Predicting Fine-tuned Performance on Larger Datasets Before Creating Them

Toshiki Kuramoto
Bridgestone Corporation / Tohoku University
kuramoto.toshiki.q1@dc.tohoku.ac.jp

Jun Suzuki
Tohoku University
jun.suzuki@tohoku.ac.jp

Abstract

This paper proposes a method to estimate the performance of pretrained models fine-tuned with a larger dataset from the result with a smaller dataset. Specifically, we demonstrate that when a pretrained model is fine-tuned, its classification performance increases at the same overall rate, regardless of the original dataset size, as the number of epochs increases. Subsequently, we verify that an approximate formula based on this trend can be used to predict the performance when the model is trained with ten times or more training data, even when the initial training dataset is limited. Our results show that this approach can help resource-limited companies develop machine-learning models.

1 Introduction

In recent years, the development of pretrained models (PMs) for natural language processing (NLP) has been growing rapidly, with the widespread availability of Transformers (Vaswani et al., 2017), a representative example. Notably, Transformers framework (Wolf et al., 2020), provided by Hugging Face[1], is capable of advanced analysis without specialized knowledge.

However, when we attempt to fine-tune such a PM for use in a business context, we are likely to face "dataset size issues", such as data size limitations and a lack of clarity in the number of datasets required for expected performance. Moreover, fine-tuning a PM with the small amount of data initially available to most businesses does not always result in ideal performance. This raises another issue: "Fine-tuning with available data did not achieve ideal performance, good, so how much data would be enough?" When the fine-tuning results are based on only a few hundred units of data, this question is a difficult one to answer. One recent study, which reviewed the performances of the latest PMs (Min et al., 2021), noted that the quantification of the required labeled data is another significant challenge. Meanwhile, Rosenfeld et al. (2020) investigated the relationship between model size and dataset size and proposed certain formulas to predict generalization errors in language models. However, similar considerations have not been made in the context of fine-tuning PMs. Solving the "dataset size issues" would be a significant contribution in this era of widespread PM use. Furthermore, data collection and annotation are time-consuming and costly processes; therefore, knowing the amount of data required to achieve specific performance goals can save time and money. Therefore, the primary objective of this study was to develop a means of determining future guidance for situations in which data are limited. More specifically, the objective was to develop a method that predicts the performance achievable when fine-tuning PMs with a large dataset using a limited dataset.

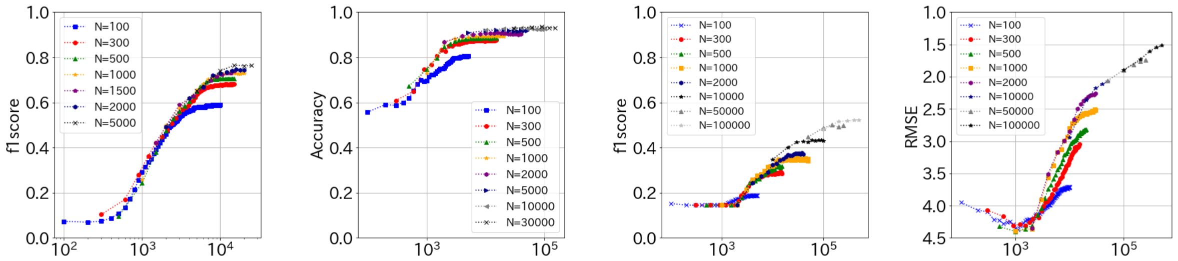
2 Related Work

Kaplan et al. (2020) explored scaling laws in Large Language Models (LLM) and demonstrated that performance extends exponentially based on three factors: model size, dataset size, and the amount of computation. These scaling laws have been observed not only in NLP but also in other fields (Henighan et al., 2020).

These facts suggest that model performance will increase indefinitely if these factors continue to be raised. In fact, performance improvement is widely pursued by scaling up to compete for the number of parameters. For example, BERT (Devlin et al., 2019), a pioneer in this field, has around 300 million parameters. Subsequent GPT series have continued to expand, with some reaching 175 billion parameters (Radford et al., 2019; Brown et al., 2020). Google's LLM, PaLM, is reported

[1] https://huggingface.co

Proceedings of the 31st International Conference on Computational Linguistics: Industry Track, pages 204–212
January 19–24, 2025. ©2025 Association for Computational Linguistics



