Active Learning for Efficient NLP Training

Stanford CS224N Custom Project

Daniel Lee
Department of Computer Science
Stanford University
jaeylee@stanford.edu

Thomas Yim
Department of Computer Science
Stanford University
yimt@stanford.edu

Ibrahim Dharhan
Department of Computer Science
Stanford University
abraham7@stanford.edu

Abstract

In this project, we investigated active learning techniques to improve the efficiency of fine-tuning the BERT-base-uncased model for natural language processing tasks, including multiple-choice question-answering using the CommonSenseQA dataset and spam detection using the Enron Spam dataset and the SMS Spam Collection dataset. Our objective was to achieve performance comparable to fully supervised training while significantly reducing the amount of labeled training data required. We evaluated several active learning strategies, including least certainty, query by committee, expected model change, and representativeness, and compared their effectiveness on three different datasets against random selection. Although none of the active learning methods yielded better performance than random selection on the Enron Spam and CommonSenseQA datasets, all methods outperformed random selection on the SMS Spam dataset. We also discovered that the active learning method may oversample a minority class to improve performance when the training subset is small. These results suggest that while active learning can be effective for NLP tasks, its success is highly dependent on the dataset used. Our findings aim to contribute to more cost-effective and scalable training methodologies for NLP applications.

1 Key Information to include

- Mentor: Arvind Mahankali
- External Collaborators (if you have any): No
- Sharing project: No

Team contributions: Ibrahim worked on implementing the query by committee method and plotting results. Thomas worked on the experiment loop, preprocessing data, the random selection method, and the expected model change method. Daniel worked on implementing the least certainty method and the representativeness method. Everyone worked together to produce the results and write this report.

2 Introduction

Deep learning models have achieved state-of-the-art performance on many natural language processing (NLP) tasks, including named entity recognition (NER), translation, question-answering, and sentiment analysis. Despite their success, these models typically require vast amounts of labeled

training data, which can be expensive and time-consuming to obtain. This dependency on large labeled datasets poses a significant challenge for scaling NLP applications, especially in domains where labeled data is scarce or costly to acquire.

Active learning presents a promising solution to this problem. Unlike traditional model training, which chooses all examples or randomly selects a subset of the data to be labeled, active learning techniques selectively choose the most informative data points to label and then train on. By focusing on the data that is expected to provide the most significant performance gains, active learning aims to maximize model accuracy while minimizing the amount of labeled data required. This approach not only reduces the annotation effort but also decreases training time and cost, making it a more efficient and scalable method for training deep learning models.

In this project, we focused on improving the fine-tuning efficiency of the BERT-base-uncased model using active learning across various datasets, including the CommonSenseQA dataset for question-answering, and the Enron Spam and SMS Spam Collection datasets for spam detection. Our objective was to demonstrate that active learning could achieve comparable or superior accuracy to traditional methods while utilizing fewer labeled data points.

We explored several active learning strategies: least certainty, query by committee, expected model change, and representativeness. These methods were tested against random selection to evaluate their effectiveness. Although our results showed that none of the active learning techniques yielded significant performance gains on two of the datasets, they all surpassed random selection on one dataset. This variability underscores the context-dependent nature of active learning's success in NLP tasks.

Our research contributes to the ongoing effort to develop more cost-effective and scalable training methodologies for NLP applications. By demonstrating the potential and limitations of active learning, we provide insights that can guide future research and practical implementations in the field.

3 Related Work

Active learning has become a topic of considerable interest in the field of NLP especially as it is increasingly expensive to label all the large corpora of free text. One of the foundational works in active learning was done by Lewis and Gale [1], who demonstrated that uncertainty sampling could reduce the amount of training data required to achieve a certain accuracy by up to 500-fold in text classification tasks. This was done by labeling the data the model was least confident in. Their study involved a simple naive Bayes model categorizing news headlines, but the concept of choosing low-certainty examples to train on has been extended to a variety of NLP tasks.

