5\}$.

We experiment with settings where the learner chooses its documents using random sampling. This is to make sure that each snippet selection strategy sees the same set of documents but differs only on the selected snippet. To further increase tractability of snippet selection, the candidate set $\mathcal{S}_i^j$ is built using only j contiguous sentences, i.e., a sliding window of size j, instead of any subset of j sentences.

4.4 Results and Discussion

In this section, we report results investigating the answers to the research questions asked in Section 4.1. We first investigate the answers to the following two research questions.

4.4.1 Does Snippet Selection Outperform Document Truncation? How Does Oracle Noise Affect Snippet Selection? Our objective is to accelerate annotation time by showing a well-chosen snippet to the oracle for annotation, allowing the student to search over snippets of all sizes. Suppose each method is allowed to select only one sentence per document to show the oracle for annotation. We can select the first sentence, i.e., FIRST-1, or find the best sentences with SS-1. Figure 4.1 compares FIRST-1 and SS-1 under varying oracle noise on IMDB data. We see that actively selecting snippets SS-1 outperforms selecting the first sentence of each document FIRST-1 across noise levels. While it may not be surprising that sentences in the middle of the document may be more indicative of the label than the first sentence, it is somewhat surprising that the learner is able to identify such sentences even early in training.

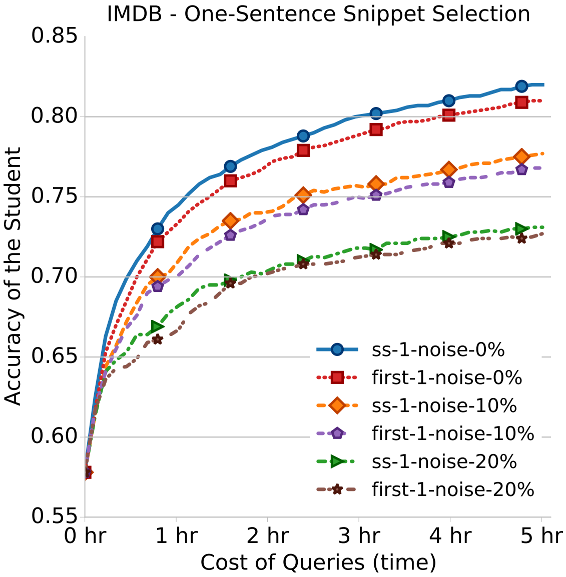
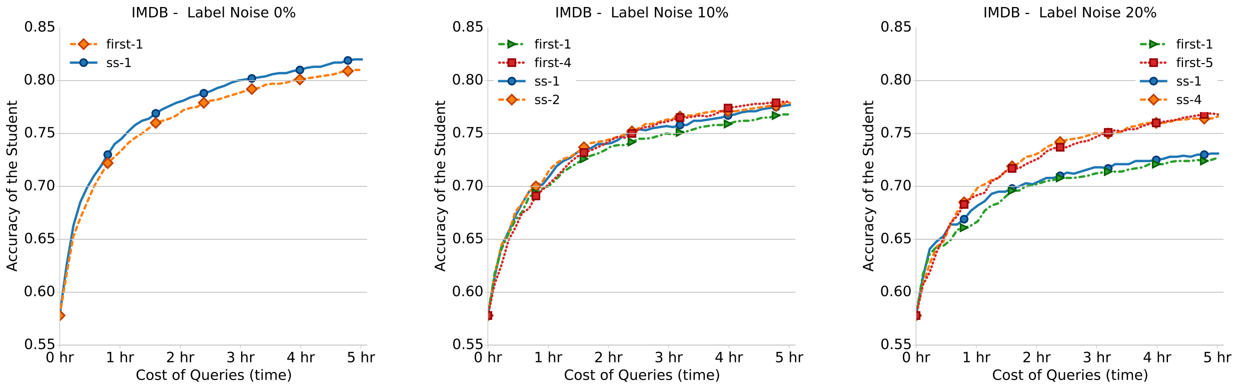


Figure 4.1. Average student accuracy on IMDB data. The methods select one-sentence snippets as first sentence FIRST-1, and best snippet SS-1. SS-1 outperforms the baselines.

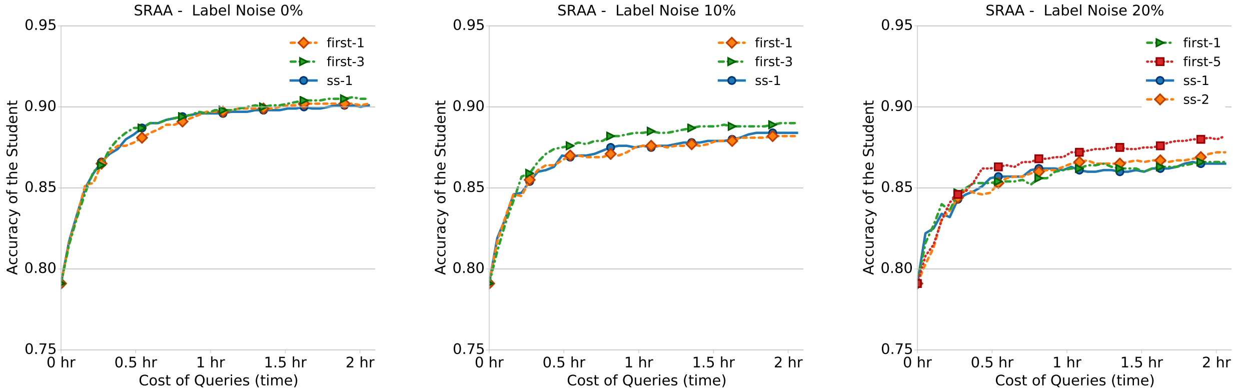
To further compare snippet selection with truncation and noise levels, we run experiments comparing FIRST-K and SS-K with $k \in \{1 \ldots 5\}$. Figure 4.2 shows the average performance of the classifier as a function of the budget on IMDB data. For readability, we display up to four methods in each graph: FIRST-1, SS-1, as well as the best performing k for each of the FIRST-K and SS-K methods.

We note that for noisier oracles, static methods that use snippets of a fixed number



(a) Noise 0% - Student Accuracy (b) Noise 10% - Student Accuracy (c) Noise 20% - Student Accuracy

Figure 4.2. Effect of snippet selection on random document on IMDB. Selecting the best snippet (SS) outperforms best performing FIRST-K baselines under low levels of noise.



(a) Noise 0% - Student Accuracy (b) Noise 10% - Student Accuracy (c) Noise 20% - Student Accuracy

Figure 4.3. Effect of snippet selection on random document on SRAA. Selecting the best snippet (SS) is comparable to best performing baselines under low levels of noise.

of sentences need longer snippets to counter the label noise. For example, Figure 4.2a shows the student performance when the oracle is always correct but may reject an answer. The best static method is FIRST-1 showing only the first sentence for annotation, whereas in Figure 4.2c at 20% label noise, the best static method is FIRST-5 which requires to show five sentences to the oracle.

Table 4.1. Average percentage of neutral responses from the oracle, and oracle accuracy on non-neutral responses, from Figure 4.2 for IMDB data. FIRST-1, and best performing FIRST-* are baselines compared to SS-1, and best performing SS-*.

Method	Oracle Neutrality			Oracle Accuracy		
	0% Noise	10% Noise	20% Noise	0% Noise	10% Noise	20% Noise
FIRST-1	0.35	0.37	0.36	1.00	0.93	0.86
FIRST-*	0.35	0.14	0.12	1.00	0.96	0.92
SS-1	0.31	0.33	0.35	1.00	0.93	0.86
SS-*	0.31	0.25	0.17	1.00	0.94	0.92

Table 4.2. Average percentage of neutral responses from the oracle, and oracle accuracy on non-neutral responses, from Figure 4.3 for SRAA data. FIRST-1, and best performing FIRST-* are baselines compared to SS-1, and best performing SS-*.

Method	Oracle Neutrality			Oracle Accuracy		
	0% Noise	10% Noise	20% Noise	0% Noise	10% Noise	20% Noise
FIRST-1	0.50	0.50	0.50	1.00	0.98	0.96
FIRST-*	0.34	0.34	0.27	1.00	0.98	0.96
SS-1	0.41	0.43	0.46	1.00	0.98	0.95
SS-*	0.47	0.43	0.48	1.00	0.98	0.95

Similarly, for active snippet selection methods (SS-K) we observe that more label noise requires longer snippets. Intuitively, a less confident oracle would require more content from a document to provide a better answer. Thus, larger snippets compensate for the lower quality of annotation. We observed similar results for SRAA in Figure 4.3.

Comparing the best performing truncation approach (FIRST-*) with the best snippet selection approach (SS-*), we see on IMDB that SS-* outperforms (at low noise levels) or matches (at high noise levels) FIRST-*. Further, we see that SS-* uses fewer sentences than FIRST-*. For example, with 10% label noise (Figure 2b), SS-* requires only two snippets whereas FIRST-* required twice as many: four sentences.

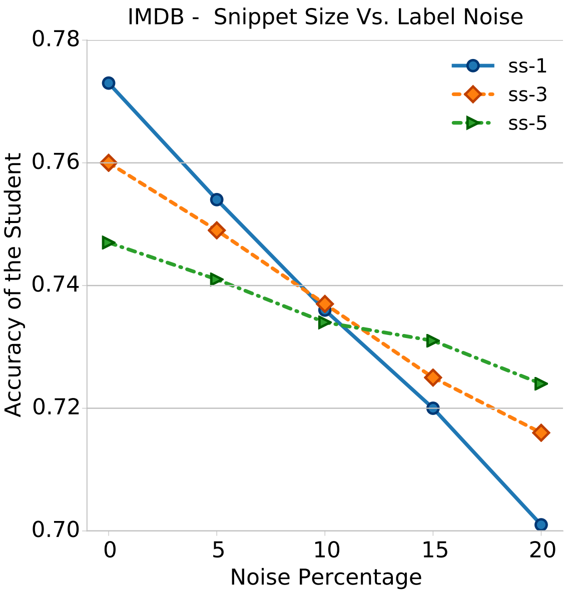


Figure 4.4. Relationship between snippet size and label noise. IMDB average student performance per label noise level. Selecting more than one sentence is better than selecting one when noise is high (e.g., 20% noise). Similar results were observed on SRAA.

We look at a deeper analysis of how the noise affect snippet size selection. Figure 4.4 shows the average student performance per level of label noise, and various snippet sizes. Note that as the label noise increases the performance of the student decreases for all methods. Furthermore, under small label noise selecting one sentence, SS-1, is better than selecting more expensive snippets of five sentences SS-5. However, selecting SS-5 outperforms SS-1 when the label noise is high. We observed similar results on SRAA data where longer snippets perform better when there is more label noise.