(a) LodgeRev

(b) MovRev

(c) ACRev

(d) DrugRev

Figure 1: Classification performance trend by datasets

to have been trained with 540 billion parameters and has shown high performance in a variety of tasks (Chowdhery et al., 2023). The subsequent PaLM2 is reported to be even more capable, although the number of parameters has not been disclosed (Anil et al., 2023). Likewise, Open AI has demonstrated high performance with GPT-4, although the number of parameters has not been disclosed either (Achiam et al., 2023). Based on the principle of scaling up, various studies have investigated how to efficiently train LLMs (Devlin et al., 2019; Aßenmacher et al., 2021). The continued development of LLMs at such scales can contribute significantly to the development of this field; however, this is only possible for a few resource-rich organizations and groups. In fact, there are several limitations for developing or using an LLM in proportion to the size of the available parameters, such as machine specifications. Hence, it is becoming crucial to find how to handle LLMs efficiently and achieve LLM-like performance with PMs with small parameters. Several studies have already addressed these topics (Schick and Schütze, 2021; Ouyang et al., 2022; Pfeiffer et al., 2020). Additionally, modern LLMs often do not make their internal mechanisms available, limiting user customization. In this respect, PMs, which are relatively lightweight and whose internal mechanisms are publicly available, present notable advantages. This study therefore aims to contribute to these efforts, examining ways to efficiently use PMs to solve the challenges mentioned above, given the limitations of a relatively small data size.

3 Task Definition

This study aimed to predict the performance that can be achieved when fine-tuning a PM with a larger dataset in a situation in which no such dataset is available.

Suppose we have a small fine-tuning dataset D_S. Moreover, suppose we plan to create a larger fine-tuning dataset D_L, which always includes D_S. Then, our task is to construct a function $f(\cdot)$ that returns the value of the predefined performance metric Y, such as the classification accuracy of the target task, given D_L, from the information in D_S (before actually creating D_L), namely,

$$Y = f_{D_S}(D_L). \tag{1}$$

This function would be highly beneficial for developing real-world systems. For example, it would allow users to estimate how much fine-tuning data would be needed to achieve the desired performance or decide whether they should reconsider building a new system before creating expensive fine-tuning data.

4 Preliminary Experiment

First, we investigated whether a particular correlation exists among the performances obtained from various sizes of fine-tuning datasets.

4.1 Data Set

This study addressed the classification task of review comments. Such a task is likely to be required in a company to develop a new product or improve service. In this study, datasets on review comments in different languages, categories and subjects were selected to provide a broad test set. We prepared four different datasets using online reviews. These included lodging reviews (LodgeRev) (Kanouchi et al., 2020), Amazon customer reviews[2] (ACRev), movie reviews (MovRev) (Maas et al., 2011), and reviews of pharmaceuticals[3] (DrugRev) (Gräßer et al., 2018).

[2] https://s3.amazonaws.com/amazon-reviews-pds/readme.html

[3] https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+/28Drugs.com/29

Datasets	Review	Language	# of	Available	Divided	Predict target
name	Contents		labels	data size	data size	data size
LodgeRev	Lodging / Accommodation	Japanese	9	500	100,300	1K,1.5K,2K,5K
MovRev	Movie	English	2	1,000	100,300,500	2K,5K,10K,30K
ACRev	Electric appliances	Japanese	5	1,000	100,300,500	2K,10K,50K,100K
DrugRev	Medical products	English	10	1,000	100,300,500	2K,10K,50K,100K

Table 1: Experimental condition (K : Thousand)
In the case of the dropping out condition, the divided data size of 100 is not used to calculate the formula.

LodgeRev consists of lodging reviews in Japanese, with nine labels. These reviews were derived from the evidence-based explanation dataset provided by Recruit Co., Ltd., (Kanouchi et al., 2020). However, as the purpose of this study was different from the purposes of the original study, the data were only partially processed. Specifically, only the review comments were taken out, and appropriate labels were assigned to them. There were nine categories of labels: meals, buildings and equipment, customer service, tourism and recreation, fares, baths, access, revisit, and others. The annotation process was conducted by two trained workers. ACRev was a set of Japanese reviews of electric appliances, categorized based on Amazon's five-star ranking system (1–5). MovRev consisted of movie reviews in English, which were assigned either a positive or negative sentiment class for each review. Originally, this consisted of 25,000 reviews each for the train and test datasets; in this study, however, they were combined . However, the 50:50 ratio of positive/negative reviews was not changed. DrugRev was a set of reviews of medicinal products in English, with ten ranks (1–10).

