Exploring the Relationship Between In-Context Learning and Finetuning for Data Selection

Avi Trost
Brown University / Providence, RI
avi_trost@brown.edu

Abstract

The capabilities of large language models (LLMs) rely heavily on the data they are trained on. Training on different training mixtures will cause an LLM to perform better on different tasks. In this work, I focus on the problem of data selection for finetuning in the scenario where validation data is available. I posit that data selection methods can benefit from the availability of a validation performance signal. I explore whether in-context learning can provide such a signal. Specifically, I hypothesize that examples that are the most beneficial as one-shot examples for in-context learning are also the most beneficial examples to train on, with respect to validation performance. Contrary to intuition, I find that this hypothesis does not generally hold true. I theorize an explanation for this result and detail further experiments to thoroughly investigate this behavior.

1 Introduction

Despite the vast out-of-the-box capabilities of LLMs, finetuning the model on custom data has a wide array of applications. Recent works have shown that quality of training data can trump quantity, especially when there is high variance or noise in the data (Liu and Wang, 2023). In particular, by training on a select subset of a dataset, downstream performance can match or exceed the performance of training indiscriminately on the whole dataset. Naturally, the extent of this effect depends on the variance present in the dataset. For instance, discarding the noise in a noisy dataset will generally be beneficial for downstream performance. This prompts the question: how can we optimally filter a dataset to maximize downstream performance?

As the data available on the Internet runs thin, and synthetic data regimes grow in popularity, this question becomes ever more pressing. Generating high quality and diverse synthetic data remains an open research question, so including a filtering step in synthetic data pipelines has become commonplace. Current methods typically generate synthetic datasets that are noisy, where some data points may be detrimental to train on. For example, a synthetic data-point for a question answering dataset may produce a false claim. Ensuring that synthetically produced data is of high quality is critical for obtaining the best downstream model.

This problem is challenging because there’s no natural way to computationally determine the quality of a piece of text. Furthermore, the definition of “quality” is loose and can mean different things in different contexts. This makes it difficult to create a universal notion of “quality” that a filtering mechanism could employ. For example, there have been efforts to use perplexity as such a notion, but this has had mixed results as a filtering heuristic (Li et al., 2024a).

Filtering heuristics such as perplexity, as well as many filtering works, at their core make assumptions on what a “quality” piece of text looks like. Specifically, the downstream task is not considered. In a hypothetical world with sufficient computational resources, the best subset to train on will be the subset that achieves the best validation performance. Realistically, it is infeasible to exhaustively compute validation performance for each possible training subset. Instead, we can attempt to estimate the performance of a subset. Just as useful, we can estimate the relative performance of subsets; better subsets should attain superior performance than worse subsets. I posit that a signal from the validation dataset is necessary to estimate this performance.

In this work, I hypothesize that for training examples, beneficial one-shot prompting performance translates to beneficial finetuning performance. That is, if an example is better than average as an in-context example, I expect it to also be better than average to finetune with. Here, better is defined relative to validation performance on the
target task. In this work, across five datasets in SuperGLUE (Wang et al., 2020), I test this hypothesis through one-shot prompting methods followed by finetuning on the desired subsets. I find that my hypothesis does not generally hold, motivating further work to identify why this seemingly intuitive hypothesis does not hold, given its potential benefits to synthetic data pipelines. I identify future experiments to be done in this vein.

2 Background and Related Work

Much of the related works to this work are related to data selection methods for LLMs. Data selection is generally the task of determining what data to include in a dataset for use; The survey work Albalak et al. (2024) does an excellent job of covering these methods, including but not limited to data selection for both finetuning and in-context learning. Many of the following works are covered in more detail in the survey.

Data Selection for LLM Finetuning

Heuristics are a common method of filtering finetuning data for quality or diversity. Some examples include word statistics, n-gram existence, or inclusion of keywords among other heuristics. For example, Raffel et al. (2023) filter out documents with specific NSFW words.

Other works like Wenzek et al. (2019) use perplexity as a filtering mechanism. Perplexity-based filtering entails computing the perplexity of generations with respect to the model that will be finetuned, e.g. Mistral 7B(Jiang et al., 2023). By discarding a subset of the data with high perplexity, the data that the model is less familiar with is discarded. Note that this method of quality filtering hinges on many assumptions about what high quality training data looks like.