We next investigate the possible reasons why the snippet selection strategy SS-* outperforms FIRST-*. When the oracle is shown a few sentences rather than the full document, it is quite possible that the chosen snippet might not contain relevant information, resulting in an ”I don’t know,” i.e., a neutral answer, or the snippet might contain misleading information, causing the oracle to return an incorrect label. Table 4.1 shows the average oracle neutral response rate on all snippets and the oracle accuracy on the non-neutral answers, for IMDB (left) and SRAA (right). For IMDB, we observe that SS-1 results in lower neutral percentage than FIRST-1 and comparable oracle accuracy at all noise levels. This result explains why SS-1 is better than or comparable to FIRST-1. Comparing SS-* to FIRST-*, even though both have comparable oracle accuracies, SS-* is able to elicit fewer neutral responses than FIRST-* at only low noise levels. Comparing the learner's accuracies in Figure 4.2 as noise levels increase, SS-* and FIRST-* have comparable results. This suggests that even though SS-* does not necessarily elicit fewer neutrals at high noise levels, it is choosing better snippets that in turn improve the learner and make up for the loss in neutrality rate.

For SRAA in Table 4.2, we observe similar results: the oracle accuracy is the same or comparable for SS-K and FIRST-K methods. SS-1 results in lower neutrality than FIRST-1 in all noise levels. Comparing SS-* to FIRST-*, SS-* results in lower neutrality rate only in noise-free case. However, for SRAA, unlike IMDB, the learner using FIRST-* outperforms SS-* in noisy cases (Figure 4.3). A possible explanation for SS-* learner's performance (Figure 4.3) is that SRAA is a newsgroup classification task and in many cases the authors of the documents quickly disclose the topic through the subject line, sometimes in combination with the first sentences of the message body, thus FIRST-K approaches are more appropriate for this domain.

As a drawback, static approaches such FIRST-K and SS-K require domain knowledge of annotation difficulty or prior knowledge about oracle quality to determine the best k value for each task. In some domains, this knowledge may be inferred easily or set by the domain expert. In others, this needs to be adaptively set, as we discuss later in 4.4.2.

4.4.2 How Does Adjusting the Snippet Size Dynamically Fare against a Fixed Size Approach? Table 4.3 and Table 4.4 show the average of the learning curve of the underlying classifier, for IMDB and SRAA dataset respectively. We include the best performing static methods FIRST-* and SS-* and the dynamic methods FIRST-1-TO-5 and SS-1-TO-5. We indicate statistical significance of the winning method with respect to the corresponding static or dynamic competitor.

We observe that dynamic versions often outperform or are comparable to their best static versions under low to moderate noise levels. That is, FIRST-1-TO-5 is better than or comparable to FIRST-* and SS-1-TO-5 is better than or comparable to SS-* under low or moderate noise levels. When the noise is higher; however, the best static approaches outperform their dynamic versions. The disadvantage of the static approaches is, however, they require the learner to set k once and for all and this might be or might not be possible depending on the domain.

Comparing dynamic versions to one another, we observe that SS-1-TO-5 significantly outperforms FIRST-1-TO-5 under low or moderate noise levels. However, when the noise is high, SS-1-TO-5 is comparable to FIRST-1-TO-5 methods, for IMDB. In contrast, for SRAA, we find that SS-1-TO-5 is not able to outperform FIRST-1-TO-5. This result is similar to what we have observed earlier for SRAA: the `first` approaches contain the subject line. Even though the SS can technically search for the subject as well, it is not able to learn it, whereas FIRST approaches benefit from the subject line implicitly.

For IMDB, we observe the best performing methods have high oracle accuracy and low neutrality percentages. This is especially noticeable with larger levels of noise. For example, in Table 4.1 best performing FIRST-* has a lower neutrality than the best SS*, although the latter is better than FIRST-Kmethods.

4.4.3 How does the Budget Size Affect Learning Efficiency? We study the effect that the budget size has on how effectively the learner balances quality and cost of annotation. So far, we have considered use of actively selecting snippets for annotation with a fixed budget (5 hours for IMDB and 2 hours for SRAA). However, in some cases the annotation budget may be more restrictive. To investigate the effect of the budget size, we examined the learning curves of each method at several fractions of the original budget.

Table 4.3. Average performance of student classifier for IMDB. Average of learning curve of best performing static methods FIRST-* and SS-*, and dynamic methods FIRST-1-TO-5, and the SS-1-TO-5 methods. Marked values with * are statistically significant with respect to their counterparts (static to static and dynamic to dynamic).

Noise	Method	Performance per budget				
		∼ 1hr	∼ 2hrs	∼ 3hrs	∼ 4hrs	∼ 5hrs
0%	FIRST-1*	0.681	0.718	0.738	0.753	0.763
	FIRST-3	0.670	0.707	0.727	0.742	0.753
	FIRST-5	0.662	0.699	0.721	0.735	0.746
	SS-1*	0.690*	0.727*	0.748*	0.763*	0.773*
	SS-3	0.676	0.714	0.735	0.750	0.760
	SS-5	0.662	0.698	0.720	0.735	0.747
	FIRST-1-TO-5	0.684	0.720	0.741	0.755	0.765
	SS-1-TO-5	0.689	0.726*	0.748*	0.763*	0.773*
10%	FIRST-*	0.656	0.691	0.711	0.725	0.736
	FIRST-1	0.664	0.692	0.709	0.720	0.729
	FIRST-3	0.654	0.685	0.704	0.718	0.728
	FIRST-5	0.656	0.689	0.710	0.724	0.735
	SS-*	0.666*	0.699*	0.717*	0.730*	0.739*
	SS-1	0.666*	0.698	0.716	0.727	0.736
	SS-3	0.659	0.692	0.712	0.726	0.737
	SS-5	0.658	0.691	0.710	0.724	0.734
	FIRST-1-TO-5	0.665	0.693	0.711	0.723	0.732
	SS-1-TO-5	0.669	0.699*	0.716*	0.728*	0.736*
20%	FIRST-1	0.643	0.665	0.680	0.689	0.696
	FIRST-3	0.646	0.673	0.690	0.701	0.710
	FIRST-5*	0.649	0.680	0.699	0.713	0.723
	SS-*	0.651	0.683	0.703	0.716	0.725
	SS-1	0.643	0.665	0.680	0.689	0.696
	SS-3	0.646	0.673	0.690	0.701	0.710
	SS-5	0.649	0.680	0.699	0.713	0.723
	FIRST-1-TO-5	0.644	0.667	0.681	0.691	0.698
	SS-1-TO-5	0.648	0.672	0.686	0.696	0.703

Table 4.4. Average performance of student classifier for SRAA. Average of learning curve of best performing static methods FIRST-* and SS-*, and dynamic methods FIRST-1-TO-5, and the SS-1-TO-5 methods. Marked values with * are statistically significant with respect to their counterparts (static to static and dynamic to dynamic).

Noise	Method	Performance per budget			
		$\sim^1/_2$ hr	$\sim$ 1hrs	$\sim 1^1/_2$ hrs	$\sim$ 2hrs
0%	FIRST-*	0.855	0.874	0.882	0.888
	FIRST-1	0.854	0.873	0.881	0.886
	FIRST-1	0.854	0.873	0.881	0.886
	SS-*	0.854	0.873	0.881	0.886
	SS-1	0.854	0.873	0.881	0.886
	SS-1	0.854	0.873	0.881	0.886
	FIRST-1-TO-5	0.854	0.873	0.882	0.887
	SS-1-TO-5	0.849	0.868	0.878	0.884
10%	FIRST-*	0.849	0.864*	0.872*	0.876*
	FIRST-1-TO-5	0.849*	0.863*	0.869*	0.872*
	FIRST-1-TO-5	0.849*	0.863*	0.869*	0.872*
	SS-*	0.845	0.859	0.865	0.869
	FIRST-1-TO-5	0.849*	0.863*	0.869*	0.872*
	FIRST-1-TO-5	0.849*	0.863*	0.869*	0.872*
	FIRST-1-TO-5	0.849*	0.863*	0.869*	0.872*
	SS-1-TO-5	0.839	0.853	0.859	0.864
20%	FIRST-*	0.837	0.851	0.859	0.863
	FIRST-1-TO-5	0.840*	0.850*	0.857*	0.860*
	FIRST-1-TO-5	0.840*	0.850*	0.857*	0.860*
	SS-*	0.832	0.845	0.852	0.856
	FIRST-1-TO-5	0.840*	0.850*	0.857*	0.860*
	FIRST-1-TO-5	0.840*	0.850*	0.857*	0.860*
	FIRST-1-TO-5	0.840*	0.850*	0.857*	0.860*
	SS-1-TO-5	0.833	0.843	0.847	0.850

Table 4.5. Average accuracy of static methods per budget with 10% label noise, on IMDB. Highest values are highlighted.

Method	Avg. Performance per Budget (hrs.)				
	1	2	3	4	5
FIRST-1	**0.664**	0.692	0.709	0.720	0.729
FIRST-2	0.663	**0.694**	**0.712**	**0.725**	**0.735**
FIRST-3	0.654	0.685	0.704	0.718	0.728
FIRST-4	0.656	0.691	0.711	**0.725**	**0.735**
FIRST-5	0.656	0.689	0.710	0.724	**0.735**

Table 4.5 shows the average performance of static methods on IMDB per budget level. We observe that with increasing budgets, the k of the best method tends to be higher. For example, with one hour of budget the best static method is $k = 1$ whereas with five hours the best k is 5 sentences (two and four sentences perform with the same average; however, the curve end point for five is best). We observed a similar trend for SS-K methods but we omitted them to avoid redundancy. On SRAA, the trends are less clear due to the small differences among the performances.

4.5 Chapter Conclusions

Typical active learning learning methods have been developed under the assumption that the annotation cost depends entirely on the oracle. Anytime active learning proposed a framework to allow the active learner to intervene in annotation by estimating the maximum amount of time needed for annotation and interrupting the oracle. We described and presented a new active annotation strategy that condenses an instance to a smaller snippet to reduce the annotation time and effort. We proposed a model-based approach capable of identifying the key snippet in a document (located in an arbitrary position) and showed that it can result in more efficient learning than fixed-snippet strategies.

CHAPTER 5
ACTIVE DOCUMENT-SNIPPET PAIR SELECTION

In this chapter, we describe an AAL that simultaneously select the documents and snippets to accelerate annotation and improve spending efficiency. In previous chapters we discussed anytime active learning as a new active learning approach that allows the active learner to manipulate the queries posted to the oracle in order to produce savings. In previous chapters, we explored AAL where given a document, the learner accelerates annotation by interrupting the oracle through truncated documents, and by extracting key pieces of information to show the expert. In contrast, in this section we will build a decision-theoretic framework that allows the active learning algorithm to choose simultaneously the best document for learning, where in the document to extract a snippet, and for how long to allow the expert to review the snippet.

5.1 Introduction

We introduce a framework in which the active learner has the ability to select the best instance for annotation, extract a snippet from the example, and ask the expert his/her best guess. For example, in document classification, we may show the expert a snippet located in an arbitrary location k of a document, and ask for the expert for his best guess at the document label.

Ideally, an active learner picks the optimal example and corresponding snippet to show the expert for annotation. The optimal example maximizes the performance of the classifier at the least cost possible, if labeled and added to the training data [50, 66]. Graphically, the optimal example allows the model to achieve the highest point in the performance learning curve (i.e., highest slope) from the current performance. In anytime active learning, the algorithm needs to find a snippet to show the expert, estimate how much time the expert will spend on that snippet, and predict if the expert will return an answer at all. Further, the algorithm would estimate the likelihood of the expert returning the correct answer. We refer to this method as *optimal anytime active learning*, since the learner selects the optimal example for annotation using anytime capabilities.

