Effective Data Collection with Active Learning

Brandon Novak

November 14th, 2021

Abstract

Active Learning is a semi-supervised topic that chooses data points to train itself. Active Learning is a design methodology which allows efficient data sampling by combining Machine Learning prediction algorithms, sampling methods, and human intervention to effectively choose points that have the weakest class prediction probability. Due to a lack of data in Natural Language Processing (NLP), Active Learning has become more prevalent. Natural Learning Processing is a popular field of Machine Learning research and researchers are finding that computers can program, understand, and predict natural languages. In order to train these models, we need an adequate amount of data, but NLP data is expensive to obtain. Active Learning allows for efficient data collection that can minimize costs. Active Learning is an iterative process of training a model on a small amount of training data, making predictions with the test data, and purposely sampling test data into the training data. Our experiment shows that Random Forest provides the most robust prediction algorithm, in terms of accuracy and run times, for our particular dataset on an email dataset that identifies messages as spam or not spam. Moreover, we find the most efficient amount of data collection when considering the costs of data collection.

Introduction

Data collection is a vital part of the data analytics process. Often times, data analysts and organizations have superb questions, but the data necessary to answer can be expensive to label, and the amount of data that is sufficient can be unclear. Both of these problems can be solved with Active Learning. Active Learning is a semi-supervised topic that chooses data points to train itself. An Active Learning framework is an iterative process of training a model on a small amount of training data and making predictions with the test data. The framework samples the test data points that will contribute the most information by knowing their label, and adds them to the training set. The data that we are using is an email dataset that contains email messages and includes an indicator variable that states if the email is spam or not spam. Using our dataset, we aim to answer the following question: What is the most effective Active Learning framework for Natural Language Processing (NLP) datasets, and how can we decide how many iterations of data collection are necessary to maximize the accuracy while minimizing the cost? We found that Random Forest produced the best results in terms of accuracy and run times. Moreover, we found that only 20% of the original dataset is truly necessary to produce an effective model.

Motivation

We chose a Natural Language Processing (NLP) data set because NLP applications are notorious for requiring a lot of data, and being very difficult and expensive to label. For example, the creator of this dataset had to collect, read, and label 5,556 emails as spam or not spam. For this reason, the results and analyses performed in this research could contribute to the overall topic of NLP. By making data collection quicker and more effective, Natural Language Processing applications and experiments can be expedited. Part-Of-Speech tagging and Name Entity Recognition are just a few examples that require a large amount of data to be trained on (Ratnaparkhi 134).

Active Learning Frameworks

We test six different Active Learning frameworks. There are two main components when creating an Active Learning framework: prediction algorithm and sampling method. The prediction algorithms implemented were K-Nearest Neighbors (K-NN), Random Forest, and Support Vector Machines (SVM). The sampling methods used were Uncertainty sampling and Entropy sampling. Therefore, we had a total of six different Active Learning frameworks. There were two performance metrics that we observe: accuracy compared to the entire dataset, and time to complete an iteration of data sampling.

Once we had data of each prediction algorithm and sampling method combination, we performed analyses on the effectiveness of the frameworks. This allowed us to answer our central question: What is the most effective Active Learning framework for NLP datasets, and how can we decide how many iterations of data collection is necessary to maximize the accuracy while minimizing the cost? With the data provided in the Active Learning frameworks, we determined the optimal amount of data sampling required to obtain a reasonable sample when considering potential cost values.

Overview of Active Learning

Active Learning is a design methodology which allows efficient data sampling by combining Machine Learning prediction algorithms, sampling methods, and human intervention. There are many reasons to implement Active Learning. The most common reason is that it is expensive or time intensive to label all of the data. The framework for all six combinations of prediction algorithms and sampling methods are the same. The framework begins by assuming that none of the dataset is labelled, and then a random 10% of the data is labelled. With the training data, representing 10% of the entire dataset, predictions are made using either K-Nearest Neighbors, Random Forest, or Support Vector Machines on the test dataset, the remaining 90% of data. Using the predictions, the framework samples 10 data points that were the most uncertain when being labelled. We used two sampling methods to determine which points were the most uncertain: Uncertainty sampling and Entropy sampling. The framework chooses the most uncertain points because they have the most potential to impact the decision boundary. Consider Figure 1 below.

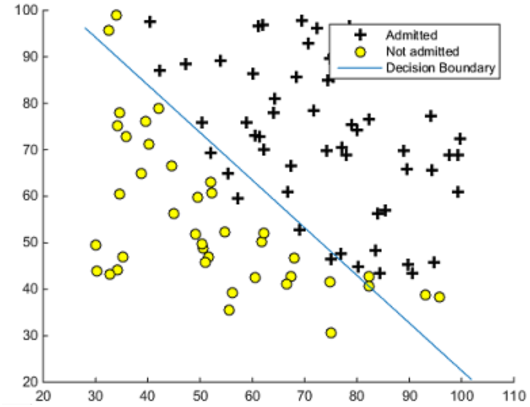

Figure 1: Example of Decision Boundary provided by Mr. Thunder

In this example, the classes are circles and crosses. As noted in the legend, the crosses indicate admitted students, and the circles represent not admitted students. The decision boundary is linear and is mostly accurate. However, there are some misclassifications where some crosses are on the left side of the decision boundary, and there are some circles on the right side of the decision boundary. Moreover, there are points that are extremely close to the decision boundary. These points contribute to the majority of the inaccuracies in models, and therefore, provide the most potential information gained of all the test data points. Each algorithm produces a decision boundary. In Active Learning, the goal is to identify the closest points to the decision boundary and label those points.

