2017-08-01

The Annotation Cost of Context Switching: How Topic Models and Active Learning [May Not] Work Together

Nozomu Okuda
Brigham Young University

The Annotation Cost of Context Switching: How Topic Models and Active Learning [May Not] Work Together

Nozomu Okuda

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Kevin Seppi, Chair
Tony Martinez
Dennis Ng

Department of Computer Science

Brigham Young University

ABSTRACT

The Annotation Cost of Context Switching: How Topic Models and Active Learning [May Not] Work Together

Nozomu Okuda
Department of Computer Science, BYU
Master of Science

The labeling of language resources is a time consuming task, whether aided by machine learning or not. Much of the prior work in this area has focused on accelerating human annotation in the context of machine learning, yielding a variety of active learning approaches. Most of these attempt to lead an annotator to label the items which are most likely to improve the quality of an automated, machine learning-based model. These active learning approaches seek to understand the effect of item selection on the machine learning model, but give significantly less emphasis to the effect of item selection on the human annotator.

In this work, we consider a sentiment labeling task where existing, traditional active learning seems to have little or no value. We focus instead on the human annotator by ordering the items for better annotator efficiency.

Keywords: active learning, topic modeling, annotation, human cost

ACKNOWLEDGMENTS

Of course, the members of my thesis committee deserve to be acknowledged as part of my work, my advisor foremost. In addition, Dr. Eric Ringger, my former advisor, deserves to be acknowledged. Drs. Jordan Boyd-Graber and Leah Findlater also helped me in this work. Lab members helped in various ways, most directly involved being Connor Cook and much wisdom from Paul Felt and Jeff Lund. The university's supercomputing facilities were a great boon to my experiments. Many open source software projects made my life much easier. Finally, university staff, especially the CS secretaries, were helpful in logistics. This research was supported by NSF Grant IIS-1409739.

Table of Contents

Chapter 1

Introduction

The amount of text available for consumption grows at an ever increasing pace. Making sense of this deluge of information is an important but intimidating task.

In this work, we are interested in the document metadata labeling task. That is, based on the text of a given document, what label should be given to this document? More specifically, we focus on ordinal labeling systems (such as star ratings and dates). Sometimes, the label might be found explicitly in the document's text; however, the document metadata labeling task must also handle documents whose text does not explicitly contain the label.

While it would be ideal, from a label correctness point of view, to have multiple expert annotators carefully label all unlabeled documents, this is often infeasible, especially as the number of documents in a corpus of interest becomes large. To cope, we might hire experts to label some subset of the corpus and then train a machine learning model on the labeled data. The hope is that the machine learning model will be able to label the remainder of the documents with satisfactory accuracy—however that may be defined.

In accepting this machine learning approach to solving the document metadata labeling task, we are left with the issue of how to choose the corpus subset to be labeled. One way to consider this issue is to work from the machine learning model's side by choosing the subset that best improves the model's accuracy. This suggests we should investigate active learning approaches. Another way to consider this issue is to work from the human annotators' side by choosing the subset that best accommodates efficient annotation. This suggests we should consider reducing cognitive load by minimizing context switching.

This work makes three main points. First, despite the potential for active learning, we found through simulation experiments that a training set of randomly selected documents is the best choice for training our machine learning model (Chapter 3). Second, we show through a first user study that context switching increases the time humans take to label documents (Chapter 4). Third, we show through a second user study that we can order documents, even when randomly selected, so as to reduce context switching and thereby reduce the time needed to annotate a fixed set of documents. (Chapter 5). To preface these points, we review fundamental concepts in machine learning, including active learning and the supervised anchor words algorithm (Chapter 2). Based on the three main points, we conclude this work by suggesting a strategy for annotating documents that is akin to cost-conscious active learning approaches (Chapter 6).

Chapter 2

Machine Learning and Topic Modeling Concepts

As promised, we begin with a review of fundamental concepts in machine learning, with a special focus on topic modeling. These concepts lay the foundation for not only the design decisions for our experiments but also the conclusions we draw from our experiments.

2.1 Machine Learning

Supervised machine learning has traditionally been the method used to train a model to predict labels based on instance features [7]. Thus, a supervised machine learning algorithm takes instance features along with the instance's associated label as input. For example, an instance in a corpus would be a document, and the features of the document might include entries like the length of the document or whether a given word appeared in the document. Based on the instances and labels given, the algorithm trains a model. The output model can then be used to predict labels, given instance features.

Note that in this traditional machine learning paradigm, the order in which the algorithm considers instances is left unspecified. In practice, training instances are typically randomly ordered.

The goodness of a machine learning algorithm is usually evaluated by measuring its prediction accuracy on a test set which has been previously reserved from the labeled data, after training on the other labeled data.[1] The higher the accuracy, the better the model. Of course, by random chance, it is possible that the model performs unusually well because of

[1]This concept is so fundamental to machine learning that it inevitably appears in every machine learning textbook, including Mitchell's [12], Marsland's [8], and Murphy's [13].

the items chosen to be in the test and training sets. Thus, it is considered good practice to account for this by testing the algorithm on multiple different training and test sets. This is typically done by cross validation [6].

2.2 Active Learning

In contrast to traditional machine learning, active learning considers the scenario of incrementally increasing the amount of labeled data available [5, 18]. Active learning is a looping framework. It begins by training a model on some initially labeled data. It then uses the model to consider which labels of the items in the unlabeled data would be most beneficial to the model. Once this is determined, the selected items are given to a human for labeling. Once the human submits the labels for these items, a new model is trained, which can then be used to choose new items for labeling, and the loop continues in this manner until some stopping criteria are met. Typically, one of the stopping criteria is related to the human cost of labeling (e.g., time, money). Thus, active learning is also interested in minimizing the human cost for training a sufficiently good model. For these reasons, the goal of active learning can be thought of as how to most improve the model with the least human cost.

An active learning selection algorithm might try to select items for which the model has difficulty making correct predictions. It is possible to measure the difficulty a model has of correct prediction without knowing the true label by considering the instances seen thus far. The most common approaches include uncertainty sampling (choose the instance for which the model has the highest variance in predicting a label) and query by committee (choose the instance which has the least agreement of prediction labels from multiple trained models). For this work, we focus on diversity sampling, which is to diversify the training set instances by deliberately choosing unlabeled instances that seem least like the already labeled instances. Presumably, the model will have difficulty correctly predicting the labels of instances which little resemble the instances the model has used for training. In fact,

Sachan et al. have shown that diversity in the training set is important to the success of active learning [17].

2.3 Topic Models

We now review key concepts in topic modeling, as imagined in latent Dirichlet allocation (LDA) [2]. A topic is a probability distribution over vocabulary words. This technical definition relates to an intuitive notion of a topic in that the most probable words in a technical topic are words that we would expect to find in a text focused on an intuitive topic. For example, if the intuitive topic is "gardening", we would expect to see such words as "soil", "plant", "season", and so forth to appear frequently; so a technical topic representing this intuitive topic would assign higher probabilities to these words than to words such as "bacon", "highway", or "classroom".

Thus, a topic model attempts to discover some number K of these topics from an input corpus. Again, the input to a topic model is a collection of documents; the output of the topic model is K topics. Learning the topics from a corpus requires some assumptions. Besides topics being distributions of words, LDA makes another assumption important to a method discussed later in this work: every document has a topic composition. In other words, document d is composed of $x\%$ of topic 1, $y\%$ of topic 2, etc. Note that by definition, the sum of these percentages is equal to 100%. There is also the implication that one or more of the topics in document d may be 0%, meaning that those topics do not occur at all in document d.

Since none of the documents are labeled with topics, this formulation of topic modeling is an unsupervised algorithm.

2.4 Anchor Words

With that review of topic modeling, we are ready to consider the anchor words algorithm [1]. The anchor words algorithm takes concepts from LDA and applies a simplifying assumption

that results in a comparatively efficient algorithm for topic modeling. Thus, like LDA, the anchor words algorithm is an unsupervised algorithm that takes a corpus of documents as input and finds topics present in them.

When originally published, the anchor words algorithm to performed a matrix decomposition that led to precision problems (a matrix inversion was required). However, the idea that the probability of the cooccurrences of words being related to the second order moment matrix of the topic matrix multiplied by the per document word count matrix, originally exploited by the matrix decomposition technique, led to the current probabilistic approach, which although eschewing direct matrix factorization, is able to recover the parts the matrix decomposition technique could, now with the precision issues greatly diminished.

Instead of belaboring every step in the anchor words algorithm, we note the importance of the conditional word cooccurrence probability matrix, $\bar{Q}$, in preparation for explaining the supervised anchor words algorithm and the supervised anchor words algorithm for regression. Essentially, $\bar{Q}$ encodes the characteristics of words in the corpus. First, consider numbering all words in the vocabulary, from 1 to V. In $\bar{Q}$, the ith word is represented as the ith row of the matrix. The jth element in the ith row is the conditional probability of seeing the jth word given that the ith word is in the document: $p(w_j \mid w_i)$. Thus $\bar{Q}$ is a $V \times V$ matrix which characterizes a word by how other words cooccur with it (the word).[2]

Using the $\bar{Q}$ matrix, the anchor words algorithm can discover the eponymous anchor words and then use both $\bar{Q}$ and the anchor words to calculate topics.[3] Note that because the anchor words algorithm relies on $\bar{Q}$ to infer topics from a corpus, word cooccurrences in the documents of the corpus determine the topics. Note also that this process does not give us topic compositions of documents, but we can compute these topic compositions relatively cheaply with the topics calculated from the anchor words algorithm.

[2]Using word cooccurrence information to characterize words is not a novel concept [3], although the probably most famous instance in current natural language processing, word2vec [11], was published the year after the anchor words paper [1].

[3]Further details on the anchor words algorithm can be found in both the anchor words paper [1] and the supervised anchor words paper [14].

2.5 Supervised Anchor Words

The supervised anchor words algorithm [14] incorporates document label information into calculating topics by augmenting the $\bar{Q}$ matrix with extra columns. Each extra column represents a label. Thus, the element at row i on the jth extra column is the conditional probability $p(l_j \mid w_i)$, the probability of the document having the jth label given that we have seen the ith word of the vocabulary in the document. This augmented matrix is called S.

The topics inferred from S can then be used to infer the topic compositions of documents. These topic compositions are used as instance features for a classifier. Thus, the supervised anchor words algorithm can produce label-aware topics and a classifier, given a corpus of labeled documents.

Note that the supervised anchor words algorithm is useful for classification tasks. Accordingly, the supervised anchor words paper [14] reports results exclusively on binary sentiment analysis.

2.6 Supervised Anchor Words for Regression

Nguyen proposed a variation on the supervised anchor words algorithm in his Ph.D. proposal [15], which allows for regression (i.e., the prediction of a real number value).

We justify using this regression model because we are interested in the document metadata labeling task with ordinal labels (e.g., star rating prediction and date prediction). Regression inherently accounts for the natural ordering of ordinal values. It could be argued that because ordinal labels are discrete, classification would be more appropriate (classification deals with discrete classes), but classification does not account for the ordering of the labels. In contrast, regression accounts for the ordering of the labels but is meant for predicting continuous values. We can easily overcome regression's non-discrete nature by choosing points along the continuum to represent a given discrete label and then being satisfied with the regression model's prediction if the prediction is within some distance from the correct label's

point. Following Walker's lead [19], we normalize all document labels to the range of $[0, 1]$. Thus, on a five star rating scale, a one star rating gets a label of 0, a five star rating gets a label of 1, and the other star ratings are scaled within this range.

We now describe the supervised anchor words algorithm for regression, which modifies the supervised anchor words algorithm's S matrix construction. Instead of adding columns for each possible label as in the original supervised anchor words algorithm, only two columns are added. The first additional column is the weighted average label value for each word in the vocabulary. Step by step, the ith value in this first additional column is computed as follows:

1. For each document, count up the number of times the ith vocabulary term appears (these counts are not summed; each count is kept separate, according to its associated document)
2. For each document, multiply the label given to the document by the number of times the ith vocabulary term appears in the document
3. Sum all of these multiplied terms together
4. Divide the sum by the total number of times the ith word of the vocabulary appears in the corpus.

