Reducing Automated Scoring Training Set Size with Alternative Sampling Methods

Shayne Miel
LightSide Labs
shayne@lightsidelabs.com

Abstract

The reliability of automated essay scoring (AES) is largely dependent on the size and quality of the training data used to build the models. This paper explores several methods for reducing the size of the required training data while increasing the reliability of the predictions. In particular, we explore whether unsupervised methods for selecting a training set from a large pool of unscored essays can lead to more reliable predictions on a holdout set than simply sampling with random selection. We also compare two active learning strategies, but focus our attention on fully unsupervised methods because of the operational difficulties of using active learning in the context of automated essay scoring. Our results show that maximum-minimum distance selection, an unsupervised method, can significantly improve prediction quality on some data sets. However, further research and development is needed to find a generalizable solution.

I. Introduction

In order to score essay-writing test items with automated scoring solutions, most existing engines rely on a training set of human-scored student responses. The reliability of the AES model is dependent on the quality of the training data, in terms of both the accuracy of the human scoring and the coverage of the answer space. In particular, it is important that the training set approximates the distribution of the various score points in the testing population and provides a representative sample of the ways those score points can be achieved. Obtaining a training set with these characteristics is a challenge given that the scores are not known at the time the training set is chosen to be hand scored. This challenge is often solved by randomly selecting a larger training set than is truly needed by the scoring engine. However, hand-scoring a set of responses to use as a training set is an expensive and time-consuming element of AES. Reducing the required size of the training set offers a potential reduction in the operational cost of AES and increases the affordability of AES for test items with small test populations.

A motivating example for this research may be illustrative. We consider the case of a client who wishes to use automated scoring to score responses to an essay item delivered via an online platform. A common scenario is one in which the client has a large collection of unscored responses (e.g. responses from an essay item on an end-of-grade test). Scoring all of these responses by hand is either cost- or time-prohibitive, so the client wishes to hand score a small subset of the responses to be used as a training set for the automated scoring engine. The AES engine can then score the remainder of the responses in a timely and cost-effective manner. Common practice dictates that we would randomly select the training set to be hand scored from the unscored responses. In order to maximize the accuracy of the automated scores, a large training set is desired. However, to minimize cost and time, it would be beneficial to reduce the number of responses that require hand scoring.

The question of how many human-scored responses are needed in a training set to build a reliable AES model can be restated as follows:

How many randomly sampled responses does it take to achieve sufficient coverage of the answer space? Past research has suggested a wide range of possible training set sizes, from 100 (Landauer, et al., 2003) to 1,800 (Shermis & Hamner, 2013). This paper explores the use of alternative sampling methods when constructing the training set for an AES model in order to reduce the number of hand-scored responses needed, while still maintaining high reliability.

The focus of this paper is on unsupervised methods of selecting the training set to be hand scored from a large pool of unscored essays. That is, we use information extracted from the essays themselves, with no knowledge of the scores a human would give, in order to select the optimal training set. In addition, we include some shallow exploration of active learning to select training sets. The machine learning community has done a considerable amount of research into active learning, a group of algorithms that iteratively select larger training sets based on the hand-scoring information from smaller sets (Cohn, Atlas & Radner, 1994), (Freund, Seung, Shamir, & Tishby, 1997), and (Settles, 2010). This process can be repeated multiple times and often achieves a much better result than simple random selection. However, the iterative hand-score/train/predict/select cycle requires technological flexibility from the hand-scoring system that may be difficult for the hand-scoring vendor to provide. A fully unsupervised method, if it is approximately as effective as active learning, would be preferable.

The data for this study comes from the Automated Student Assessment Prize, Phase One (ASAP) data.[1] Three source-dependent prompts were selected: essay items 3, 5, and 6. Table 1 gives further details about the prompts. For each prompt, we combine the training, validation, and test sets into a single group and use the final human scores for the model building and evaluation process.