Once the framework identifies the most uncertain points, the frameworks returns them to the human (or oracle) to be labelled, and added to the training set. The training set then becomes (*10% of data + 10 data points*), and the test set then becomes (*90% of data - 10 data points*). This process of predicting, sampling, and labelling continues until a threshold is met. In our experimentation, we chose to repeat the process until 60% of the data is labelled. A training set of 60% is reasonable for future data analysis. Figure 2 provides a diagram which visualizes the whole process of the Active Learning framework.

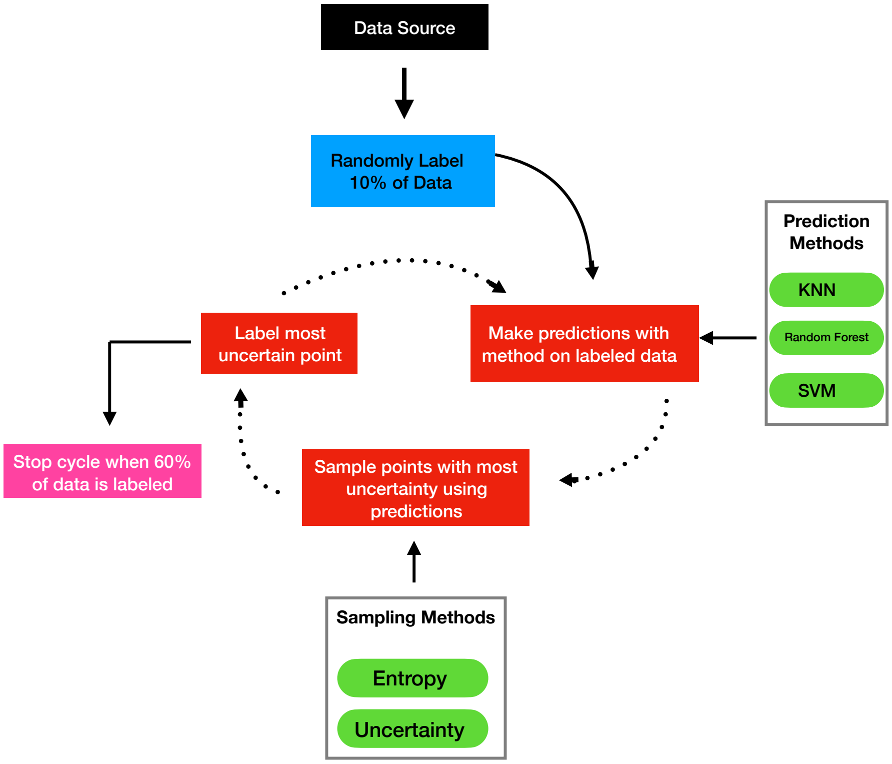
Data Source
Randomly Label 10% of Data
Prediction Methods
KNN
Random Forest
SVM
Label most uncertain point
Make predictions with method on labeled data
Stop cycle when 60% of data is labeled
Sample points with most uncertainty using predictions
Sampling Methods
Entropy
Uncertainty

Figure 2: Active Learning Framework Diagram

Maximum Utility

Utility maximization is a concept of economics that states organizations should seek to attain the most satisfaction possible from their decisions (Clark 95). A company may consider collecting data either on the population, their organization, or clients. Data collection provides a cost and benefit. More data provides a company with a higher representation of data, and therefore, more confidence and accuracy on models. However, data collection is a very expensive task. It is likely companies will outsource data collection to other companies or create entire data governance teams within the company. Organizations need a way to find the maximum amount of data that results in high accuracies while minimizing their overall data collection. Utility maximization is the modern economic theory to do so. The equation below finds the maximum utility of a product.

$$\text{Maximum Utility} = \frac{\text{Marginal Utility of Product}}{\text{Price of Product}}$$

In our experimentation, the "Marginal Utility of Product" is the marginal accuracy of a data sampling iteration, and the "Price of Product" is the cost of each iteration of data sampling. The marginal accuracy is calculated by $A_n - A_{n-1}$, where A_n is the accuracy of the current iteration's training set and A_{n-1} is the accuracy of the previous iteration's training set. The initial cost of data sampling is \$10 and each additional sampling is \$.25 where each sampling increases the training data by 10 points. We multiplied the data sampling iteration with the accuracy for each iteration and find the maximum value. The iteration with the maximum value will be the recommended amount of data collection. We chose to have an initial cost of \$10 and each additional sampling cost \$.25 because these appear to be within scale of data collection prices within the Natural Language Processing field. However, this is just an estimate, and it is impossible to know without a real consultation. In fact, many data services, such as IBM, would provide the service completely free because our data is very limited ("Pricing").