Another common filtering mechanism is to use an LLM to judge the data and output a prediction on the text’s quality (Huang et al., 2024). LLM-based filtering entails using a separate LLM to produce a score judging how “good” it thinks an instruction/output pair is. Here, a pair is “good” if the output is a proper and well-constructed answer to the corresponding input query. For example, this process in Li et al. (2024) is facilitated using a prompt that describes a 5-point scale for the LLM to judge the examples, keeping the examples ranked highly. This is meant to filter out the examples where the output may be incorrect, as deemed by the LLM. Note that depending on the prompt used as the rubric for the LLM, the input query may not be well-formed, or it may be unrelated to the domain of interest, or it may be too easy of a question to provide a meaningful training signal. The LLM judges purely based on the prompt, and thus encodes the biases of the prompt writer as to what dictates a “good” example.

A similar flaw of human bias is inherent in perplexity based filtering, as we are assuming, without rigorous backing, that lower perplexity of training examples leads to better data to train a model with. I posit that this problem will always exist in a zero-shot setting. Without knowing the target task, any attempts at training data filtering will be more of an engineering than a science. We can mitigate the gap between filtering methods and downstream tasks based on our own heuristics, but we will never be able to optimize our data mixture towards a specific output distribution if we don’t know what that distribution is.

Data Selection for LLM In-Context Learning

This work hinges on the fact that there is variance amongst examples used for in-context learning when evaluating validation performance (Liu and Wang, 2023). Work has been done in the realm of determining which examples are best to use for in-context learning with the objective of improving performance. Lee et al. (2024) use perplexity as a proxy for how much “knowledge” a model has of examples, and finds a correlation between lower example perplexity and improved performance, to an extent. Liu et al. (2021) uses embedding cosine similarity to select in-context demonstrations, showing that it improves performance. Other work such as Lu et al. (2022) emphasize the effect of different orderings of in-context demonstrations on the model’s performance.

3 Hypothesis

For data filtering settings where the target task distribution is known and we have validation benchmark data, we should be able to do better than solely relying on zero-shot heuristics. This motivated us to consider ways that this targeted filtering can be achieved. In an environment with abundant computational resources, one would be able to finetune an LLM with every combination of training data they have available, benchmark the models, and keep the model with the highest validation performance. While this isn’t feasible, I ask if it’s possible to find a more computationally effective
method to achieve an estimate of the same task. That is, does there exist a cheap proxy for determining the ideal fine-tuning subset of the training data, catered towards the specific downstream task?

My hypothesis is that few-shot prompting performance of a training example can be a proxy for the training signal that example provides. In other words, if example $X$ consistently improves few-shot prompting performance of a model on a particular benchmark compared to other examples, we would expect example $X$ to be a particularly good example to fine-tune the model with. If this hypothesis holds, we can determine an ordering of the best training examples, with respect to the target benchmark. The key part here is that, instead of utilizing some heuristic, we are optimizing directly for higher performance on the objective we are interested in: validation performance.

If the hypothesis holds, and few-shot performance carries over to training performance, we will be able to effectively and efficiently filter fine-tuning datasets for LLMs in settings where we have access to validation data and training data could be noisy. This is a relatively common LLM fine-tuning setting, so this method has the potential of widespread use if it works.

Besides its practical benefits for LLM practitioners, this method will be useful for studying the quality of natural language data for fine-tuning LLMs in general. It would open the door to experiments that could rely on the fact that we know that a particular subset of data is ideal for a specific task.

4 Methods

To determine if beneficial in-context learning examples translate to beneficial fine-tuning examples, I first must determine, for each example in the training datasets, how beneficial of an in-context example it is. I do this for a particular language model. In my case, I pick Mistral 7B for its well-roundedness (Jiang et al., 2023). I estimate this through one-shot prompting. Specifically, I compute the average accuracy of a random subset of the validation set when using a particular example in a one-shot prompting format. Given the variation of in-context learning examples, some examples will average higher accuracy than others (Liu and Wang, 2023). My hypothesis posits that these examples will be better to train on than a random subset of the dataset.

After obtaining the average in-context accuracy for each training example, I form an ordering of these examples. I split the training examples into different subsets based on their performance. I form a subset for the best, worst, and middle performing subsets. I additionally form a randomly selected subset of the same size as a control.

I then fine-tune the same base language model, i.e. Mistral 7B, on these subsets in a standard supervised manner for causal language models. I then evaluate each of these fine-tuned language models in a zero-shot manner on their corresponding validation datasets. The accuracy of these evaluations for each subset is the metric of interest to us. If the accuracy of the “best” subset’s fine-tuned model is the highest, this is evidence to support my initial hypothesis.