Note that because we imposed the restriction that all labels are in the range $[0, 1]$, all values in this first additional column will also be in the range $[0, 1]$. Each value in the second additional column is one minus the corresponding value in the first additional column. In other words, the ith value in the second additional column is equal to one minus the ith value in the first additional column. The other parts of the algorithm remain unchanged.

We have found subjectively that this algorithm performs reasonably accurately. We came to this conclusion by comparing results from this algorithm with results from SLDA [9]. Although accuracy results were comparable, timing results were not: the supervised anchor words algorithm for regression completed experiments were faster than SLDA. This

is in line with the claims the supervised anchor words paper makes regarding its speed and accuracy performance in comparison with SLDA [14]. This speed up convinced us to to use this supervised anchor words algorithm for regression instead of SLDA.

2.7 Evaluation of Models

We have already reviewed the paradigm of splitting labeled data into test and training sets. This section will be devoted to explaining how to calculate the evaluation score, given a model and a test set.

Following the precedent set by McAuliffe and Blei [9], we will use predictive R^2 (pR^2), also known as the coefficient of determination, as one measurement for evaluating trained models. The most commonly known form of the coefficent of determination, R^2, has the property of being in the range $[0, 1]$ because R^2 deals with how well a regression model fits the data used to induce the model. In contrast, pR^2 can take any value in the range $(-\infty, 1]$, because data not used in inducing the model is used to compute the goodness of fit of the model. The definition of pR^2 is as follows:

$$\text{pR}^2 = 1 - \frac{\sum_d (y_d - \hat{y}_d)^2}{\sum_d (y_d - \bar{y})^2}, \tag{2.1}$$

where y_d is the true label value for document d, $\hat{y}_d$ is the predicted label value for document d, and $\bar{y}$ is the mean label value for all labels in the test set. Note that the numerator of the fraction is the squared prediction error and the denominator of the fraction is the variance of the set of interest. Thus, pR^2 should be close to one when either the prediction error is low or the variance of the test set is extremely high. For pR^2 to be a highly negative value, either the prediction error must be extremely high or the variance of the test set must be extremely small.

Due to the above quirks of pR^2, we also follow the example of Walker [19] in reporting the generalized zero-one loss, which is defined as follows:

$$\text{Zero-One} = \frac{1}{N} \sum_{d} \begin{cases} 1 & \text{if } |y_d - \hat{y}_d| < \Delta \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

where N is the number of instances in the test set, y_d is the true label for document d (just like in Eq. 2.1), $\hat{y}_d$ is the predicted label value for document d (likewise as in Eq. 2.1), and Δ is some predetermined threshold. In prose, the generalized zero-one loss says that the model should get credit when its prediction is off by some amount less than some threshold; otherwise, it gets no credit. Note that this measure of accuracy better fits the ideas in our discussion regarding why to use regression instead of classification for ordinal label tasks. However, generalized zero-one loss does not automatically account for the difficulty of predicting labels in the test set (although by adjusting Δ, we can manually account for difficulty); pR^2 accounts for the difficulty by including the variance of the test set as part of its calculation (the larger the variance, the more difficult we expect making correct predictions to be).

Chapter 3

Active Learning Exploration

Now that we have reviewed fundamental concepts and introduced our machine learning approach of choice (supervised anchor words for regression), we are ready to think about what subset of the data should be labeled in order to train a good model. Active learning seems like a good candidate for determining this. In this chapter, we describe our attempt at discerning the best active learning method for the task of interest. This description includes our experimental methodology and a discussion of the results from these experiments.[1] We find that labeling a random sampling of the documents is the best method for training a good model.

3.1 Exploration Methods

Before settling on the experimental design presented here, we tried many active learning approaches with SLDA [9] (as mentioned earlier). An abbreviated table of selection methods tried in the pre-exploration is found in Table 3.1. Further details are available in Appendix A. Based on these earlier experiments, we found two specific flavors of diversity selection to be promising active learning methods. Thus, for our experiments with supervised anchor words for regression, we decided to compare the two flavors of diversity selection and random selection, which is similar in spirit to traditional supervised machine learning. Since random selection is trivially easy to implement, we now discuss how to implement diversity selection.

[1] The code used for these experiments is available at `https://github.com/n0kuda/activetm`.

Selection Method	Description
uncertainty sampling	choose the unlabeled document the model predicts with the highest variance
least confidence	choose the unlabeled document whose kernel density estimate over its predicted values has the lowest peak
query by committee	choose the unlabeled document that, given a set of models, has the highest variance in mean prediction among the set of models
diversity sampling by response value *	choose the unlabeled document whose uncertainty is highest from among documents predicted to be in the same bin of the range of predicted values
diversity sampling by top topic *	choose the unlabeled document whose top topic is least represented in the labeling set
diversity sampling by topic composition *	choose the unlabeled document whose topic composition is least like the topic composition of the labeling set
diversity sampling by topic-weighted word distribution *	choose the unlabeled document whose topic-weighted word distribution is least like the topic-weighted word distributions of the labeled documents; a topic-weighted word distribution is calculated by weighting each topic (i.e., word distribution) by that topic's prevalence in the document (which can be found from the document's topic composition) and taking the sum of these weighted topics
diversity sampling by top topic with balanced centroid and JSD	choose the document that scores highest at being different from the labeled set but similar to the unlabeled set—so there are two centroids to calculate with in this scheme
selection by shortest/longest document	choose the shortest/longest document

Table 3.1: An abbreviated table of selection methods attempted in the pre-exploration. Where the selection method ends with *, multiple variants were attempted. More details are available in Appendix A.

In order to define diversity, we need a way to represent documents. We choose to represent documents as vectors. There are various ways to represent a document as a vector, but for this exploration, we focus on two ways. The first is to represent a document by the top topic. By top topic, we mean the topic with the highest percentage in the topic composition of a document. Recall that a topic is a probability distribution over vocabulary words. Thus, this representation is a vector in $\mathbb{R}^V$ (where V is the number of words in our vocabulary). Moreover, with the top topic representation, any one document is represented as only one of K possible vectors, since a topic model produces only K topics and every document analyzed with this topic model is composed of only these K topics.

Perhaps a more natural document representation is topic composition, which is a vector in $\mathbb{R}^K$. In this representation, the ith entry in the vector is the percentage of the ith topic in the topic composition of the document represented by the vector.

Now that we can represent documents as vectors, we should next be concerned with how to compare vectors. For if we can compare vectors, we can compare document representations. We choose to compare vectors using Jensen-Shannon divergence (JSD), which is defined as

$$JS(\vec{u} \parallel \vec{v}) = \frac{1}{2} KL\left(\vec{u} \parallel \frac{\vec{u}+\vec{v}}{2}\right) + \frac{1}{2} KL\left(\vec{v} \parallel \frac{\vec{u}+\vec{v}}{2}\right), \tag{3.1}$$

where

$$KL(\vec{x} \parallel \vec{z}) = \sum_i x_i \log \frac{x_i}{t_i}. \tag{3.2}$$

In prose, JSD is a symmetric version of the Kullback-Leibler divergence (KLD). KLD is a way of measuring how different one probability distribution is from another. Thus, the input vectors to these divergence equations must be normalized to be like probability distributions. Conveniently, the document representations we chose already are normalized.[2]

Finally, we need a way to compare one document against all documents in the labeled set. We make this comparison with one document against the centroid of the labeled set. In

[2]Obvious alternatives to JSD would be L1-norm and L2-norm. See Appendix A for further discussion on this matter.

other words, given one document representation, we compared this vector against the vector resulting from averaging the representations of all documents in the labeled set.[3]

With these tools, we build a simulated active learning environment. Within this environment, the labeled set begins with 20 documents. We then run the supervised anchor words algorithm for regression (with 20 topics) on the labeled set and evaluate the trained classifier (produced by the supervised anchor words algorithm for regression) on a held-out test set. An unlabeled document is then selected for labeling. The procedure for selecting the next document to be labeled is one of the following:

1. random selection (this is the baseline)
2. diversity selection by top topic
3. diversity selection by topic composition

The newly selected document is then taken out of the unlabeled set and added to the labeled set with its true label, and the active learning loop continues until the model trained on 300 labeled documents is evaluated. There are 1000 documents reserved in the test set. The initial partitioning of test, labeled, and unlabeled sets is determined randomly.[4]

Note that of the three selection methods, random selection is the cheapest in terms of computation. This is because the diversity selection methods must compute the per document topic compositions, the centroid of the labeled set, and the vector comparisons. In contrast, random selection does not perform these computations.

In order to cut down the computation time of the diversity selection methods, the simulation randomly chooses 500 unlabeled documents as candidates for selection. The topic composition and vector comparison computations are performed only for the documents in the labeled set and the candidate set. We justify this method of computational savings by

[3]It is also conceivable to compare a given document representation against the representation of each document in the labeled set separately. Again, see Appendix A for further discussion.

[4]If you are wondering why this process happens only once, it is because this process happens only once per run of experiments. If you were expecting repeated resampling of test and training sets, you will find this at the end of this section.

the intuitive argument that randomly selecting a subset of the unlabeled set will yield a distribution of documents in the candidate set that is similar to the unlabeled set, given that we sample enough documents. Justifying that 500 documents is enough is mostly an exercise in subjective judgment.

Also noteworthy is the decision to have each document selected for labeling to be given its true label. If diversity selection cannot improve the model faster than random selection in the ideal situation of perfect labeling, it is doubtful that diversity selection could improve the model faster than random selection in the unideal situation of imperfect labeling. This is because diversity selection uses label information to choose the next document to label. Thus, diversity selection should perform optimally when documents are labeled perfectly. In contrast, since random selection does not use document labels for document selection, random selection does not perform worse or better whether the label is correct or incorrect.

In order to have confidence in the evaluation results, we run the simulation for each selection method 100 times, making sure that the test set and initial labeled sets are the same for corresponding simulation runs of each method (e.g., the first run of random selection has the same test set and initial labeled set as the first run of diversity selection by top topic; likewise, the first run of diversity selection by topic composition has the same test and initial labeled set). Note that for each of these 100 times, all data is resampled.[5]

If active learning improves model learning efficiency purely from diversity of labeled documents, we expect the evaluation results from the diversity selection methods to have a greater rise in accuracy in fewer labeled documents than the random selection baseline. Alternatively, we may think of active learning in terms of a resource-constrained task. Knowing that we have resources only enough to label some number of documents, we would like to

[5]This method of evaluation is most similar to what Kohavi calls random subsampling [6]. His main argument against the use of random subsampling is that it can evaluate the classifier as having an accuracy that is not accurate of the classifier's true potential. In our case, we are not interested in measuring the classifier's true accuracy; we want to see the relative merits of using different selection methods in an active learning context. As such, the deficiencies of random subsampling should affect the classifier equally for the selection methods we are comparing, thereby not affecting the comparison of the selection methods. Thus, Kohavi's criticisms of random subsampling do not apply to our comparison of selection methods.

	source	size	rarewords	common	smalldoc
Amazon	Nguyen et al. [14]	39400	5	30000	5
Yelp	Nguyen et al. [14]	25459	5	20000	5
Chunked SotU	Dan Walker [19]	7134	5	1500	5
Frus	Allison Chaney[6]	64477	50	30000	5

Table 3.2: Corpus information. Source refers to where the corpus was obtained, size refers to the number of documents within a corpus, rarewords is the threshold used for rare word removal, common is the threshold for common words removal, and smalldoc is the threshold used for small document elimination.

know whether active learning will yield a better model after that number of documents has been labeled.