Literature Review

Our project is built off decades of prior research. The prediction algorithm of Active Learning that is standardly used the most is Support Vector Machines. This literature review will discuss prior research, debates, and knowledge gaps in the field of Active Learning. Dr. Jason Baldridge and Dr. Miles Osborne provide the basic benefits and limitations of annotating data through Active Learning. In their paper, “Active Learning and the Total Cost of Annotation”, they discuss different sampling methods, such as uncertainty sampling, random sampling, and query committee. They find that each sampling method has a valuable contribution to certain datasets. For example, they mention that random sampling, randomly selecting points to be labelled, is more reliable for improving reusability of models than any other Active Learning sampling strategy. Meanwhile, uncertainty sampling, which is the method of labelling points closest to the decision boundary, has shown to dramatically reduce the cost of annotation of highly informative datasets such as speech and language data (Baldridge 208). In this research, we chose experiment with uncertainty sampling due to its high performances in studies.

On the same topic of sampling, Dr. Sanjoy Dasgupta and Dr. Daniel Hsu discuss sampling bias and current unsupervised learning techniques to obtain strong sampling in their paper, “Hierarchical Sampling for Active Learning”. Their thesis is that Active Learning heuristics choose data points to label but change the underlying data distribution. Dasgupta argues against Baldridge that there is not a correct way to select points because each model carries its own bias. Therefore, they propose using hierarchical clustering and choosing random points from each cluster to begin the Active Learning process (Dasgupta 209). In these two papers by Baldridge and Dasgupta, there are multiple ways to sample the unlabeled data. Different sampling methods should be used for different scenarios and the limitations of each query strategy should be considered.

As stated earlier, the current method to create the decision boundary within a limited dataset is Support Vector Machines. Dr. Klaus Brinker discusses the successes of Support Vector Machines in many fields such as document classification and computational chemistry in his paper, "Incorporating Diversity in Active Learning with Support Vector Machines". In the Active Learning paradigm, Support Vector Machines select a training example from a finite set of unlabeled examples in each iteration of training. While this is very effective, it is also very time consuming. Brinker presents an approach to construct batches of new training examples from the finite set. His results yield Support Vector Machines with sampling batches that produce better results and training data than Support Vector Machines with single extraction (Brinker 6).

A research team led by Dr. Yukun Chen found in their study that Active Learning can accelerate the use of electronic health records (EHR). They decided to experiment with uncertainty sampling and random sampling. This is a useful study because electronic health records can be time-consuming and costly to review and label. Similar to Brinker's paper, they use Support Vector Machines to be their prediction model. In their study, they predicted rheumatoid arthritis, colorectal cancer, and venous thromboembolism. Their results concluded that Active Learning with SVM classifiers could reach high AUC scores of over .90 with a few dozen annotated samples, meanwhile strategies with Random Sampling only reach AUC scores of .6 (Chen 253). Their results are encouraging for Active Learning research in the healthcare domain. The hope is that the medical field's results can be indicative to NLP datasets.

In most recent research of Active Learning regarding algorithms, neural networks and tree ensemble methods have been a very popular topic. In Dr. Ozan Sener and Dr. Silvio Savarese's paper "Active Learning for Convolutional Neural Networks: A Core-Set Approach", the authors discussed an approach using convolutional neural networks to be the prediction algorithm on image datasets. They proved that a batch solution is impossible for neural networks to complete. Therefore, they redefined Active Learning as a core-set problem. A core-set is a small set of points that can estimate the shape of larger points. Therefore, Sener and Savarese optimized their neural networks to return a set of points that are most representative of the entire data sets (Sener 7). Their results suggested that Convolutional Neural Networks can be used effectively to predict image datasets. Although we do not research into neural networks in this experiment, we find it important that many prediction algorithms are being utilized in Active Learning research.

Dr. Rasoul Karimi discusses the possibility of decision trees within Active Learning. In Karimi's paper, "A Supervised Active Learning framework for recommender systems based on Decision Trees", he experimented with two types of tree-based methods to predict movie recommendations for Netflix users. He found that Bagging was the most effective method in his experimentation. His query strategy was to label the item that would maximize the reduction of the mean square error (MSE) of the whole model. This is the first sampling method discussed where an error metric is being maximized (Karimi 40). Karimi did not use batch sampling and decided to use single sampling in each iteration. Karimi's research is encouraging as we implement a decision tree ensemble method (Random Forest) for our dataset.

In Dr. Zhenfeng Zhu and Dr. Xingquan Zhu's paper, "Transfer Active Learning", they discuss the possibilities of not having unlabeled samples from the same domain as the labeled samples. Zhu states it is possible that there remains data from an auxiliary target domain. In their paper, they seek to know if an Active Learner can label samples from auxiliary domains to benefit the target domain. Using a specialized Support Vector Machine, they determined that an auxiliary domain can be used to predict instances of the original target domain (Zhu 2169). Interestingly, they found a Support Vector Machine worked significantly better than an ADA Boosting algorithm. Both the Support Vector Machine and ADA Boosting algorithm had a prediction accuracy of 100% but the ADA Boosting required far more iterations to reach 100% accuracy. It is notable that the Support Vector Machine outperformed a Decision Tree ensemble method in another domain area.

