# COST/BENEFIT ANALYSIS OF FINE-TUNING WORSE LLMS FOR SPECIFIC INSTRUCTIONS

**Authors**
Khoi Nguyen

## ABSTRACT

This study investigates the potential for cost and time efficiency in fine-tuning less advanced language models to perform domain-specific tasks, traditionally the province of more sophisticated models. Specifically, we explore whether Ada, when fine-tuned, can achieve comparable performance to Curie in the task of text elaboration, without incurring the higher costs and longer processing times associated with the latter. This research fills a notable gap in knowledge regarding the cost-performance trade-offs between these OpenAI models. Our preliminary data suggests that it is indeed feasible to achieve similar performance metrics using fine-tuned Ada models, with a notable reduction in cost. The 'Ada' model, '1k Ada', and '10k Ada' achieved average semantic scores of 0.7899, 0.7528, and 0.7842 respectively, indicating a comparable level of performance to the Curie model in text elaboration and an indication of improved performance based on additional fine-tuning. In terms of total cost, the Ada models have relatively low fixed costs, so advanced models will overtake the total costs after relatively low usage. Our findings thus far suggest that fine-tuning Ada for specific text generation tasks can be a cost-effective and time-efficient alternative to employing higher-order models. This has significant implications for machine learning applications, particularly those on a budget or under time constraints.

## 1 INTRODUCTION

OpenAI offers a variety of advanced models for users to access, with the most renowned being Davinci-03. While its superior performance outshines other models, Davinci-03's slower speed and higher cost pose limitations. In this study, we aim to explore whether adapting a less advanced model, such as Ada, for specific text generation applications can achieve comparable results to Davinci-03 at a lower cost and faster rate. There is currently limited data available on the cost-performance trade-offs between fine-tuning Ada in this manner as opposed to utilizing Davinci-03. Our research seeks to address this gap in knowledge. For the purpose of this experiment, we will investigate if Ada could be fine-tuned to a similar performance as Curie for the domain specific task "Elaborate on the following sentence" since experimenting with these models will not incur as many costs as experimenting with Davinci, but should reveal relevant data and insights.

### 1.1 HYPOTHESIS AND QUESTION

Is it possible to fine-tune Ada to be as effective as Curie for the domain specific task "Elaborate on the following sentence"?

We hypothesize that it is feasible and preferable to fine-tune Ada to perform at a similar level as Curie without incurring the higher costs and longer processing times associated with more advanced models. In essence, the fine-tuning of a less sophisticated model could yield comparable or superior outcomes than relying on complex models for domain-specific tasks.

### 1.2 LITERATURE REVIEW

In recent years, Natural Language Processing (NLP) has witnessed significant progress with the introduction of large pre-trained language models, such as BERT, GPT-4, and Transformer, that have considerably enhanced the performance of various NLP tasks. NLP has also been popularized after the release of ChatGPT 1, which has been used by millions of users (Citation Needed). Nonetheless, these models are typically trained on extensive general-domain data and may not yield optimal results for domain-specific tasks. To tackle this constraint, fine-tuning has emerged as a prevalent approach for modifying these models to excel in specific domains. In this literature review, we delve into the intricacies of fine-tuning NLP models for domain-specific tasks and assess contemporary research efforts in this field (NerdyNav).