Applying these active learning techniques to deeper neural networks presents various challenges. Most active learning techniques are designed for contexts with fewer examples, while deep learning is typically dependent on large amounts of data. Gal et al. [2] explored Bayesian approaches and active learning for image classification on the MNIST dataset, which eventually led to the query-by-committee active learning technique we explore in this paper.

Then Sener and Savarese [3] explored the concept of selecting a core set of examples and then using a model trained on that data to then choose the remaining data points to label. This significantly outperformed the existing approaches in image classification thus we adopted this approach to our NLP classification tasks. We attempted to implement and evaluate a variety of active learning methods across different Natural Language Processing tasks. In doing so we hoped to understand what types of tasks these methods would work best on and how the class distribution of the chosen examples affected performance.

4 Approach

In this project, we fine-tuned the BERT-base-uncased model on the CommonSenseQA dataset, Enron Spam dataset, and SMS Spam Collection dataset to explore various active learning techniques, aiming to enhance the efficiency of the model with respect to labeled training data. Our experimental framework trained and then evaluated a new model for each training subset size for every active learning method. To minimize noise from different random initial core sets of examples, we trained the model using five different seeds and computed the average accuracy at each interval. As a baseline,

we fine-tuned the model without any active learning techniques, training it on the next available random set of data. We then used the following set of active learning techniques [4] to try and improve the model accuracy:

1. Least Certainty: This is the most simple and frequently used active learning technique. This technique chooses to train on examples that the model is least certain about how to label. Least Certainty for a training example x can be represented as

$$LC(x) = 1 - P_\theta(\hat{y}|x)[4]$$

where $\hat{y} = \text{argmax}_y P_\theta(y|x)$. The next subset of training examples to be selected are those with the highest least certainty scores.

2. Query by Committee (QBC): Here we used the entropy of the distribution of classifications to find which data the model should be further trained on, selecting the examples with the highest entropy. Allowing k to be the number of models in our committee and $V(c, e)$ to be a function defining the number of models that classified training example e as class c, we define the vote entropy of an example e as

$$\text{Entropy}(e) = \Sigma_c \frac{V(c,e)}{k} \log(\frac{V(c,e)}{k}) \text{ [5]}$$

Instead of instantiating and training multiple models, we emulated a committee by applying different dropout masks on our original model, which is more computationally and memory efficient. However, it's worth noting that dropout masks are more likely to create a less diverse and rich ensemble of models compared to instantiating independent models.

3. Expected Model Change: By hallucinating an example's label to be the one with the highest probability, we can do a backward pass over the final classification layer of the model to obtain the magnitude of the gradient change. Let

$$\nabla_\theta(x, \text{argmax}_y P(y|x))$$

be the gradient with respect to the model parameters from an example x and its hallucinated label y. The examples with higher gradient norms indicate more informativeness and represent the next set of examples that the model should be trained on.

4. Representativeness: This involves choosing a diverse subset that will be representative of all our data, which can be achieved using K-means clustering. To select the subset of examples $\hat{S}$, we can define the following equation:

$$\hat{S} = \cup_{k=1}^{K} \{x_i \in \mathcal{X} | x_i = \text{argmin}_{x_j \in C_k} ||x_j - \mu_k||\}$$

where $\hat{S}$ is the selected subset of representative samples, K is the number of clusters, $\mathcal{X}$ is the entire dataset, C_k is the set of samples in the k-th cluster, x_i is a sample in the dataset, μ_k is the centroid of the k-th cluster, and $||x_j - \mu_k||$ is the Euclidean distance between sample x_j and the centroid μ_k. To get the embeddings of each sample, we used the sentence-transformers/all-mpnet-base-v2 model from HuggingFace, a sentence transformer model that maps sentences to a 768-dimensional dense vector space, since this model was created for the purpose of clustering. This equation ensures that we select the most representative samples in each cluster based on the k-means clustering algorithm, ensuring that the selected samples are those closest to the centroids of their respective clusters. We can follow this equation until we have satisfied the total number of samples needed. For the CommonSenseQA dataset, we chose K to be $10, 100,$ and 1000 to identify which value yielded the best result, as there was no clear way to categorize the questions. For the Enron Spam dataset and the SMS Spam Collection dataset, we chose $K = 2$ since there was a clear way to categorize the input text (spam or not spam), and we tested $K = 10$ to see if having a larger value of K would yield better results. Additionally, for the Enron Spam dataset, we computed the clusters based on the combined text of email subject + body, as well as only on the subject, as the body text was very messy compared to the subject text.