3.2 Corpora

For our experiments, we will investigate model performance on four datasets. They are 1) Amazon, 2) Yelp, 3) Chunked SotU, and 4) Frus. All contain English documents exclusively.

Each corpus has received some amount of preprocessing treatment, as shown in Table 3.2. The preprocessing options are stopword removal, rare word removal, common word removal, and elimination of small documents. The stopwords list comes from `http://www.ranks.nl/stopwords`, containing English stopwords. Rare words are those words which occur in only some specified number or less of documents. Common words are those words which occur in at least some specified number of documents. Finally, small document elimination takes out any documents left in the corpus which contain some specified number or less of words. The number of words contained in a document is determined after removal of stop, rare, and common words. The numbers specified for rare word removal, common word removal, and small document elimination are completely arbitrary.

Both the Amazon and Yelp datasets are reviews with star ratings as metadata labels. Both the Chunked SotU and Frus datasets are documents with publication dates. Thus these corpora present two major tasks: sentiment analysis and dating.

[6]She cites `https://history.state.gov/historicaldocuments` as the source.

3.3 Exploration Results

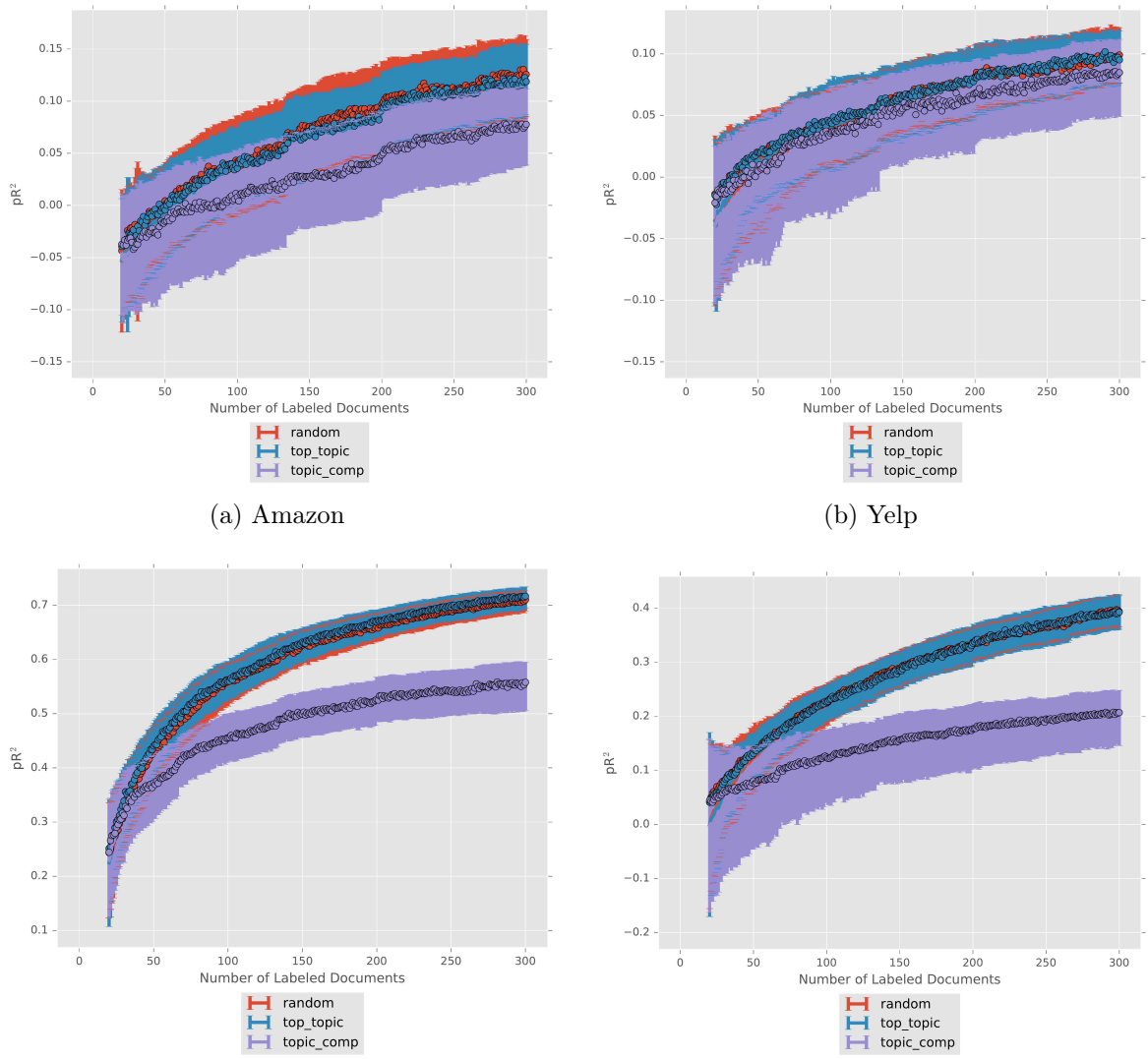
The results are shown in Figs. 3.1 and 3.2. For all of these figures, the x-axes count how many labeled documents were available to train the model. For Fig. 3.1, the y-axes measure the pR^2 scores the models were able to attain for a given number of labeled documents. For Fig. 3.2, the y-axes measure the generalized zero-one losses of the models. The colored circles with black outlines show median scores. The error bars show the first and third quartiles of the results per labeled set size. The labels corresponding with the colors tells which selection method was used. Thus, "random" refers to random selection, "top_topic" refers to diversity selection by top topic, and "topic_comp" refers to diversity selection by topic composition.

For the zero-one loss plots (Fig. 3.2), the threshold chosen for each dataset is noted in the y-axis label. For the Amazon and Yelp datasets, this threshold was chosen to be 0.1. As for the Chunked SotU and Frus datasets, the arbitrary threshold of 0.01 was chosen to say that if the prediction is close enough to the date of the true label, then it counts.[7]

Before making more specific comments about the plots, note the occlusion caused by the error bars, making discernment between selection methods difficult at points. This indicates that there is no clear winner among the hard to discern selection methods at those points. Where there is a clear distinction between selection methods, then it becomes clear which selection method loses.

While diversity selection by topic composition may have sounded more reasonable than diversity selection by top topic, we see from the results that in all plots except for the Amazon zero-one loss plot (Fig. 3.2a), top topic yields higher model results than topic composition. Why topic composition yields better results for zero-one loss on Amazon alone is still somewhat mysterious. However, we can see that topic composition performed no

[7]Note that whereas Amazon and Yelp have only five points along the range that mark correctness, Chunked SotU has 220 (Presidential addresses from 1790 to 2010) and Frus has potentially 15795 (number of days from 1945 to 1988). Thus, the zero-one threshold choice for Amazon and Yelp indicates an expectation that the five-label task will be easy. The zero-one threshold choice for Chunked SotU and Frus indicates an expectation that date prediction will be more difficult and therefore that some margin of error is acceptable: about two years for Chunked SotU and about five months for Frus.



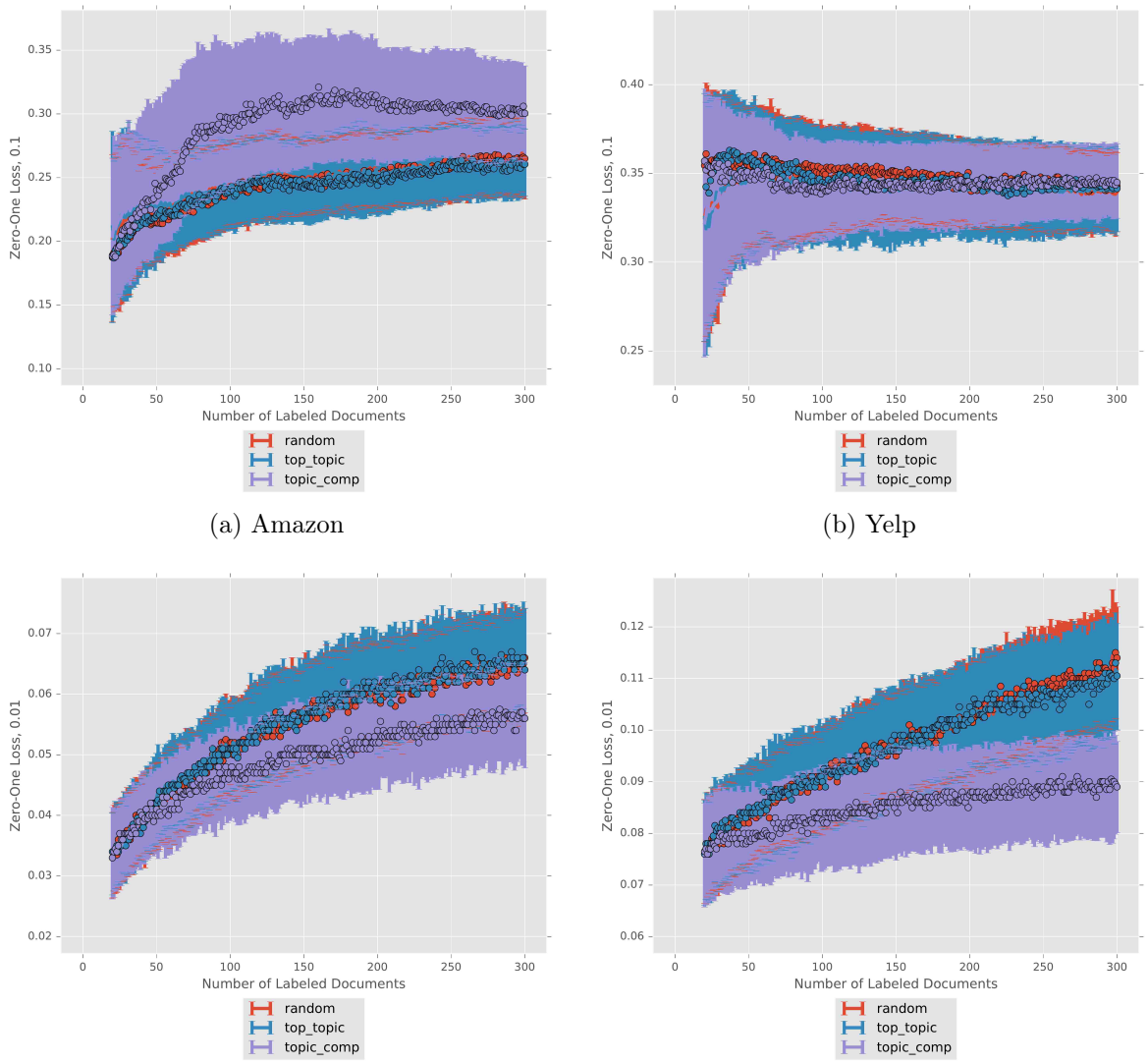
(a) Amazon

(b) Yelp

(c) Chunked SotU

(d) Frus

Figure 3.1: pR^2 results. "random" refers to random selection. "top_topic" refers to diversity selection by top topic. "topic_comp" refers to diversity selection by topic composition. Note that in all cases, "top_topic" occludes "random", making it hard to discern between the two and therefore implying that the two methods perform equivalently in terms of pR^2.