Of course, in order to find the example that yields the highest improvement, the learning algorithm needs access to the evaluation metric and a validation test for measuring improvement. However, in some cases the validation set may not be available or may not be representative of the domain data. Furthermore, annotating documents through snippets may cause the expert to select the incorrect label. Our active learning framework, thus models the tradeoff between the value of the (possibly incorrectly labeled or not answered) instance, the cost of annotating a snippet (time to review the snippet), and the relevance of the snippet (to convey the instance label). At each iteration, the algorithm searches over document-snippet pairs to optimize this tradeoff — for example, to decide between asking the human expert to spend more time on the current document or move on to another document. We build upon a decision-theoretic formulation [38], where the value of a document-snippet pair is the expected improvement in performance after the instance is added to the training set. The pair with the highest improvement is used for annotation.

While previous active learning research has considered the cost-benefit ratio for selecting instances [23], as well as the annotation error [40], our method considers which snippet to use for annotation, and whether the expert may return a correct response or reject the request. Though closely related, our framework differs from other look-ahead methods [66]; in our method we also consider which snippet and what size of snippet to show the expert to guarantee a cost-effective annotation. We experimented with several text classification tasks to test our method empirically. In particular, we provide an answer to the following research questions:

RQ1. How do optimal methods compare to passive and active fixed methods? We found that our method is able to outperform random sampling with fixed snippets of various sizes. Furthermore, OPTIMAL-AAL performs well when compared to active learning with fixed snippets. For example, for a sentiment analysis dataset our method is 8% more efficient in average than the best fixed baseline.

RQ2. How does length of the document affect the snippet selection? We found that in general anytime active learning methods are able to balance cost, by selecting shorter snippets, and utility of selecting document-snippet pairs. Our method outperforms or is comparable to the active learning baselines.

RQ3. How does cost and response rate affect the learning efficiency of the student? We found mixed results related to the effect of cost, likely dependent of the domain, where in some domains our method is able to handle a steep cost function better than the baselines.

RQ4. How does the validation set affect learning efficiency? We found that the validation set used to compute utility measures has a big influence on the performance of our method. We empirically show that larger validation sets prove better for final results.

5.2 Methodology In this section, we first present the problem formulation and formally discuss our proposed method. We detail our method components and elaborate on implementation considerations.

5.2.1 Active Document-Snippet Pair Selection. An optimal anytime active learning strategy selects the document-snippet pair $\left\langle \mathbf{x}_i, s_i^k \right\rangle$ that is expected to produce the highest improvement, once its labeled document is added to the training set. This decision-theoretic approach is illustrated graphically by selecting the document-snippet pair that is expected to produce the maximum improvement in the classifier learning curve, i.e., the largest slope in the next active learning iteration. Note that if the slope is less or equal to zero, i.e., using the pair will not help the classifier's performance, the algorithm may decide to skip such query even if it incurs in the cost of annotation. If the slope is greater than zero, then the learner can decide which pair yields the highest performance (i.e., highest in y-axis) at lowest cost (i.e., lowest in x-axis).

5.2.2 Expected Utility for Anytime Active Learning. The best document-snippet pair $< \mathbf{x}_i^*, s_i^* >$ maximizes the expected utility of the query over all possible labels of s_i^k. We assume the expert may reject a query or answer with a label which may be incorrect. More formally, let $y \in \{y^0, y^1, n\}$ be the possible labels of snippet s_i^k according to a reluctant expert. The expert can provide a label y^0, or y^1, or reject the query with an "I don't know" answer or neutral label n. Let P_E be a probabilistic classifier that models the oracle responses. Let $Utililty_{\mathcal{V}}(\mathbf{x}_i, y)$ be a utility function that measures the benefit of adding document $\mathbf{x}_i$ with label y to the training set. The objective is to find the optimal pair $< \mathbf{x}_i^*, s_i^* >$ that maximizes the expected utility of knowing the label of document $\mathbf{x}_i^*$:

$$\operatorname*{argmax}_{s_i^k \in \mathcal{C}} \mathbb{E}[(\mathbf{x}_i, s_i^k)] = \operatorname*{argmax}_{(\mathbf{x}_i, s_i^k) \in \mathcal{C}} \sum_{y}^{\mathcal{Y}} \frac{P_E(y|s_i^k) \times Utility(\mathbf{x}_i, y)}{C(s_i^k)} \tag{5.1}$$

where $\mathbf{x}_i \in \mathcal{U}$ is a candidate document, $s_i^k \in \mathbf{x}_i$ is a snippet of the candidate document, $P_E(y|s_i^k)$ is the posterior probability of y_i given s_i^k, $Utility(\cdot)$ is a measure of improvement should document $\mathbf{x}_i$ be labeled, and $C(\cdot)$ is the annotation cost (e.g., the time the expert reviews a snippet). For simplicity of explanation, we assume a binary classification task; however, this assumption can be easily generalized to multiple label values.

In contrast to our previous formulation, where the learner dynamically selects documents and its truncation point, the objective in Equation 5.1 allows the algorithm to find a snippet to show the expert, estimate how much time the expert will spend on that snippet, and predict if the expert will return a neutral answer.

Next, we describe the components of the objective function and describe implementation details of the algorithm.

5.2.3 Utility Function. In general, the utility function tells us how good a classifier is and allows us to compare the current classifier with future versions when acquiring new labels. Our utility function $Utility(\cdot)$ is given by the improvement on the classifier performance per unit cost [35]; however, alternative definitions can be used depending on the domain. Formally, the improvement of performance is the difference between the current classifier train on $\mathcal{L}$ and a future version of the classifier trained on an extended training set $\mathcal{L} \cup (\mathbf{x}_i, y)$, given by:

$$Utility_{\mathcal{V}}(\mathbf{x}_i, y) = Perf_{\mathcal{V}}(\mathcal{L} \cup (\mathbf{x}_i, y)) - Perf_{\mathcal{V}}(\mathcal{L}) \tag{5.2}$$

where $Perf_{\mathcal{V}}(\mathcal{L})$ is the performance measure of a classifier trained on $\mathcal{L}$, and $Perf_{\mathcal{V}}(\mathcal{L} \cup (\mathbf{x}_i, y))$ is the performance measure of the classifier when $(\mathbf{x}_i, y)$ is added to the training set. These type of approaches are referred to as look-ahead methods because they evaluate the effect of adding an instance into the current training set. The performance of the classifier is measured over a validation set $\mathcal{V}$ of labeled instances. We will discuss this in more detail in Section 5.2.4.

The classifier performance can be measured with any metric of interest depending on the domain. This measure should be aligned to the testing measure for optimal results. For example, in text classification classifier accuracy is often used as the preferred performance measure. However, it may be trivial to obtain a high score if the label distribution is imbalanced.

Typically, the performance measure $Perf_{\mathcal{V}}(P_{\mathcal{L}})$ is defined as a *loss function*, $L(P_{\mathcal{L}}(y|x))$, and the objective is to minimize said loss:

$$\begin{aligned} Perf_{\mathcal{V}}(P_{\mathcal{L}}) &= \mathbb{E}\left[L(P_{\mathcal{L}}(y|x)\right] \\ &= \int_x L(P_{\mathcal{L}}(y|x))P(x) \\ &\approx \frac{1}{|\mathcal{V}|}\sum_{x\in\mathcal{V}} L(P_{\mathcal{L}}(y|x)) \end{aligned}$$

However, when using a loss function as a performance metric, Equation 5.2 should be $-Utility(\cdot)$ so that it measures the improvement properly. A common loss function is 0/1 loss to measure the accuracy of classifier. However, because we do not know the true label of the instances in the unlabeled set $\mathcal{V}$, we need to use proxies for the loss function L. For example, a proxy for 0/1 loss is:

$$Perf_{\mathcal{V}}(P_{\mathcal{L}}) = \frac{1}{|\mathcal{V}|}\sum_{x\in\mathcal{V}} 1 - \max_{y^j} P_{\mathcal{L}}(y^j|x) \tag{5.3}$$

One problem with this proxy in practice is that it trivially achieves 0 loss when all the instances are classified into one class with probability 1. Thus, the reliability of this loss function depends on the calibration of the student probabilities. We could use AUC as a performance measure to obtain a more effective classifier.

Another alternative measure is predicting probabilities of the underlying classifier. This measure provides a smoother function of the degree of the classifier's forecasting abilities, in contrast with a 0/1 loss measure which works as a step function of correct predictions. A simple formulation using the predicted probabilities is as follows:

$$Perf_{\mathcal{V}}(P_{\mathcal{L}}) = \frac{1}{|\mathcal{V}|}\sum_{x\in\mathcal{V}} 1 - P_{\mathcal{L}}(y^*|x) \tag{5.4}$$

where y^* is the true label of instance x. This measure of performance is aimed to directly aligned the classifier performance to the oracle performance.

Replacing the expected utility Equation 5.2 in Equation 5.1, the updated objective function is as follows:

$$\underset{\mathbf{x}_i, s_i^k \in \mathcal{C}}{\operatorname{argmax}} \mathbb{E}[(\mathbf{x}_i, s_i^k)] = \underset{\mathbf{x}_i, s_i^k \in \mathcal{C}}{\operatorname{argmax}} \sum_{y}^{\mathcal{Y}} \frac{P_E(y|s_i^k) \times (Perf(\mathcal{L} \cup (\mathbf{x}_i, y)) - Perf(\mathcal{L}))}{C(s_i^k)}$$

where the cost and probabilities in the expectation depend on the snippet s_i^k and the performance only depends on adding a document and the possible labels. These considerations may be used to efficiently implement the formulation.

5.2.4 Validation Set. The utility function uses a validation set, $\mathcal{V}$, to determine the future performance of the underlying classifier. The validation set is typically a held-out labeled data or the test set itself. The validation set should be large enough to be representative of the test set, and to prevent overfitting of the underlying model. However, a validation set is not always available when labeled instances are scarce or instances are hard to obtain as well. In such cases, the available training labeled data $\mathcal{L}$ is a proxy for the validation set. Typically, the evaluation measure is computed as the average of a cross-validation score to approximate the future performances. For example, a candidate $\mathbf{x}_i$ is added to all training folds, and the performance is measured on the corresponding testing fold. The performance measure is an average over the fold measures.

5.2.5 Revisiting Early Decisions. Note that in Algorithm 4, our search space is updated by eliminating the snippet used for annotation [82]. Other snippets of the same document $\mathbf{x}_i$ remain in the pool as candidates. At early states of active learning, we expect the classifier $P_{\mathcal{L}}$ to be less reliable; thus, it is more likely to induce annotation error by selecting a less than optimal snippet. When $P_{\mathcal{L}}$ improves, it may select a different snippet from a document that already belongs to $\mathcal{L}$. If a document is still considered to produce improvement in the student classifier, then the algorithm may request a second label for a known document through a different snippet. Sheng et al. [82] showed that in many cases, re-labeling instances can produce significant improvement in learning efficiency. This revisiting strategy is particularly useful when the training is significantly small, the cost of obtaining instances (even if unlabeled) is high, or the underlying classifier is highly sensitive to label noise; however, relabeling is not effective in all cases [49].