Although not exhaustive, these methods represent a comprehensive exploration of the different ways the next subset of data can be chosen using active learning techniques.

5 Experiments

5.1 Data

5.1.1 CommonSenseQA

We used the CommonsenseQA dataset [6] containing 9741 training and 1220 test examples of multiple-choice questions that require common sense knowledge to answer correctly. For each question, there are 5 potential answers with one correct and four distractor answers. Here is an example:

Question: Google Maps and other highway and street GPS services have replaced what?

Options: A. united states, B. mexico, C. countryside, D. atlas, E. oceans

Answer: D

5.1.2 Enron Spam

The Enron Spam dataset [7] was collected from emails of Enron employees that were publicly released after an investigation into the company. We pulled from the HuggingFace SetFit/enron_spam dataset. The dataset includes 31700 training emails and 2000 test emails. However, we limited the size of the training data to be 2000 examples as the model's performance began to plateau at that size. An example of a non-spam and spam email can be found below:

Text: funding deal louise, we have a customer that wants to monetize a couple of itm deals...
Label: Not Spam

Text: looking for property in spain ? looking for property in spain ? don't waste your time !
Label: Spam

5.1.3 SMS Spam Collection

The SMS Spam Collection [8] is a public set of SMS labeled messages that have been collected by UC Irvine for mobile phone spam research. The dataset contains 3865 training and 1709 test examples, but we again limited the training data to a size of 2000 for our experiments. Examples of non-spam and spam messages are provided below:

Text: Nah I don't think he goes to usf, he lives around here though
Label: Not Spam

Text: Had your mobile 11 months or more? U R entitled to...
Label: Spam

5.2 Evaluation method

We utilized the BERT-base-uncased model as our pre-trained model architecture and fine-tuned it on three different datasets: CommonSenseQA, Enron Spam, and SMS Spam Collection, employing various active learning techniques. To pre-process the CommonSenseQA data, we created five sentences each with the question and one of the five potential answers. The model then generates a score for each sentence and softmaxes them to output a probability distribution. For the spam datasets, the input text was passed into a model and a probability was generated for how likely it is spam. The process for applying active learning techniques was as follows:

1. Fine-tune the BERT model on a small subset of the training dataset.
2. Run a forward pass of the fine-tuned model on the remaining unlabeled data from the training set.
3. Use one of the active learning techniques to select a batch of informative samples from the unlabeled data.
4. Obtain labels for the selected batch of samples.
5. Retrain the model on the combined dataset consisting of the initial labeled data and the newly labeled batch.

6. Repeat steps 2-5 for a fixed number of iterations.

We measured each model's accuracy on the test set by calculating the percentage of correct classifications across the entire test set. This quantitative metric provided a clear basis for comparing the performance of different active learning methods. For the CommonSenseQA dataset, we retrained the model from scratch at each iteration using training examples of 100, 500, 1000, 2000, 5000, and 9741, then evaluated its performance on the full test set. For the Enron Spam dataset, we retrained the model with 25, 50, 100, 500, and 1000 training examples, and for the SMS Spam Collection dataset, we used training examples of 25, 50, 100, 200, 300, 400, 500, and 1000. These sample sizes were chosen to allow us to thoroughly understand the accuracy trends and the impact of active learning techniques across different dataset sizes. Thus, for each dataset, we are comparing the accuracy at each subset size across the active learning methods.