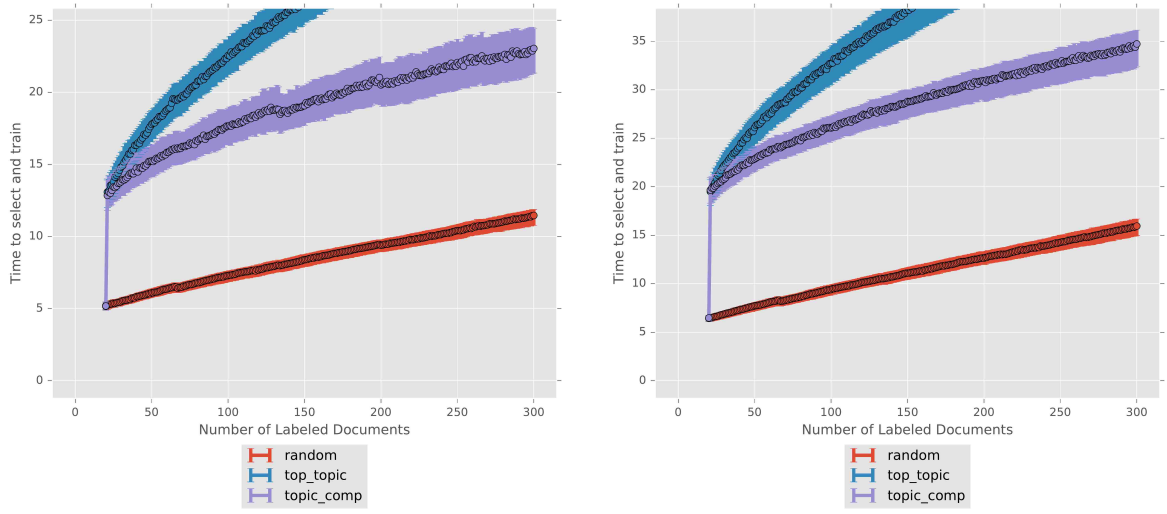
(a) Amazon

(b) Yelp

(c) Chunked SotU

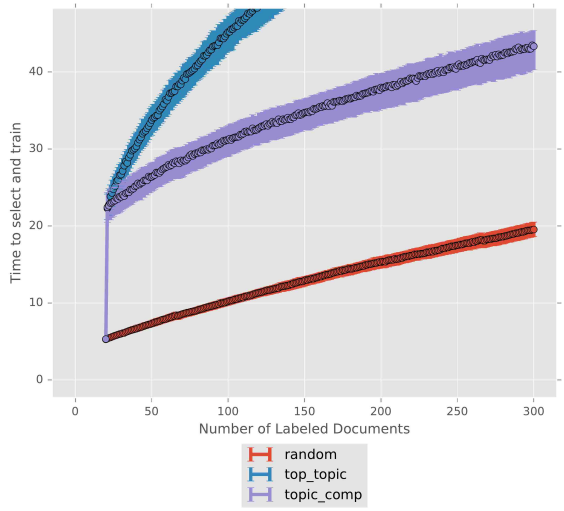
(d) Frus

Figure 3.2: Zero-one loss results. "random" refers to random selection. "top_topic" refers to diversity selection by top topic. "topic_comp" refers to diversity selection by topic composition. Note that in all cases, "top_topic" occludes "random", making it hard to discern between the two and therefore implying that the two methods perform equivalently in terms of zero-one loss.

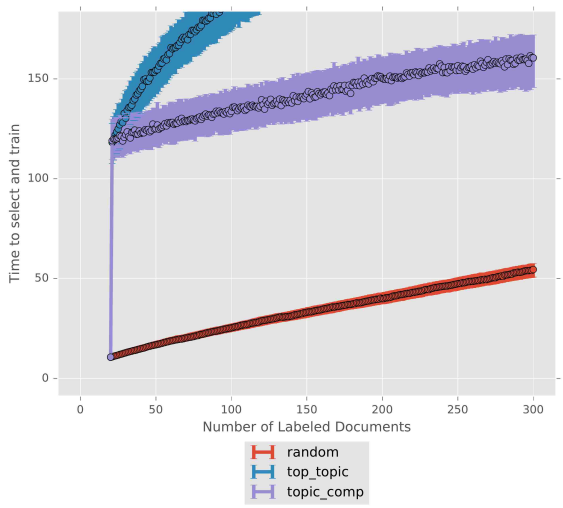


(a) Amazon

(b) Yelp



(c) Chunked SotU



(d) Frus

Figure 3.3: Time to complete training of model and selection of document to be labeled. These timings assume zero labeling time. Note that in all cases, random selection is the indisputable winner. Also interesting is how much longer top topic takes compared to topic composition; this is explainable in the fact that top topic is a larger vector (the size of the vocabulary, at least a couple thousand) whereas topic composition is a smaller vector (the number of topics, which is only 20).

better than top topic (or random) in the Yelp zero-one loss plot (Fig. 3.2b), whereas topic composition clearly performed worse than top topic (and random) in the other plots (Fig. 3.2c and Fig. 3.2d). Looking at the pR^2 plots (Fig. 3.1), we see that topic composition yields lower pR^2 than top topic (and random), although the topic composition does not lose by as much in the cases of Amazon and Yelp (Fig. 3.1a and Fig. 3.1b). These results suggest that the five-label task (which is the task for Amazon and Yelp) puts topic composition at less of a disadvantage than does the dating task (which is the task for Chunked SotU and Frus). Thus, in all but one case, top topic is a superior active learning method to topic composition.

Note also that diversity selection by top topic is no more effective than random selection. There are a few possibilities as to why this might be. First, deliberately selecting for diversity may be unnecessary. We already know from previous work that diversity is important to the success of active learning [17], but these results suggest that random selection chooses a diverse enough set of documents to train a good model. In other words, it may not be worth doing the extra work to select specifically for diversity. Second, topics and topic composition may not be good measures of diversity. Third, the computed topic compositions may not have been sufficiently correct. Fourth, the topics produced by the topic model may not have been correct. Whatever the reason, it stands that the extra computation necessary for diversity selection by top topic is more costly than random selection. We see this clearly in Fig. 3.3, which plots the combined training and selection times.

Based on these results, we can conclude that the diversity selection methods we tried in our exploration are no better at consistently improving a model in an active learning context than random selection. Moreover, because of the computational overhead of diversity selection, random selection is preferable.

Chapter 4

Context Switch User Study

The active learning exploration demonstrated that deliberate selection by diversity has mixed results for improving the model faster than random selection, with more cases favoring random selection. So it seems that the documents we ask annotators to label is best chosen at random. However, causing humans to undergo context switches as they label one document with a given top topic followed by another document with a different top topic, which is likely to occur often if the randomly chosen documents are revealed in the order they are chosen, may incur a time cost that could be avoided by minimizing the number of context switches. In order to see if there truly is an increase in time in labeling after a context switch, we recruited humans for a first user study.

4.1 Context Switch User Study Methods

For this first user study task, we asked participants to determine the star rating of 80 Amazon product reviews. When the review was written, the writer gave a star rating to indicate his or her opinion of the product, where one star indicated much dissatisfaction with the product, five stars indicated much satisfaction with the product, and star ratings in between meant satisfaction levels in between. Each review (with profanity removed, as best we could) was shown one at a time, and the participants were shown their progress on the task, including how closely the labels they gave matched the labels given by the review writers. Unbeknownst to the participants, the product reviews were ordered by top topic. That is, the first 16 reviews shared the same most prevalent topic, the next 16 share another same most prevalent

topic, and so forth. We will refer later to a given group of 16 documents with the same most prevalent topic as a topical group. The list of topical groups was randomly generated for every participant. More specifically, the topic for every topical group was chosen randomly. The documents within a topical group were also chosen randomly, with the restriction that they could be chosen only if their top topic matched the topic of the topical group and the participant had not previously labeled the document. By equating a topical group change to a context switch, we can use data collected from this user study to see what effect context switching has on labeling times.

The topic model determining the topic compositions of the reviews was computed beforehand using the anchor words method [1] which, as mentioned before, is an unsupervised algorithm. We chose to ignore the labels in the documents, since in the scenario we wished to simulate (namely, labeling an unlabeled corpus), labels would not be available even though the text is available. We chose to run the topic model on documents with at least 30 words, just as Nguyen et al. [14] did; therefore, only reviews with at least 30 words were shown to participants. We also removed words that occurred in less than 50 documents. In addition, we ran the topic model with $K = 80$, since this is the number of topics that Nguyen et al. [14] found to be optimal for the anchor words method.[1]

Finally, it should be noted that Poursabzi-Sangdeh et al. [16] performed a user study that measured the time it took humans to cluster documents with and without the help of a topic model. To determine which documents were shown on the user interface, a number of methods were tested, including according to topic composition, with special weight for the top topic. That user study bears similarity to this context switch user study if we think of documents which the user put in the same cluster as having been given the same label. However, the interface used in that user study showed multiple documents at once. In contrast, this context switch user study shows only one document at a time. This deliberate

[1]This treatment of the Amazon dataset is different from the Amazon dataset referred to earlier in the exploration experiments. We ran the exploration experiments with fewer topics so that experiment run times would be shorter; we were willing to wait the extra time once to compute the 80 topics for the user study, not the hundreds of times required for running experiments.

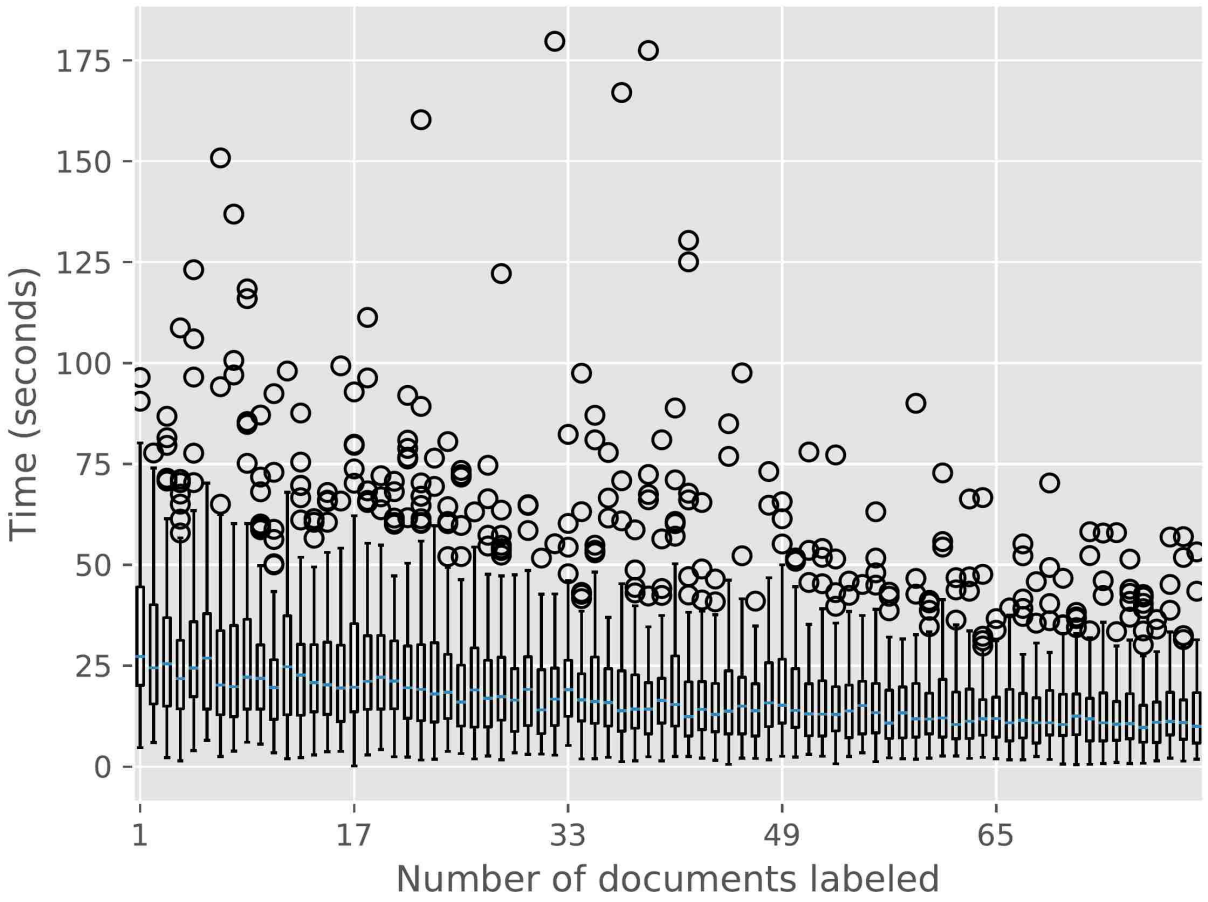


Figure 4.1: Aggregate labeling time results. Note that the x-axis ticks mark the beginnings of topical groups. As these are box plots, the middle blue bar indicates the median time to label, the rectangle marks the interquartile range, the bars extend over the range of times, excluding times that are quite far from the other times; these far away times are marked by the circles.

difference in user study design allows us to directly measure document labeling time, thus allowing us to use the results of this context switch user study to say something about how topic change affects the human time cost of labeling documents.