5.2.6 Cost Function. The cost function $C(s_i^k)$ determines how much effort is needed to inspect a snippet s_i^k. This cost depends on factors, such as instance features, oracle expertise, task, and how much information s_i^k includes. In active learning literature, there are various methods to infer the cost function, such as basing it on domain knowledge, or using a proxy to expert quality. In this chapter, we make a simplifying assumption and assume that the cost of annotation depends on how many words are presented to the oracle.

In general, annotation time increases at a faster rate for short documents than for long documents, as we observed in our user studies. For example, the cost difference between five to 10 words is larger than the difference between 100 and 105 words. To avoid the need for user studies for every new dataset, we define a cost function for text classification tasks as follows:

$$Cost(w) = a \times \log_b w + c \tag{5.5}$$

where w is the number of words and $w \in [1, \inf]$, a represents the slope of the cost function and c represents the minimum cost of annotating one word.

5.2.7 Implementation Details. In this section, we provide details on how we implemented our objective function Equation 5.1 and describe the search space of the function.

5.2.7.1 Decision Process. Consider requesting a label for a snippet pair $\langle \mathbf{x}_i, s_i^k \rangle$ by showing s_i^k to the oracle. The oracle may reject the query (if the snippet is irrelevant), in which case the utility will come from obtaining a neutral label (this utility may be zero). Let's consider this a small constant utility ϵ, where after obtaining the neutral answer the student classifier remains unchanged (i.e., student oracle is not re-trained). However, if the query is answered, the utility of labeling s_i^k is derived from obtaining a label y^0 or y^1, and updating the train set as $\mathcal{L} \cup (\mathbf{x}_i, y_i)$. Note that each possible label obtained from snippet s_i^k, the full instance $\mathbf{x}_i$ may take any of the class labels and the utility depends on the document.

Figure 5.1 illustrates the steps in decision process:

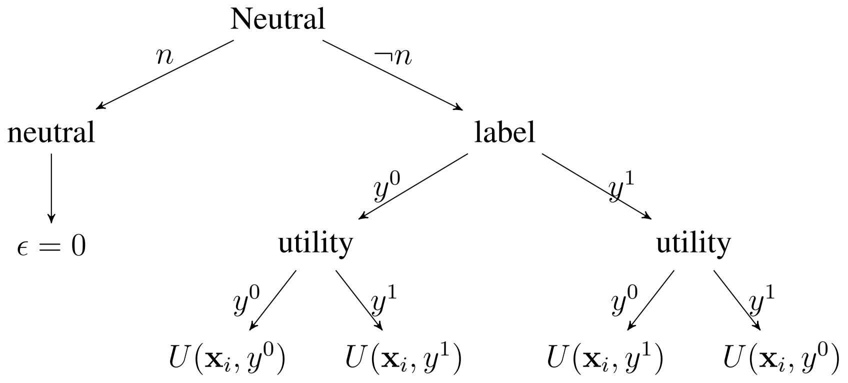


Figure 5.1. Decision-theoretic approach to pick an instance to query. $U(\cdot)$ is the measure of improvement.

The neutral response depends on what snippet the oracle sees as well as the label obtained as an answer. The utility to the underlying classifier depends on the full instance added to the training set with a label and the annotation cost of the snippet.

5.2.7.2 Search Space. In general, at every iteration the algorithm picks the best $< \mathbf{x}_i, s_i^k >$ for learning and requesting the label. Let $\mathcal{C} = \{< \mathbf{x}_i, s_i^k > \in \mathcal{U}\}_{i=1}^{s}$ be the search space that contains all instance-snippet pairs available to the learning algorithm. If we organize the candidate pairs where each row represents the pairs of a document $\mathbf{x}_i$ we can illustrate the search space as follows:

$$\begin{array}{ccccccc}
\multicolumn{2}{c}{\text{First-k}} & & & & & \text{Full} \\
x_1, s_1^1 & x_1, s_1^2 & \dots & x_1, s_1^j & \dots & x_1, s_1^k & x_1 \\
 & & \vdots & & \vdots & & \\
x_2, s_2^1 & x_2, s_2^2 & \dots & x_2, s_2^j & \dots & x_2, s_2^k & x_2 \\
 & & \vdots & & \vdots & & \\
x_i, s_i^1 & x_i, s_i^2 & \dots & x_i, s_i^j & \dots & x_i, s_i^k & x_i \\
 & & \vdots & & \vdots & & \\
x_n, s_n^1 & x_n, s_n^2 & \dots & x_n, s_n^j & \dots & x_n, s_n^k & x_n
\end{array}$$

where s_i^k is a snippet from $\mathbf{x}_i$. Note that we can use a function to generate snippets systematically, e.g., based on domain knowledge. For example, in text classification one can use natural language processing techniques to create document summaries as snippets for the documents. A well thought function to generate the search space in $\mathcal{C}$ can greatly reduce the computations needed to find the best document-snippet pair.

5.2.8 Other Considerations. An optimal AAL requires more computations and resources. A naive implementation may increase the computational complexity of the approach. We point to some additional considerations when using look-ahead strategies:

- Some models allow incremental training, and potentially reduce the overhead of retraining a classifier for every possible label. For example, some implementation of naive bayes classifiers allow training in batches. However, incremental training is only available on some models.
- The expected utility incurs in the same computations whether the approach uses full documents or all possible snippets, because snippets are only used to obtain the labels and only full documents are added to $\mathcal{L}$. A careful implementation of expected utility can reuse utility values for snippets of the same document. Furthermore, some computations may be skipped based on similarity of the instances. However, this

consideration depends on the domain.

- Ideally, a large validation set $\mathcal{V}$ is available during training, however, this may not be case. The initial labeled set may be split into validation and bootstrap to compensate for the lack of validation data. Depending on the test measure, an unsupervised utility measure may be available that can be computed on the unlabeled set.

- The choice of utility function should, however, be aligned to the performance measure the underlying classifier is being tested on. This allows the active learner to directly optimize over the target measure. Our formulation allows to plug-in a significant measure according to the task.

5.3 Experimental Evaluation

In this section, we describe the datasets, evaluation methodology, provide details of simulations, and define the baseline methods used to evaluate our strategy.

5.3.1 Datasets. We ran our experiments on four text classification datasets. (1) **Amazon fine foods:** is a collection of food product reviews from amazon.com with positive and negative sentiment collected by [56], we subsample reviews with ratings greater and less than three stars. (2) **Arxiv:** is a collection of abstracts from research papers published in the arxiv.org website under artificial intelligence (cs.AI) and machine learning (stat.ML) categories. (3) **IMDB:** is a collection of movie reviews from imdb.com labeled with positive and negative sentiment [53]. (4) **20News:** is a subset of the 20 newsgroups dataset that includes documents from *alt.atheism* and *talk.religion.misc* categories [65]. Table 5.1 summarizes the datasets characteristics.

We pre-process these datasets using a binary bag-of-words representation and stemmed one-grams. We eliminate all terms that appear in less than five documents in the corpus, and we eliminate all empty documents. We only use the body of the documents, ignoring subject lines and titles, to make the classification task more realistic.

Table 5.1. Dataset statistics for document-snippet pair selection. Number of documents in training set (Train), label distribution (Dist.), average number of sentences per document (Sent/doc), average number of words per sentence (Words/sent), average number of words per document (Words/doc), and total number of sentences in the data (Num.Sent). Standard deviation included (±)

Dataset	Train	Dist.	Sent/doc	Words/sent	Words/doc	Num. Sent.
Amazon	164K	50%	5 (±4)	16 (±13)	82 (±8)	842 K
Arxiv - ML	6.9K	49%	6 (±2)	23 (±10)	144 (±54)	44 K
IMDB	25K	50%	12 (±9)	19 (±13)	232 (±172)	311 K
20NG - Religion	828	43%	13 (±28)	16 (±2)	214 (±498)	10 K

5.3.2 Evaluation Methodology. We compute the accuracy of each method by average over five random train-test splits of the data. Our experiments are simulated as follows:

5.3.2.1 Oracle Simulation. For our large scale experiments, we use a trained multinomial naive bayes classifier to simulate an oracle. We used all available training data to induce a classifier using default parameters. When the oracle is asked a label, we returned the predicted label from this classifier, thus producing a noisy response depending on the instance. Like with a human oracle, the simulated oracle only has access to the snippet selected by the active learning method to produce a label.

To simulate the oracle neutral response, we use the oracle confidence on a query to return a neutral label. If the confidence on a label ($\max P_{\mathcal{L}}(y|\mathbf{x})$) is lower than a set threshold T the oracle will return a neutral label; otherwise, we return the predicted label according to the classifier. We tested high and low neutral rate levels using $T = \{0.7, 0.6\}$. For example, for a threshold $T = 0.7$ the oracle should have a confidence higher than 0.7 to return a label.

5.3.2.2 Student. The underlying classifier tested and reported in the learning curves is a multinomial naive bayes classifier with default parameters, trained on the queried documents at each iteration. Use a multinomial naive bayes classifier to model the oracle probability distribution on the snippets P_E; this classifier is trained on an additional labeled set of the bootstrap snippets, and the queries and their corresponding labels.

Our baselines use a utility-cost ratio formulation to select document and fixed snippet strategies to select the queries. The methods use variable cost, and fixed interruption methods for snippet selection. The following are the baselines to our proposed work:

First-k: These methods select snippets composed of the first k sentences of the document. We used values of k in the $[1, 5]$ range. These methods have a reduced and fixed search space.

Uncertainty-k: These methods apply a fixed interruption to an uncertainty sampling strategy. In this case, the documents are selected based on the uncertainty of the underlying classifier, and the snippet is fixed to be the first k sentences.

Our optimal anytime active learning method, OPTIMAL-AAL, selects document-snippet pairs according to Equation 5.1. We randomly subsampled the unlabeled instances to a pool of 25 candidate documents to allow faster computation of utilities. We compute the expected utility of a candidate document using the available labeled data. The utility is computed as the average measure of utility improvement and cost ratio on a held-out validation set. We use half of the bootstrap labeled instances as our validation set. Note that the baselines use the full bootstrap set as the seed of active learning, whereas OPTIMAL-AAL only uses half of that. All methods use a 300 instance bootstrap.