5.3 Experimental details

We fine-tuned an uncased BERT model with an added classification layer to generate a score for each class. For the CommonSenseQA dataset, we used the HuggingFace AutoModelForMultipleChoice, while for the Enron Spam and SMS Spam Collection datasets, we employed the HuggingFace AutoModelForSequenceClassification. The fine-tuning process was carried out using the Trainer API with default HuggingFace parameters: a learning rate of `2e-5`, batch size of `16`, and weight decay of `0.01`, for 3 epochs on varying subsets of the data. We deliberately did not engage in hyperparameter tuning, as the primary objective of this project was to assess the performance improvement of active learning methods over random selection under arbitrary, constant hyperparameter settings. A complete training job on the entire CommonSenseQA dataset took approximately 40 minutes. Training with 1000 examples on the spam datasets took 10 minutes. To ensure robust results, we trained and evaluated the model on the predefined subset sizes across five different random seeds. This approach helped mitigate the influence of the initial randomly chosen examples and the randomly initialized classifier weights, providing a more reliable comparison of the active learning methods.

5.4 Results

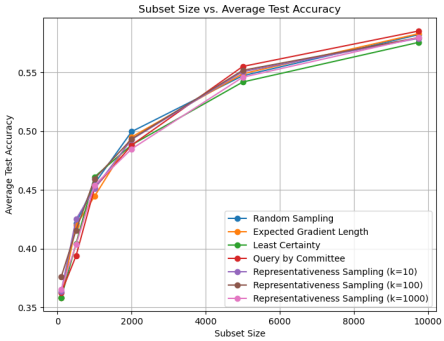

Figure 1: Test accuracies on the CommonSenseQA dataset.

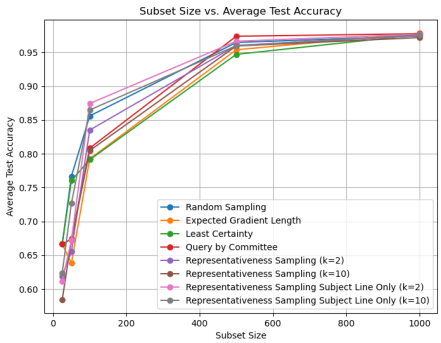

Figure 2: Test accuracies on the Enron Spam dataset.

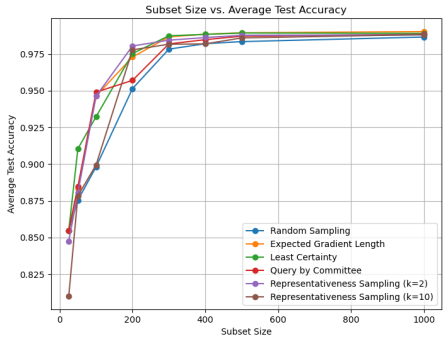

Figure 3: Test accuracies on the SMS Spam Collection dataset.

While none of the active learning models resulted in positive performance on the CommonSenseQA dataset and the Enron Spam dataset, all active learning techniques employed outperformed random selection on the SMS Spam Collection dataset. While we expected active learning methods to work across all datasets, we still showed that there is some benefit to using active learning techniques on some datasets.

We had expected to see the accuracy improvements we saw on the SMS Spam dataset on the other datasets as well. On SMS spam, we see all active learning techniques surpassing random selection's performance on 1000 examples (0.9866) when trained on an informative set of 300-400 examples. By 100 examples, methods like query by committee (0.9491) and representativeness sampling with two clusters (0.9463) exceed random sampling performance by $\sim 5\%$ (0.8981).