4.2 Context Switch User Study Results: Quantitative

Let us turn now to the labeling time data collected. We begin with Fig. 4.1. Recall our hypothesis: a context switch is more costly than no context switch; in addition, we consider a change in topical group to be a context switch. Fig. 4.1 shows aggregated labeling times

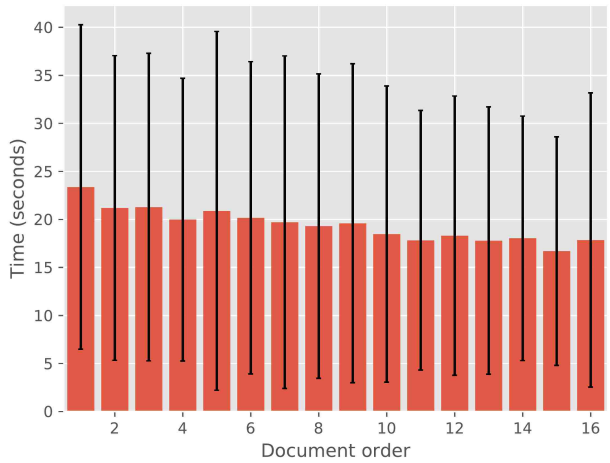
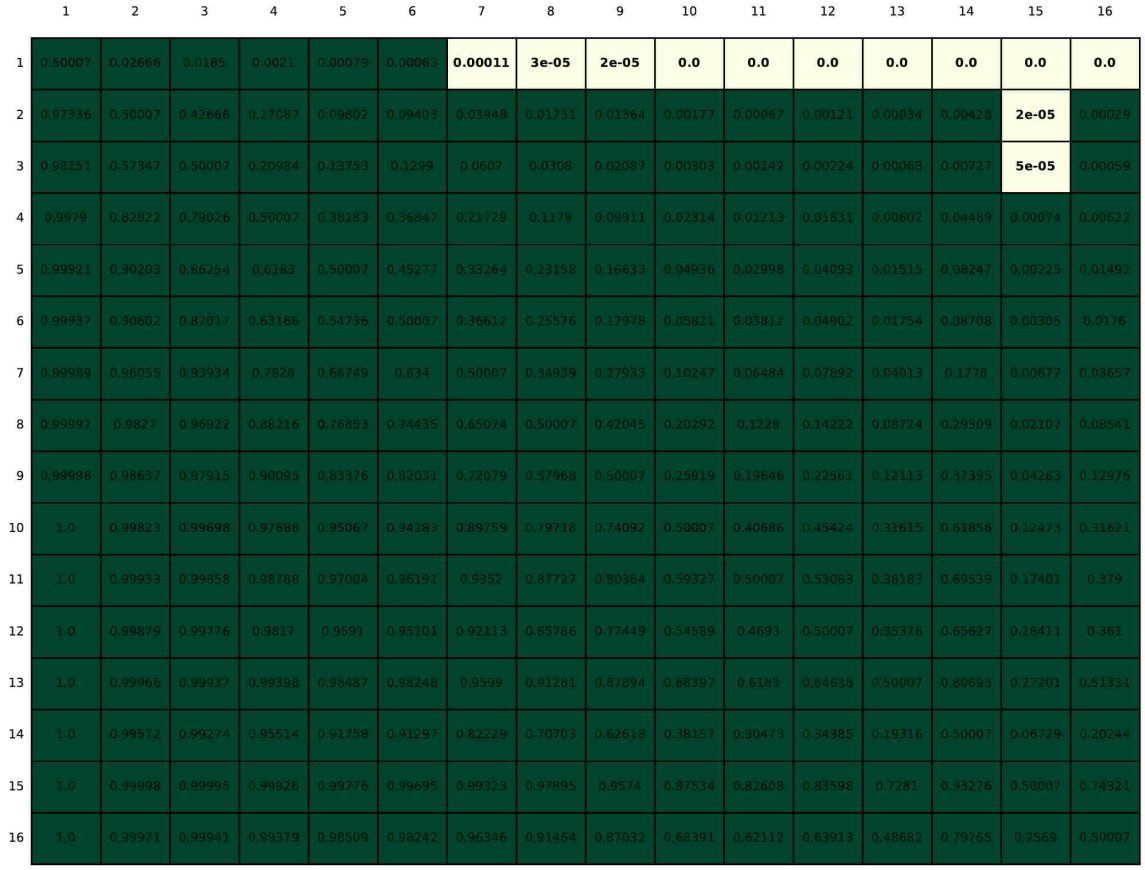


Figure 4.2: Mean time (in seconds) to label a document within a topical group. The x-values refer to the positions of documents within a topical group. Thus, the first document labeled after a topic switch corresponds to the 1 on the x-axis. The error bars extend out by the standard deviations of the samples.

of all participants across the entirety of the task given. Because the x-axis ticks mark the beginnings of topical groups, we would expect, according to the hypothesis, a jump up in labeling times at these x-axis ticks. Although there seems to be a learning effect (i.e., that participants are able to label documents faster the more documents they have labeled), it is difficult to tell in Fig. 4.1 whether labeling times spike up at context switches.

In order to ease the process of statistical testing, we have chosen to further aggregate labeling times according to order within document groups. This aggregated data is shown in Fig. 4.2 in terms of means (the bars) and standard deviations (the error bars). With the data organized as presented in Fig. 4.2, we would expect, according to the hypothesis, to see the time of the first document labeled within a topical group to be higher than the second, third, etc. labeled documents within a topical group. The mean labeling times suggest that this might be true to some extent, but the error bars are quite large.

We use the Wilcoxon-Mann-Whitney test [4] (which is inherently one-sided) to see whether the distribution of times for labeling some ith document in a topical group is significantly larger than the distribution of times for labeling some jth document in a topical group. The results of this test are shown in Fig. 4.3. The test shows that the distribution of



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.50007	0.02666	0.0185	0.0021	0.00079	0.00063	**0.00011**	**3e-05**	**2e-05**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
2	0.97336	0.50007	0.42666	0.17087	0.09802	0.09403	0.03948	0.01731	0.01364	0.00177	0.00067	0.00121	0.00034	0.00428	**2e-05**	0.00029
3	0.98151	0.57347	0.50007	0.20984	0.13753	0.1299	0.0607	0.0308	0.02087	0.00303	0.00142	0.00224	0.00063	0.00727	**5e-05**	0.00059
4	0.9979	0.82922	0.79026	0.50007	0.38183	0.36847	0.21729	0.1179	0.09911	0.02314	0.01213	0.01831	0.00602	0.04489	0.00074	0.00622
5	0.99921	0.90203	0.86254	0.6183	0.50007	0.45277	0.33264	0.23158	0.16633	0.04936	0.02998	0.04093	0.01515	0.08247	0.00225	0.01492
6	0.99937	0.90602	0.87017	0.63166	0.54736	0.50007	0.36612	0.25576	0.17978	0.05821	0.03812	0.04902	0.01754	0.08708	0.00305	0.0176
7	0.99989	0.96055	0.93934	0.7828	0.66749	0.634	0.50007	0.34939	0.27933	0.10247	0.06484	0.07892	0.04013	0.1778	0.00677	0.03657
8	0.99997	0.9827	0.96922	0.88216	0.76853	0.74435	0.65074	0.50007	0.42045	0.20292	0.1228	0.14222	0.08724	0.29309	0.02107	0.08541
9	0.99998	0.98637	0.97915	0.90095	0.83376	0.82031	0.72079	0.57968	0.50007	0.25919	0.19646	0.22561	0.12113	0.37395	0.04263	0.12976
10	1.0	0.99823	0.99698	0.97688	0.95067	0.94183	0.89759	0.79718	0.74092	0.50007	0.40686	0.45424	0.31615	0.61856	0.12473	0.31621
11	1.0	0.99933	0.99858	0.98788	0.97004	0.96191	0.9352	0.87727	0.80364	0.59327	0.50007	0.53083	0.38183	0.69539	0.17401	0.379
12	1.0	0.99879	0.99776	0.9817	0.9591	0.95101	0.92113	0.85786	0.77449	0.54589	0.4693	0.50007	0.35376	0.65627	0.16411	0.361
13	1.0	0.99966	0.99937	0.99398	0.98487	0.98248	0.9599	0.91281	0.87894	0.68397	0.6183	0.64636	0.50007	0.80693	0.27201	0.51331
14	1.0	0.99572	0.99274	0.95514	0.91758	0.91297	0.82229	0.70703	0.62618	0.38157	0.30473	0.34385	0.19316	0.50007	0.06729	0.20244
15	1.0	0.99998	0.99995	0.99926	0.99776	0.99695	0.99323	0.97895	0.9574	0.87534	0.82608	0.83598	0.7281	0.93276	0.50007	0.74321
16	1.0	0.99971	0.99941	0.99379	0.98509	0.98242	0.96346	0.91464	0.87032	0.68391	0.62112	0.63913	0.48682	0.79765	0.2569	0.50007

Figure 4.3: Wilcoxon-Mann-Whitney test p-value results on distributions of labeling times of documents at a given position in a topical group. The value in the ith row and jth column is the p-value result of testing whether the time distribution for labeling the ith document in a topical group tends to be greater than the time distribution for labeling the jth document in a topical group. The time distribution for labeling the ith document in a topical group is composed of the labeling times collected from user study participants on the ith document for each topical group they labeled. Cells containing p-values that are statistically significant at the Bonferroni corrected α-value of $0.05/(16 \times 16)$ are highlighted and contain bolded numbers. All values are rounded to the fifth decimal place.