We randomly sampled review texts from these datasets and then created eight different sizes of fine-tuning and evaluation data for each dataset.

4.2 Method

We selected the BERT model as the pretrained model for the fine-tuning experiments. Two experiments were planned: one each for the Japanese- and English-language datasets. The model used for the Japanese datasets was cl-tohoku/bert-base-japanese-whole-word-masking[4] and that for the English datasets was bert-base-uncased[5]. The hyperparameters for fine-tuning in both experiments were consistently set as follows: Token size = 128, Batch size = 32, and Learning rate = 2e-5.

For the analysis, we continued to update epochs until no further improvements in performance appeared. Accuracy, F1 Score, and root mean squared error (RMSE) were adopted as evaluation metrics. The closer the accuracy and F1 score were to 1 and the closer the RMSE was to 0, the better the performance. In this study, "learning amount" was used as a measure of the scale of training when fine-tuning a model. This value was calculated as a multiplier of the dataset size and the number of epochs. For example, if the dataset size was 100 and a model was fine-tuned by 10 epochs, the learning amount was 1,000 (100 examples $\times$ 10 epochs) Similarly, if the dataset size was 500 and a model was fine-tuned by 2 epochs, the amount of learning was 1,000 (500 examples $\times$ 2). Both examples theoretically indicate a model that has been fine-tuned with a dataset size of 1,000. In other words, the learning amount was defined as the total amount of data used to train the model.

4.3 Results

Figure 1 shows the learning curve of the classification performance obtained by fine-tuning the same pretrained model with each prepared data size. The vertical axis shows the score of the evaluation metric, that is, accuracy, F1 value, or RMSE. Meanwhile, the horizontal axis shows the learning amount on a logarithmic scale. Each plot indicates the **average score of five runs**, varying the random seeds given the fact that performance can vary significantly depending on seeds during fine-tuning (Dodge et al., 2020).

An increase in the learning amount resulted in better scores. Interestingly, the slopes did not depend on the size of the fine-tuning datasets; rather, they advanced in a similar manner, particularly when taken on a logarithmic scale and were close to linear. In contrast, where performance saturates look proportional to the fine-tuning data size. These phenomena were observed regardless of metrics, tasks, and PMs, at least within these preliminary experiments.

In summary, the following noteworthy findings

[4] https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

[5] https://huggingface.co/bert-base-uncased

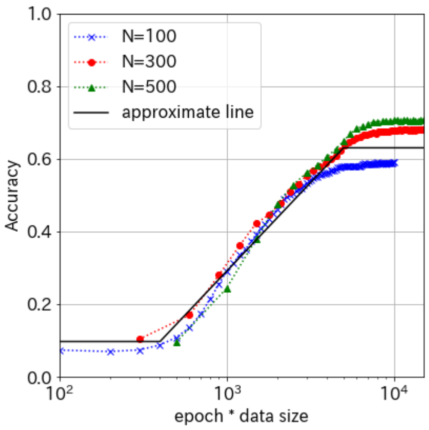


Figure 2: Approximate formula for performance growth Approximate line in the figure is the pseudo (negative) ramp loss function calculated based on Eq. (2), and this slope is used for prediction..

were made based on the preliminary experiment: (1)When taking the log scale, it was observed that performance improved at a constant rate as the learning amount increased. This had a similar trend in slope regardless of the dataset size. (2)The timing at which performance saturated depended on the dataset size. These findings are used to attempt to predict performance in the following experiment.

5 Proposed Method

Based on the findings of the preliminary experiments, this section proposes a method to predict the evaluation score when the number of fine-tuning data is increased using evaluation scores obtained from much smaller sets of fine-tuning data.

More specifically, the proposed method attempts to predict the accuracy from the fine-tuning of 100,000 units of data (D_L) from a set of accuracies obtained from the fine-tuning of less than 1,000 units of data (D_S). For this purpose, we assume that each line in Figure 1, namely, the increase in performance against the learning amount increase, can be approximated by a simple (negative) ramp loss function (Collobert et al., 2006), $f(x)$, which can be written as follows:

$$f(x) = \begin{cases} w_0 + w_1 \log_{10} t_{min} & \text{if } x < t_{min} \\ w_0 + w_1 \log_{10} t_{max} & \text{if } x > t_{max} \\ w_0 + w_1 \log_{10} x & \text{otherwise} \end{cases} \quad (2)$$

where t_{min} and t_{max} represent the start and end points of performance growth in the learning amount. Based on Eq. (2), the learning amounts of t_{min} and t_{max} are calculated where the RMSE with the actual value of the fine-tuned model is minimum. This slope w_1 is used to predict performance improvement. Figure 2 shows an example of applying the ramp loss function to the LodgeRev result.