Similar to Zhu and Zhu, Dr. Piyush Rai and Dr. Hal Daume researched how one domain can be used to predict another domain. In their paper, "Domain Adaptation meets Active Learning", they discuss the challenges of finding complete datasets in niche fields. Domain adaptation is the goal of using labeled data from a different source domain to be predictive of the target domain. They use a completely different sampling method to collect their labeled samples. They use stream-based selective sampling which is when each unlabeled point is examined one at a time evaluating its certainty of each item against its query parameter. While they had success with these methods, they also found a large limitation. Their drawback is that an initial pool of labeled target domain data must be available to create an initial target domain classifier (Rai 32).

This literature review has given a glimpse into the field of Active Learning. We have seen the benefits, limitations, and experimentations with the two main aspects of Active Learning: predicting and sampling. We have seen the classical methods of Support Vector Machines and explored research of tree ensemble methods. Further, we have explored the efficiencies of each sampling method including uncertainty sampling, random sampling, query committee, and stream-based selective sampling. Unfortunately, we were unable to find any sources that discusses the K-Nearest Neighbors (K-NN) algorithm. Hopefully, our research will be able to provide insights how K-NN performs in an Active Learning framework. With the review of prior work, we will continue our own research into effective data collection.

Data and Methodology

The general methodology of the research project was to collect many combinations of prediction algorithms and sampling methods. In our experimentation, we collected two performances statistics for each iteration of data sampling, model accuracy against entire data and time to complete each iteration. From these two measurements, we identified which combination provided the most efficient Active Learning framework. Additionally, the data collected allowed us to detect the optimal value of data to collect. We find the maximum utility by calculating the marginal accuracy and cost values in each iteration (Clark 92).

Data

The data set was a collection of email messages that are labelled as spam or not spam. The dataset was collected by the Hewlett Packard Labs in Palo Alto, CA. They allowed the data to be publicly available and there are no other ethical concerns with this data because the identities of the senders and recipients are unknown. Spam messages are emails sent to a very large group of people for commercial and advertising purposes. Many email services provide an automatic spam filter which will read the content of your email and decide if the email should go the inbox or to a designated spam folder. The purpose of the spam filter is to alleviate the amount of unnecessary and aggravating emails that a user may receive. An example of a spam message in the dataset is below.

> WINNER!! As a valued network customer you have been selected to receive a £900 prize reward!
> To claim call ##########. Claim code KL341. Valid 12 hours only.

In this experiment, we recreated the spam filter using many Active Learning frameworks. There are 5,567 observations with 13.4% identified as spam messages, and the remaining 86.6% labelled as not spam. Our framework includes prediction algorithms that will mimic the spam filter by predicting if an email is spam or not spam. Our initial size of the training set is 556 and incrementally increases by 10 data points in each data sampling iteration.

Pre-Processing Data with NLP

As mentioned in the motivation portion in the introduction, we applied Active Learning frameworks to a Natural Language Processing with the hopes that it will be generalizable for many NLP datasets. We used the Python Library nltk, Natural Language Toolkit, to perform many pre-processing steps (Bird). The nltk library allowed for easy feature extraction in a natural language dataset. Using its stop words library and stemming tools, we eliminated many words that carry no important context, such as pronouns, transition words, and articles (a, as, the), and removed words that are very similar that carry similar meaning and convert them to their stem. For example,“winning” and “won” are converted into “win”. Then, we vectorize the dataset by converting the messages into a frequency of each stemmed word in the dataset. Therefore, we convert our $5567 \cdot 2$ matrix into a $5567\cdot$ (*# of stemmed words in training set*). Since the initial training set is randomized every time, the number of stemmed words will be slightly different. Typically, the number of stemmed words in the dataset is between 2,000 and 2,050 words. Table 1 provides an example of clean data that we input to our Active Learning frameworks.

index	body_len	punct%	0	1	2	3	...	2000	2001	2002
805.0	37.0	10.8	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
1548.0	7.0	42.9	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
4078.0	38.0	2.6	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
4308.0	31.0	9.70	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0
4726.0	62.0	4.8	0.230	0.0	0.0	0.0	...	0.0	0.0	0.0
1229.0	28.0	10.7	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0

Table 1: Example of Clean Data

K-Nearest Neighbors

The first prediction algorithm used was K-Nearest Neighbors. The K-Nearest Neighbors algorithm computes the distances of each point from the training set to the testing point. The algorithm then sorts the training data points based off of the computed distances. The algorithm takes the closest k amount of elements and the class that has the highest frequency becomes the testing point's predicted value. K is a parameter that is decided by the user. After trialing many values of k, we found $k = 5$ was the optimal value at maximizing the accuracy. The K-NN algorithm is extremely helpful as a predictor for datasets that have a smaller amount of predictors. Typically, a dataset with more predictive variables incurs the curse of dimensionality. With more dimensions, the model becomes less predictive and statistical significance dramatically decreases. Therefore, the K-NN algorithms may struggle with this particular dataset due to its high dimensionality.