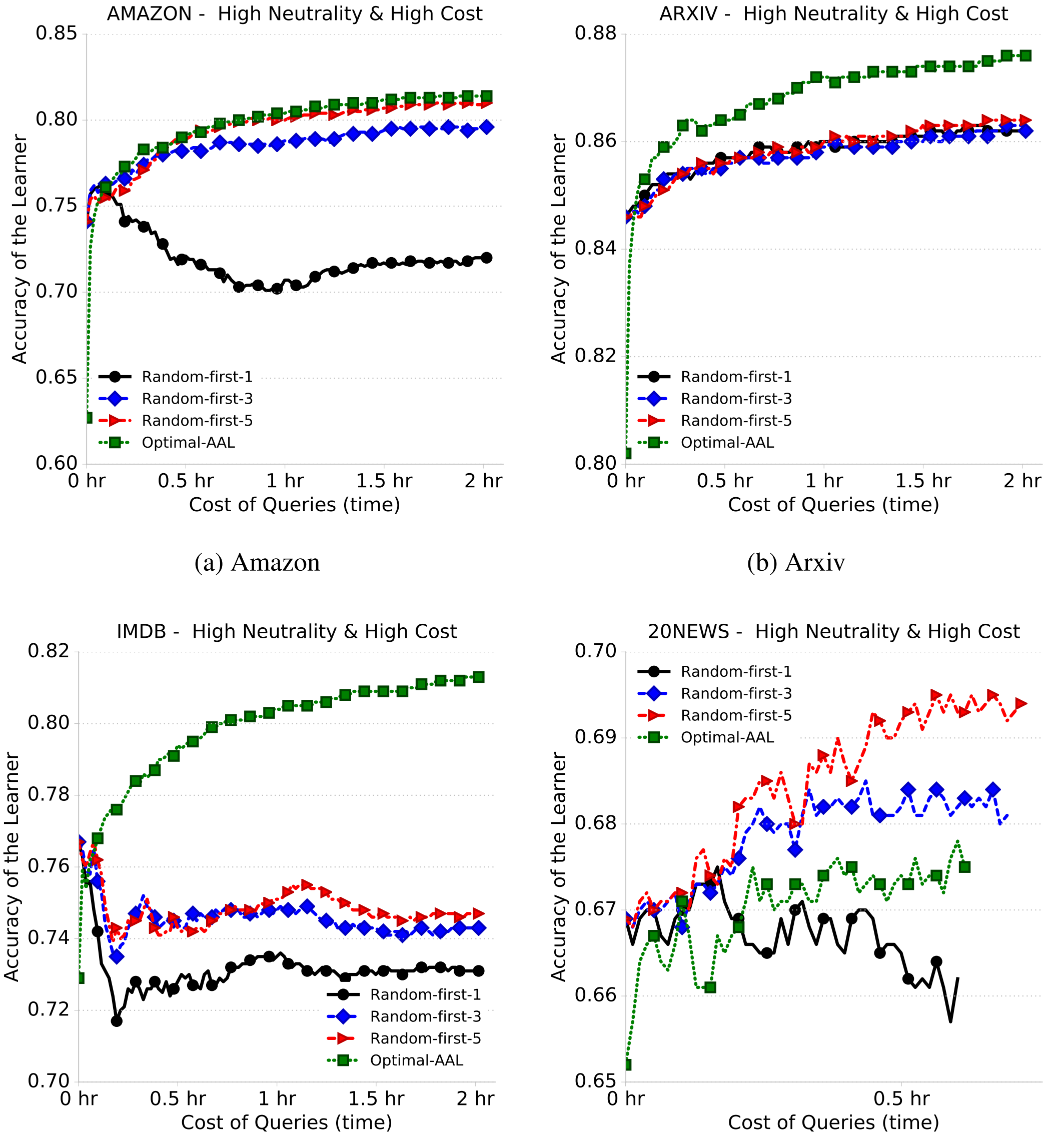
All methods use the same heuristic to generate the possible snippet of a document. We obtain all sentences in a document and generate snippets by sliding a size window of size k. In our proposed method, we generate snippets of size $k = 1$ sentence, $k = 2$

sentences, up to $k = 5$ sentences.

5.4 Results

In this section, we will discuss the experiment results comparing our proposed method OPTIMAL-AAL with the defined baselines. We will discuss our research questions and present the relevant results to those questions.

5.4.1 How does OPTIMAL-AAL Compare to Random Fixed-K Methods? An easy way to determine whether there is actual learning is to use a random sampling strategy. We defined FIRST-K as a basic baseline to compare our active strategy. Figure 5.2 shows the student performance on all datasets when the cost function is low (parameter is $A = 0.5$), and the neutrality is low (confidence threshold $T = 0.6$). Similar results were observed for the remaining cost-neutrality combination scenarios we tested. In most cases, our proposed method is able to outperform all FIRST-K baselines as expected.



(a) Amazon

(b) Arxiv

(c) IMDB

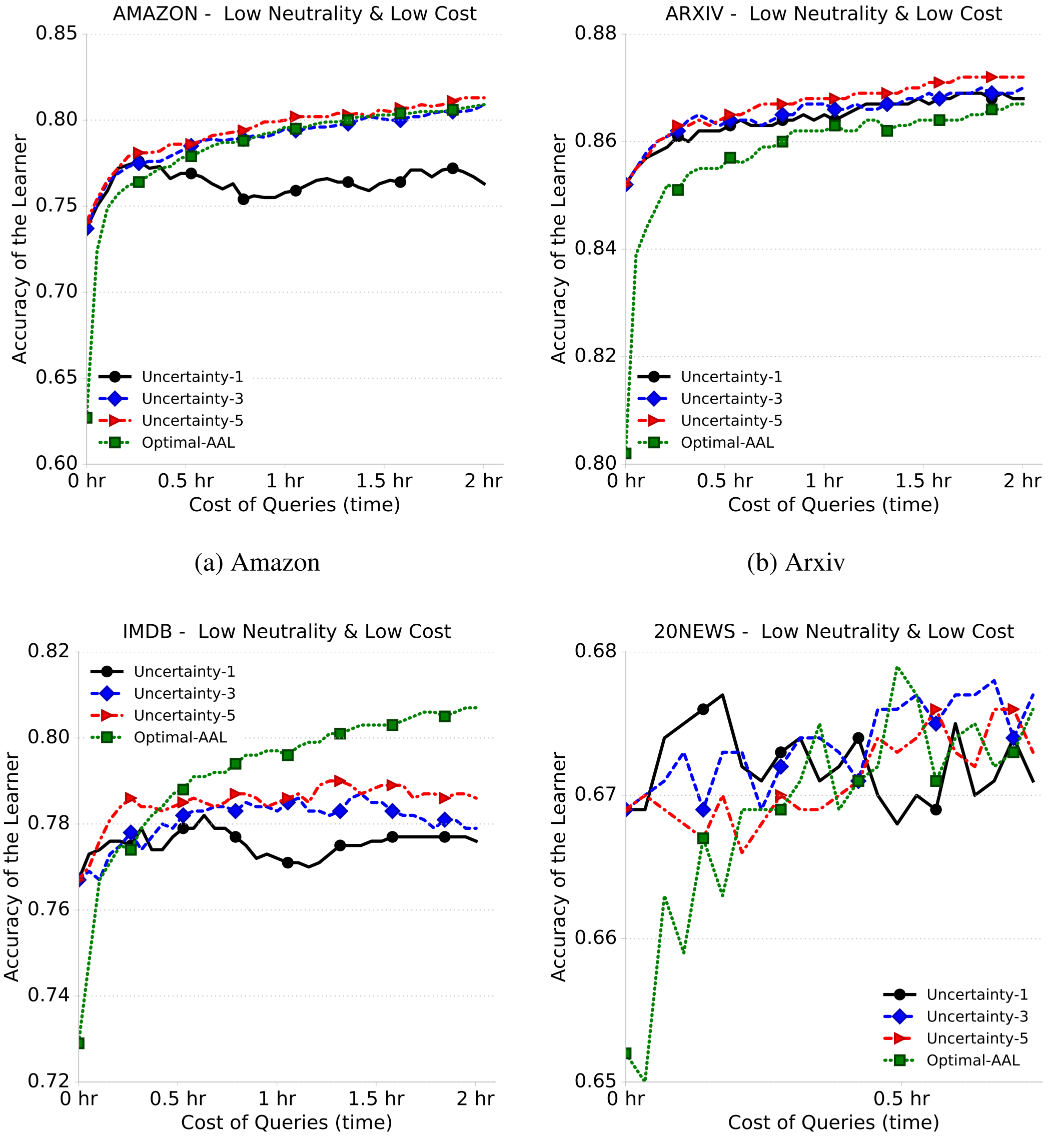
(d) 20 newsgroups

Figure 5.2. Comparing OPTIMAL-AAL to baselines on four datasets on low cost and low neutrality.

5.4.2 How does OPTIMAL-AAL Compare to other Expected Utility Methods?

We further contrasted our proposed method with active strategies. We compared UNC-K baselines using fixed values of k from 1 to 5 to form the snippet shown to the oracle. Consistent with results of previous chapter (Chapter 4), we observed that the best performing k varies across neutrality and cost conditions. In general, we observe that OPTIMAL-AAL outperforms or is comparable to UNC-K. OPTIMAL-AAL is able to effectively find the best snippet of an arbitrary size and improve learning efficiency. Figure 5.3 shows the performance of the learner for UNC-K and OPTIMAL-AAL on all four datasets with a high cost ($A = 3$) and high neutrality level ($T = 0.7$). OPTIMAL-AAL outperform the baselines on IMDB, and is comparable on Amazon and 20NG. Furthermore, on IMDB OPTIMAL-AAL is able to improve over the baselines ever when the baselines have reached a plateau. An advantage is that OPTIMAL-AAL does not need to set a fixed snippet size in advance in contrast to the baselines.

However, we note that OPTIMAL-AAL has a lower starting point in the learning curve because the method only uses half of bootstrap for training, giving an advantage at the beginning in favor of the baselines by starting with a better classifier. Despite this disadvantage, our method is still able to quickly catch up with the baselines and continue learning. Furthermore, when evaluating how well each method optimizes the utility performance measure, OPTIMAL-AAL outperforms or is comparable to the baselines on most cases (14 out of 16 cases), as expected because it is directly optimizing the measure. For example, Figure 5.4 shows the utility measure of all methods on Arxiv dataset where OPTIMAL-AAL is comparable to the baselines, even though other performance measures are not better. Further analysis shows that even though uncertainty is not designed to optimize the measure, it is able to produce good results. This may be due to a good underlying model in general.



(a) Amazon

(b) Arxiv

(c) IMDB

(d) 20 newsgroups

Figure 5.3. Comparing OPTIMAL-AAL to baselines on four datasets.

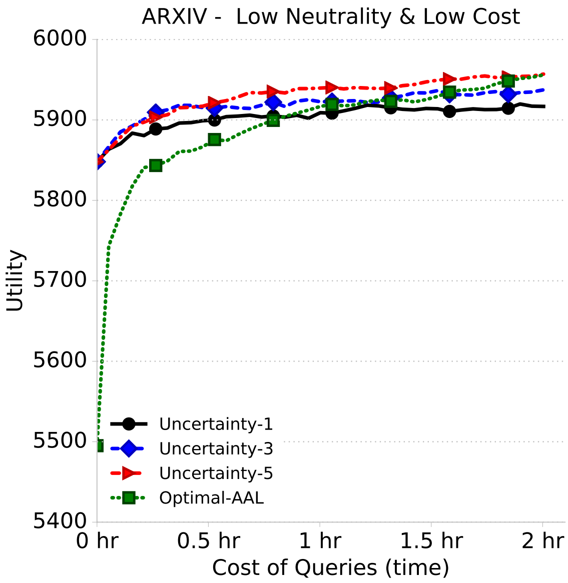


Figure 5.4. Learner utility measure on Arxiv dataset with $T = 0.6$ and $A = 0.5$. OPTIMAL-AAL is able to outperform the baselines.