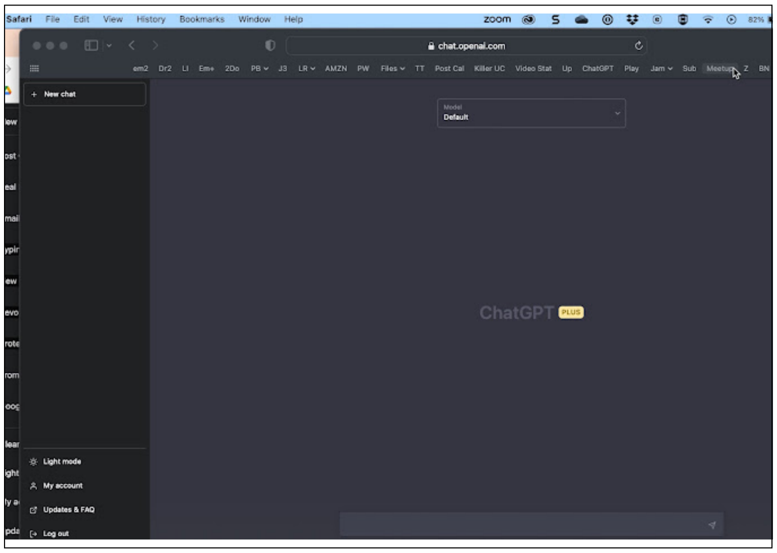

Figure 1: This is ChatGPT, a web-app powered by GPT3 and GPT4 by OpenAI. It has popularized the use of LLMs for instruction based tasks.

Fine-tuning entails the process of adapting a pre-trained machine learning model to focus on a particular task. While this technique offers a cost-effective and time-efficient solution for designing models suited for specific tasks, it also presents corresponding challenges and limitations.

A primary obstacle in fine-tuning is the substantial computational cost linked with training deep neural networks. This process necessitates researchers to possess considerable data reserves and high-performance computing resources, potentially incurring significant expenses and time. Another challenge lies in the risk of overfitting when fine-tuning a model. Striking a balance between refining the model for a new task and retaining the original model's generalization capabilities is crucial to avoid overfitting, which can cause the model to be overly specialized to the training data, ultimately leading to diminished performance with unfamiliar data (Tetko et al., 1995).

In spite of these challenges, fine-tuning remains a widely employed and efficient method for developing machine learning models, particularly in the context of NLP. However, the cost-benefit analysis regarding fine-tuning is currently limited. As noted by OpenAI, "in general, we've found that each doubling of the dataset size leads to a linear increase in model quality" (OpenAI).

The implications of this research range from offering insights into the applicability and drawbacks of fine-tuning NLP models to guiding future developments in the field. By understanding the benefits and limitations of fine-tuning, researchers can make informed decisions about model adaptation and better tailor large pre-trained language models to suit specific domain tasks. This knowledge contributes to the ongoing advancement of NLP technology and its applications in diverse areas.

## 2 Experimental Design

Please refer to the GitHub repository's README for a low level description of the experimental design used for this research as well as the instructions to complete the project. In this paper, we will discuss at a high level the description of the experimental design 2

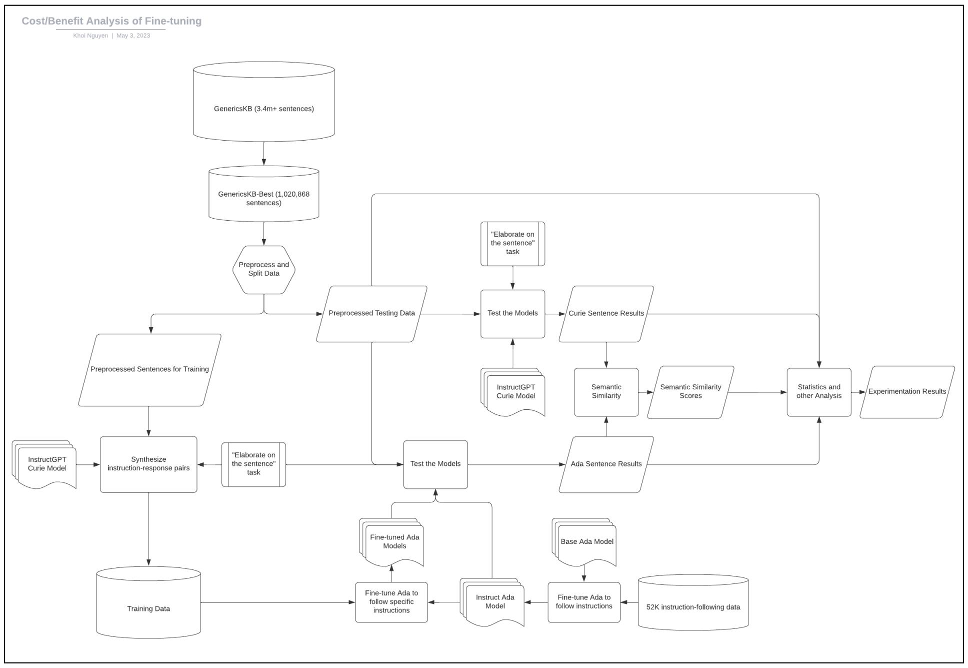
Cost/Benefit Analysis of Fine-tuning
Khoi Nguyen | May 3, 2023
GenericsKB (3.4m+ sentences)
GenericsKB-Best (1,020,868 sentences)
Preprocess and Split Data
Preprocessed Sentences for Training
Preprocessed Testing Data
"Elaborate on the sentence" task
Test the Models
Curie Sentence Results
InstructGPT Curie Model
Semantic Similarity
Semantic Similarity Scores
Statistics and other Analysis
Experimentation Results
InstructGPT Curie Model
Synthesize instruction-response pairs
"Elaborate on the sentence" task
Test the Models
Ada Sentence Results
Fine-tuned Ada Models
Base Ada Model
Training Data
Fine-tune Ada to follow specific instructions
Instruct Ada Model
Fine-tune Ada to follow instructions
52K instruction-following data