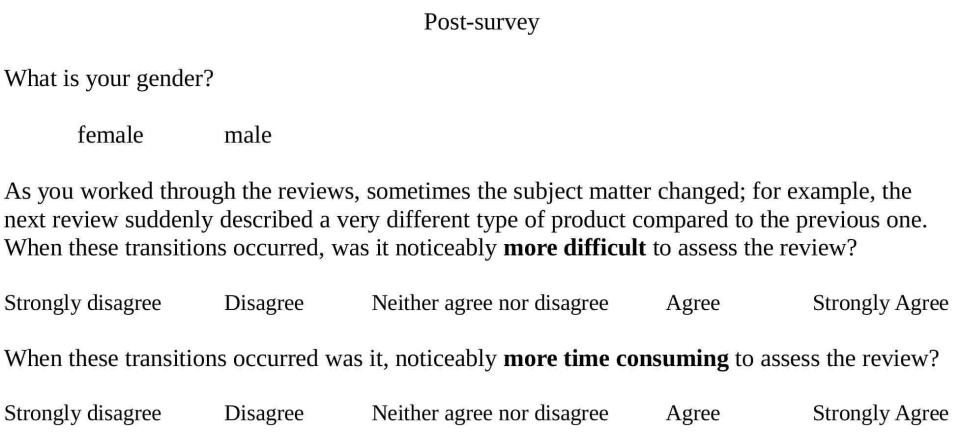
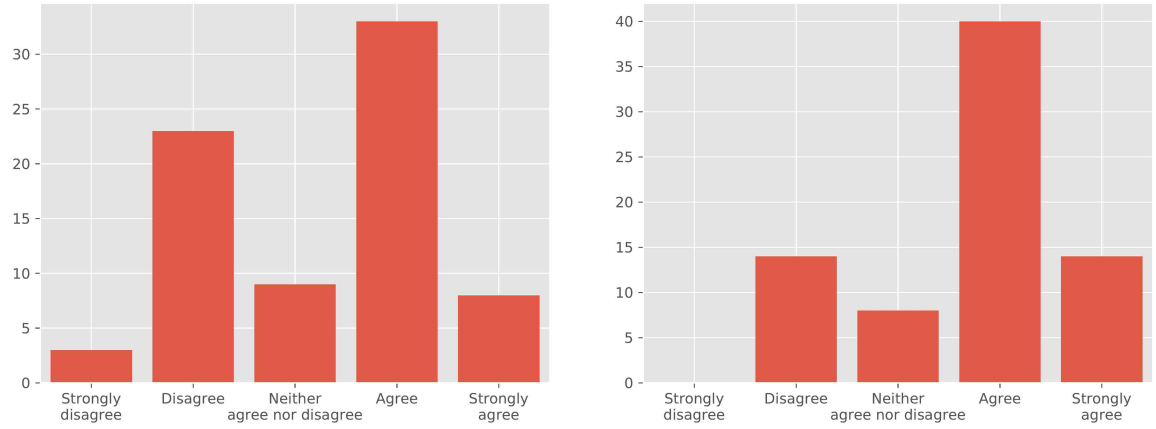


Figure 4.4: Post-survey form

times it took a participant to label the first document in a topical group does not yield values which tend to be larger than the values from the distribution of times it took a participant to label the second document in a topical group. This pattern continues until the sixth document. However, when comparing labeling times of first documents against labeling times of seventh documents, the test results show that the labeling times of the first documents tend to be larger than the labeling times of the seventh documents, with statistical significance. This pattern continues for the rest of the documents in a topical group. In other words, the test shows that labeling times for the first documents tend to be longer than labeling times for document positions seven and greater. This is sufficient to support the hypothesis that a context switch is more costly than no context switch. Thus, we conclude that there is some labeling time increase when a context switch occurs.

4.3 Context Switch User Study Results: Subjective

After completing the task, participants filled out a post-survey asking them three multiple choice questions. The post-survey form is shown in Fig. 4.4. It consists of a gender declaration and two questions to be answered on a five-level Likert scale concerning the perceived difficulty and time cost of labeling a document which seemed to have changed topic.



(a) More difficult?

(b) More time consuming?

Figure 4.5: Context switch post-survey responses

We had 76 participants, of whom there were 28 females and 48 males. Their post-survey responses have been charted in Fig. 4.5. Based on Fig. 4.5a, there seems to be a bimodal distribution regarding participants' responses to the question of whether a topic change made labeling the document more difficult. Even so, more participants agreed that a topic change made labeling the document more difficult. Fig. 4.5b shows that most participants agreed that a topic change made labeling the document more time consuming.

We should note that because of the format of the post-survey, we have no way of knowing what a participant perceived to be a change of topic. This would be problematic if participants did not perceive change of topic at the designated context switches built into the user study. However, we saw earlier that change of topical group tends to make document labeling more time consuming. At the least there is a correlation between the participants' responses regarding time consumption of perceived topic change and labeling time of documents after a topical group change.

Thus, we see that based on subjective response and corroborated by empirical data collected, a topic switch tends to be more costly than staying in the same topic. In other words, a context switch increases document labeling time.

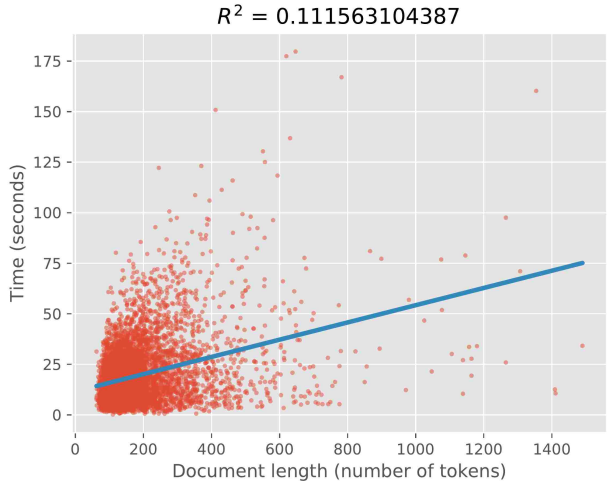


Figure 4.6: Scatter plot of document length versus time to label document. The linear regression line is included.

4.4 A (Lousy) Time Cost Model

Presented with these numbers and plots, we might still question whether document length has an effect on the increased time seen for labeling a document after a topic switch. After all, it takes longer to read a document if the document is longer. Fig. 4.6, which is a scatter plot of document length and time to label with a linear regression performed on those points ($R^2 = 0.11$) suggests that our intuition is not totally trustworthy. This might partially be explained by the nature of the task: participants may not have needed to read the entirety of the document in order to come up with a label.

Suppose, nonetheless, that we choose to believe that there is some positive correlation between document length and time to label document. Then we might wonder whether the first documents in a topical group tended to be longer than documents at other places in a topical group. To test this hypothesis, we use the Wilcoxon-Mann-Whitney test (again, with Bonferroni corrected significance level of $0.05/(16 \cdot 16)$). The results show that document lengths at a given position in a topical group are not longer in a statistically significant way compared to document lengths at another position in a topical group.

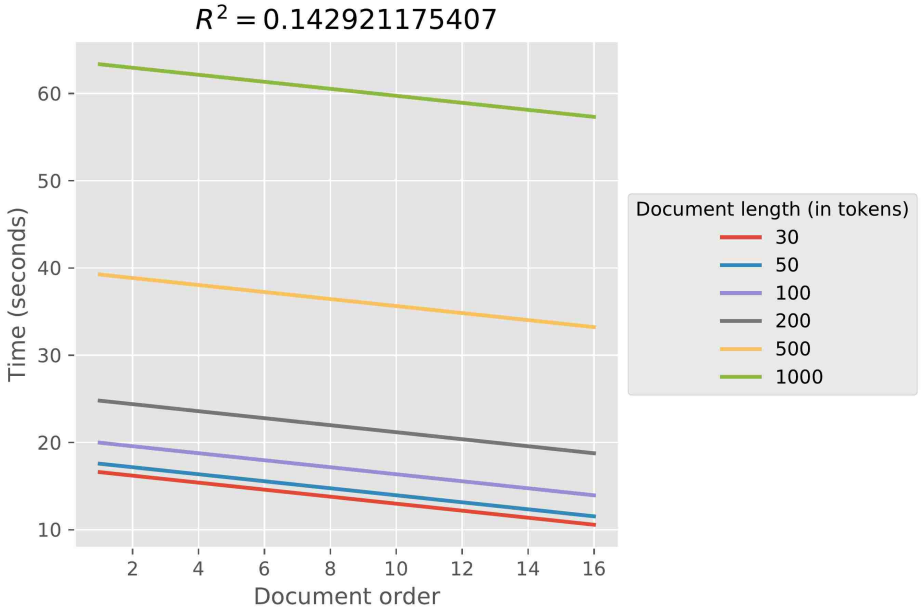


Figure 4.7: Ridge regression ($\alpha = 1.0$) model predictions after fitting to collected data. Each line corresponds to a different document length. Note that for every document length, the difference of predicted labeling time between the first and 16th documents in a topical group is 6.02762577314 seconds. Compare the R^2 value of this ridge regression over both document position and document length with the R^2 value of the linear regression over document length (Fig. 4.6).

One more check we can make to see whether the data suggest greater labeling times for first documents in topical groups is to fit a regression model to the data and look at its predicted values. Ridge regression yields the results shown in Fig. 4.7, which shows (as expected) that the longer the document, the greater the labeling times are. Moreover, the model predicts labeling time for the first document to always be higher than the 16th document. With $R^2 = 0.14$, this regression is not particularly trustworthy, although it does suggest that the labeling time for the first document in a topical group will be longer than the labeling time for the 16th document in a topical group.

Chapter 5

Paired Treatment User Study

From the computational experiments, we found that random selection is a fine choice for training a machine learning model. The context switch user study suggests that topic switching increases labeling time. With these two observations, we could try the following strategy: randomly select the subset of the data for labeling, then sort the documents into topical groups. We can compute these topical groups without document labels because computing topical groups relies only on inferring topics for the corpus (which is an unsupervised task) and on inferring the topic mixtures for the documents (which requires only the documents and the inferred topics). This strategy avoids the pitfall of diversity selection while minimizing the number of topic switches an annotator must experience. In other words, this strategy might benefit both model and human. In order to verify the effectiveness of this strategy, we conduct a second user study, with a new sample of participants.

5.1 Paired Treatment User Study Methods

For this second user study task, we asked participants to determine the star rating of 90 Amazon product reviews. By nature of the recruitment process, some participants from the first study also participated in this second user study. Again, reviews were shown one at a time to participants, and various statistics were recorded. In contrast to the previous user study, this user study had two phases. One phase, the random treatment, showed documents in a random order; the other phase, the ordered treatment, showed documents ordered by top topic. The first five documents in both treatments were shown as practice tasks, in

order to account for a learning curve, and so were not considered in the post-study analysis. Thus, we collected 40 labeling times per treatment from each participant for analysis. In order to ensure that we had near-equal representation of random treatment first and ordered treatment first, we implemented a balanced design by assigning first treatment by the order in which the participants began the user study: the first participant had random treatment first, the next participant had ordered treatment first, the next had random first, the next had ordered first, etc.

We used the topic model from the previous user study to determine top topics for documents. To build the lists of documents for each participant's treatments, we first partitioned the corpus in halves. One half was simply shuffled, and the last five documents of this half were given as the five practice documents to each participant. Then, a participant was allocated the first 40 unassigned documents from this shuffled half for random treatment.

The other half of the corpus was segmented into 1000 document chunks (except for the last chunk, which contained the remainder of the documents). In every chunk, the documents were clustered by top topic; these clusters are analogous to the topical groups of the previous study. Then, the clusters were ordered by next unseen topic with smallest JSD (Jensen-Shannon divergence, by comparing word distributions), with the first topic cluster being chosen at random. The first five documents in the ordered treatment for every participant was the last five documents of this ordered half. Finally, each participant was allocated the first 40 unassigned documents from the ordered half for ordered treatment.

We justify this topic clustering approach for the ordered treatment with a few reasons. First, selecting 40 documents at random and then ordering them by top topic using a model with 80 topics results in participants getting only a couple documents with the same top topic in a row at most, making such an ordered experience almost indistinguishable from the random treatment. Second, in a use case where some agency distributes tasks to annotators, it is plausible for the agency to collect the work of multiple annotators in order to train a machine learning model. We chose 1000 documents as a convenient number by subjectively

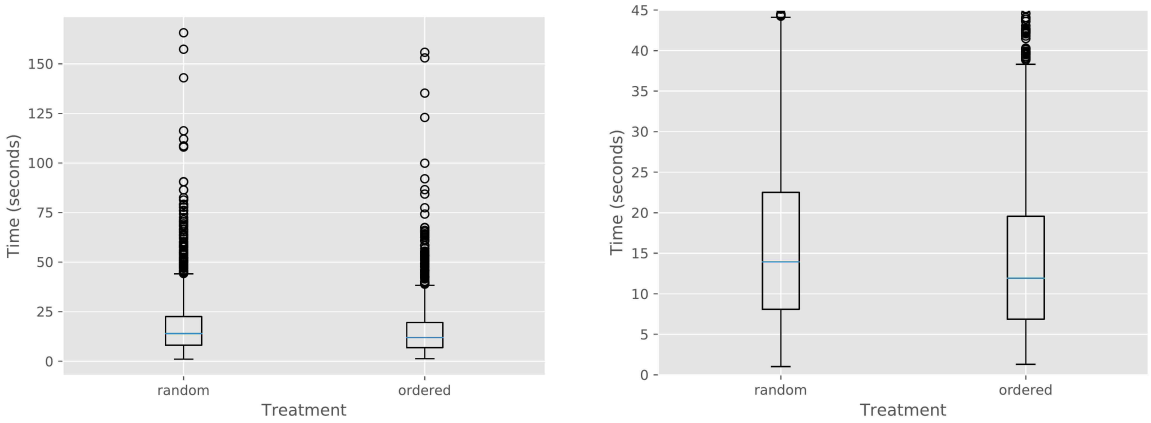
judging the number of consecutive documents with the same top topic as being long enough. Third, we chose the cluster ordering approach in an attempt to make the transition from one topic group to another smooth. We leave investigation of transition smoothness to future work.