5.4.3 How Does Length of the Document Affect the Snippet Selection?

Table 5.2 shows the average number of queries and the average snippet size per method per dataset, when using cost parameter $A = 3$, and neutrality threshold $T = 0.7$. We found that typically, OPTIMAL-AAL selects smaller snippets than its uncertainty counterpart, with the exception of result in Amazon data. When comparing the number of queries asked to the oracle, OPTIMAL-AAL is able to request more labels than the baselines, consistent with selecting smaller snippets. However, on Amazon data OPTIMAL-AAL is able to request more labels than the baseline although longer snippets. This may be because the distribution of snippet size for OPTIMAL-AAL is more long-tailed than for uncertainty methods. Further analysis also shows that UNC-5 produces two percent points more neutral responses than our method. This tells us that OPTIMAL-AAL is able to request better document-snippet pairs than the baseline.

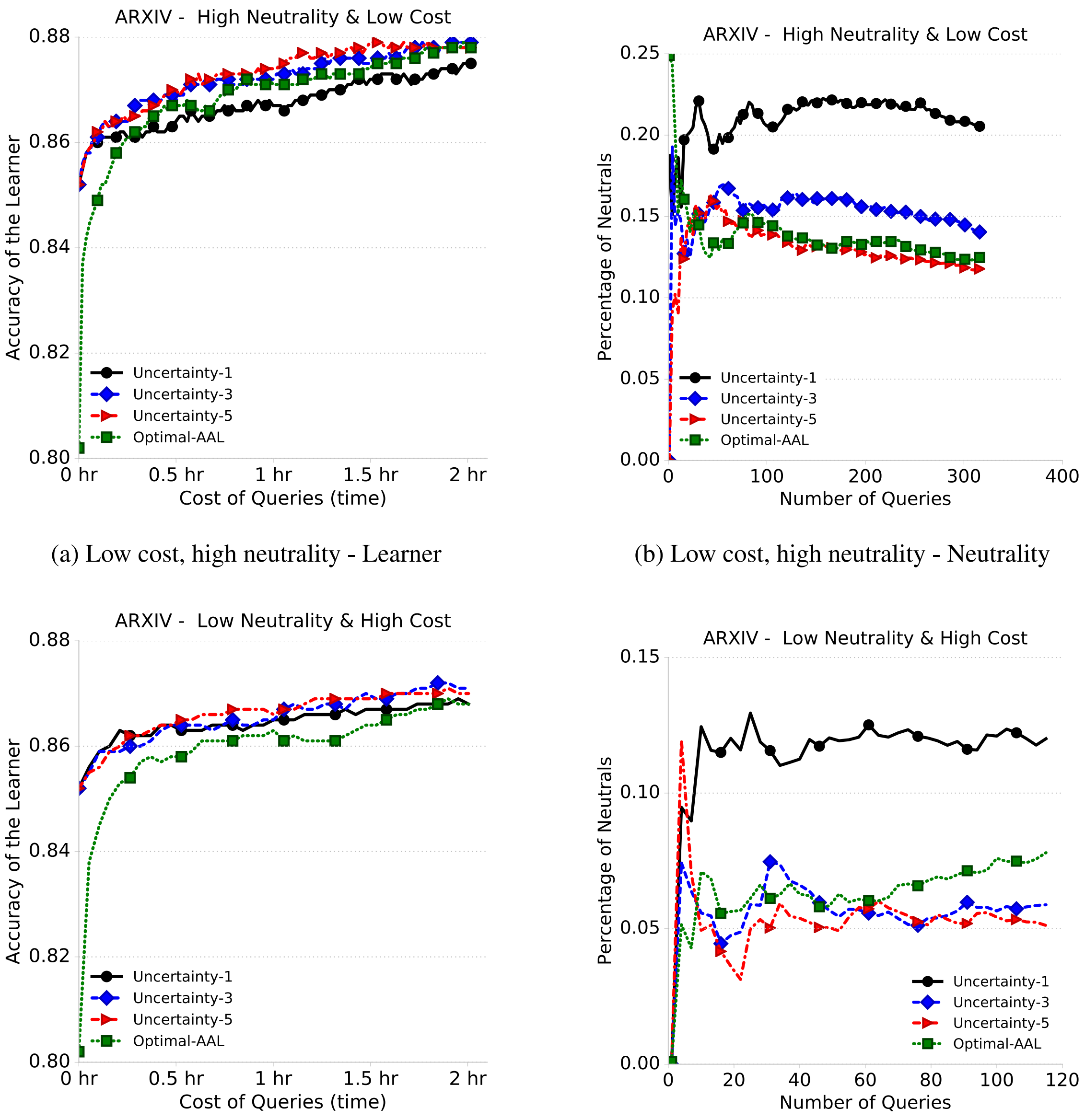
5.4.4 How Does Cost and Response Rate Affect the Learning Efficiency of the Student?

We tested four scenarios as the combination of high and low neutrality and cost. In

Table 5.2. Average snippet size and number of queries

Dataset	Number of Queries		Snippet Size (words)	
	OPTIMAL-AAL	UNC-5	OPTIMAL-AAL	UNC-5
Amazon-Food	640	240	12.1 (± 10)	9.7 (± 6)
Arxiv - ML	520	460	18.3 (± 9)	20.5 (± 9)
IMDB	720	620	11.9 (± 11)	14.1 (± 12)
20NG - Religion	1400	520	14.7 (± 21)	16.4 (± 14)

general, a high neutrality rate affects all methods slowing down the learning curve. In all cases, the neutral response is consistent throughout the tested settings. However, the effect of cost is mixed depending on the domain. Figure 5.5 shows the student performance on Arxiv dataset with two cost and neutrality settings. Figure 5.5a and Figure 5.5c show that higher neutrality and cost negatively affects OPTIMAL-AAL compared to the baselines. For example, when the cost is low, difference in cost per snippet size is smaller (cost with $A = 0.5$), OPTIMAL-AAL performs better than when the difference in cost is larger (cost with $A = 3.0$). However, UNC-K is also affected by the low cost making its learning curve less efficient. In contrast, Amazon dataset shows a different effect of cost. Figure 5.6 shows that OPTIMAL-AAL performs better when the cost is high, and the cost difference among snippet sizes is larger. We conjecture that the differences may be because Arxiv dataset documents have less number of sentences in average compared to Amazon.



(a) Low cost, high neutrality - Learner

(b) Low cost, high neutrality - Neutrality

(c) High cost, low neutrality - Learner

(d) High cost, low neutrality - Neutrality

Figure 5.5. Effect of cost and response rate on active learning methods on Arxiv dataset. Low cost uses $A = 0.5$ and high cost uses $A = 3$; high neutral response rate uses $T = 0.7$, and low response rate uses $T = 0.6$.

5.4.5 How does the Validation Set Affect Learning Efficiency? One challenge when using a look-ahead method is the use of a validation set in order to calculate the expected utility of future queries. We tested various alternatives for $\mathcal{V}$ using cross-validation on the training set, splitting the bootstrap instances, and a disjoint held-out validation set. Overall, held-out data provided the best results among the methods tested; other methods tend to overfit the validation set if $\mathcal{V}$ is too small. Figure 5.7 shows the effect of size in the average performance of OPTIMAL-AAL. As the size of the held-out data increases, the overall performance of the classifier increases as well. Additional challenges emerge from finding held-out data for validation and will be left for future work.

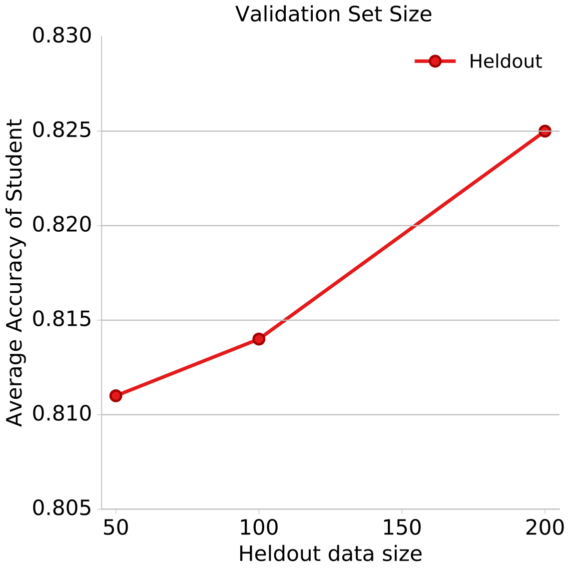
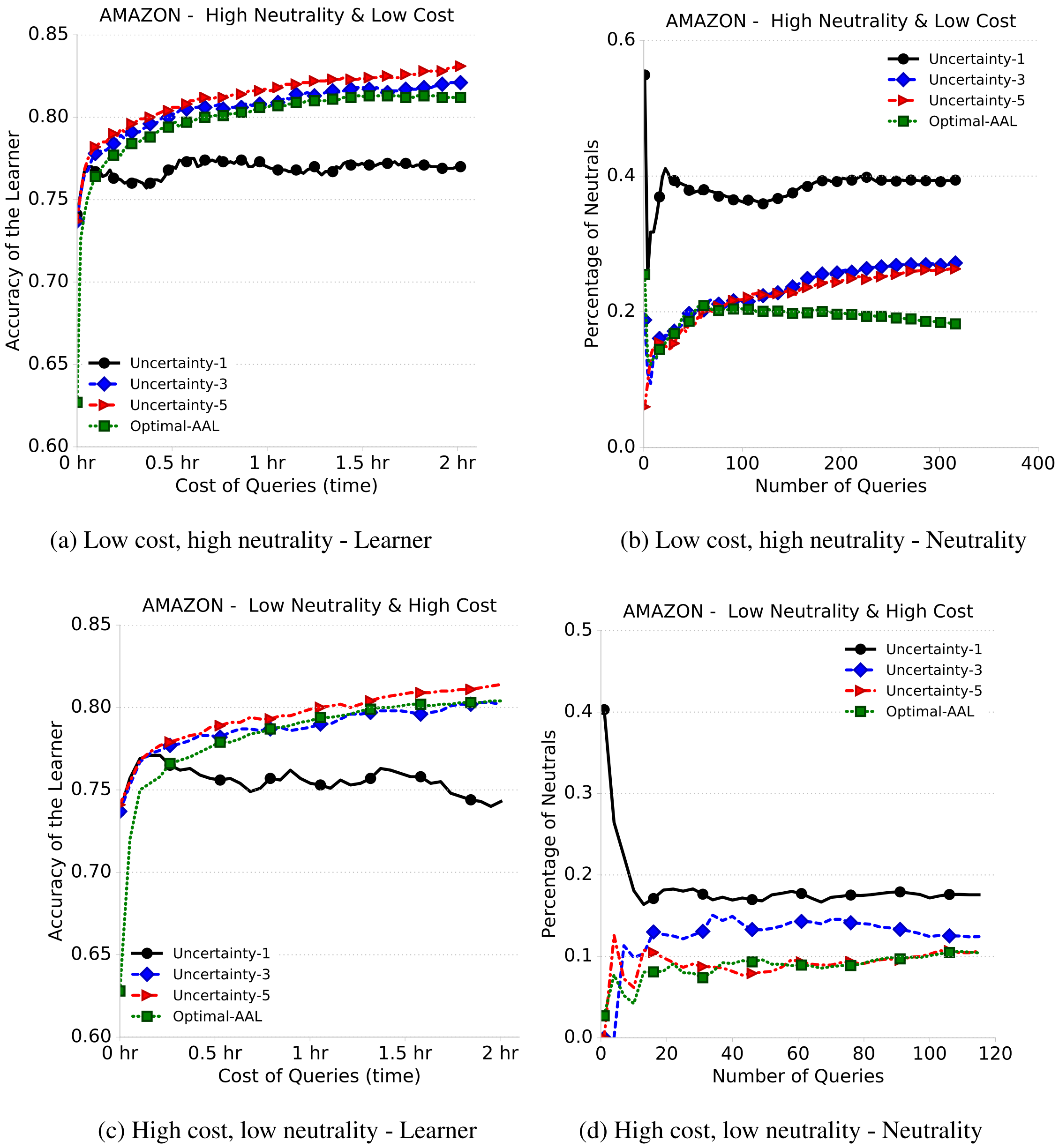


Figure 5.7. Effect of validation set size in OPTIMAL-AAL performance.