Figure 2: This is the visualization of the project's experimental design.

## 2.1 Datasets

### 2.1.1 GenericsKB

The GenericsKB dataset encompasses over 3.4 million generic sentences pertaining to the world, featuring statements expressing general truths such as "Dogs bark" and "Trees remove carbon dioxide from the atmosphere." Generics hold potential as a knowledge repository for AI systems necessitating a comprehensive understanding of the world. The GenericsKB represents the first large-scale resource compiling naturally occurring generic sentences, as opposed to extracted or crowdsourced triples, and is abundant in high-quality, general, and semantically complete statements (Citation Needed). The primary sources for extracting generics include the Waterloo Corpus, selected portions of Simple Wikipedia, and the ARC Corpus. Additionally, a filtered, high-quality subset called GenericsKB-Best is available, comprising 1,020,868 sentences, which can be accessed through this Google Drive link. Thus, for this project, we will employ the GenericsKB-Best dataset, downloadable from the aforementioned Google Drive link.

### 2.1.2 52K Instructions from Stanford Alpaca

Given that the base Ada model is not inherently designed to follow instructions (in contrast to the InstructGPT model), we will proceed to fine-tune the base model using the 52K dataset provided by the Stanford Alpaca repository. This data consists of 52,000 instruction and completion pairs as to be used for fine-tuning Meta's LLaMA model to be instruction following. The resulting instruction-following Ada model, which is fine-tuned on this 52K dataset, will serve as the base model for subsequent experimentation in this research.

Table 1: Example of an instruction-completion pair from the 52K dataset

| **KEY** | **VALUE** |
|---|---|
| Instruction | Identify the odd one out. |
| Input | Twitter, Instagram, Telegram |
| Output | Telegram |

### 2.2 Design

The design of the experiment can be summarized as preparing the datasets and the training and testing data, fine-tuning the models, testing the models on the testing data, performing semantic similarities on the results, and then doing analysis on the results.

Execute `prepare_instruct_ada.ipynb` to reorganize and integrate the 52K dataset, a collection of instructional data points, enabling the fine-tuning of the Ada model to enhance its instruction-following capabilities.

Execute `preprocess_data.ipynb` to perform data preprocessing tasks on the GenericsKB-Best dataset, ensuring its structural coherence and format compatibility with the project requirements. This process includes data cleaning and encoding transformations to facilitate seamless integration into the experimentation pipeline.

Execute `split_data.ipynb` to partition the GenericsKB-Best dataset into distinct training and testing sets, ensuring a fair distribution of data samples. This strategic segmentation enables model evaluation by reserving a portion of the dataset exclusively for performance assessment after training. Moreover, the splitting process accounts for various factors, such as class balance and stratification, to maintain data integrity and avoid potential biases in model training and testing.

Execute `prepare_training_data.ipynb` to organize the training data required for fine-tuning. Training data plays a pivotal role in guiding GPT models towards generating desired outputs by exposing them to examples that align with the intended content. By utilizing the text completion endpoint, we will synthesize the training data, transforming it into a `.jsonl` document format. This transformation involves creating a set of prompt-completion pairs, with each line of the `.jsonl` file representing a training example.