5.2 Paired Treatment User Study Results: Quantitative

Originally, we intended to analyze the collected data with the paired t test by comparing the sums of the labeling times of the two treatments. However, we found that neither treatment's labeling time sums were distributed normally, using the Shapiro-Wilk test at an α-value of 0.05 (p-value of 3.68×10^{-6} for random sums, 6.31×10^{-5} for ordered sums). As the lack of normality violates an assumption of the paired t test, we must choose a different statistical test.

So we return to the Wilcoxon-Mann-Whitney test (one-sided by nature) to analyze all random labeling times against all ordered labeling times. At an α-value of 0.05, the Wilcoxon-Mann-Whitney test finds that the random labeling times tend to be greater than the ordered labeling times, with statistical significance (p-value of 3.48×10^{-7}). Thus, we see that the strategy of ordering a random selection of documents by top topic tends to yield faster labeling times compared to simply showing the documents in random order.

But how much real world time will be saved by using this approach? Fig. 5.1 shows that making an exact answer to that question is difficult because of the wide spread of labeling times. Nevertheless, the mean time for labeling documents in random order, about 17.76 seconds, is greater than the mean time for labeling documents in top topic order, about 15.42 seconds. By extrapolation, a savings of about 2.34 seconds per document makes a 40 document labeling task about one and a half minutes shorter (93.6 seconds) under the ordered treatment, compared to the random treatment. Extrapolating to the 1000 documents setting (proposed as justification for the user study design), the ordered treatment saves about 39 minutes (2340 seconds). Another way to measure the time savings is as a percentage; thus,



(a) Full view

(b) Zoomed view

Figure 5.1: Box plots of labeling times per treatment. The full view (Fig. 5.1a) shows the entire spread of labeling times by treatment; the zoomed view (Fig. 5.1b) shows a region that makes seeing the difference in medians easier. The median labeling time for the random treatment is 13.9445 seconds; for ordered, 11.9165 seconds.

the mean labeling time for a document in top topic order is $2.34 \div 17.76 \approx 13.2\%$ faster than mean labeling time for a randomly ordered document.

Of course, improved labeling times is no good if faster labeling leads to worse accuracy in labeling. To check this, we performed a paired t test on average accuracies per treatment per participant. This time, both distributions of accuracies were deemed to be normally distributed. We fail to reject the null hypothesis that the two distributions are the same, at an α-value of 0.05 (paired t test p-value of 0.8586). The mean average accuracy for the random treatment is about 0.638; for the ordered treatment, about 0.641. Thus, labeling documents in top topic order is not only faster, it does not cause any loss in accuracy.

5.3 Paired Treatment User Study Results: Subjective

After completing the task, the participants filled out the same post-survey as the one used in the previous user study (Fig. 4.4).

For this user study, we had 44 participants, of whom there were 18 females and 26 males. Their post-survey responses have been charted in Fig. 5.3. Once again, there seems

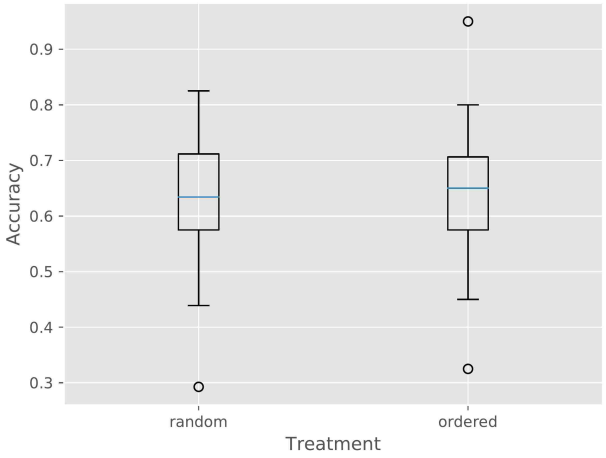
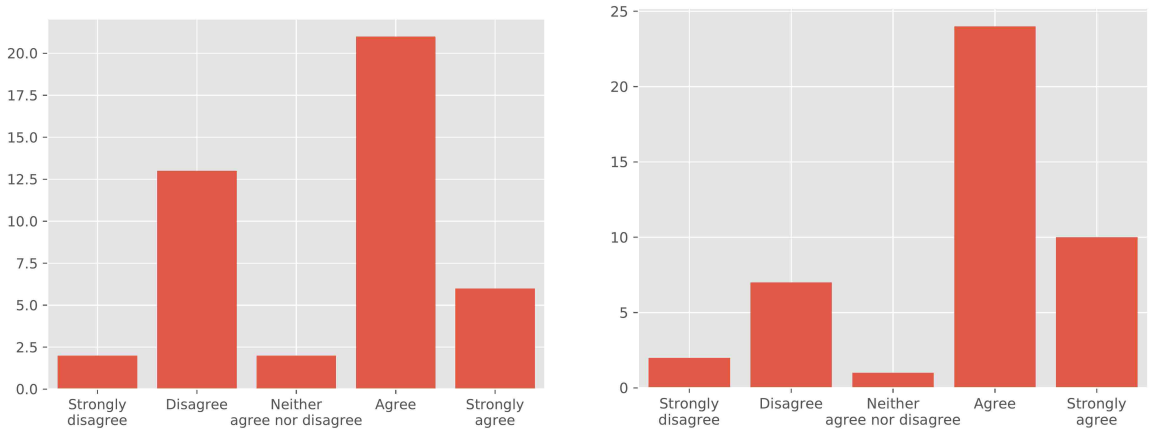


Figure 5.2: Box plots of labeling accuracies per treatment. The median accuracy for the random treatment is about 0.634; for the ordered treatment, 0.65.



(a) More difficult?

(b) More time consuming?

Figure 5.3: Paired treatment post-survey responses

to be a bimodal distribution of participants' opinions on whether topic change made the task more difficult (Fig. 5.3a). Unlike the previous user study, there seems to be a more pronounced bimodal distribution of participants' opinions on whether the topic change made the task more time consuming (Fig. 5.3b, cf. Fig. 4.5b), but the prevailing opinion is that a topic change makes document labeling more time consuming.

Chapter 6

Conclusion

First, we discovered by computational experimentation that diversity selection does not improve model accuracy compared to random selection. This suggests that a randomly chosen training set is sufficient (and preferable) for training a machine learning model for the document metadata labeling task. Second, we discovered by the context switch user study results that labeling time for the first document in a topical group tends to be greater than labeling time for the sixteenth document in a topical group. This suggests that topic switching increases labeling time. These two observations suggest that we should randomly select a training set for machine efficiency but order the training set by top topic for human efficiency. To verify the effectiveness of this strategy, we conducted the paired treatment user study and found that labeling times for documents ordered by top topic tend to be lower than labeling times for randomly ordered documents. Comparing means, labeling times for documents ordered by top topic are a little over 10% faster than labeling times for randomly ordered documents. Together, these findings support the effectiveness of the randomly selected but topically ordered training set approach to training a machine learning model for the document metadata labeling task.

Note that the randomly selected but topically ordered training set approach fulfills the goals of an active learning method. Not only does it account for what is best computationally, it also accounts for the human cost of obtaining the training set. Thus, this approach is akin to cost-conscious active learning.

Appendix A

Active Learning Pre-Exploration

Before settling on the experimental protocol described in Section 3.1, we explored various active learning approaches with SLDA. This section actually reports results from using semi-supervised LDA (which we refer to as ssLDA). The original impetus for using ssLDA was as an attempt to obtain active selection results that performed better than random selection.

In the pre-exploration experiments, the two corpora used are Chunked SotU and Frus, although preprocessed in a slightly different way compared to what is shown in Section 3.2. Significantly, the corpora were first randomly sampled to contain only a subset of all the documents. Thus, the Chunked SotU in these pre-experiments contains only 25% of the documents of the full Chunked SotU corpus; similarly, the Frus in these pre-experiments contains only 1.5% of the documents of the full Frus corpus. The stopwords list came from MALLET[10]. Words that appeared in fewer than 1% of the document or more than 50% of the documents were eliminated from the vocabulary. Empty documents after this preprocessing were eliminated. Numbers were normalized to a special number token.

The pre-experiments were repeated 16 times. Since selection computations took so long to complete, multiple documents were chosen per labeling cycle: 16 unlabeled documents were labeled per iteration of the active learning loop for Chunked SotU; 15 for Frus.

The unabridged tables of the various approaches explored in this pre-exploration is as follows:

Selection Method	Description
uncertainty sampling	choose the unlabeled document that the model predicts has the highest variance
least confidence	choose the unlabeled document whose kernel density estimate over its predicted values has the lowest peak
query by committee	choose the unlabeled document that, given a set of models, has the highest variance in mean prediction among the set of models
selection by highest variance of predicted values	choose the unlabeled document whose predicted labels have a mean within some range of the label space and have the highest variance; the ranges are determined by binning the label space into contiguous and mutually exclusive regions; the range chosen from is the first topic of the topic model; in the case where multiple unlabeled documents are chosen at a time, each non-empty range will be chosen from for their highest variance predicted document
selection by lowest variance of predicted values	like above, except with lowest variance
diversity sampling by top topic with centroid and L1 distance	choose the unlabeled document whose top topic is least similar to the centroid of the top topics of the labeled documents, according to L1 distance
diversity sampling by top topic wth centroid and L2 distance	like above, except with L2 distance
diversity sampling by top topic wth centroid and JSD	like above, except with Jensen-Shannon divergence
diversity sampling by top topic with sum of L1 distances	choose the unlabeled document whose top topic is least similar to the the top topics of the labeled documents, calculated by finding the greatest sum of the L1 distances of the unlabeled document's top topic from the the top topics of the labeled documents
diversity sampling by top topic with sum of L2 distances	like above, except with L2 distances
diversity sampling by top topic with sum of JSD's	like above, except with Jensen-Shannon divergences
diversity sampling by topic composition with centroid and L1 distance	choose the unlabeled document whose topic composition is least similar to the centroid of the topic compositions of the labeled documents, according to L1 distance
diversity sampling by topic composition with centroid and L2 distance	like above, except with L2 distance
diversity sampling by topic composition with centroid and JSD	like above, except with Jensen-Shannon divergence

Selection Method	Description
diversity sampling by topic composition with sum of L1 distances	choose the unlabeled document whose topic composition is least similar to the topic compositions of the labeled documents, calculated by finding the greatest sum of the L1 distances of the unlabeled document's topic composition from the topic compositions of the labeled documents
diversity sampling by topic composition with sum of L2 distances	like above, except with L2 distance
diversity sampling by topic composition with sum of JSD's	like above, except with Jensen-Shannon divergence
diversity sampling by topic-weighted word distribution with centroid and L1	choose the unlabeled document whose topic-weighted word distribution is least like the centroid of the topic-weighted word distributions of the labeled documents, according to L1 distance; a topic-weighted word distribution is calculated by weighting each topic (i.e., word distribution) by that topic's prevalence in the document (which can be found from the document's topic composition) and taking the sum of these weighted topics
diversity sampling by topic-weighted word distribution with centroid and L2 distance	like above, except with L2 distance
diversity sampling by topic-weighted word distribution with centroid and JSD	like above, except with Jensen-Shannon divergence
diversity sampling by topic-weighted word distribution with sum of L1 distances	choose the unlabeled document whose topic-weighted word distribution is least like the topic-weighted word distributions of the labeled documents, calculated by finding the greatest sum of the L1 distances of the unlabeled document's topic-weighted word distribution from the topic-weighted distributions of the labeled documents
diversity sampling by topic-weighted word distribution with sum of L2 distances	like above, except with L2 distances
diversity sampling by topic-weighted word distribution with sum of JSD's	like above, except with Jensen-Shannon divergences

Selection Method	Description
diversity sampling by top topic with balanced centroid and JSD[1]	choose the unlabeled document that scores high not only at being different from the labeled set but also at being similar to the unlabeled set, where scoring is determined by the Jensen-Shannon divergence of a document's top topic from the centroid of the top topics of the given set of documents
selection by shortest document	choose the shortest unlabeled document
selection by longest document	choose the longest unlabeled document

The plotted results of these pre-exploration experiments are shown in Fig. A.1. Notably, random selection wins in Chunked SotU. The next best is diversity selection by top topic with JSD from the centroid, which is the same method as the diversity selection by top topic analyzed in Chapter 3. However, Frus has slightly different results. At the beginning, diversity selection by top topic with the sum of L1 distances seems to have done best. Towards the middle, diversity selection by topic composition with sum of L1 distances performs well. By the end, diversity selection by topic-weighted word distributions with the L1 distance from the centroid is winning. Random selection seems to be second tier in the Frus results. To make this easier to see, we show only these winning selection methods in Fig. A.2. Note also that the maximum number of labeled documents reported here is about half as far out as the one reported in Chapter 3.