Moreover, from the results of the preliminary experiment, it can be inferred that the saturation point varies depending on the size of the data used. As the Eq. (2) is a linear equation, it can be interpreted that performance will improve as the learning amount increases. In reality, however, performance should saturate at some point, where the PM should reach the limit of its learning. Therefore, an equation for predicting the learning amount at saturation is proposed below. First, the maximum (or minimum, in the case of RMSE) score is extracted for each small dataset. The maximum value is fitted to Eq. (2) and the learning amount t_{max} for each smaller dataset is calculated backward. This is the estimated learning amount at saturation for each small dataset. Additional linear regression equations are calculated with the estimated learning amount as the objective variable and each smaller dataset size as the explanatory variable. This regression equation is then used to calculate the amount of learning amount at saturation for an arbitrary dataset size. The saturated learning amount depending on dataset size can thus be calculated as follows:

$$t_{max}(D) = \theta_0 + \theta_1 \log_{10} D, \quad (3)$$

where θ_0 and θ_1 are assumed to be estimated from a set of performances on smaller datasets. Finally, the proposed method estimates performance after fine-tuning with data size D by calculating $f(t_{max}(D))$ in Eq. (2) with the condition $t_{max} = t_{max}(D)$.

The proposed method is shown in Figure 3. The maximum possible of performance can therefore be predicted from limited data by combining Eq. (2), which predicts the improvement in performance according to the learning amount, and Eq. (3), which estimates the learning amount at saturation.

6 Experiment

We conducted an experiment to verify the effectiveness of the proposed methodology.

6.1 Experimental Conditions

Using the same data as the preliminary experiment, we experimented with the proposed method

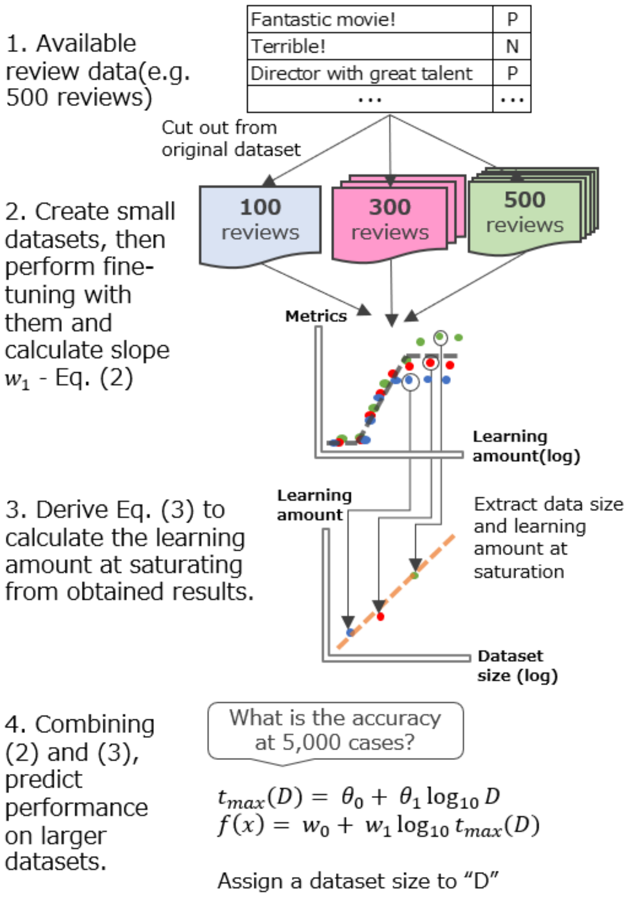

1. Available review data(e.g. 500 reviews)
Fantastic movie! | P
Terrible! | N
Director with great talent | P
...
Cut out from original dataset
2. Create small datasets, then perform fine-tuning with them and calculate slope w_1 - Eq. (2)
100 reviews
300 reviews
500 reviews
Metrics
Learning amount(log)
3. Derive Eq. (3) to calculate the learning amount at saturating from obtained results.
Learning amount
Extract data size and learning amount at saturation
Dataset size (log)
4. Combining (2) and (3), predict performance on larger datasets.
What is the accuracy at 5,000 cases?
$t_{max}(D) = \theta_0 + \theta_1 \log_{10} D$
$f(x) = w_0 + w_1 \log_{10} t_{max}(D)$
Assign a dataset size to "D"

Figure 3: Proposed method and application

to predict future performance, assuming a situation in which the amount of available data is limited, specifically, where approximately 1,000 samples of data are available at most. The specific data size conditions are shown in Table 1.