5 Experiments

I perform the main experiments of my method on the five smallest subsets of SuperGLUE: cb, copa, rte, wic, and wsc (Wang et al., 2020). The base model I use is Mistral 7B. To obtain one-shot accuracies of the training dataset, I sample the performance of each example with respect to 30 random validation examples. I form subsets of size equal to 25% of the total training set size for each dataset.

5.1 Training Details

I fine-tune Mistral-7B using QLoRA due to memory constraints (Dettmers et al., 2023). The 4-bit QLoRA hyperparameters used are: rank($r$) of 64, alpha($\alpha$) of 4, dropout of 0. I train for 4 epochs using AdamW (Loshchilov and Hutter, 2019) with a peak learning rate of 5e-05 and weight decay of 0.1. I train using a cosine learning rate scheduler with a warmup ratio of 0.1. I clip the gradient norm to 0.3. I train using an effective batch size of 16. I train for 4 epochs.

6 Results

I report my results on the SuperGLUE datasets in Table 1, showing accuracy, and in Table 2, showing F1.

The data does not provide evidence to generally support my hypothesis. While it provides evidence that my method works for RTE, this behavior is not consistently replicated in the other four SuperGLUE datasets I tested.
### Table 1: Accuracy results for different tasks in SuperGLUE and their respective subsets generated by my method.

<table>
<thead>
<tr>
<th>Task</th>
<th>Worst</th>
<th>Middle</th>
<th>Best</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSC</td>
<td>0.683</td>
<td>0.692</td>
<td>0.683</td>
<td><strong>0.702</strong></td>
</tr>
<tr>
<td>WIC</td>
<td><strong>0.730</strong></td>
<td>0.705</td>
<td>0.712</td>
<td>0.719</td>
</tr>
<tr>
<td>RTE</td>
<td>0.841</td>
<td>0.856</td>
<td><strong>0.877</strong></td>
<td>0.841</td>
</tr>
<tr>
<td>COPA</td>
<td><strong>0.910</strong></td>
<td>0.900</td>
<td><strong>0.910</strong></td>
<td>0.900</td>
</tr>
<tr>
<td>CB</td>
<td>0.696</td>
<td><strong>0.857</strong></td>
<td>0.804</td>
<td>0.839</td>
</tr>
</tbody>
</table>

### Table 2: F1 results for different tasks in SuperGLUE and their respective subsets generated by my method.

<table>
<thead>
<tr>
<th>Task</th>
<th>Worst</th>
<th>Middle</th>
<th>Best</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSC</td>
<td>0.676</td>
<td>0.681</td>
<td>0.631</td>
<td><strong>0.691</strong></td>
</tr>
<tr>
<td>WIC</td>
<td><strong>0.730</strong></td>
<td>0.702</td>
<td>0.709</td>
<td>0.718</td>
</tr>
<tr>
<td>RTE</td>
<td>0.839</td>
<td>0.856</td>
<td><strong>0.877</strong></td>
<td>0.841</td>
</tr>
<tr>
<td>COPA</td>
<td><strong>0.910</strong></td>
<td>0.900</td>
<td><strong>0.910</strong></td>
<td>0.900</td>
</tr>
<tr>
<td>CB</td>
<td>0.597</td>
<td><strong>0.599</strong></td>
<td>0.561</td>
<td>0.586</td>
</tr>
</tbody>
</table>

### 7 Analysis

I have a potential theory for why I did not see evidence of my hypothesis in evaluating my method on the SuperGLUE datasets. My hypothesis hinges on the assumption that in-context learning performance of examples is directly related to the specifics of the validation dataset. However, other works have contradictory claims for the source of in-context learning performance of examples. Specifically, Lee et al. (2024) claim that the LLM’s “knowledge” of in-context learning examples is a strong indicator of the performance of a particular example. In this work, the authors use perplexity of the example text with respect to the model as a proxy for its “knowledge” of the text. To a limited extent, they find that lower perplexity of in-context examples leads to better validation performance.

Naturally, this finding and my hypothesis contradict: I hypothesize that the relationship between in-context and validation examples contribute to validation performance, whereas they find that the relationship between the model used and the in-context examples contributes to validation performance. Without more depth the the hypothesis and claim, it logically follows that both arguments cannot simultaneously hold. To see why, both claims being true would imply that a perplexity is a good indicator for which data is beneficial to train on in order to improve validation performance. However, we know this is not true: perplexity is not generally a good data selection mechanism for finetuning (Li et al., 2024a).

Assuming the validity of their findings, and the results of my experiments, does this mean my hypothesis is incorrect? In its current form: it likely is. However, these results can educate the next form it takes on. One part of the hypothesis that is being called into question is the relationship between in-context learning examples and the validation dataset. I will detail future experiments that I plan on conducting.