Random Forest

Random Forest is a tree ensemble method that builds trees by selecting a random subset of attributes from the original dataset. Each subtree is a decision tree that uses different features of the dataset. Decision trees are effective supervised learning techniques to predict a target variable. Decision trees' nodes start as observations and conclude as predictions. As a single observation traverses down the tree, the prediction is the value of the leaf node at the bottom of the tree. The tree is built by splitting data into subgroups that decrease a specified error metric. In a Random Forest model, the number of attributes is $\sqrt{d}$, where d is the number of attributes. For every observation, each tree will produce a prediction, and the class that was predicted the most in the subtrees will be the overall prediction.

Support Vector Machines

Support Vector Machines is the final prediction algorithm that we implemented in our Active Learning frameworks. The main objective of a Support Vector Machine is to identify the n-dimensional space that distinctly classifies the data, where n is the number of features. Each feature space can produce a line or a plane (if $n > 2$) that separates the classes. In SVMs, the algorithm chooses the plane that maximizes the marginal distance between data points of each class. It does this so that future data points can be predicted with more certainty. Support Vector Machines are the most common algorithm to use in a Active Learning framework because it is highly efficient with a limited amount of data.

Sampling Methods

As mentioned earlier, there are two sampling methods used in the our Active Learning frameworks: Uncertainty and Entropy. Uncertainty is the simplest measure as it measures the probability of uncertainty of its class prediction. This is calculated by the following equation.

$$U(x) = 1 - P(\hat{x}|x)$$

where x is the prediction and $\hat{x}$ is the most likely prediction

In this method, the Active Learning framework selects the samples that have the highest uncertainty probability to their predicted class.

Entropy sampling is another method that measures the amount of uncertainty in class prediction. Entropy is a measure of disorder in an environment, and prediction algorithms seek to minimize the amount of disorder or uncertainty in a model. Below is the equation to measure entropy.

$$E(x) = -\sum p_k * log(p_k)$$

In the entropy equation, p_k is the probability of the sample belong to the kth class. The Active Learning framework samples the points that produce the highest entropy because they have the most information to give about the decision boundary of a dataset.

Results

Prediction Algorithms

The overall results provided some interesting insights as it appears that Random Forest performs very well while Support Vector Machines are under-performing despite being the current standard of practice (see page 16). In Figure 3 that compares the algorithms' accuracy over the iteration, it is clear that Random Forest outperforms K-NN and SVM significantly. It should be noted that these results were the aggregated scores for each sampling method. Random Forest's initial accuracy with a 10% training set was 95% and continues to improve to 98% with 30% training data. After which, the accuracy plateaus and remains stagnant at 98%. K-NN had the lowest initial accuracy with 86%, and steadily climbs to 91% with 30% labelled data. SVM's initial accuracy was 87% and sees very little improvement despite gaining more data. It only increases 2 percentage points to an 89% accuracy score. It is clear that Random Forest significantly produced the most accurate models.