The predicted performances were evaluated by comparing them with the actual performances when fine-tuning with 2 $\sim$ 100 times more data sizes. The values given here refer to the best possible scores estimated when fine-tuned with each dataset.

As a further validation, a condition was added in which some data were not used. As seen in Figure 1, the learning curve in a particular dataset size does not overlap well with the ones in other datasets. This is the case for the 100-sample datasets for MovRev or ACRev. Generally, it can be assumed that prediction performance can be improved by excluding indicators that show outlier values. Therefore, based on this assumption, cases in which some data were dropped were also added to the validation conditions. Specifically, this proposed method was applied while excluding the results of the 100-sample dataset in which performance did not improve after fine-tuning. Other experimental conditions followed the preliminary experiment.

(a) LodgeRev

Metrics	D	Actual	Predict	diff	Predict (w/drop)	diff (w/drop)
F1	1K	0.731	0.744	+.013	0.735	+.004†
	1.5K	0.738	0.763	+.025	0.749	+.011†
	2K	0.746	0.775	+.028	0.758	+.012†
	5K	0.765	0.809	+.044	0.784	+.020†
Acc	1K	0.779	0.789	+.011	0.775	−.004†
	1.5K	0.788	0.804	+.016	0.783	−.005†
	2K	0.793	0.814	+.021	0.789	−.004†
	5K	0.814	0.842	+.028	0.806	−.008†

(b) MovRev

Metrics	D	Actual	Predict	diff	Predict (w/drop)	diff (w/drop)
F1	2K	0.908	0.916	+.007	0.903	−.005†
	5K	0.920	0.933	+.013	0.915	−.006†
	10K	0.931	0.944	+.014	0.922	−.008†
	30K	0.935	0.960	+.024	0.934	−.002†
Acc	2K	0.909	0.914	+.005	0.903	−.005†
	5K	0.921	0.930	+.009	0.914	−.006†
	10K	0.931	0.940	+.010	0.922	−.009†
	30K	0.936	0.954	+.019	0.933	−.003†

(c) ACRev

Metrics	D	Actual	Predict	diff	Predict (w/drop)	diff (w/drop)
F1	2K	0.375	0.370	−.006	0.375	−.001†
	10K	0.434	0.402	−.032	0.407	−.027†
	50K	0.500	0.425	−.075	0.429	−.071†
	100K	0.523	0.433	−.090	0.436	−.087†
RMSE	2K	1.007	1.012	+.004	1.012	+.004†
	10K	0.938	0.943	+.006†	0.953	+.015
	50K	0.880	0.895	+.015†	0.913	+.033
	100K	0.846	0.877	+.032†	0.899	+.053

(d) DrugRev

Metrics	D	Actual	Predict	diff	Predict (w/drop)	diff (w/drop)
Acc	2K	0.423	0.411	−.012	0.421	−.002†
	10K	0.453	0.424	−.029	0.438	−.015†
	50K	0.495	0.433	−.062	0.449	−.046†
	100K	0.538	0.436	−.102	0.452	−.086†
RMSE	2K	2.262	2.368	+.106	2.322	+.059†
	10K	2.074	2.061	−.014†	2.010	−.065
	50K	1.742	1.837	+.095	1.794	+.052†
	100K	1.518	1.756	+.238	1.718	+.200†

Table 2: comparison between prediction and actual result (D:dataset size, K:Thousand)
Daggers show the closer of the two prediction conditions to the actual measurements.

