ADVERSARIALLY ROBUST LLM FINE-TUNING VIA CORESET SELECTION WITH SPIN

Pratham Kheskwani, Sahil Dharod & Aditya Kumar *
Department of Electrical Engineering
Indian Institute of Bombay
Mumbai,400076, India
{21d070051,210070026,210070003}@iitb.ac.in

1 BACKGROUND AND MOTIVATION

Neural networks have achieved great success in the past decade. Today, they are one of the primary candidates in solving a wide variety of machine learning tasks, from object detection and classification to photo-realistic image generation and beyond. Further, with the advancements in large language models (LLMs), they have also become instrumental in natural language processing, enabling tasks such as language generation, translation, and understanding. But, training large language models (LLMs) poses significant resource challenges. These models demand substantial computational power and memory, requiring access to high-performance computing infrastructure. Also despite the great success, it has come to light in recent years,that the use of adversarial attacks can lead the model to fail miserably. These attacks which are capable of altering model behaviour by changing the input by small imperceptible perturbations, pose significant challenges to the reliability of deployed systems causing serious concerns to security-critical areas of deployment.

Adversarial training has emerged as a promising defense mechanism, improving the model robustness but at an additional cost of increased computational complexity and time due to the need for generating adversarial examples across the entire training dataset and increased training time. Leveraging upon the theory of coreset selection, we show how selecting a small subset of training data that can effectively capture the diversity of dataset, provides a principled approach to reducing the time complexity of robust training, thus also reducing the computational burden. Utilizing curriculum learning, along with coreset selection, we aim to reduce the data and computation cost. Also using SPIN (Self-Play Mechanism) which is a technique used in training large language models (LLMs) which involves training through self-play further optimizes the use of data. Merging all these techniques together, we aim to create more robust models with limited data-usage.

2 CONTRIBUTIONS

- Proposal of a hardness definition for curriculum learning to be used along with random sampling for better self-play fine tuning(SPIN) of LLMs.
- Proposal of a facility coverage maximization-based coreset selection method for optimizing SPIN fine-tuning of LLMs, built upon the foundation of CORDS IITB.
- Proposal of incorporating adversarial training with the aforementioned techniques to enhance the robustness of large language models (LLMs) trained under constraints of limited data and computational resources.

3 PROBLEM SETTING

We consider a Large Language Model (LLM) parameterized by and denoted by p. The model takes as input a sequence $= [x_1, \ldots, x_n]$, commonly referred to as the prompt, to generate the corresponding response $= [y_1, \ldots, y_m]$. The response is therefore considered as a sample from the conditional probability distribution $p(\cdot|)$. In LLMs, x_i and y_j represent individual tokens from a predetermined vocabulary within the sequences and , respectively. The auto-regressive model p generates tokens

*Footnotes.

sequentially for a given position, leveraging only the sequence of previously generated tokens. This model therefore constitutes a Markov process, where the conditional probability distribution $p(|)$ can be expressed through a decomposition as follows:

$$p(y|x) = \prod_{j=1}^{m} p(y_j|x, y_{<j}),$$

where $y_{<1}$ is null and $y_{<j} = [y_1, \ldots, y_{j-1}]$ for $j = 2, \ldots, m$.

Two major fine-tuning methods used for LLMs previously are : supervised fine-tuning(SFT) and reinforcement learning fine-tuning(RLHF). The problem with SFT is the requirement of a large corpuse of good dataset, while the problem with RLHF is requirement of a good reward function which relies of training over a preference dataset, which again requires large high-quality data. Evaluating over various datasets and models, it is seen that the expressive power of data is not utilised fully with these models, SPIN builds on these techniques to achieve better results.Chen et al. (2024)

The basic algorithm of SPIN involves random sampling of examples, but using facility-location for subset selection(maximum representation based selection) helps in effective generalized learning of data distribution.

Also, the main bottleneck in the time/computational complexity of adversarial training stems from constructing adversarial examples for the entire training set at each epoch. Instead of using a faster adversarial example generator, here, we take a different, *orthogonal* path and try to reduce the training set size effectively. This way, the original adversarial training algorithm can still be used on this smaller subset of training data.Dolatabadi et al. (2022) This approach can reduce the time/computational complexity, while optimizing for robustness similar to full adversarial training.