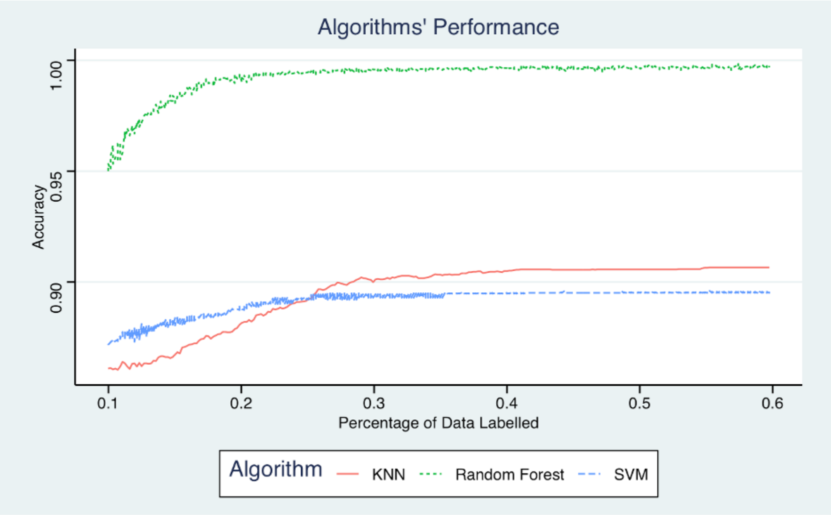

Figure 3: Algorithms' Accuracy Performance over each Iteration

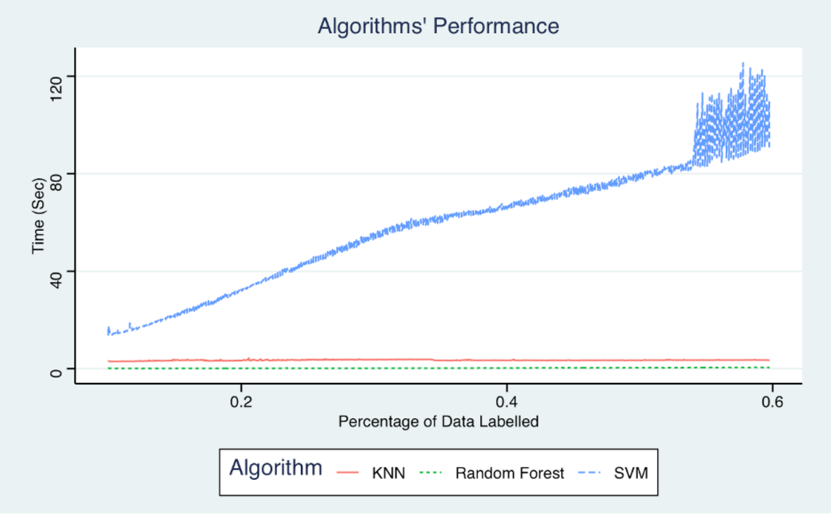

Figure 4: Algorithms' Time Performance over each Iteration

As for the time graph in Figure 4, SVM performs the worst of the three algorithms. SVM's initial run time was 13 seconds and increases relatively linearly with a $\frac{1}{3}$ second increase at each iteration. This linear trend continues until 55% of the data is labelled. After which, the time has sharp increases and decreases that reach as high as 125 seconds, but never less than 95 seconds. K-NN's initial run time was 3 seconds and never surpassed 4 seconds. This is to be expected that K-NN's run time would remain consistent throughout the entire process. The nature of K-NN's algorithm requires calculating the same amount of distances over the training and test sets so it is not surprising that the run time has no significant change over iterations. Random Forest had the most impressive run time of the three algorithms. Its initial run time was .22 seconds and only increased to .55 seconds. Similar to the accuracy metric, Random Forest significantly outperforms the other algorithms.

It is clear that the Random Forest algorithm is the superior prediction algorithm for this Natural Language Processing dataset. Next, we will check which sampling method performed better. We only evaluated the accuracy and time of each sampling method with the Random Forest algorithm. We continue our analysis with only Random Forest because of its significantly strong results for both its accuracy and run times.

Sampling Methods

The results of each sampling method were very similar. In Figure 5's accuracy graph, there did not appear to be any significant difference between the sampling methods. They were not choosing the exact same points since their accuracies were slightly different. On the other hand, they were not choosing radically different points because their accuracies were very similar. Uncertainty sampling's initial accuracy was 95% meanwhile Entropy sampling's initial accuracy was 95.5%. Both accuracies continued to increase to 98% at the same pace. We concluded that there was not a sampling method that we could consider more efficient than the other.