Table 1: *ASAP Prompts*

ID	Grade	# of Responses	Rubric
3	10	2858	0-3
5	8	3006	0-4
6	10	3000	0-4

We use LightSide, an open-source AES engine developed originally at Carnegie-Mellon University and now maintained by LightSide Labs, to do all of the feature extraction, model building, and prediction work in this paper. LightSide is provided for free (open source GPLv3 license) as a desktop tool for researchers.[2]

II. Methods

I. Experimental Design

In order to measure whether a sampling method can improve upon random selection, we use the method to select training sets of 100, 200, 300, 400, 500, and 600 essays. Each training set is used to build a LightSide model and predict scores on a hold-out set of 30% of the essays (approximately 1,000 responses), selected at random. The quality of a given training set is measured by the quadratic weighted kappa of the predictions that a model built from the training set gives on the hold-out set. The hold-out set is constant across training set sizes and sampling methods so that a valid comparison can be made. To control for randomization in the selection algorithms, this process is repeated 5 times at each training set size, and to control for randomization in the hold-out set selection, the entire experiment is run 3 times on each data set. Because this data has already been hand-scored, when selecting the training sets we explicitly ignore the human-assigned scores in order to mimic the motivating example described in the introduction. The steps of the experiment are described in Algorithm 1.

```
for each prompt do
    for each of 3 runs do
        holdout = 30% of the essays
        available = other 70% of essays
        for each selection method do
            for each of 5 trials do
                for m = 100 through 600 do
                    select m from available
                    build a model
                    predict on holdout
                end
            end
        end
    end
end
```

Algorithm 1: Experiment

Once all trials are performed (1890 trials in total), we analyze the data in two ways. First, for each prompt, we calculate the quadratic weighted kappa difference between each method and random selection at each training set size, averaging across all 5 trials in all 3 runs (15 trials per method per training set size). Those differences are given in Tables 3, 4, and 5, with statistical significance from a two-tailed T test indicated at $p < 0.05$, $p < 0.01$, *and* $p < 0.001$.

We also attempt to estimate how many training set essays would be needed when sampled under the various strategies in order to achieve a quadratic weighted kappa equivalent to random selection. To estimate this data, we fit logarithmic curves of the form given in Equation 1 to the scores from the 15 trials at each size for all seven sampling methods. In this equation, y is the quadratic weighted kappa for the trial, x is the training set size, and the variables a, b, c, and d are fit to the data using a non-linear least squares curve fitting approach (Levenberg, 1944). Using the random selection curves for each prompt, we estimate the expected quadratic weighted kappa for a training set of 600 essays. We then use the curves for the other sampling methods to determine how many training essays would be needed to achieve an equivalent quadratic weighted kappa. These estimates are given in Table 2.

$$y = a(\ln(b(x - c))) + d \tag{1}$$

II. Sampling Methods

The sampling methods under comparison are:

- Random Selection
- Stratified Selection on Length
- K-Means Center/Border Selection
- Maximum-Minimum Distance Selection
- Active Learning

Random Selection
Random selection of the available data is the industry standard and is presented here as the baseline selection strategy.

Stratified Selection on Length
Motivation: We want our training set to have a representative distribution of all of the score points. Essay length tends to have a strong correlation with human-assigned scores (Breland, Bonner, & Kubota, 1995), so we use length of the essay as a proxy for score and perform stratified selection.

Description: To select m essays from the available pool with this sampling method we sort the essays by length, divide the list into m equally sized bins, and select a random essay from each bin.

K-Means Center/Border Selection
Motivation: The AES engine needs data points that are most representative of the various score points, as well as data points that help it find the dividing hyperplane between the score points. We begin with the assumption that we can cluster the data into groups that resemble the score point groupings by clustering on the features alone. If this

assumption holds, selecting from the center of the clusters would approximate finding the essays that are most representative of the score points, and selecting from the borders between the clusters would approximate finding essays that help define the dividing hyperplane.

Description: K-means is an unsupervised clustering algorithm that groups a data set based on how close each essay is to the centers of a proposed set of clusters. These clusters are generated by considering the essays as points in an n-dimensional feature space, where n is equal to the number of features measured on each essay, and trying to minimize the variance of the clustering. Center/border selection is an algorithm that selects evenly from those essays that are close to the center of the clusters and those essays that are on the border between two or more clusters (Lughofer, 2012).