#### 7.1 Possible Ablations

One logical step forward is to evaluate more broadly my method. While evidence thus far has not supported my initial hypothesis, I can nonetheless collect more data to base a larger analysis on. This will aid both in informing the scientific community of a negative result, and in informing future hypotheses.

I have compiled a list of possible ablations as next steps.

#### 7.1.1 Measuring Variation Amongst Subsets

It could be useful to measure quantifiable differences between the different subsets of text identified by my method. Features to measure include but are not limited to:

1. **Perplexity**
   - It would be highly valuable to analyze the perplexity of each of these subsets with respect to the LLM used as part of
the experiments. This is one way to test the claim of Lee et al. (2024), and I expect the better-performing subsets to be of lower perplexity, as supported by their results. Their experiments focused on a few-shot setting, so this analysis would be key for identifying if the pattern holds in the one-shot setting, as well. Seeing a trend in perplexity from the best subset to the worst subset would be a strong result.

2. Embedding similarity to each other and to the target distribution
   • It would be enlightening to compute and visualize the embeddings of the different text subsets. This can be done using a text embedding model such as SimCSE (Gao et al., 2022). Specifically, it would be interesting to see whether or not they form clusters in the embedding space. The validation dataset should also be visualized in the same manner. Additionally, we can compute a distance between the validation dataset and each subset, e.g. average distance or distance between clusters. A trend correlating inverse performance to distance here would partially support my initial hypothesis. This is because it would provide evidence of the relationship between in-context learning examples and the validation dataset.

3. Statistics on the text distribution such as text length
   • The text distribution itself of each subset is another feature worth investigating. It’s possible that simple statistics such as text or word length could be correlated to in-context performance on the validation dataset.

7.1.2 Expanded Scope of Datasets
Evaluating my method on a wider variety of datasets will be insightful, considering the fact that all of my experiments were conducted on datasets with SuperGLUE (Wang et al., 2020). Notably, I opted to use its smaller datasets due to computational costs. To capture a bigger picture, I should experiment using datasets with a variety of sizes, task types, and data collection methods. For instance, I could expand my scope to include dialogue-based datasets, or other datasets with tasks that differ from the more traditional tasks featured in SuperGLUE.

7.1.3 Expanded Scope of Models
My experiments thus far have focused solely on a single pretrained LLM: Mistral-7B. Conducting additional experiments with both instruction-tuned models and preference-aligned models via RLHF methods will allow us to collect more data to determine if my results still hold true for these types of models. A discrepancy here would spark further investigation into why that is the case.

Additionally, I could investigate these results using models of different scales. It has been shown that in-context learning behaviors can vary depending on the size of the LLM in question (Wei et al., 2023).

7.1.4 Method Detail Ablations
In addition to the training hyperparameters, there are several variables that can be ablated in the method pipeline.

1. Subset size
   • In my experiments, every subset was of a size equal to 25% of the total training dataset size. Changing this variable could shine light on if the hypothesis is dependent on the number of training examples used for finetuning. For instance, there could be either too little for the differences to make any impact, or there could be too many, and the differences are smoothed over by scale of the finetuning dataset. The latter would be the case if fine-grained distinctions are needed.

2. Number of subsets
   • Another possible ablation is to increase the number of subsets formed. Note that this wouldn’t modify the existing data, but rather provide more information on possible trends that occur throughout the training dataset. In the case where fine-grained distinctions may be needed, this would assist in representing the differences.

3. Number of times each training example is sampled
In the experiments performed, the ordering of training examples was determined by sampling the validation performance of each training examples 30 times. This was done due to the linear scaling of computational costs. However, analysis needs to be done on whether 30 samples is sufficient to determine an accurate ordering of training examples. It may be the case that significantly more samples are needed, in which case the computational resources needed to perform my method will likely become a bottleneck that will need to be addressed. The magnitude of error caused by this can be estimated through measuring the variance between orderings made from different random seeds. A high variance would be indicative that many more samples are needed, whereas a low variance would indicate that more samples would not significantly affect the outcome.

4. Size of validation dataset

- Related to the previous point, it could be interesting to analyze how the size of the validation set used affects the variance of the ordering generated. Currently, the entire validation set is used per dataset. Suppose I instead sampled a subset of the validation split and evaluated how my results changed. If there is any configuration that provides positive results, this variable will be one of the most important to practitioners. Then, they can be informed about how much validation data is required to achieve a notable performance boost, if any.