6.2 Results

Table2 shows the predicted performances obtained from the proposed method and the actual results achieved when using the review datasets. For

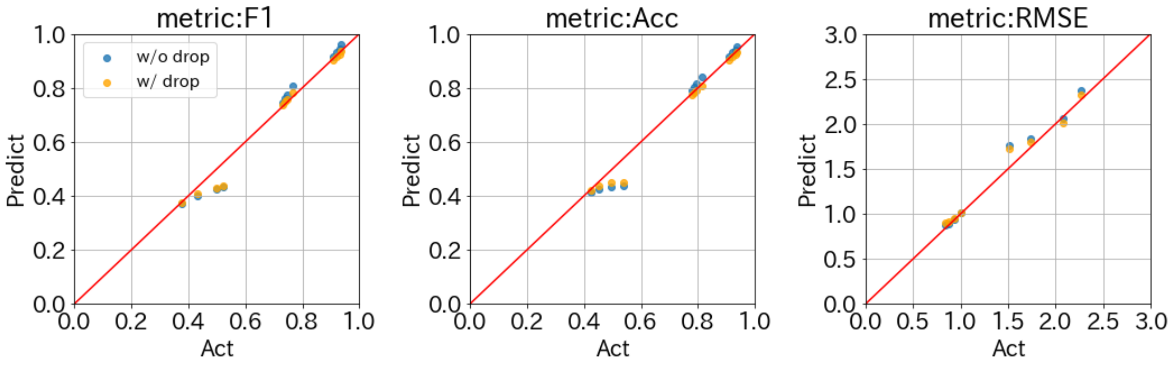


Figure 4: Prediction results comparison by metrics

LodgeRev, the predictions were $0.01 \sim 0.045$ higher than the actual results in F1 value and accuracy . Meanwhile, for ACRev and DrugRev, the predictions were lower than the actual results, especially for the datasets of 50K and 100K samples, where the predictions differed widely. Overall, the larger the hypothetical dataset size (and thus the farther away from the available dataset size), the lower the prediction accuracy. Although the RMSE predictions were generally larger than the actual performance, the variability in predictions and actual differences did not necessarily increase proportionally to the dataset size.

Next, we examined the effect of data exclusion based on the results for each experimental condition. In this experiment, the prediction results for most conditions were slightly better when the data from the 100-sample datasets were excluded. As this method fit a linear regression model based on the observed data, excluding possible outliers may improve the fit.

Figure 4 shows the predicted performances obtained from the proposed method and the actual results by metrics. Each dot plots the actual performance value when fine-tuning the PM with data from 1K- to 100K-sample datasets and the predicted value from this method based on the limited available data. The closer each dot is to the line, the higher the prediction accuracy. As can be seen from this figure, the prediction achieved a good approximation of the actual results.

To better understand the difference between the predicted and actual results, RMSEs were calculated for each metric between them. The RMSE of RMSE prediction may be somewhat confusing , but these were calculated to verify the discrepancies when each metric is considered as a mere numerical indicator. The results are presented in Table 3. Overall, the predictions showed good performance. These results demonstrate that excluding outliers results in better prediction. In some cases, the slope of the performance improvement based on the small datasets was gentler than those based on larger datasets. Therefore, excluding smaller datasets—in this case, the 100-sample dataset—would lead to a better fit for the predictive model. However, while the 100-sample dataset was dropped in this case, it may not always be sufficient to exclude the smallest dataset. Although neither condition was able to predict the results perfectly, even a simple linear regression-based method could predict the performance of fine-tuned PMs with increased dataset size.

	F1	Acc	RMSE
w/o drop	.040	.038	.099
w/ drop	.034†	.029†	.083†

Table 3: RMSE between predicted and actual values in each metric

6.3 Simulation

Finally, we verified the effectiveness of our proposed method by making some assumptions and estimating the costs of implementing it.

Let us consider a case in which a company builds a model that automatically classifies customer reviews about its products, in line with the setting of the above experiments. Model implementation requires not only a model but also data for training and testing. The task of data collection for a model can be further subdivided into data collection itself and annotations for machine learning. If there is a review site available, such as those created for restaurants, it is possible to acquire review data through methods such as crawling. However, in general, acquiring evaluation data on products is costly and time-consuming for many companies. Of course, it is also not easy to accurately define survey costs given the differences in the various types of businesses, situations, and customs across countries. However, as an example, let us consider the cost of a survey conducted by a Japanese marketing research firm. The actual name of this company has been withheld, but it is a well-known and popular research firm in Japan. The cost of a 10-question survey by this research firm is approximately US$1,700 for 500 samples and US$2,600 for 1,000 samples (converted at US$1 = JPY148.21). This excludes the cost of annotation for the survey data. The cost of annotations using crowd workers was estimated to be between US$0.13 and 0.41 per annotation. Assuming a median of US$0.27 as a standard value, annotating 1,000 samples would cost a total of US$270, and US$2,870 would be required to collect and annotate 1,000 data samples. There are further costs associated with this process, but for the sake of simplicity, we only consider the costs of collecting and annotating survey data.