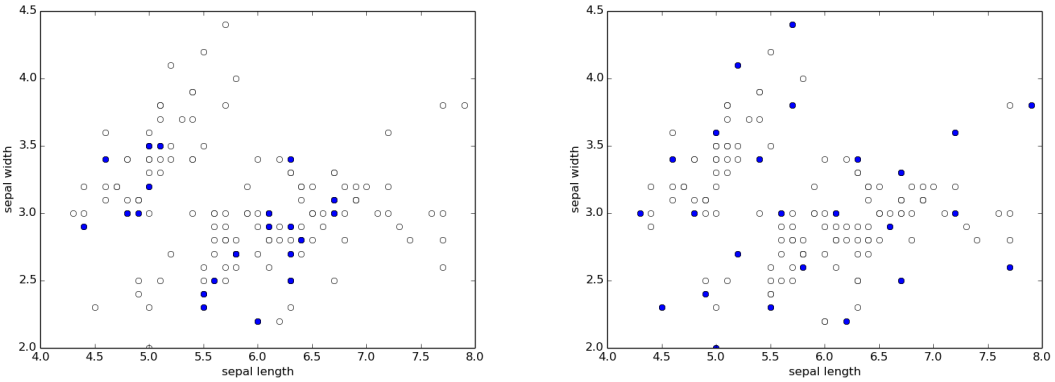
K-means is an efficient clustering algorithm, but does not work well when used on high-dimensional data (Sun, Wang, & Fang, 2012). LightSide generates boolean features indicating the presence or absence of word, character, and part of speech n-grams, which number into the tens of thousands. To reduce these to a usable number, we first transform them via tf-idf into real valued features and then reduce them to 10 principal components using Randomized PCA, which has been shown to be a convenient transformation for the k-means algorithm (Ding & He, 2004). Once each essay has been transformed, we use k-means clustering to group the essays into k clusters, where k is the number of possible score points for that data set. We then attempt to select 50% of the training set from the center of the clusters, spread evenly across the clusters, and 50% of the training set from regions between clusters.

Maximum-Minimum Distance Selection

Motivation: Assuming that the extracted features properly describe the essays, we want to select a sample such that as much of the described feature space as possible is covered. Maximum-minimum distance (MMD) selection is an algorithm that ensures the selected essays are all as far from one another as possible in the feature space for a given distance metric. The result of this sampling strategy is that the selected set is spread evenly over the n-dimensional feature space that describes the data points. Figures 1a and 1b illustrate the difference between MMD selection and random selection, using a Euclidean distance metric (Equation 2) on two dimensions from the Iris data set.[3]

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{2}$$

Description: MMD (Algorithm 2) is an iterative algorithm that selects a set of data points such that the distance from each selected point to its nearest neighbor in the selected set is maximized. MMD begins by selecting m points at random (where m is the desired sample size). It then considers each selected point, and if there is an unselected point such that the distance between the unselected point and the unselected point's nearest neighbor in the selected set is greater than the distance between the selected point and *its* nearest neighbor in the selected set, the currently selected point is replaced with the unselected point.[4] The algorithm repeats this process for every selected point until convergence is reached when no points are updated. As proof that this algorithm converges, we define S_t, the selected set after t updates have been made, and D_t (Equation 3), the sum of the distances between all points in S_t. Note that $D_{t+1} > D_t$ for all t. Since D_t is monotonically increasing and there are a finite set of points to select from, the algorithm must eventually converge. Of course, this is only true if $d(\mathbf{p}, \mathbf{q})$ is a proper distance metric. In particular, $d(\mathbf{p}, \mathbf{q})$ must satisfy the



(a) *Random Selection*

(b) *Maximum-Minimum Distance Selection*

Figure 1: *Random selection vs. Maximum-Minimum Distance selection on two dimensions of the Iris data set*

triangle inequality.

$$D_t = \sum_{\mathbf{p} \in S_t} \sum_{\mathbf{q} \in S_t} d(\mathbf{p}, \mathbf{q}) \tag{3}$$

We examine MMD with two distance metrics: Jaccard distance (Equation 4) on the set of boolean features extracted by the LightSide engine and Euclidean distance (Equation 2) on those same features after applying tf-idf.

$$d(\mathbf{p}, \mathbf{q}) = 1 - \frac{\sum_{i=1}^{n} p_i \wedge q_i}{\sum_{i=1}^{n} p_i \vee q_i} \tag{4}$$