6 Analysis

As seen in the figures, the active learning techniques we employed on the CommonSenseQA dataset and the Enron Spam dataset all achieved marginally lower accuracies compared to random sampling. This was an unexpected result, as active learning techniques are supposed to be selecting the next most informative samples to train on and should at least match if not outperform random sampling. There are several potential reasons the active learning techniques did not outperform random sampling on the CommonSenseQA dataset and the Enron Spam dataset:

1. The Enron Spam dataset contained a significant number of malformed examples. We hypothesize that the active learning techniques tended to select these malformed examples for training due to the model's inability to confidently classify them. Training on this malformed data likely hindered the model's performance as it was forced to learn from noisy and invalid examples. In contrast, random sampling would have a better proportion of well-formed to malformed training examples, mitigating the negative impact of training on malformed data and in turn achieving a higher accuracy compared to active learning.
2. The CommonSenseQA dataset poses a significant challenge for a relatively small model such as the BERT-base-uncased model to achieve high accuracy. Reaching a high accuracy on this task requires a deep understanding of general knowledge and common sense, which is likely out of the processing power of the BERT-base-uncased model. We hypothesize that the difficulty of this task, in combination with the limited learning power of our model, hindered the effectiveness of active learning, making it essentially equal to random sampling.
3. The training examples our active learning techniques chose were often skewed and centered on one class. This would result in a polarizing and uneven distribution of the types of examples our model would train on, resulting in a lower test accuracy. Our model would eventually make up for this by focusing on examples it hasn't trained on, but this polarization likely hindered the model's performance.

The active learning techniques used on the SMS Spam Collection dataset worked as expected, all achieving a significant increase in the model's accuracy compared to random sampling. We believe

that active learning was effective on this dataset due to two reasons: the simplicity of the task and the lack of malformed data. We found least certainty, the simplest but most commonly used active learning technique, to be the most effective by the end of training. Expected gradient length and QBC were a close second and achieved a higher accuracy compared to least certainty up to a subset size of 300.

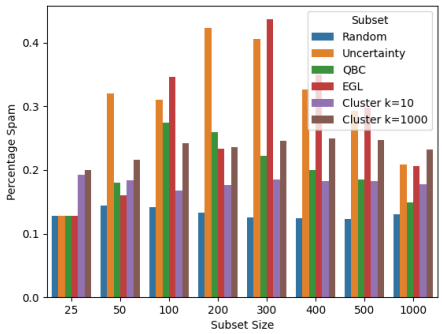

Figure 4: Spam percentage of chosen subsets across active learning methods

The difference in performance from the active learning examples can be partially explained by the above in Figure 4. Since the pool of possible training examples was 13% spam in the SMS dataset, we noticed that the randomly selected subsets maintained this approximate distribution throughout the experiment. This was expected, as the core set of examples was chosen randomly (with the exception of those chosen via clustering techniques).
As the experiment progressed, the distributions for each method began to diverge. Notably, both uncertainty sampling and expected gradient length methods showed a tendency to oversample from the spam class. This led to a notable increase in the proportion of labeled spam examples, reaching over 40% in the subsets of 200 examples. This oversampling phenomenon can be attributed to the inherent bias of these methods towards selecting examples where the model exhibits the most uncertainty or expects the greatest learning impact, which often coincided with the less frequent spam examples. As the model in the smaller-subset evaluations would predict "Not Spam" for everything and achieve 87% accuracy, the oversampled training set could be boosting performance through increased representation of the spam minority class to learn from.

7 Conclusion

In this project, we investigated various active learning techniques to improve the efficiency of fine-tuning the BERT-base-uncased model for NLP tasks, specifically focusing on multiple-choice question-answering with the CommonSenseQA dataset and spam detection with the Enron Spam and SMS Spam Collection datasets. Our objective was to reduce the amount of labeled training data required while maintaining or improving model performance compared to fully supervised training.

Main Findings and Achievements

- **Performance on SMS Spam Collection Dataset**: All active learning methods tested (least certainty, query by committee, expected model change, and representativeness) outperformed random selection on the SMS Spam Collection dataset. This demonstrates the potential of active learning to enhance model performance while reducing labeling effort in certain contexts.
- **Challenges with Enron Spam and CommonSenseQA Datasets**: None of the active learning strategies yielded positive performance on the Enron Spam and CommonSenseQA datasets. These results highlight the context-dependent nature of active learning's effectiveness.
- **Insight into Active Learning Techniques**: Our analysis identified several factors that may contribute to the limited success of active learning in specific datasets, such as the

presence of malformed data in the Enron Spam dataset and the inherent difficulty of the CommonSenseQA task for the BERT-base-uncased model.