To generate the training data for models undergoing fine-tuning, we will extract the training sets from each ada folder, using Curie to create completion pairs that correlate with desired model outputs. This process ensures that the prepared training data effectively contributes to enhancing the performance of the Ada models during the fine-tuning phase. Once the training data is prepared, we use the OpenAI CLI to fine-tune the 52K Instruct Ada.

Once the models are fine-tuned, the next step is to test the models on the test set and evaluate the performances of the models. Performance is determined by how well the Ada models can generate sentences that are semantically similar to the Curie model. This is accomplished by running `run_models.ipynb` and `semantic_similarity.ipynb`. The last step of analysis is entirely done in the `analysis.ipynb`.

## 3 Results

In the context of fine-tuning language models, this study performed a cost-benefit analysis by examining both the financial cost and the performance gains associated with three models: Ada, 1k_Ada, and 10k_Ada. These models represent different levels of fine-tuning, with Ada receiving the least and 10k_Ada receiving the most.

### 3.1 Observations

During the synthetic creation of data with Curie, we noticed that some of the prompt-completion pairs that were produced were not ideal. For instance, we would expect that ”Elaborate on the following sentence: Tarantulas are spiders” would return a completion like ”Tarantulas are spiders because they have 8 legs”; however, we noted that the completion was ”This sentence talks about spiders” or something similar with analyzing the sentence.

### 3.2 Data and Analysis

The benefit of fine-tuning was evaluated based on the average semantic score of the models. The semantic score reflects the quality of the output from the language models, with a higher score indicating better performance.

Table 2: Average Semantic Scores

| **Model** | **Average Semantic Score** |
|---|---|
| Ada | 0.7899 |
| 1k_Ada | 0.7528 |
| 10k_Ada | 0.7842 |

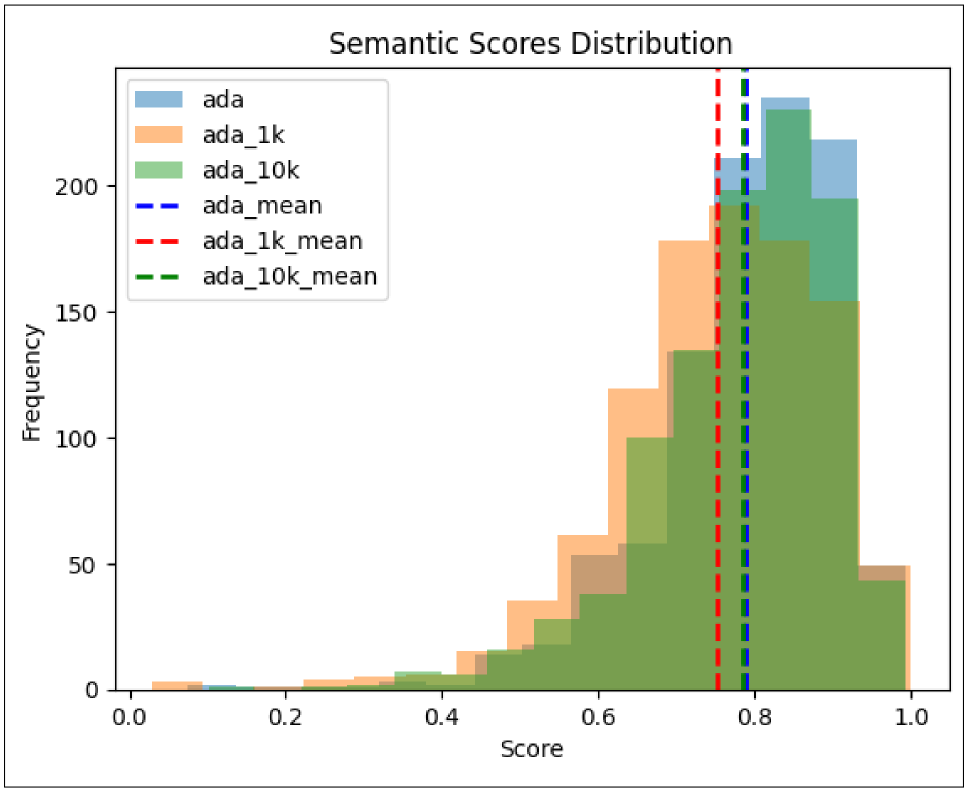

Figure 3: This is the distribution of the semantic scores based on the test results of each of the fine-tuned Ada models.