Input: A, m
Output: The m most distant points

Let $S = \{x_0, x_1, ..., x_m \mid x \in A\}$

done = False
while *not done* **do**
  done = True
  for p *in* S **do**
    $S' = S \setminus p$
    $i = \underset{i \in N \setminus S'}{\text{argmax}} \ \underset{j \in S'}{\text{argmin}} \ d(i, j)$
    $S = S' \cup i$
    if $i \neq p$ **then**
      done = False
    end
  end
end
return S

Algorithm 2: Maximum-minimum distance selection algorithm for selecting m points from a set A

Active Learning

Motivation: For any given training set, an automated scoring system will have more information about some essay types and less information, or potentially confusing information, about others. If we are able to identify those essays for which the automated

scoring engine has the least amount of information, then we could be sure to add those essays to the training set to be hand-scored. Active learning is a broad category of machine learning algorithms that iteratively selects more data with which to train, based on information gathered from training on a smaller set and predicting on the remainder of unscored data from which it is selecting. Settles (2010) provides an overview of many of the common methods for active learning.

Description: In this experiment, after training a model on 100 randomly selected responses, the active learning algorithm predicts on both the hold-out set and the responses not selected in the first batch of 100. The sampling strategy then selects an additional 100 responses from that remainder set, based on some heuristic over the predictions, to include (with the original 100) in the $m = 200$ training set. We compare two common strategies for active learning: confidence and committee. The active learning by confidence algorithm selects those data points for which it is least certain about its predictions. The LightSide engine is powered by a naive Bayes classifier, which outputs a probability distribution over the possible scores for each essay. The predicted score is the one with the highest probability. At each step, we take the 100 essays with the lowest probability given for the predicted score. Active learning by committee, on the other hand, uses multiple models to predict on each essay and selects those essays for which the predictions differ the most. We use a logistic regression model to compare with the naive Bayes model and select the 100 essays with the greatest difference between the two predictions at each step.

III. Results

No particular sampling method outperformed random selection on all three prompts. However, several methods were able to reduce the average training set size by up to 45% on some of the prompts (Table 2). Stratified selection on length was approximately equal to random selection on all three prompts (Figure 2), as was active learning by committee (Figure 5). K-means center/border selection and MMD selection with both distance metrics outperformed random selection on prompts 3 and 6, but underperformed random selection on prompt 5 (Figures 3 and 4). Finally, active learning by confidence was approximately equal to random selection on prompts 3 and 5, but outperformed random selection on prompt 6 (Figure 5). Further study is needed to understand why certain prompts were amenable to these training set reduction strategies, while others were not.

I. Prompt 3

On prompt 3, MMD selection with Jaccard distance and MMD selection with Euclidean distance outperformed random selection when training set sizes were $m \geq 300$ essays, showing an increase in average quadratic weighted kappa of 0.015 to 0.027 (Table 3). The expected reduction in training set size would be 36% for MMD with Jaccard distance and 37% for MMD with Euclidean distance (Table 2). K-means center/border selection showed slight improvement over random at $m = 200$, as did active learning by confidence (Δqwk of 0.011 and 0.014, respectively). Stratified selection on length and active learning by committee showed no difference with respect to random selection.

II. Prompt 5

On prompt 5, k-means center/border selection and MMD selection with either distance metric performed worse than random selection at all training set sizes, showing up to a 0.012 to 0.082 drop in average quadratic weighted kappa (Ta-

Table 2: *Expected training set size to achieve a quadratic weighted kappa equivalent to that of 600 randomly selected essays. "n/a" indicates that the expected size is larger than 600 (no reduction).*

	Prompt 3	Prompt 5	Prompt 6
Stratified on Length	n/a	593	577
K-means	591	n/a	497
MMD Jaccard	386	n/a	429
MMD Euclidean	376	n/a	399
Active Confidence	n/a	531	331
Active Committee	n/a	518	454

Table 3: *Prompt 3: Mean quadratic weighted kappa difference between sampling method and random selection at six training set sizes, averaged over 15 trials. (two-sample t-test, *p<.05, **p<.01, ***p<.001)*

