Model-Agnostic Dataset Pruning for Sentiment Analysis Fine-Tuning

Han Goh
University of California, Los Angeles
andygoh5@ucla.edu

Abstract

As recent natural language datasets have continuously increased in size, fine-tuning large language models (LLMs) for downstream tasks requires extra computation costs. Recent works in LLM pre-training and fine-tuning have shown that dataset pruning is effective in reducing the amount of computation needed to train the model while retaining similar final performance. Pruning unimportant examples in dataset pruning is frequently based on model-dependent metrics, such as perplexity and Error L2-Norm score, which typically require warm-up epochs and dynamic updating of the scores throughout training. To avoid these computational overheads, I devised and tested three model-agnostic metrics for sentiment analysis tasks: the length of the sentence, number of clusters in a word embedding space, and the mean distance between words in a word embedding space. All metrics performed better than random pruning when keeping the examples with the lower scores, which corresponds to easy examples. The mean distance metric achieved the best performance, retaining 91% of the full accuracy while training on only 30% of the data. The code can be found at https://github.com/andygoh5/Dataset-Pruning-Sentiment-Analysis.

1 Introduction

Recent trends and progress in LLM training has seen an explosion in the size of datasets. Pre-training and fine-tuning LLMs require large amounts of data, which directly corresponds the amount of computational resources to train the model. And although training on larger datasets has been one of the major foundations of LLM development, more is not necessarily better. Specifically, previous research has shown that not all training samples are created equal: not every training sample contributes equally to significant training of the model, and can be ignored (Katharopoulos and Fleuret, 2018).

This leads to the practice of dataset pruning, which aims to find those non-important samples in the dataset and remove them. The goal is to find a subset of the full dataset that, when trained on the subset, preserves the performance of the model compared to the full dataset. Dataset pruning has been successful in both the computer vision domain, where a significant portion of the training set could be pruned without big losses in performance (Sorscher et al., 2022; Marion et al., 2023).

Dataset pruning is centered around first scoring and ranking every training example by a pruning metric, and then pruning the subset according to the ranking. A pruning metric should measure how difficult or informative a particular sample is. Some notable pruning metrics in the natural language domain include perplexity (Marion et al., 2023) for pre-training examples, which measures how probable a sentence is based on the language model. Another pruning metric is the Error L2-Norm (EL2N) Score, which is the norm of error vector of the training example(Paul et al., 2021). EL2N has been effective as a pruning metric for NLP classification fine-tuning datasets (Attendu and Corbeil, 2023).

However, perplexity and EL2N scores are model-dependent metrics: they require the model in order to be calculated. Specifically, perplexity is calculated by:

$$\exp\left\{-\frac{1}{N}\sum_{i=1}^{N}\log_2(P(w_i|w_1,...,w_{i-1})\right\} \quad (1)$$

where $P(w_i|...)$, the probability of a word given previous words, is dependent on the model. In addition, the EL2N score is calculated by:

$$\|p(\theta_t, x) - y\|_2 \quad (2)$$

where $p(\theta_x, x)$, the output of x when inputted through a model parameterized by θ_t, is of

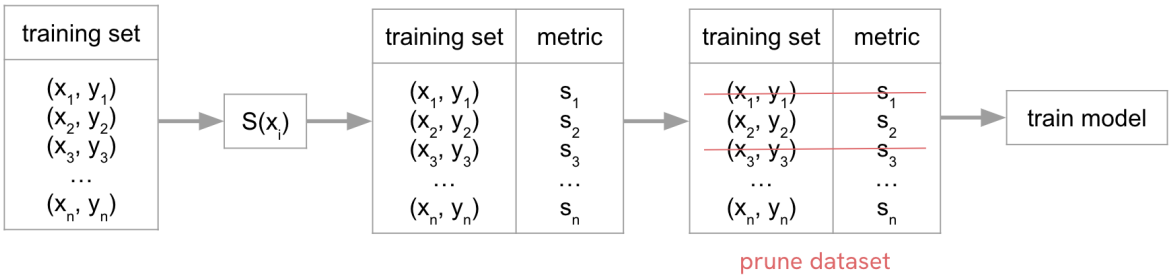

Figure 1: Illustration of dataset pruning pipeline. The function $S(x_i)$ calculates the pruning metric score for each training example x_i.