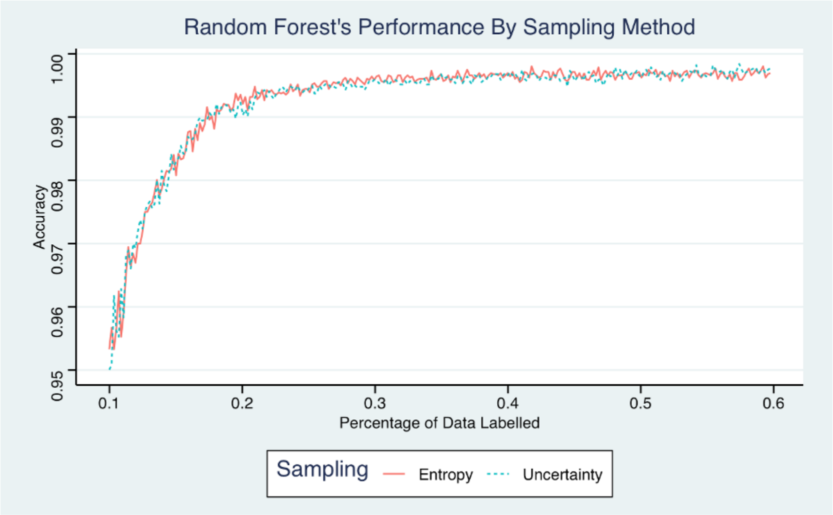

Figure 5: Random Forest's Accuracy Performance for each Sampling Method

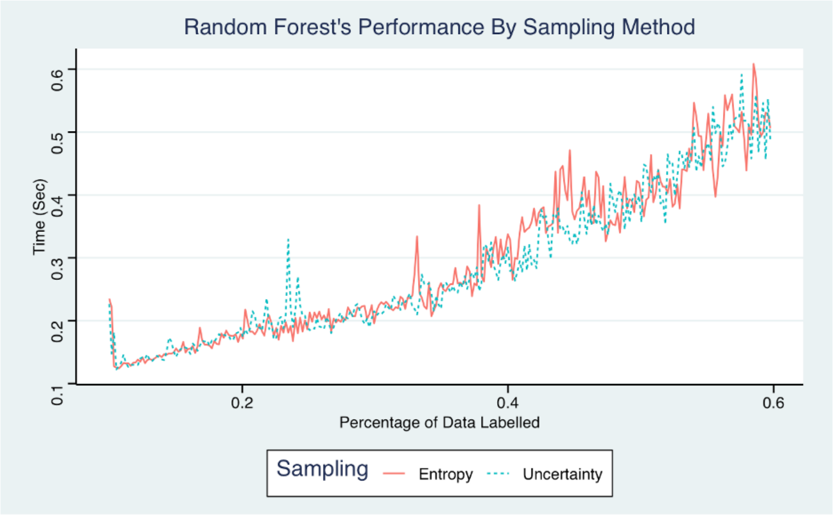

Figure 6: Random Forest's Time Performance for each Sampling Method

For run time in Figure 6, the result was very similar. Both sampling methods' initial run time began at .22 seconds and steadily increased to .50 seconds. Each had some interesting outliers as it would randomly increase by .1 second and decrease by the same amount. Similar to accuracy, there is no clear advantage of one sampling method to another. Since neither sampling produced a result that concluded one sampling method was more efficient than the other, we decided to continue with Uncertainty sampling. We chose Uncertainty sampling due to its evidence to work on many data sets in variety of fields (Sharma 165). Therefore, to make our experiment as generalizable as possible, we continue to use Uncertainty sampling with the Random Forest prediction algorithm as our Active Learning framework.

Utility Maximization

Since we identified the preferred Active Learning framework as Random Forest with Uncertainty sampling, we identified the optimal number of data sampling iterations. Figure 7 provided the necessary information to find the recommended amount of data sampling. In general, the smoothed graph showed a typical cost and utility graph. The marginal increase of accuracy decreased exponentially as each new data sampling provided less information. As we can see in the graph below, the Maximum Utility function recommended 60 data iterations to optimize cost and accuracy. As a reminder, the initial size of the data was 5,567 so the initial training size was 556. Each sample of data increased the training size by 10. Therefore, the optimal amount of data to effectively maximize the accuracy and minimize cost of data collection was 1,156 data points.

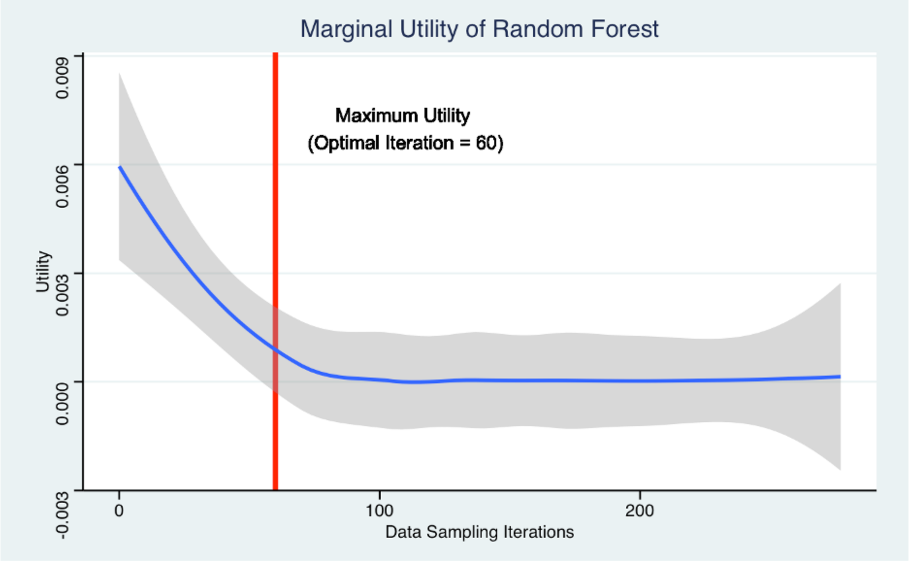

Figure 7: Max Utility of the Random Forest (Uncertainty Sampling) Active Learning Framework

Discussion

The Random Forest prediction algorithm provided the highest accuracy and lowest run time of our three algorithms. This provided a strong basis to continue our analysis with only the Random Forest. This finding was very surprising to us considering that Support Vector Machines are the standard practice. Random Forest could a better prediction algorithm in the context of Natural Language Processing datasets, but we are no position to confirm that claim. There was not a significant difference between Entropy and Uncertainty sampling in either metric of accuracy and time. In support of prior research, we decided to continue with Uncertainty sampling in hopes to make the experiment more generalizable. Our results showed that the optimal data sampling iterations was 60 which meant that 1,156 data points was sufficient in maximally increase the accuracy while decreasing the cost of data collection.

The results of our experiment indicate that we can still have extremely high accuracy with less data than the original dataset. It appears our experiment agrees with some prior research. In our results, we found that Random Forest was an extremely accurate dataset and this aligns with Dr. Karimi's research. Karimi found that decision-tree-based Active Learning frameworks were most effective on recommender systems on Netflix data (Karimi 40). This shows that decision trees and Random Forest models have the potential to increase generalizability in the Active Learning research community. Moreover, our Active Learning framework did produce a less intensive data collection process and decreased the cost of data annotation. This is also conclusive with Baldridge's research that Active Learning has shown to drastically reduce the cost of annotation of highly informative datasets such as natural language data (Baldridge 208).