	100	200	300	400	500	600
Stratified on Length	0.003	0.004	0.005	0.002	-0.005	-0.005
K-means	-0.010	0.011^{**}	-0.002	0.004	0.006^{*}	0.000
MMD Jaccard	-0.011	-0.006	0.026^{***}	0.026^{***}	0.016^{***}	0.015^{***}
MMD Euclidean	0.004	0.000	0.019^{***}	0.027^{***}	0.022^{***}	0.020^{***}
Active Confidence	-0.007	0.014^{*}	0.006	0.001	0.004	0.000
Active Committee	0.009	0.006	0.003	0.003	-0.003	-0.005

ble 4. Active learning by confidence and active learning by committee showed slight increases at $m = 600$ (Δqwk of 0.006 and 0.007 respectively), while stratified selection on length showed no difference with respect to random sampling. The expected reduction of training set size with the active methods would be 12% to 14% (Table 2).

III. Prompt 6

On prompt 6, active learning by confidence outperformed random selection when training set sizes were $m \geq 300$ essays, showing an increase in average quadratic weighted kappa of 0.017 to 0.021 (Table 5). MMD selection with Jaccard distance was worse than random selection on small training set sizes (when $m = 100$, $\Delta qwk = -0.080$), but steadily rose until it began outperforming random selection when $m \geq 400$, eventually reaching $\Delta qwk = 0.022$ when $m \geq 500$. MMD selection with Euclidean distance showed similar performance, with less dramatic drops in average quadratic weighted kappa when $m \leq 200$ (Δqwk of -0.026 to -0.033), and higher gains in average quadratic weighted kappa when $m \geq 400$ (Δqwk of 0.016 to 0.023). K-means center/border selection showed erratic performance, with lower average quadratic weighted kappa when $m \leq 200$ (Δqwk of -0.030 to -0.048), no difference when $m = 300$, 400, or 600, and slight increase when $m = 500$ ($\Delta qwk = 0.009$). Stratified selection on length and active learning by committee showed no significant improvement. On prompt 6, however, all methods showed an expected reduction in training set size, from stratified selection on length (4%) to active learning with confidence (45%).

IV. Discussion

The ultimate goal of the exploration in this paper was to find an unsupervised method for selecting data that could reduce the re-

Table 4: *Prompt 5: Mean quadratic weighted kappa difference between sampling method and random selection at six training set sizes, averaged over 15 trials. (two-sample t-test, *p<.05, **p<.01, ***p<.001)*

	100	200	300	400	500	600
Stratified on Length	−0.004	0.001	−0.003	−0.002	−0.003	0.005
K-means	−0.127***	−0.073***	−0.070***	−0.053***	−0.041***	−0.012**
MMD Jaccard	−0.073***	−0.070***	−0.052***	−0.048***	−0.026***	−0.013***
MMD Euclidean	−0.082***	−0.058***	−0.062***	−0.056***	−0.031***	−0.018***
Active Confidence	−0.010	0.004	0.002	0.002	0.001	0.006*
Active Committee	0.000	−0.010	−0.001	0.000	0.002	0.007*

Table 5: *Prompt 6: Mean quadratic weighted kappa difference between sampling method and random selection at six training set sizes, averaged over 15 trials. (two-sample t-test, *p<.05, **p<.01, ***p<.001)*

	100	200	300	400	500	600
Stratified on Length	0.006	0.000	−0.007	−0.003	0.002	0.002
K-means	−0.030***	−0.048***	0.005	0.005	0.009*	−0.005
MMD Jaccard	−0.080***	−0.054***	−0.022**	0.013**	0.022***	0.022***
MMD Euclidean	−0.026**	−0.033***	0.000	0.016**	0.022***	0.023***
Active Confidence	0.008	−0.001	0.017***	0.017***	0.020***	0.021***
Active Committee	0.000	−0.006	0.002	0.007	0.009	0.005