Thus, despite our best efforts, we found that diversity selection was superior to the more common active learning approaches of uncertainty sampling and query by committee. Moreover, these first results showed that we could not say that one selection algorithm would be best in all cases when measuring model accuracy by number of documents labeled. However, as we reasoned in Section 3.3, random selection is preferred because it requires no extra computation time, compared to the active selection algorithms.

[1]This experiment was actually done with supervised anchor words, not SLDA.

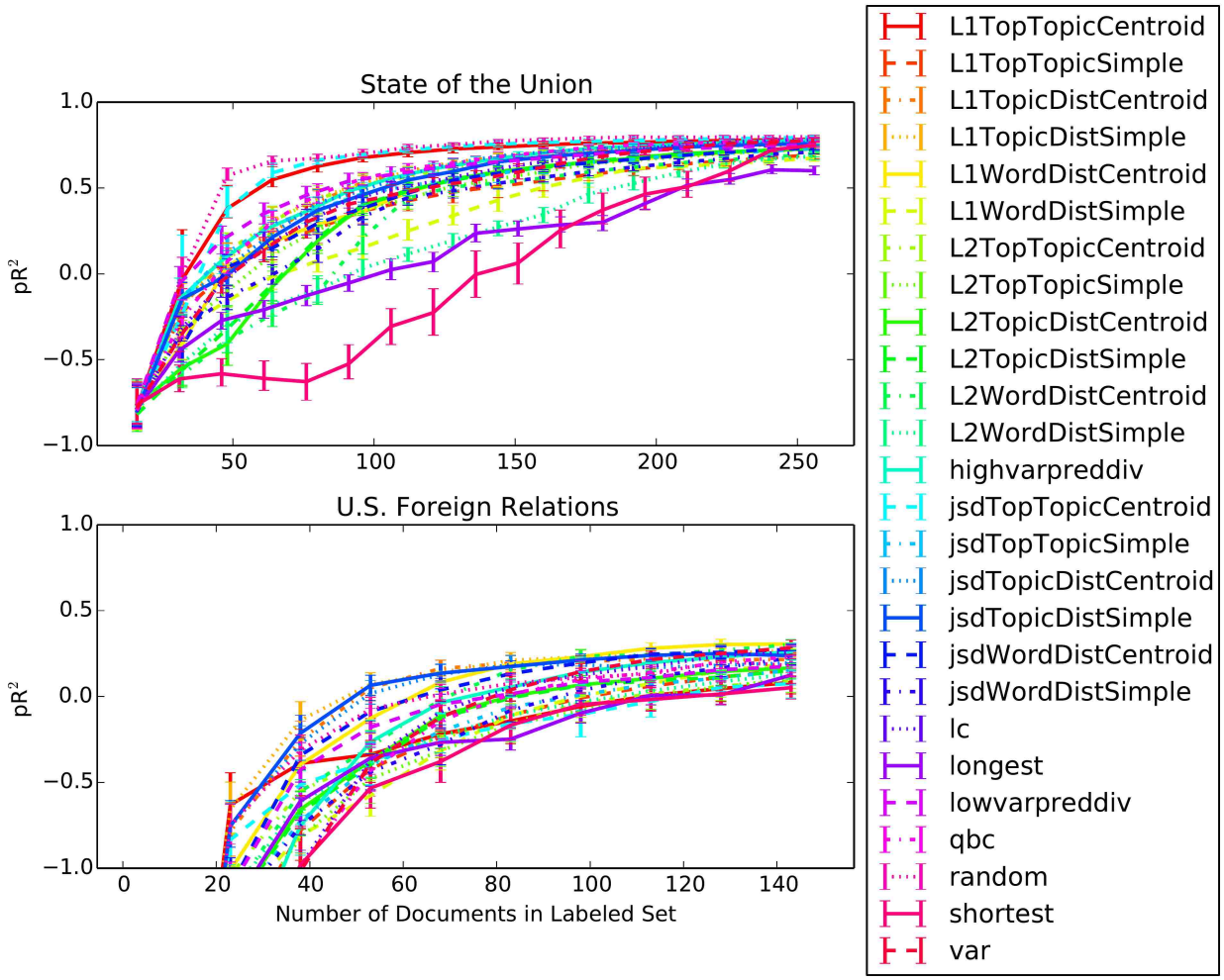


Figure A.1: An aggregation of all results gathered during the pre-exploration experiments. The abbreviations used to identify which lines correspond to which selection method was used, although cryptic, should be recognizable once compared against the selection methods listed in the long table of Appendix A.

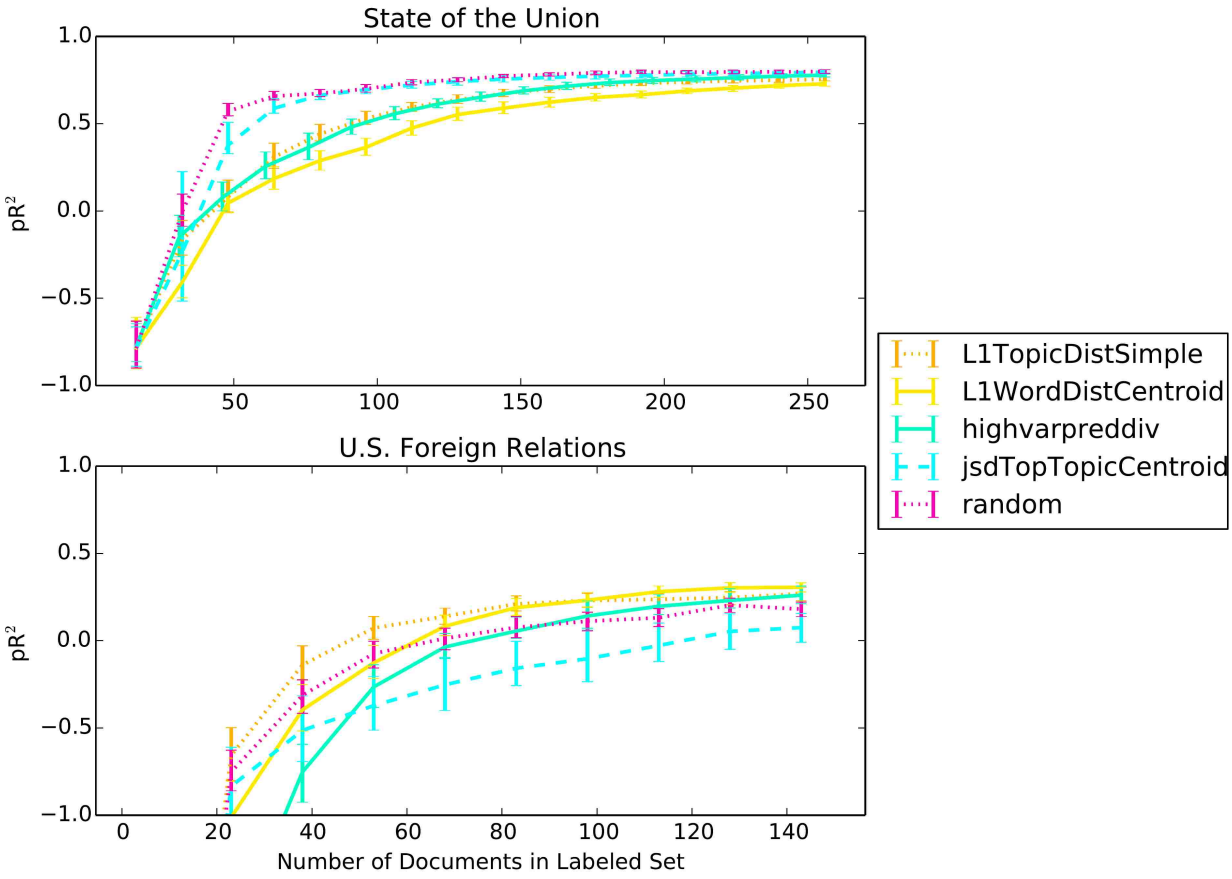


Figure A.2: A selection of pre-exploration results, showing the methods that performed best. The results shown here are the same as the equivalent results shown in Fig. A.1.

References

[1] Sanjeev Arora, Rong Ge, Yoni Halpern, David M. Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proceedings of the International Conference on Machine Learning*, pages 280–288, 2013.

[2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003.

[3] John A. Bullinaria and Joseph P. Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3): 510–526, 2007.

[4] Michael P. Fay and Michael A. Proschan. Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics Surveys*, 4:1–39, 2010.

[5] Robbie A. Haertel. *Practical Cost-Conscious Active Learning for Data Annotation in Annotator-Initiated Environments*. PhD Thesis, Brigham Young University, 2013.

[6] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1137–1143, 1995.

[7] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques. In I Maglogiannis, K Karpousiz, BA Wallace, and J Soldatos, editors, *Emerging Artificial Intelligence Applications in Computer Engineering*, pages 3–24. IOS Press, 2007.

[8] Stephen Marsland. *Machine Learning: An Algorithmic Perspective*. CRC press, 2015.

[9] Jon D. Mcauliffe and David M. Blei. Supervised topic models. In *Advances in Neural Information Processing Systems 20*, pages 121–128, 2007.

[10] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119, 2013.

[12] Tom M Mitchell. *Machine Learning*. McGraw Hill, 1997.

[13] Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

[14] Thang Nguyen, Jordan Boyd-Graber, Jeff Lund, Kevin Seppi, and Eric Ringger. Is your anchor going up or down? Fast and accurate supervised topic models. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–755, 2015.

[15] Thang-Dai Nguyen. *Enriching Spectral Methods for Topic Modeling*. PhD Proposal, Department of Computer Science, University of Maryland, College Park, 2017.

[16] Forough Poursabzi-Sangdeh, Jordan Boyd-Graber, Leah Findlater, and Kevin Seppi. Alto: Active learning with topic overviews for speeding label induction and document labeling. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1158–1169, 2016.

[17] Mrinmaya Sachan and Eric P Xing. Easy questions first? A case study on curriculum learning for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 453–463, 2016.

[18] Burr Settles. *Active Learning*. Synthesis lectures on artificial intelligence and machine learning. Morgan & Claypool, 2011.

[19] Daniel David Walker. *Bayesian Test Analytics for Document Collections*. PhD Thesis, Brigham Young University, 2012.