5.5 Chapter Conclusions

Anytime active learning methods enhance the abilities of the learning algorithm to control the time an oracle spends on a given query. We combined an optimal search of instances for annotation and the best snippet representation to obtain the labels. We presented empirical results comparing the proposed method with commonly used active learning methods, and showed that in general, our learning algorithm is able to find balanced document-snippet pairs improving learning efficiency.



(a) Low cost, high neutrality - Learner

(b) Low cost, high neutrality - Neutrality

(c) High cost, low neutrality - Learner

(d) High cost, low neutrality - Neutrality

Figure 5.6. Effect of cost and response rate on active learning methods on Amazon dataset. A parameter controls cost function (higher means steeper cost function), and T parameter controls neutral rate (higher means more neutral responses)

CHAPTER 6
CONCLUSION

In this thesis, we discussed the general problem of cost-sensitive active learning and introduced the concept of interrupting the oracle during annotation time to induce annotation savings. Here, we summarize the contributions of this body of work, describe future research directions, and conclude.

6.1 Summary

In this document, we discussed oracle interruption as a way to accelerate annotation during active learning. First, we introduced a scenario where the active learning truncates documents as a proxy for interrupting during annotation time. Our contributions for this proposed framework are:

- **Anytime active learning through document truncation:** We presented a novel active learning framework for sequential annotation scenarios. In this framework, the learner interrupts the expert during annotation time and requests the best guess up until that point. We use interruption at the first k words with fixed and dynamic values.

- **User studies:** We conducted user studies to analyze the effect of interrupting the oracle during annotation, and informed large scale active learning experiments.

Second, we presented a more complex scenario where the learner can only extract information from the document to show the oracle without having had a choice to select a document. For example, this applies to situations where all examples need to be annotated. In this scenario, our main contributions are:

- **Faster annotation through snippet extraction:** We presented a framework to actively select document snippets to show an oracle for annotation. In this framework, the learner presents condensed versions of queries and trades off the risk of forcing the oracle to err and the cost of annotation. We used a technique similar to summarization to build the condensed queries.
- **Empirical results:** We presented an extensive analysis of the effects of selecting snippets to accelerate annotation and how the method behaved under various stressing conditions of label noise and neutral response rate.

Third and final, we proposed a composite method that jointly selects the best document-snippet pair for faster annotation. From this section the main contributions are:

- **Document-snippet pair Selection:** We provided an intuitive formulation to select snippets, where the snippets not only convey the whole instance label but also are likely to be correctly labeled by the expert.
- **Empirical results:** We showed through our experiments that selecting document-snippet pairs produce significant savings for learning.

6.2 Future Work on AAL

This work presents many interesting research directions on anytime active learning. In this section we discuss four possible avenues in this section.

6.2.1 Self-Contained Evaluation. Our experiments showed that held-out data provided the best results when estimating the expected utility. However, acquiring a separate validation dataset is not always possible and may require equal effort to annotate as the active learning itself. In addition, cross-validation on the labeled set is computationally expensive and at higher risk of overfitting the underlying model if the data is too small. An interesting research path to remedy these problems is a self-contained method to define a proper utility measure for AAL.

6.2.2 Complex Oracle Models. Our formulation presents a general approach applied on text classification task. Even though our formulation allows application on any domain for which an interruption can be defined, a proper model of the oracle response to snippet queries is vital for the overall performance. Some annotation tasks such as sentiment analysis have been proved difficult to learn because of factors such as irony and context, which are challenging to represent in a general simple model.

An interesting direction is to elaborate a model to select snippets more effectively by considering context. For example, when selecting a sentence, the model may consider the surrounding sentences as context to calibrate the probability of response from the oracle [57]. Furthermore, the model can incorporate additional information regarding the position of the sentence, and value more the first and last sentences because they may contain more information [103].

6.2.3 Crowdsourcing. Thus far, we have explored active learning scenarios where there is only one oracle available for annotation. A natural progression in complexity for this scenario is to consider various oracles with variable cost, expertise, and response rates. This scenario is particularly helpful for domains where crowdsourcing is a suitable annotation source. Additional considerations include: (1) balancing workload of the oracles and benefit-cost ratio of the queries[96]; and (2) allowing the oracle to be trained prior to the active learning loop. The learner can decide what type of queries are best for an oracle, and introduce the oracle to the interruptions gradually.

6.2.4 Application on Feature-based Data Domains. Our proposed methods have been extensively tested using text classification tasks. However, selecting snippets or summarizing a feature-based instance is not a trivial task. In general, the active learning cycle develops as an interaction between a learning algorithm and a human expert. We observed that during annotation, the expert focuses on distinctive features of an example more than others to identify the label. For instance, when a fraud analyst reviews an insurance claim, he/she will check first for specific features such as type of claim, amount insured, or history of claims. If necessary, the analyst will review in detail every item in the client file before emitting a concept. Note that different cases may require different sets of features to be reviewed in detail, depending on the difficulty of the case, or amount of information known about the claim. Further research in this area will address how to provide an anytime learner with capacity to affect annotation time. Some research has been done to select relevant features that are significant to human oracles [63]. Exploring the use of these methods is an interesting step in generalizing anytime active learning methods for faster annotation.

6.3 Conclusion

In this document, we introduced a novel active learning approach that provides the active learner with enhanced capabilities to control annotation budget by interrupting the oracle during annotation time. We discussed three scenarios where interruption, in the form of truncation or more thoughtful snippets, is used to accelerate annotation. We discussed how practical factors influence the performance of the methods and present empirical evidence of our findings. The rate at which we are able to annotate data, fundamental resource to build accurate machine learning models, is greatly overwhelmed by the rate at which we are able to obtain the unlabeled data; methods to accelerate annotation or reduce annotation cost go a long way towards obtaining the best machine learning model possible.

BIBLIOGRAPHY

[1] Abe, N. and Mamitsuka, H. (1998). Query learning strategies using boosting and bagging. In *Proceedings of the 15th International Conference on Machine Learning*, volume 1 of *ICML'98*, pages 1–9.

[2] Abe, N., Zadrozny, B., and Langford, J. (2006). Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 504–509.

[3] Angluin, D. (2001). Queries revisited. In *International Conference on Algorithmic Learning Theory*, pages 12–31.

[4] Angluin, D. (2004). Queries revisited. *Theoretical Computer Science*, 313(2):175–194.

[5] Banko, M. and Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 26–33.

[6] Barzilay, R. and Elhadad, M. (1999). Using lexical chains for text summarization. *Advances in automatic text summarization*, pages 111–121.

[7] Baxendale, P. B. (1958). Machine-made index for technical literature: an experiment. *IBM Journal of Research and Development*, 2(4):354–361.

[8] Beineke, P., Hastie, T., Manning, C., and Vaithyanathan, S. (2004). Exploring sentiment summarization.

[9] Bilgic, M. (2012). Combining active learning and dynamic dimensionality reduction. In *SIAM International Conference on Data Mining*, SDM '12, pages 696–707.

[10] Bilgic, M. and Getoor, L. (2010). Active inference for collective classification. In *Proceedings of the 24th Conference on Artificial Intelligence (AAAI NECTAR Track)*, AAAI '10, pages 1652–1655.

[11] Bilgic, M., Mihalkova, L., and Getoor, L. (2010). Active learning for networked data. In *Proceedings of the 27th International Conference on Machine Learning*, ICML '10.

[12] Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media, Inc.

[13] Bobicev, V. and Sokolova, M. (2008). An effective and robust method for short text classification. In *Proceedings of the 23rd Conference on Artificial Intelligence*, AAAI'08, pages 1444–1445.

[14] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.

[15] Brinker, K. (2003). Incorporating diversity in active learning with support vector machines. In *Proceedings of the International Conference on Machine Learning*, volume 3, pages 59–66.

[16] Chao, C., Cakmak, M., and Thomaz, A. L. (2010). Transparent active learning for robots. In *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction*, HRI '10, pages 317–324.

[17] Chen, M., Jin, X., and Shen, D. (2011). Short text classification improved by learning multi-granularity topics. In *Proceedings of the International Joint Conference on Artificial Intelligence*, IJCAI '11, pages 1776–1781.

[18] Cohn, D., Atlas, L., and Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

[19] Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145.

[20] Culotta, A. and McCallum, A. (2005). Reducing labeling effort for structured prediction tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 5 of *AAAI '05*, pages 746–751.

[21] Dagan, I. and Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. In *Proceedings of the International Conference on Machine Learning*, ICML '95, pages 150–157.

[22] Dietterich, T. G. (2000). *Proceedings of First International Workshop on Multiple Classifier Systems*, chapter Ensemble Methods in Machine Learning, pages 1–15. MCS '00. Springer Berlin Heidelberg.

[23] Donmez, P. and Carbonell, J. G. (2008). Proactive learning: Cost-sensitive active learning with multiple imperfect oracles. In *Proceeding of the 17th ACM conference on Information and Knowledge Mining*, CIKM '08, pages 619–628.

[24] Donmez, P., Carbonell, J. G., and Schneider, J. (2009). Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 259–268.

[25] Du, J. and Ling, C. X. (2010). Active Learning with Human-Like Noisy Oracle. In *2010 IEEE International Conference on Data Mining*, pages 797–802.

[26] Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285.

[27] Ertekin, S., Huang, J., Bottou, L., and Giles, L. (2007). Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 127–136. ACM.

[28] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

[29] Fang, M., Zhu, X., and Zhang, C. (2012). Active learning from oracle with knowledge blind spot. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, AAAI '12, pages 2421–2422.

[30] Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139.

[31] Freund, Y., Schapire, R. E., et al. (1996). Experiments with a new boosting algorithm. In *Proceedings of the International Conference on Machine Learning*, volume 96, pages 148–156.

[32] Freund, Y., Seung, H. S., Shamir, E., and Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2):133–168.

[33] Fu, Y., Zhu, X., and Li, B. (2012). A survey on instance selection for active learning. *Knowledge and Information Systems*, 35(2):249–283.

[34] Guo, Y. and Schuurmans, D. (2008). Discriminative batch mode active learning. In *Advances in neural information processing systems*, NIPS '08, pages 593–600.

[35] Haertel, R., Ringger, E., Seppi, K., Carroll, J., and Mcclanahan, P. (2008). Assessing the costs of sampling methods in active learning for annotation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 65–68.