Thus, the problem we aim to address revolves around making LLM training more accessible,robust and efficient, particularly in settings where data availability is limited and computational resources are scarce.

4 METHODS

In this section, we will discuss the development of our algorithm from base SPIN algorithm step by step.

Algorithm 1 Base Self-Play Fine-Tuning ()

Input: $\{(x_i, y_i)\}_{i\in[N]}$: SFT Dataset, p_{θ_0}: LLM with parameter θ_0, T: Number of iterations.
for $t = 0, \ldots, T-1$ **do**
  for $i = 1, \ldots N$ **do**
    Generate synthetic data $y_i' \sim p_{\theta_t}(\cdot|x_i)$.
  end for
  Update $\theta_{t+1} = \arg\min_{\in} \sum_{i\in[N]} \ell\Big(\lambda \log \frac{p(y_i|x_i)}{p_t(y_i|x_i)} - \lambda \log \frac{p(y_i'|x_i)}{p_t(y_i'|x_i)}\Big)$.
end for
Output: θ_T.

The basic SPIN algorithm can be understood as below:Chen et al. (2024)
Let us consider a two-player game, where the main player's objective is to distinguish the responses generated by the LLM and those generated by the human. Meanwhile, the opponent's role is to generate responses that are indistinguishable from the human's responses. The core of our method is the self-play mechanism, where both the main player and the opponent are the same LLM, but from different iterations. More specifically, the opponent is the old LLM from the previous iteration, and the main player is the new LLM to be learned in the current iteration. Taking into account all this, we arrive at the end-to-end objective function which has to be minimized.

$$L(\theta, \theta_t) = E_{x\sim q(\cdot), y\sim p_{\text{data}}(\cdot|x), y'\sim p_{\theta_t}(\cdot|x)}\left[\ell\left(\lambda \log \frac{p_\theta(y|x)}{p_{\theta_t}(y|x)} - \lambda \log \frac{p_\theta(y'|x)}{p_{\theta_t}(y'|x)}\right)\right]. \quad (1)$$

where ℓ is a monotonically decreasing function.

More can be found out in the appendix. Now, first we modify the approach of training execution, by using curriculum based learning for which we use length of prompt as a proxy of hardness. Thus first we feed the model with samples of smaller length prompts and increase it step by step. It is essential to note that here we still sample the subset randomly; just the order in which the samples will be used for training is fixed.Nagatsuka et al. (2021)

$$Hardness = \text{Length of Prompt} \tag{2}$$

Now, further we change the method of sampling, by using subset selection based on representation maximization (using facility coverage as submodular maximization criterion). This is useful as using representation functions ensures that we find cluster representatives (X) such that the most similar cluster based similarity summed across all points is maximized. In our case, we find cluster representatives of batches having similar gradients; thus giving our model points across which it's performance varies a lot.(As output text generated is of varied length so are the gradients, so we truncated the gradients to minimum length between the two for which distance is being calculated and then normalized with respect to minimum length to ensure uniformity across lengths ; the distance metric used is Euclidean distance)Kaushal et al. (2022)

$$\text{Facility Coverage Objective}= \sum_{i \in Vertices} max_{k \in X_{subset}} S_{ik} \tag{3}$$

where S_{ik} is negative of normalised mean-square difference between gradients of i,k batches (basically a similarity function).
Now, finally we also incorporate adversarial attack into our algorithm;

Algorithm 2 Final Self-Play Fine-Tuning

Input: $\{(x_i, y_i)\}_{i \in [N]}$: Subset obtained from SFT Dataset after facility coverage representation, p_{θ_0}: LLM with parameter θ_0, T: Number of iterations.
for $t = 0, \ldots, T-1$ **do**
  for $i = 1, \ldots N$ **do**
    Generate adversarial prompt $x_i' = ATTACK(x_i, y_i)$
    Generate synthetic data $y_i' \sim p_{\theta_t}(\cdot|x_i')$.
  end for
  Update $\theta_{t+1} = \arg\min_{\in} \sum_{i \in [N]} \ell\Big(\lambda \log \frac{p(y_i|x_i)}{p_t(y_i|x_i)} - \lambda \log \frac{p(y_i'|x_i)}{p_t(y_i'|x_i)}\Big)$.
end for
Output: θ_T.