Primary Limitations

- **Data Quality Issues**: The Enron Spam dataset contained a significant number of malformed examples, which likely misled the active learning algorithms into selecting less informative samples.
- **Model Capacity Constraints**: The BERT-base-uncased model may lack the necessary capacity to handle the complexity of the CommonSenseQA dataset, impacting the effectiveness of active learning.

Avenues for Future Work

- **Data Preprocessing Improvements**: Implementing more robust data cleaning and preprocessing techniques to handle malformed examples could improve the efficacy of active learning methods.
- **Exploration of Advanced Models**: Testing active learning techniques with more powerful models, such as BERT-large or other state-of-the-art NLP models, might yield better results on challenging datasets like CommonSenseQA.
- **Broader Dataset Evaluation**: Extending the evaluation to include a wider variety of datasets could provide a more comprehensive understanding of the conditions under which active learning techniques are most effective.

Overall, our research contributes valuable insights into the practical application of active learning for NLP tasks. While we demonstrated the benefits of active learning on the SMS Spam Collection dataset, the mixed results on other datasets underscore the importance of considering dataset characteristics and model capabilities when employing active learning strategies.

8 Ethics Statement

As with training or fine-tuning any large language model, there is the risk that the model will learn and express the same bias in the data. As we reduce the number of data points, there is a further risk that it can maximize performance by over-sampling from one group thus exaggerating the bias. And if we apply this method to a real system, we won't have the model trained on full data to compare against so we won't know for sure if we missed something important in the data.

As a mitigation strategy, we could combine informativeness and representativeness-based subsetting. If we cluster the data and make sure we are pulling examples from across the space we can reduce the bias. This can be combined with the informativeness measures using the density-weighted methods:

$$x_{\text{chosen}} = \text{argmax}_x \phi_A(x) \times (\frac{1}{U} \sum_{u=1}^{U} \text{sim}(x, x^{(u)}))^\beta$$

Where ϕ represents the informativeness score and the second part of the product is a score representing how similar it is to the data we have yet to label (in set U).

Another risk involves algorithmic transparency, as active learning methods can be complex and opaque. For example, for the least certainty method, the subset of the data the model deems to be "least certain about" depends on the output from a model trained on random examples before. For the representativeness method, different values of K, as well as the initial random state of the k-means algorithm, can impact the data selection. We do not know what shared properties in the data are forming these clusters. Additionally, the iterative nature of active learning, where the model continuously updates and reselects data, further complicates transparency, as it becomes hard to trace the exact reasons behind a model's selection at each step. To mitigate this, we can generate transparency reports that summarize the active learning process, including the selection strategies used, data points chosen, model performance changes, and any identified biases in class distribution after labeling. This will help stakeholders understand and trust the model's decision-making process, ensuring greater accountability and reliability in the application of active learning methods.

References

[1] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. 1994.

[2] Gal et al. Deep bayesian active learning with image data. 2017.

[3] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. 2017.

[4] Burr Settles. Active learning survey. In *Computer Sciences Technical Report - University of Wisconsin, Madison*, 2010.

[5] Ido Dagan Shlomo Argamon-Engelson. Committee-based sample selection for probabilistic classifiers. In *Journal of Artificial Intelligence Research 11*, 1999.

[6] Commonsenseqa. `https://www.tau-nlp.sites.tau.ac.il/commonsenseqa`.

[7] Enron spam. `https://huggingface.co/datasets/SetFit/enron_spam`.

[8] Sms spam collection. `https://archive.ics.uci.edu/dataset/228/sms+spam+collection`.