[36] Hoi, S. C., Jin, R., and Lyu, M. R. (2006a). Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th International Conference on World Wide Web*, pages 633–642.

[37] Hoi, S. C., Jin, R., Zhu, J., and Lyu, M. R. (2006b). Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd International conference on Machine learning*, pages 417–424.

[38] Howard, R. A. (1966). Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26.

[39] Hsueh, P.-Y., Melville, P., and Sindhwani, V. (2009). Data Quality from Crowdsourcing : A Study of Annotation Selection Criteria. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*, HLT'09, pages 27–35. Association for Computational Linguistics.

[40] Kapoor, A., Horvitz, E., and Basu, S. (2007). Selective supervision: Guiding supervised learning with decision-theoretic active learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 7 of *IJCAI '07*, pages 877–882.

[41] Katayama, T., Utsuro, T., Sato, Y., Yoshinaka, T., Kawada, Y., and Fukuhara, T. (2009). An empirical study on selective sampling in active learning for splog detection. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, pages 29–36.

[42] King, R. D., Rowland, J., Oliver, S. G., Young, M., Aubrey, W., Byrne, E., Liakata, M., Markham, M., Pir, P., Soldatova, L. N., et al. (2009). The automation of science. *Science*, 324(5923):85–89.

[43] Komurlu, C. and Bilgic, M. (2016). Active inference and dynamic gaussian bayesian networks for battery optimization in wireless sensor networks. In *Proceedings of AAAI Workshop on Artificial Intelligence for Smart Grids and Smart Buildings*.

[44] Komurlu, C., Shao, J., and Bilgic, M. (2014). Dynamic bayesian network modeling of vascularization in engineered tissues. In *Proceedings of the 11th UAI Workshop on Bayesian Modeling Applications*.

[45] Kupiec, J., Pedersen, J., and Chen, F. (1995). A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '95, pages 68–73.

[46] Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, ICML '01, pages 282–289.

[47] Larsen, B. (1999). A trainable summarizer with knowledge acquired from robust NLP techniques. *Advances in Automatic Text Summarization*, page 71.

[48] Lewis, D. D. and Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 3–12.

[49] Lin, C. H., Mausam, and Weld, D. S. (2014). To Re(label), or Not To Re(label). In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing*, HCOMP'14.

[50] Lindenbaum, M., Markovich, S., Rusakov, D., et al. (1999). Selective sampling for nearest neighbor classifiers. In *Proceedings of the National Conference on Artificial Intelligence*, pages 366–371.

[51] Liu, A., Jun, G., and Ghosh, J. (2009). Spatially cost-sensitive active learning. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, SDM '09, pages 814–825.

[52] Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

[53] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150.

[54] Mahapatra, D., Schüffler, P. J., Tielbeek, J. A., Vos, F. M., and Buhmann, J. M. (2013). Semi-supervised and active learning for automatic segmentation of crohn's disease. In *Medical Image Computing and Computer-Assisted Intervention*, MICCAI '13, pages 214–221. Springer.

[55] Mazzoni, D., Wagstaff, K., and Burl, M. (2006). Active learning with irrelevant examples. In *Proceedings of the European Conference on Machine Learning*, ECML '06, pages 695–702.

[56] McAuley, J., Pandey, R., and Leskovec, J. (2006). Inferring networks of substitutable and complementary products. In *Proceedings of the European Conference on Machine Learning*, pages 695–702.

[57] Mcdonald, R., Hannan, K., Neylon, T., Wells, M., and Reynar, J. (2007). Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, page 432.

[58] Melville, P. and Mooney, R. J. (2004). Diverse ensembles for active learning. In *Proceedings of the International Conference on Machine Learning*, ICML '04, pages 584–591.

[59] Melville, P., Saar-Tsechansky, M., Provost, F., and Mooney, R. (2005). An expected utility approach to active feature-value acquisition. In *Proceedings of 15th IEEE International Conference on Data Mining*, ICDM '05, pages 745–748. IEEE.

[60] Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. (1998). Learning to classify text from labeled and unlabeled documents. In *Proceedings of the 15th National/10th Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, pages 792–799.

[61] Ramirez-Loaiza, M. E., Culotta, A., and Bilgic, M. (2013). Towards anytime active learning: Interrupting experts to reduce annotation costs. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*, IDEA '13, pages 87–94.

[62] Ramirez-Loaiza, M. E., Culotta, A., and Bilgic, M. (2014). Anytime active learning. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, AAAI '14, pages 2048–2054.

[63] Rashidi, P. and Cook, D. J. (2011). Ask me better questions: Active learning queries based on rule induction. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 904–912.

[64] Rattigan, M., Maier, M., and Jensen, D. (2007). Exploiting network structure for active inference in collective classification. In *ICDM Workshop on Mining Graphs and Complex Structures*, pages 429–434.

[65] Rennie, J. (2008). 20 newsgroups.

[66] Roy, N. and McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the International Conference on Machine Learning*, ICML '01, pages 441–448.

[67] Saar-Tsechansky, M., Melville, P., and Provost, F. (2009). Active feature-value acquisition. *Management Science*, 55(4):664–684.

[68] Sassano, M. (2002). An empirical study of active learning with support vector machines for japanese word segmentation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 505–512.

[69] Schein, A. I. and Ungar, L. H. (2007). Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265.

[70] Schütze, H., Velipasaoglu, E., and Pedersen, J. O. (2006). Performance thresholding in practical text classification. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 662–671.

[71] Sculley, D. (2007). Online active learning methods for fast label-efficient spam filtering. In *Conference on Email and Anti-Spam (CEAS).*

[72] Segal, R., Markowitz, T., and Arnold, W. (2006). Fast uncertainty sampling for labeling large e-mail corpora. In *Conference on Email and Anti-Spam.*

[73] Settles, B. (2012). *Active learning*, volume 6. Morgan & Claypool Publishers.

[74] Settles, B. and Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079.

[75] Settles, B., Craven, M., and Friedland, L. (2008). Active learning with real annotation costs. In *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, pages 1–10.

[76] Seung, H. S., Opper, M., and Sompolinsky, H. (1992). Query by committee. In *Proceedings of the 5th ACM Annual Workshop on Computational Learning Theory*, COLT '92, pages 287–294.

[77] Shannon, C. E. (1948). A mathematical theory of communication. *Bell Systems Technical Journal*, 27:379–423,623–656.

[78] Sharma, M. and Bilgic, M. (2013). Most-surely vs. least-surely uncertain. In *Proceedings of the IEEE 13th International Conference on Data Mining*, ICDM '13, pages 667–676.

[79] Sharma, M., Zhuang, D., and Bilgic, M. (2015). Active learning with rationales for text classification. In *North American Chapter of the Association for Computational Linguistics - Human Language Technologies*, pages 441–451.

[80] Shen, D., Chen, Z., Yang, Q., Zeng, H.-J., Zhang, B., Lu, Y., and Ma, W.-Y. (2004). Web-page classification through summarization. In *Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval*, SIGIR '04, page 242.

[81] Sheng, V. S. and Ling, C. X. (2006). Feature value acquisition in testing: a sequential batch test algorithm. In *Proceedings of the International Conference on Machine Learning*, ICML '06, pages 809–816.

[82] Sheng, V. S., Provost, F., and Ipeirotis, P. G. (2008). Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 614–622.

[83] Sriram, B. and Fuhry, D. (2010). Short text classification in twitter to improve information filtering. *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 841–842.

[84] Sun, A. (2012). Short text classification using very few words. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 1145–1146.

[85] Täckström, O. and McDonald, R. (2011). Discovering fine-grained sentiment with latent variable structured prediction models. In *Advances in Information Retrieval*, pages 368–374. Springer.

[86] Thompson, C. A., Califf, M. E., and Mooney, R. J. (1999). Active learning for natural language parsing and information extraction. In *Proceedings of International Conference on Machine Learning*, pages 406–414.

[87] Titov, I. and McDonald, R. T. (2008). A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*, pages 308–316.

[88] Tomanek, K. and Hahn, U. (2009). Semi-supervised active learning for sequence labeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 2, pages 1039–1047.

[89] Tomanek, K. and Hahn, U. (2010). A comparison of models for cost-sensitive active learning. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1247–1255.

[90] Tomanek, K. and Olsson, F. (2009). A web survey on the use of active learning to support annotation of text data. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, HLT '09, pages 45–48.

[91] Tong, S. and Chang, E. (2001). Support vector machine active learning for image retrieval. In *Proceedings of the 9th ACM International Conference on Multimedia*, MULTIMEDIA '01, pages 107–118.

[92] Vijayanarasimhan, S. and Grauman, K. (2009). What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '09, pages 2262–2269. IEEE.

[93] Vijayanarasimhan, S. and Grauman, K. (2011). Cost-sensitive active visual category learning. *International Journal of Computer Vision*, 91(1):24–44.

[94] Wallace, B., Trikalinos, T., Lau, J., Brodley, C., and Schmid, C. (2010a). Semi-automated screening of biomedical citations for systematic reviews. *BMC bioinformatics*, 11(1):55.

[95] Wallace, B. C., Small, K., Brodley, C. E., and Trikalinos, T. A. (2010b). Active learning for biomedical citation screening. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 173–182.

[96] Wallace, B. C., Small, K., Brodley, C. E., and Trikalinos, T. A. (2011). Who should label what? Instance allocation in multiple expert active learning. In *Proceedings of the SIAM International Conference on Data Mining*, SDM '11, pages 176–187.

[97] Whitehill, J., Wu, T. T.-f., Bergsma, J., Movellan, J. R., Ruvolo, P. L., Wu, T. T.-f., Bergsma, J., and Movellan, J. R. (2009). Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems*, 22(1):1–9.

[98] Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354. Association for Computational Linguistics.

[99] Xu, Z., Akella, R., and Zhang, Y. (2007). Incorporating diversity and density in active learning for relevance feedback. In *Proceedings of the 29th European Conference on IR Research*, ECIR '07, pages 246–257.

[100] Xu, Z., Yu, K., Tresp, V., Xu, X., and Wang, J. (2003). Representative sampling for text classification using support vector machines. *Advances in Information Retrieval*, pages 11–11.

[101] Yan, Y., Fung, G., Moy, L., and Schmidt, M. (2010). Modeling annotator expertise: Learning when everybody knows a bit of something. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS-10)*, volume 9 of *AISTATS'10*, pages 932–939.

[102] Yan, Y., Fung, G. M., Rosales, R., and Dy, J. G. (2011). Active learning from crowds. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1161–1168.

[103] Yang, Y. and Nenkova, A. (2014). Detecting information-dense texts in multiple news domains. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*.

[104] Zelikovitz, S. and Hirsh, H. (2000). Improving short text classification using unlabeled background knowledge to assess document similarity. In *Proceedings of the seventeenth international conference on machine learning*, volume 2000, pages 1183–1190.

[105] Zhang, J. J. and Fung, P. (2012). Active learning with semi-automatic annotation for extractive speech summarization. *ACM Trans. Speech Lang. Process.*, 8(4):6:1–6:25.

[106] Zheng, Y., Scott, S., and Deng, K. (2010). Active learning from multiple noisy labelers with varied costs. In *Proceedings of 13th IEEE International Conference on Data Mining*, pages 639–648.

[107] Zhu, J. and Hovy, E. (2007). Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 783–790.