The Ada model itself results in an average semantic similarity score of 0.7899. Noting that a semantic similarity score of 1 would indicate an identical sentence, we could see that our Instruct Ada performs well as it alone produces response that are 78% similar to the results of the Curie model.

It was observed that the 1k_Ada model, which was fine-tuned on limited data, performed worse than the other models 2. Although it's difficult to identify the reason to why the 1k_Ada model performed worse than the base Instruct Ada model, we believe that the model performed worse as it was trained on way too limited data, and as a result of that, over-fitted poor results and demonstrated worse performance 3.1. When the model was subsequently trained on 10k data, the results improved to be similar to that of the base Ada model.

Table 3: Intersection values of Curie and Ada

| **Model** | **Tokens** | **Total Cost** |
|---|---|---|
| Ada | 17503 | 35.00 |
| 1k_Ada | 18572 | 37.14 |
| 10k_Ada | 24447 | 48.89 |

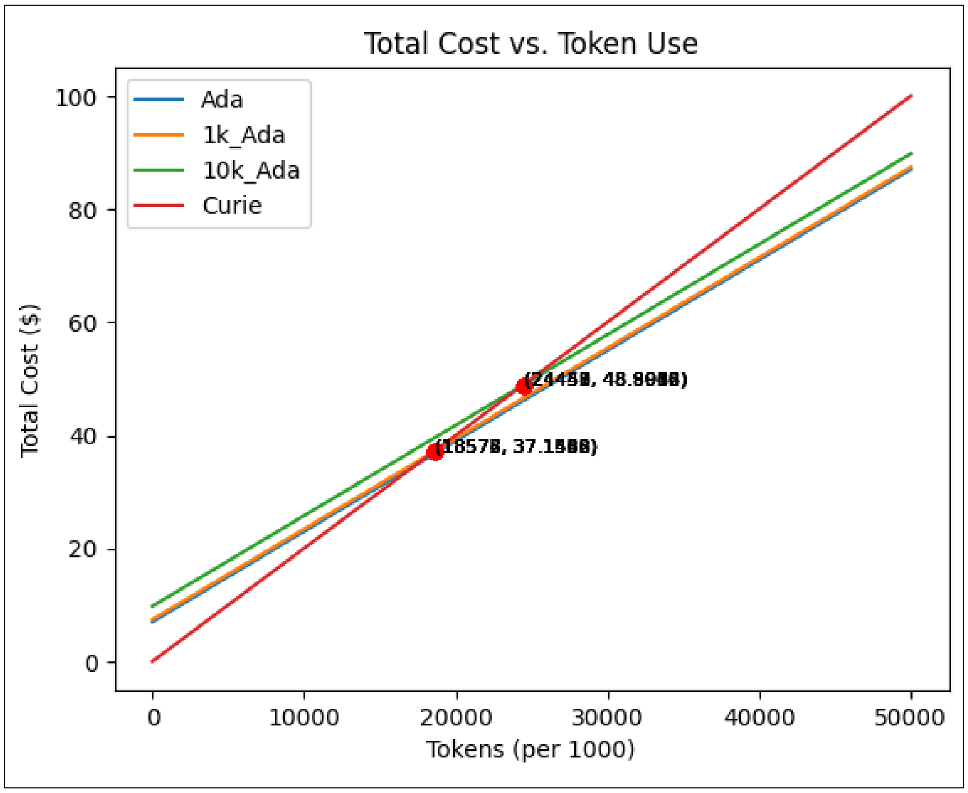

Figure 4: This is the visualization of costs over time for each of the fine-tuned Ada models.

The financial cost of using these models is two-fold: fixed costs and variable costs. The fixed costs, which are a one-time expenditure, were calculated for each model. These costs increase with the level of fine-tuning, reflecting the additional computational resources required for this process.