While it is great that our research is representative of prior Active Learning research, our results could hopefully build on two different domains. The first being the Active Learning research domain at large. Hopefully, we added to the body of knowledge within the Active Learning space, and contributed some findings that could be used in future work. The second domain is data collection decision-making. It is our hope that our experiment could provide as an example to organizations on efficient data gathering. As organizations are becoming more data-driven, they must also make smart decisions on how to collect their data and how they should do so in an appropriate and advantageous manner.

Limitations

There were a few glaring limitations in this experimentation. A portion of our central question was finding the most effective Active Learning framework for NLP datasets. We were only able to experiment with one NLP dataset. Therefore, we cannot generalize that our framework will be the most effective for every Natural Language Processing dataset. Another limitation of our experiment was that price was considered as the only cost when optimizing the amount of data to be collected. In a real-world scenario where an organization would be trying to collect data, they must consider more cost values than just price such as time to acquire data and time to clean data. The last major limitation was our constraint on the number of models we could experiment. There are many other prediction algorithms and sampling methods that could have been tested. While we are very confident in our results, we believe there is still more value to be gained if more frameworks can be tested.

Future Considerations

Building off our limitations, the future considerations are too mitigate the restraints. Our first suggestion for future work is simulate variable costs and time to gather data. It is likely that each data sampling will be a consistent flat rate. Therefore, we believe that there will be value in researching average data collection costs and implement those into the maximum utility model. Additionally, we think an aggregated cost value could be more realistic to use in the future. Data collection's two main cost values are price and time. Simulating time to gather data could be valuable and insightful for practical use. The new cost value could then be an aggregated combination between price and time of data sampling. The second consideration is to experiment with more frameworks. Due to the high performance of Random Forest, it would be valuable to research other Decision Tree Ensemble methods such as the various Bagging and Boosting algorithms. However, research into neural networks, discriminant analyses, and even perhaps Principal Component Analysis could prove to be very effective as well. Lastly, we recommend to experiment with more sampling methods. Uncertainty and Entropy sampling are just the basic measures of calculating uncertain points close to the decision boundary. There could be high success with other sampling methods such as classification margin, hierarchical sampling (Dasgupta 210), Least Confidence sampling and Margin of Confidence sampling.

Work Cited

Baldridge, Jason, and Miles Osborne. "Active Learning and the Total Cost of Annotation." ACL Anthology, https://aclanthology.org/W04-3202/.

Bird, S., Klein, E. Loper, E. Natural language processing with Python: analyzing text with the natural language toolkit, 2009.

Brinker, Klaus. "Incorporating Diversity in Active Learning with Support Vector Machines." Incorporating Diversity in Active Learning with Support Vector

Chen, Yukun, et al. "Applying Active Learning to High-Throughput Phenotyping Algorithms for Electronic Health Records Data." Journal of the American Medical Informatics Association, vol. 20, no. e2, 2013, https://doi.org/10.1136/amiajnl-2013-001945.

Clark, Colin W., and Gordon R. Munro. "The Economics of Fishing and Modern Capital Theory: A Simplified Approach 1." Fisheries Economics, 2019, pp. 47–61.

Dasgupta, Sanjoy, and Daniel Hsu. "Hierarchical Sampling for Active Learning." Proceedings of the 25th International Conference on Machine Learning - ICML '08, 2008, https://doi.org/10.1145/1390156.1390183.

"IBM Watson Natural Language Understanding - Pricing." IBM, https://www.ibm.com/cloud/watson-natural-language-understanding/pricing.

Karimi, Rasoul, et al. "A Supervised Active Learning Framework for Recommender Systems Based on Decision Trees." User Modeling and User-Adapted Interaction, vol. 25, no. 1, 2014, pp. 39–64., https://doi.org/10.1007/s11257-014-9153-z.

Rai, Piyush, et al. “Domain Adaptation Meets Active Learning.” Domain Adaptation Meets Active Learning — Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing, 1 June 2010. Sener, Ozan, and Silvio Savarese. “Active Learning for Convolutional Neural Networks: A Core-Set Approach.” ArXiv.org, 1 June 2018, https://arxiv.org/abs/1708.00489.

Ratnaparkhi, Adwait. ”A maximum entropy model for part-of-speech tagging.” Conference on empirical methods in natural language processing. 1996.

Sharma, Manali, and Mustafa Bilgic. “Evidence-Based Uncertainty Sampling for Active Learning.” Data Mining and Knowledge Discovery, vol. 31, no. 1, 2016, pp. 164–202.

Zhu, Zhenfeng, et al. “Transfer Active Learning.” Transfer Active Learning — Proceedings of the 20th ACM International Conference on Information and Knowledge Management, 1 Oct. 2011, https://dl.acm.org/doi/abs/10.1145/2063576.2063918.

Appendix

All code and datasets can be found at this GitHub Public Repository