course model-dependent. Although these model-dependent metrics give accurate orderings of example difficulty, they require additional computational overhead through *warm-up epochs* and *dynamic updating*. In particular, the model must be trained for a few warm-up epochs before the metric is calculated. And since the model changes throughout training, the metrics also change. This requires the dynamic updating of the metrics, where they are re-calculated every few epochs to re-prune the training set (Attendu and Corbeil, 2023).

To that end, I propose the use of *model-agnostic* pruning metrics, i.e., pruning metrics that do not depend on the model for computation, bypassing the need for warm-up epochs and dynamic updating. I devised and tested three such metrics specifically for *sentiment analysis* tasks: (1) Length of the sentence, or the number of words in the sentence, (2) number of clusters formed in the sentence, when embedded into a word-embedding space such as GloVe, and (3) mean distance between the word of the sentence, when embedded into a word-embedding.

The findings of this paper are as follows:

1. All three proposed metrics perform better than random pruning.
2. Using mean distance as the pruning metric yielded the best results, retaining 91% of the full accuracy while pruning away 70% of the training set.
3. Pruning to keep the easy or hard examples drastically changes the final performance of the model.

2 Methods

Given the original training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, we aim to find a subset $\mathcal{D}_p \subset \mathcal{D}$ such that the performance of the model trained on $\mathcal{D}_p$ is similar to that trained of $\mathcal{D}$, where p represents the pruning technique. The pruning technique p is determined not only by the pruning metric function $S(x)$, but also by which percentile and how much of $\mathcal{D}$ is kept. $\mathcal{D}_p$ can be formed by pruning to keep the examples with the lowest scores (bottom percentile), or the ones with the highest scores (top percentile). Then, we train the model on the pruned dataset $\mathcal{D}_p$, and compare the performance with the model trained on the full dataset $\mathcal{D}$. Figure 1 illustrates the overall process.

2.1 Pruning Metrics

This section describes in detail the three proposed model-agnostic pruning metrics.

2.1.1 Sentence Length

The sentence length is calculated by the number of words in the sentence, where a word is defined as a sequence of characters separated from others via a space character.

$$SL(x) = \text{number of words in } x \tag{3}$$

2.1.2 Number of Clusters

The number of clusters is calculated by first embedding each word w_i in sentence x into a word-embedding space such as GloVe, resulting in a set of word vectors $\{v_i\}_{i=1}^{m}$. Then, DBSCAN is run on this set of word vectors to calculate the number of clusters.

$$NC(x) = DBSCAN(Embed(x)) \tag{4}$$

where $Embed(x) = \{v_i\}_{i=1}^{m}$.

2.1.3 Mean Distance

Similar to the number of clusters, the mean distance also requires embedding each word into a word-

Experimental Axes	Choices
Pruning Metric	Sentence Length, Number of Clusters, Mean Distance
Pruning Percentage	30%, 50%, 70%, 90%
Pruning Subset	Keep Lowest Score, Keep Highest Scores

Table 1: Overview of different pruning techniques tested in this paper.

embedding space. Then, we take the average of all the pairwise Euclidean distances of the v_i.

$$MD(\{v_1, ..., v_m\}) = \frac{1}{\binom{n}{2}} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \|v_i - v_j\|_2 \tag{5}$$

3 Experiments

3.1 Model

I use the DistilBERT model as the base model to fine-tune and evaluate, which is 40% smaller than the original BERT model while retaining 97% of its performance (Sanh et al., 2019). I chose DistilBERT for its capabilities on text classification tasks, while being a more light-weight model.

3.2 Data

I use a random sample of 1000 training examples and 1000 testing examples from the IMDb Large Movie Reviews Dataset from Stanford (Maas et al., 2011). This dataset includes polarizing movie reviews for binary sentiment analysis tasks.