Algorithm 3 ATTACK

Input: Input example $(\mathbf{x_i}, \mathbf{y_i})$, $\epsilon = maxpertubation$, number of steps K.
Initialize perturbation $\eta = \mathbf{0}$.
for $k = 1$ **to** K **do**
  Compute gradient: $\nabla_\eta L(f(\mathbf{x} + \eta), y_{\text{true}})$.
  Project gradient onto L2 ball: $\eta \leftarrow \eta + \alpha \cdot \text{clip}(\nabla_\eta L(\mathbf{x} + \eta, y_{\text{true}}), \delta)$.
  Clip perturbation to ensure it stays within budget: $\eta \leftarrow \text{clip}(\eta, \epsilon)$.
end for
Output: Perturbed example $\mathbf{x}' = \mathbf{x} + \eta$.

We feed in the model and adversarial prompt that only differs slightly from the original prompt and then train the model to generate data similar to given data distribution(SFT data distribution)Geisler et al. (2024).

5 RESULTS

We utilized the Qwen1.5-0.5B model Bai et al. (2023), a transformer-based decoder-only pre-trained language model, along with the Ultrachat200k dataset. The Ultrachat200k subset, selected from the

larger UltraChat corpus containing approximately 1.4 million dialogues generated using OpenAI's Turbo APIs, represents a high-quality subset for our experiments. We conducted five experiments, two of which served as baselines:

1. In the first experiment, we assessed our pre-trained model's performance by randomly sampling 1,000 prompts from the UltraChat corpus multiple times and generating synthetic responses using the model.
2. In the second experiment, we randomly selected 1,000 prompts from the UltraChat corpus, introduced adversarial perturbations using modified FGSM, and then evaluated the model's performance based on the synthetic responses.
3. For the third experiment, we randomly sampled 1,000 prompts from the UltraChat corpus and trained the model using the SPIN technique over three iterations. Subsequently, we evaluated the model's performance.
4. In the fourth experiment, we employed facility coverage-based submodular maximization to sample 1,000 prompts from the UltraChat corpus. We then trained the model using the SPIN technique over three iterations and evaluated its performance.
5. Finally, in the fifth experiment, we sampled 1,000 prompts from the UltraChat corpus using facility coverage-based submodular maximization. After introducing adversarial perturbations using modified FGSM, we trained the model using the SPIN technique over three iterations and assessed the robustness of the resulting model.

Table 1: Information of the results obtained in all the experiments

Experiment no.	Exp-1	Exp-2	Exp-3	Exp-4	Exp-5
BLEU-1 Score	0.12	0.09	0.22	0.24	0.20

Figure 1: Example of ground truth completion compared to the fine-tuned model generation at iteration 0 and 3.

6 CONCLUSIONS

Our project delves into innovative methodologies aimed at fine-tuning large language models (LLMs) within the constraints of limited data and computational resources. We introduced a curriculum learning-based technique(using length of input prompt as a proxy to hardness), seamlessly integrated with self-play fine-tuning(SPIN), to enhance the accuracy of LLMs. Additionally, we incorporated a subset selection method based on facility coverage to optimize SPIN fine-tuning, further enhancing model performance. Furthermore, we established a baseline for adversarial prompt attacks on LLMs and demonstrated their adversarial robustness by integrating modified FGSM attacks into SPIN iterations. These comprehensive experiments underscore the importance of tailored training strategies in effectively addressing the challenges presented by resource constraints and adversarial attacks encountered during LLM training.