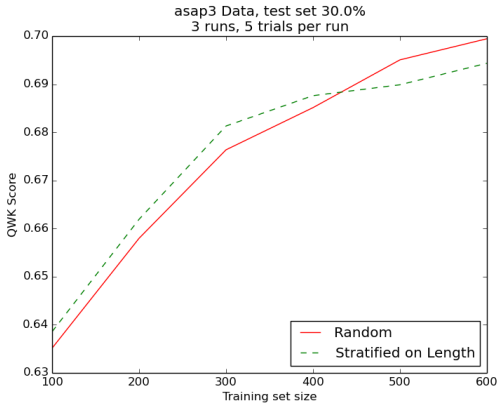
quired training set size in comparison to random selection. On prompts 3 and 5, MMD selection with a Euclidean distance metric was able to reduce the size of the training set by 37% and 34%, respectively, indicating that this algorithm may be an appropriate method to use. However, this method's poor performance on prompt 5 shows that more research needs to be done to understand the limitations of the algorithm. An interesting thing to note is that MMD tended to underperform random selection when training set sizes were very small ($m \leq 200$) and then rapidly climb in performance as training set sizes increased (Figure 4). This is likely due to the effect of outliers, which tend to be selected more often than inliers with this algorithm when the selection size is much smaller than the available pool of unscored essays. A possible improvement to this algorithm would be to include a means of reducing the number of outliers in the selected set. One promising aspect of this algorithm is that it appears to be unaffected by the curse of dimensionality, and is able to identify a strong training set even though the distances are computed between points in a very high-dimensional space. Further research is needed to understand how this algorithm behaves on low-dimensional data.

As noted in the results, stratified selection on length showed no improvement over random sampling. This is likely because, even though length does correlate with score, length is a poor proxy for score. Even if we had a good proxy for score (e.g. scores from a model trained on similar data or ability metrics for the student from other sources), we suspect that this algorithm would still not perform as well as MMD. Having a good representation of each score point is important, but it does not necessarily give you a sufficient coverage of the answer or feature space.

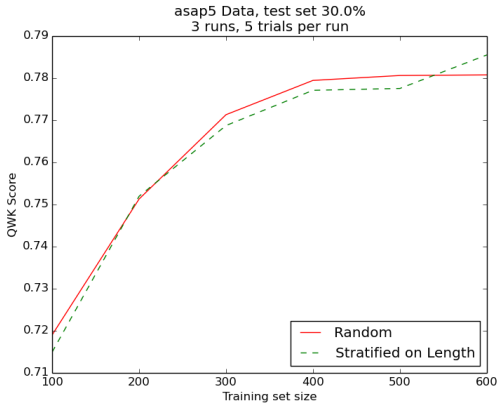
In contrast, k-means center/border selection, which also showed little noticeable improve-

ment over random selection, would benefit from more informative features (i.e. features that served as better proxies for the score). The features generated by the LightSide engine - word, character, and part of speech n-grams - are deliberately *not* intended to be proxies for score. Rather, they are designed to be as descriptive of the essays as possible, and LightSide waits for the scores in the training set to inject the meaning into those descriptions. This makes the engine a more general purpose tool, but means that the k-means center/border selection strategy is not a good match. However, there are engines which attempt to generate features from the essays that *are* proxies for score, and for them the k-means center/border selection strategy should work much better. This would also explain why Lughofer (2012) had more success with the algorithm than we did in this paper.

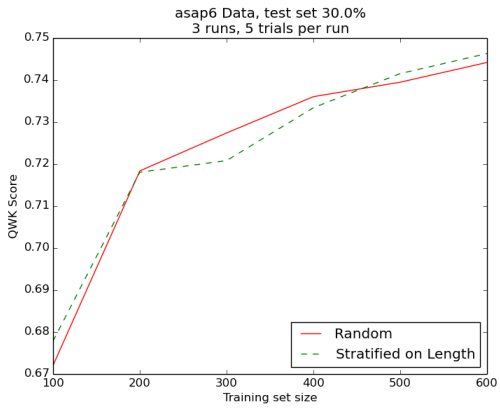
Finally, even though active learning via confidence worked well on Prompt 6, it is known that active learning strategies work better when they are able to select one new response at a time, rather than select them in batches as was done in this experiment (Settles, 2010). When selecting in batches, they tend to identify a group of very similar responses as the ones for which they need more information, slowing down the potential accuracy growth. Further work should include using MMD as an add-on to active learning in batches to force a spread of new responses with each iteration. On the other hand, active learning by committee showed little improvement over random selection. This may be due to the way that we let the second model contribute to the selection strategy without contributing to the predictions. Perhaps using an ensemble method to include the additional information from logistic regression would have aided this selection strategy's performance.