3.3 Experiment Setup

In the experiments, I compare the models trained on datasets pruned via the three pruning metrics: sentence length, number of clusters, and mean distance. I test the performances when pruning away 30%, 50%, 70%, and 90% of the data, also while changing which percentile of the data to keep. Table 1 summarizes the different experimental axes. This experimental design and summarization is inspired by Marion et al. (2023)

3.3.1 Baselines

The first baseline is the model trained on the full training set, which allows us to see how much of the original performance is retained on the models trained on pruned data. In addition, we train a model trained on *randomly pruned* data for each of the pruning percentages. This allows us to compare if the pruning metrics perform better than just choosing random examples to prune.

3.3.2 Evaluation Metric

The performance of the model is evaluated using *test accuracy*, the number of correctly classified examples in the test set. As the chosen dataset is balanced, accuracy serves as an appropriate evaluation metric.

3.3.3 Word Embedding

I used pre-trained GloVe (Global Vectors for Word Representation) embeddings to as the $Embed()$ function used to calculate the number of clusters and the mean distance (Pennington et al., 2014). Specifically, I used the smallest GloVe embeddings with 6 billion tokens with 50 dimensions for ease of computation.

3.4 Results

For ease of comparison, I split the comparison of results first by the pruning subset:

3.4.1 Keeping Lowest Scores

Figure 2 shows the final test accuracies when keeping the examples with the lowest scores when pruning, which corresponds to keeping easier examples. All three pruning metrics perform better than random pruning, and shows the most effectiveness at 70% pruning [1].

Using the mean distance pruning metric can retain 91% of the full accuracy while pruning away 70% of the training set. See Table 2 in Appendix A for all of the accuracies.

3.4.2 Keeping Highest Scores

Figure 3 shows the final test accuracies when keeping the examples with the highest scores when pruning, which corresponds to keeping difficult examples. Unlike pruning to keep the lowest scores, all three pruning metrics perform worse or around equal to random pruning. See Table 3 in Appendix A for all of the accuracies.

[1] We ignore the effects at 30% and 50% pruning, as random pruning does as well as the full model– this suggests that the model is already saturated from just 50% of the data. We focus on the effects of the pruning metrics at 70% pruning instead, where the performance of random pruning starts to fall drastically.

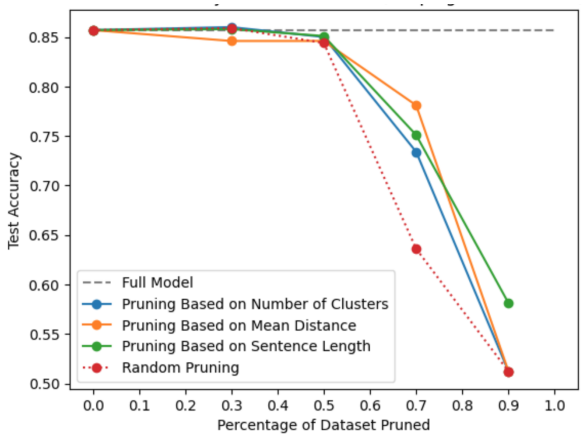

Figure 2: Final Test Accuracies of Pruned Models (Keeping Lowest Scores)

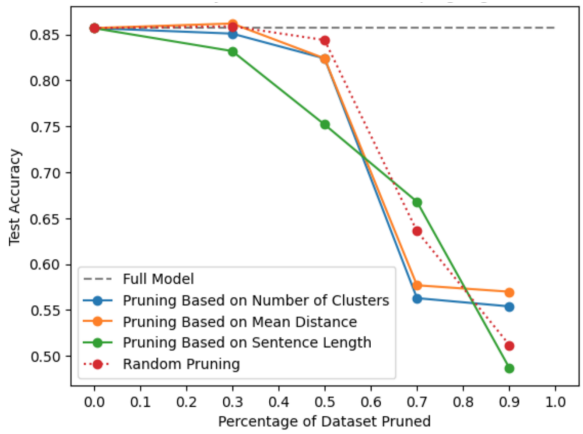

Figure 3: Final Test Accuracies of Pruned Models (Keeping Highest Scores)