In addition to the fixed costs, there are variable costs associated with the usage of the models, specifically the cost per 1,000 tokens processed. The models all exhibited a linear increase in total cost (fixed + variable) with increasing token usage.

The models were compared against the Curie model, which has a higher usage cost per 1,000 tokens, but no additional fixed costs. The intersection of the cost lines for each Ada model with the Curie model line represents the point at which the total cost of using the Ada model equals the total cost of using the Curie model.

For the Ada model, this intersection occurred at 17,503 tokens with a total cost of $35.00. For the 1k_Ada model, the intersection was at 18,572 tokens and $37.15, while for the 10k_Ada model, it was at 24,447 tokens and $48.89.

These intersection points can be interpreted as the break-even point for token usage. Beyond this point, the Ada model becomes more cost-effective.

## 4 Conclusion

### 4.1 This Study

The primary objective of this study was to examine the feasibility and potential benefits of fine-tuning a less advanced language model, Ada, to match the performance level of a more advanced model, Curie, for a specific domain task - text elaboration. Our guiding hypothesis proposed that it might be possible to achieve comparable or even superior results via fine-tuning Ada, without the incurred higher costs and longer processing times associated with the more complex model, Curie.

The evidence gathered throughout our research lends substantial support to this hypothesis, albeit with some context-dependent considerations. Our data demonstrated an incremental improvement in performance with increased fine-tuning, as evidenced by the superior average semantic score of 10k_Ada (0.7842) over 1k_Ada (0.7528).

From a cost perspective, despite the initially increased investment for fine-tuning (with 10k_Ada costing more than 1k_Ada), the comparative costs remained significantly lower than those associated with the use of a more advanced model. In an industrial context, where large-scale applications could rapidly escalate variable costs of more advanced models, the initial fixed cost of fine-tuning Ada appears to be a more economical choice.

This study highlights the potential value of model fine-tuning as a strategic choice in the deployment of language models for domain-specific tasks. By investing in the up-front cost of fine-tuning a less advanced model like Ada, organizations may achieve performance levels comparable to more advanced models, such as Curie, while significantly reducing their ongoing cost base.

These findings, although promising, should be interpreted within the context of the specific task of text elaboration. Further research is needed to confirm the generalizability of these findings to other domain-specific tasks. Furthermore, future investigations could elucidate the precise mechanisms and optimal strategies for fine-tuning that yield the best performance-to-cost ratio. This research is a stepping stone towards a more cost-effective, efficient, and strategic use of AI language models in various industries and applications.

### 4.2 Future Work

As part of our research, we elected not to fine-tune the 100k_Ada model, a decision primarily driven by budgetary considerations. The projected costs for this task surpassed $70, a budget line that we deemed excessive for the scope of this project. Furthermore, the risk of potential failure during the fine-tuning process, as we had experienced in multiple instances throughout this study, played a significant role in our decision-making. Each failed attempt at fine-tuning would have escalated the costs further without yielding any beneficial results, a risk we were not prepared to take for this particular research.

During our investigation, we also analyzed the repository to determine if there were any correlations between sentence length and semantic score, among other potential relationships. Although the data appeared to be somewhat random and did not indicate any clear correlations or trends, we believe there may still be some valuable insights hidden within this seeming randomness.

### Acknowledgments

It is my pleasure to acknowledge the invaluable contributions of several organizations that have enabled my research and helped to bring my ideas to fruition. Without their support, this project would not have been possible. I want to extend my deepest gratitude to OpenAI, Harvard College, and Hugging Face for their instrumental role in making this research a reality.

## References

NerdyNav. 101 quotable chatgpt statistics user numbers in may 2023 (gpt-4, plugins update). URL https://nerdynav.com/chatgpt-statistics/.

OpenAI. Fine-tuning. URL https://platform.openai.com/docs/guides/fine-tuning.

Igor V. Tetko, David J. Livingstone, and Alexander I. Luik. Neural network studies. 1. comparison of overfitting and overtraining. *Journal of Chemical Information and Computer Sciences*, 35(5):826–833, 1995. doi: 10.1021/ci00027a006. URL https://doi.org/10.1021/ci00027a006.