(a) *Prompt 3*

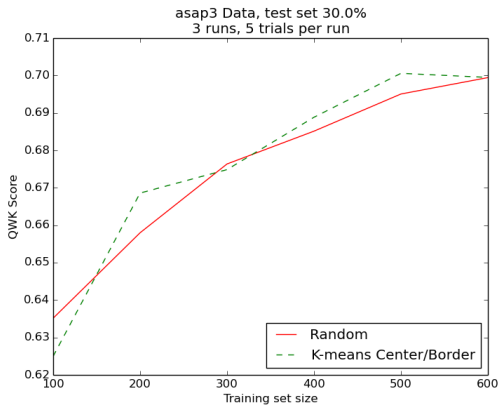


(b) *Prompt 5*

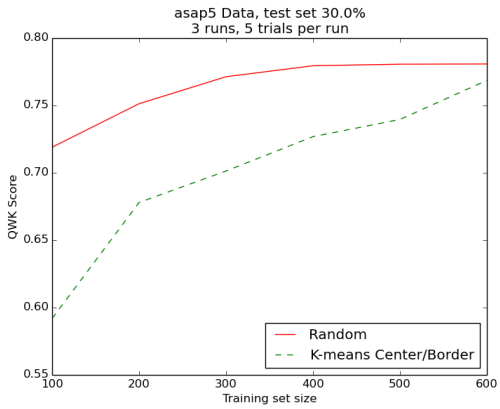


(c) *Prompt 6*

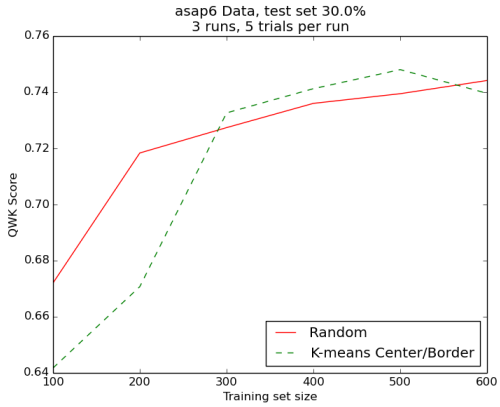
Figure 2: *Random selection vs. Stratified selection on length. QWK Score is quadratic weighted kappa for a training set selected with the given method, averaged over 15 trials.*



(a) *Prompt 3*

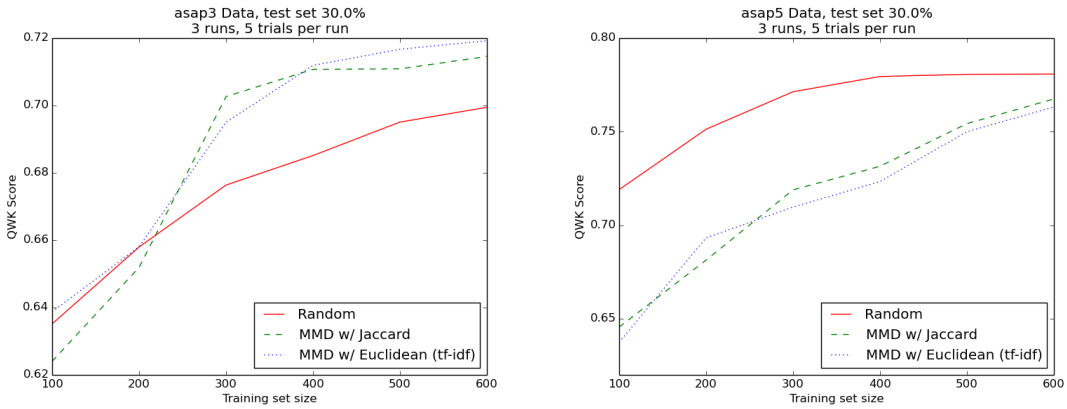


(b) *Prompt 5*



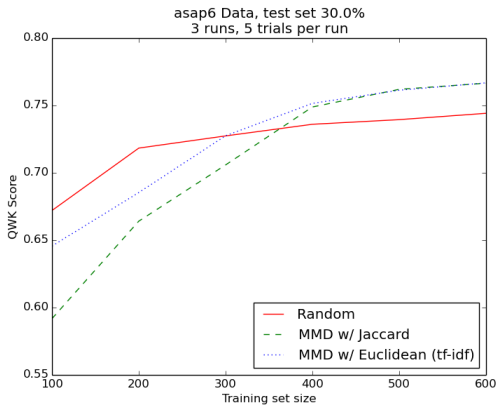
(c) *Prompt 6*

Figure 3: *Random selection vs. K-means center/border selection. QWK Score is quadratic weighted kappa for a training set selected with the given method, averaged over 15 trials.*