Below, some simulations are performed under these cost assumptions, assuming the interested company wants to build a classification model with an 80% accuracy or F1 score and that the earlier experimental results have been obtained. For example, in the case of ACRev or DrugRev, let us assume that the company paid US$2,600 to collect 1,000 survey samples. If we want to construct a model with an 80% F1 score, we can expect not to be able to reach the performance target based on the proposed method even if 100,000 samples are available. This would save the company an unrealized cost of around US$284k that would have been incurred by collecting additional data, and allow them to proceed with other strategies.

In contrast, let us apply the same consideration to LodgeRev. In this case, if there are 5,000 data samples, it is likely that an 80% accuracy can be achieved. Subsequently, additional investments can be made only to obtain the quantity necessary (i.e., 4,000 more data samples) without incurring extra investment costs.

Thus, this method for predicting future performance and required quantities allows for optimizing data collection costs and making quicker decisions.

7 Conclusion

This study proposed a method for predicting performance improvement using a limited dataset by examining the characteristics of performance trends when fine-tuning PMs from the relationship between dataset size and learning amounts and using these characteristics to formulate predictions. We verify that it is possible to accurately predict a certain degree of performance by combining simple linear formulas. The study was limited to the classification task of NLP, but it nevertheless demonstrated that if there are about $500 \sim 1,000$ data samples, it is possible to predict future performance by taking advantage of trends in performance growth. These predictions are very useful when facing the challenge of small datasets in practice. Even with limited data, this approach can accurately predict the performance expected and the data collection needed to achieve this performance, thus allowing for rapid and cost-effective decision-making.

Limitation

This study has dealt with a very basic classification task in NLP, but it remains to be seen whether this method can be applied to other tasks as well. There is also room for various improvements to this method. For example, in this study, each seed was changed five times, and calculations were run until the performances were saturated. Even though the PM was relatively lightweight and dataset sizes were small, it still required time and appropriate machine specifications. Prior research has explored various methods for refining the fine-tuning process itself (Sun et al., 2020; Dodge et al., 2020; Mosbach et al., 2021; Aghajanyan et al., 2021). Therefore, it may be possible to utilize such methods to further improve the efficiency of learning. Future research should refine the definitions of the saturation point and various operations to further improve performance. The results also demonstrate that excluding outliers improved the model's fit, but the selection of such outliers should be contextualized. In this study, improvement was achieved by excluding the results of a 100-sample dataset, but selection methods should be considered when predicting the performance of larger datasets. While this study has intentionally focused on a simple linear regression model, there is room to improve the equation. Finally, a BERT-based PM was used for this study, but the results should be verified using other PMs.

Ethics Statement

The review datasets used in this study are widely available and do not identify individuals or violate their privacy.

Acknowledgements

This work was partly supported by JSPS KAKENHI Grant Numbers JP24H00727.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Matthias Aßenmacher, Patrick Schulze, and Christian Heumann. 2021. Benchmarking down-scaled (not so large) pre-trained language models. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*, pages 14–27, Düsseldorf, Germany. KONVENS 2021 Organizers.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou. 2006. Trading convexity for scalability. In *Proceedings of the 23rd international conference on Machine learning*, pages 201–208.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *ArXiv*, abs/2002.06305.

Felix Gräßer, Surya Kallumadi, Hagen Malberg, and Sebastian Zaunseder. 2018. Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In *Proceedings of the 2018 International Conference on Digital Health*, page 121–125, New York, NY, USA. Association for Computing Machinery.

Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. 2020. Scaling laws for autoregressive generative modeling. *CoRR*, abs/2010.14701.

Shin Kanouchi, Masato Neishi, Yuta Hayashibe, Hiroki Ouchi, and Naoaki Okazaki. 2020. You may like this hotel because ...: Identifying evidence for explainable recommendations. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 890–899, Suzhou, China. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

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.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heinz, and Dan Roth. 2021. Recent advances in natural language processing via large pre-trained language models: A survey. *ArXiv*, abs/2111.01243.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jonathan S. Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. 2020. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations*.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification? *Preprint*, arXiv:1905.05583.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.