3.4.3 Analysis: Why Keep Lowest Scores?

Pruning to keep the examples with the lowest scores gave drastically better performance than those with the highest scores. Keeping the lowest-scored examples is equivalent to keeping the easiest examples in the training set: the sentences with the fewest number of words, lowest number of clusters, or lowest mean distance. Sorscher et al. (2022) explored when it is better to keep easy or hard examples on a ResNet18 trained on CIFAR-10, and found that keeping easy examples is better when the size of the training set is roughly less than $N = 10000$. This is because the model fails to learn the patterns in the data well on a limited number of *difficult* examples only. On the other hand, keeping difficult examples is better when the size of the dataset is large, as to avoid training on the easy examples over and over again.

As the models in this experiment were trained on a full training set of $N = 1000 < 10000$ examples, the results align with the findings of Sorscher et al. (2022). That is, keeping the easier examples (lower scores) yields better final performance than keeping difficult examples (higher scores).

4 Conclusion

In summary, this paper investigates the effectiveness of three model-agnostic pruning metrics, which removes the need for warm-up epochs and dynamic updating of the metrics. All three proposed metrics – sentence length, number of clusters, and mean distance – perform better than random pruning when keeping the easier examples, but worse when keeping the harder examples. This behavior is expected given the small size of the training set used in the experiments. The best performing pruning technique was 70% pruning using mean distance to keep the easy examples, which retained 91% of the original accuracy.

Some future directions include running the same experiments on a larger dataset, and see if keeping the harder examples will indeed give better results according to Sorscher et al. (2022). It also verifies if the three proposed pruning metrics scale to larger datasets. In addition, running the experiments on a different model (other than DistilBERT) will allow us to compare the results across models as well.

References

Jean-Michel Attendu and Jean-Philippe Corbeil. 2023. Nlu on data diets: Dynamic data subset selection for nlp classification tasks. *arXiv preprint arXiv:2306.03208*.

Angelos Katharopoulos and François Fleuret. 2018. Not all samples are created equal: Deep learning with importance sampling. *CoRR*, abs/1803.00942.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. 2023. When less is more: Investigating data pruning for pretraining llms at scale. *arXiv preprint arXiv:2309.04564*.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep learning on a data diet: Finding important examples early in training. *CoRR*, abs/2107.07075.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536.

A Test Accuracies

	Pruning Percentage	Accuracy	Retained Accuracy
No Pruning	0%	0.857	100%
Random	30%	0.859	100.2%
	50%	0.844	98.5%
	70%	0.636	74.2%
	90%	0.512	59.7%
Sentence Length	30%	0.859	100.2%
	50%	0.845	98.6%
	70%	0.761	88.8%
	90%	0.582	67.9%
Number of Clusters	30%	0.860	100.4%
	50%	0.850	99.2%
	70%	0.734	85.6%
	90%	0.512	59.7%
Mean Distance	30%	0.846	98.7%
	50%	0.846	98.7%
	70%	0.781	91.1%
	90%	0.512	59.7%

Table 2: Final test accuracies of pruned models when keeping the lowest-scored examples. Retained accuracy compares the accuracy of the pruned model with the accuracy of the full (no-pruning) model.

	Pruning Percentage	Accuracy	Retained Accuracy
No Pruning	0%	0.857	100%
Random	30%	0.859	100.2%
	50%	0.844	98.5%
	70%	0.636	74.2%
	90%	0.512	59.7%
Sentence Length	30%	0.832	97.1%
	50%	0.752	87.7%
	70%	0.668	77.9%
	90%	0.487	56.8%
Number of Clusters	30%	0.851	99.3%
	50%	0.824	96.1%
	70%	0.563	65.7%
	90%	0.554	64.6%
Mean Distance	30%	0.862	100.6%
	50%	0.824	96.1%
	70%	0.577	67.3%
	90%	0.570	66.5%

Table 3: Final test accuracies of pruned models when keeping the highest-scored examples. Retained accuracy compares the accuracy of the pruned model with the accuracy of the full (no-pruning) model.