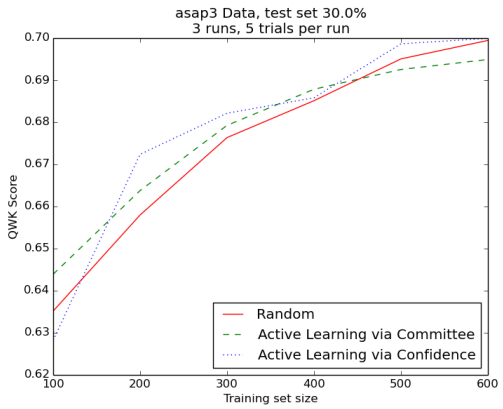
(a) *Prompt 3*

(b) *Prompt 5*



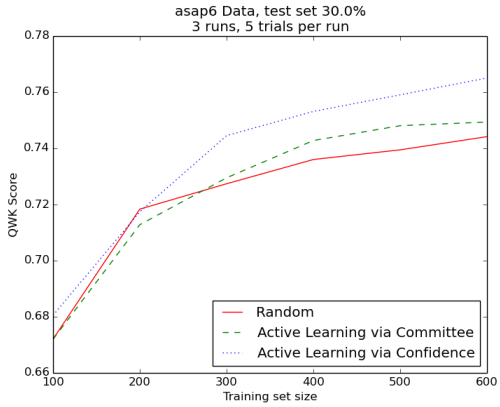
(c) *Prompt 6*

Figure 4: *Random selection vs. MMD Selection with two distance metrics. QWK Score is quadratic weighted kappa for a training set selected with the given method, averaged over 15 trials.*



(a) *Prompt 3*

(b) *Prompt 5*



(c) *Prompt 6*

Figure 5: *Random selection vs. Active learning with two strategies. QWK Score is quadratic weighted kappa for a training set selected with the given method, averaged over 15 trials.*

References

Breland, H. M., Bonner, M. W., & Kubota, M. Y., (1995). Factors in performance on brief, impromptu essay examinations. *College Board Report No. 95-4*.

Cohn, D., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine learning, 15(2)*, 201-221.

Ding, C., & He, X. (2004, July). K-means clustering via principal component analysis. *In Proceedings of the twenty-first international conference on Machine learning (p. 29)*. ACM.

Freund, Y., Seung, H. S., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine learning, 28(2-3)*, 133-168.

Landauer, T. K., Laham, D., & Foltz, P. W. (2003). Automated Essay Scoring: A Cross Disciplinary Perspective. In Mark D. Shermis and Jill C. Burstein (Eds.), *Automated Essay Scoring and Annotation of Essays with the Intelligent Essay Assessor*.

Levenberg, K. (1944). A Method for the Solution of Certain Non-Linear Problems in Least Squares. *Quarterly of Applied Mathematics, 2*, 164-168.

Lughofer, E. (2012). Hybrid active learning for reducing the annotation effort of operators in classification systems. *Pattern Recognition, 45(2)*, 884-896.

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison, 52*, 55-66.

Shermis, M. D., & Hamner, B. (2013). Contrasting State-of-the-Art Automated Scoring of Essays. *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, 313.

Sun, W., Wang, J., & Fang, Y. (2012). Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electronic Journal of Statistics, 6*, 148-167.

Notes

[1]Available at `http://www.scoreright.org/asap.aspx`

[2]The engine can be downloaded from `http://lightsidelabs.com/what/research/`

[3]The Iris data set is a classic machine learning data set, available at `http://archive.ics.uci.edu/ml/datasets/Iris`

[4]There are a few optimizations to this algorithm that enable it to run in a reasonable amount of time, but discussing those optimizations and proving a bound on running time are beyond the scope of this paper.