7.2 Synthetic Data

One theory as to why my method didn’t support my hypothesis is that it may be most prominent when the distribution of the training and validation data are different. In my SuperGLUE experiments, this was not the case. The training and validation data were from the same distribution. In situations where they are different, my method could identify training examples that are most similar to the validation distribution’s data.

This theory complements Lee et al. (2024) under the assumption that its findings are valid when the two data distributions are the same. In this case, with the assumption that the source distributions are identical, it could be valid to disregard the relationship between specific in-context examples and the validation dataset; perplexity by itself would be sufficient. However, when there is a distribution mismatch, we cannot rely on perplexity. It doesn’t matter how familiar the LLM is with the in-context data if the validation data is sufficiently different.

This primarily applies to data pipelines involving synthetic data. Here, it is unlikely that any synthetically generated training data comes from the exact same distribution as the validation data. The synthetic data could be noisy, biased, lack coverage, or otherwise misaligned with the target distribution. A followup hypothesis could be that in-context learning could be used for a pretrained LLM to identify out-of-distribution (OOD) data that is beneficial to finetune on with respect to the validation data. Note that this is different from typical OOD detection, a task which pretrained LLMs have been shown to be proficient at (Uppaal et al., 2023). This is because despite coming from different distributions, the examples in my training data may still be beneficial to train on. This has been shown to be such the case for sufficiently good synthetic data (Nayak et al., 2024).

If this followup hypothesis holds, the result will be just as useful for the intended purpose of data selection for finetuning with synthetic data. Thus, I will continue to conduct experiments in this direction to determine whether or not it holds.

7.2.1 Possible Synthetic Data Experiments

There are several different ways I can rigorously test this followup hypothesis. The difference comes down to how the synthetic data is generated. Some datasets, like PubMedQA, have heuristic-generated training splits (Jin et al., 2019). These datasets would be straightforward to use this method with. However, PubMedQA is a large dataset with over 200,000 examples in its training split. This computational barrier can be circumvented by sampling a smaller training subset from this distribution.

Another way I can test this hypothesis is to generate my own synthetic data. Using the Bonito model released in Nayak et al. (2024), I can generate synthetic instruction tuning data for any domain that we have unannotated documents for. For instance, I can generate my own dataset of questions from PubMed abstracts, similar to PubMedQA’s train split, except generated by an LLM. This kind of synthetically generated data has been shown to
be beneficial to finetune with for downstream task performance.

With Bonito, we have a few different choices for the set of unannotated documents to feed into the model. In addition to specialized domain documents like PubMed abstracts, there is an option to use documents from less specialized domains. For instance, we could use something that would be more similar to documents in the training data for Mistral 7B, such as recent Wikipedia articles. This may present different results than with the specialized domains.

Another alternative is to generate the synthetic datasets using an off-the-shelf LLM such as GPT-4 (OpenAI et al., 2024). While this method is more flexible, Nayak et al. (2024) demonstrated that this method gets eclipsed by Bonito in terms of performance on the downstream task of interest.

8 Conclusion

In conclusion, I present a hypothesis about the relationship between in-context learning and fine-tuning. Specifically, the hypothesis claims that examples which provide superior validation performance when used for in-context learning are also examples which provide superior finetuning performance, compared to a random baseline. While I was unable to find evidence of the hypothesis holding from my experiments on SuperGLUE, I offer an explanation as to why I could not find evidence supporting it on the datasets tested. I propose a followup hypothesis suggesting it could hold for synthetic datasets, along with additional experiments to be performed to evaluate it.

I additionally propose a data selection mechanism that will be valuable for any datasets in which the hypothesis holds. This is precisely because it is intended to utilize a signal representing downstream performance. By performing data selection based on the validation dataset, I can potentially outperform zero-shot data selection methods that are not dependent on the target distribution.

References


Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

Hui Huang, Yingqi Qu, Jing Liu, Muyun Yang, and Tiejun Zhao. 2024. An empirical study of llm-as-a-judge for ilm evaluation: Fine-tuned judge models are task-specific classifiers.


Yoonsang Lee, Pranav Atreya, Xi Ye, and Eunsol Choi. 2024. Crafting in-context examples according to LMs’ parametric knowledge.

Ming Li, Yong Zhang, Shhwai He, Zhitaio Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024a. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning.


Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity.


OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Alman, Diogo Almeida, Janko Alten Schmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiqing Bao, Mohammad Bavarian, Jeff Belghm, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko,


Rheeya Uppaal, Junjie Hu, and Yixuan Li. 2023. Is fine-tuning needed? pre-trained language models are near perfect for out-of-domain detection.