7 FUTURE WORK

Some possible avenues for future research are as below:

1. **Enhanced Adversarial Training for Robustness Against Stronger Attacks:** Investigating advanced adversarial training techniques, such as ensemble adversarial training or adversarial training with gradient regularization, to improve the robustness of LLMs against stronger and more sophisticated attacks. By exploring novel methods for generating and incorporating adversarial examples, we can strengthen the model's defenses and enhance its resilience to adversarial manipulation.
2. **Exploration of Novel Measures of Hardness for Curriculum Learning:** Research alternative measures of hardness for curriculum learning, such as uncertainty-based measures or diversity-based measures, to better guide the selection of training examples. By incorporating more nuanced measures of example difficulty, we can tailor the curriculum to the specific learning needs of the model, leading to more effective and efficient training.
3. **Dynamic Target Data Distribution for Improved Performance:** Investigate techniques for dynamically adjusting the target data distribution during training to adapt to changing conditions and challenges. By continuously monitoring and updating the target distribution based on real-time feedback, we can better align the training process with the evolving requirements of the task, potentially elevating the performance of LLMs to unprecedented levels.
4. **Exploration of Alternative Subset Selection Techniques:** Explore alternative subset selection techniques, such as adversarial landscape coverage, to identify representative subsets of data that effectively cover the adversarial landscape. By selecting subsets of data that capture the diversity and complexity of the input space, we can provide more comprehensive training signals to the model, leading to improved generalization and robustness.

ACKNOWLEDGMENTS

We thank Prof. Ganesh, and Teaching Assistants of the course CS769 at IIT Bombay for the necessary feedback during the course of this work.

8 APPENDIX

8.1 SELF-PLAY FINE-TUNING

Let us consider a two-player game, where the main player's objective is to distinguish the responses generated by the LLM and those generated by the human. Meanwhile, the opponent's role is to generate responses that are indistinguishable from the human's responses. The core of our method is the self-play mechanism, where both the main player and the opponent are the same LLM, but from different iterations. More specifically, the opponent is the old LLM from the previous iteration, and the main player is the new LLM to be learned in the current iteration.

In iteration $t+1$, the opponent is the LLM from the previous iteration, denoted by p_t, which generates responses $'$ for those prompts in the SFT dataset according to $p_t(\cdot|)$. Our method, therefore, consists of the following two steps at iteration $t+1$: (1) training the main player, and (2) updating the opponent player.

Training the Main Player. We begin with illustrating how we expect a main player is trained to distinguish LLM responses from human responses. Motivated by integral probability metric (IPM), we formulate our objective function such that the main player f_{t+1} maximizes the expected value gap between the target data distribution p_{data} and the opponent player's distribution p_t:

$$f_{t+1} = \arg\max_{f \in F_t} E_{x\sim q(\cdot), y\sim p_{\text{data}}(\cdot|x), y'\sim p_{\theta_t}(\cdot|x)}\big[f(x,y) - f(x,y')\big], \tag{4}$$

where F_t is a sequence of highly expressive function classes that we will determine in later deduction. The subscript t in F_t is due to that the function class is dependent on p_{θ_t}. Given such a f_{t+1} and a response sequence y to the prompt x, the value of $f_{t+1}(x,y)$ reflects the main player's degree of belief that originates from p_{data} rather than p_{θ_t}. Ideally, the main player f_{t+1} should yield a high value when $y \sim p_{\text{data}}(\cdot|x)$ and a low value when $y' \sim p_{\theta_t}(\cdot|x)$, where p_{θ_t} is the opponent's distribution. Instead of solving equation 4, we can also solve the following more general optimization problem,

$$f_{t+1} = \arg\min_{f \in F_t} E_{x\sim q(\cdot), y\sim p_{\text{data}}(\cdot|x), y'\sim p_{\theta_t}(\cdot|x)}\big[\ell\big(f(x,y) - f(x,y')\big)\big], \tag{5}$$

where $\ell(\cdot)$ is a loss function that is both monotonically decreasing and convex. For example, a linear loss function $\ell(t) = -t$ reduces equation 5 to the minimization version of equation 4. However, the use of a linear loss function results in an unbounded objective value, which, during continuous training, leads to a negative infinite value of $f(x,y')$ on the opponent player's responses. Therefore, in our work, we choose the logistic loss function $\ell(t) := \log(1+\exp(-t))$ for its non-negativity, smoothness, and exponentially decaying tail as $t \to \infty$. Such a choice of loss function aids in preventing the excessive growth in the absolute value of f.

Updating the Opponent Player. Previously we have discussed the training of f_{t+1} given the opponent player's distribution p_{θ_t}. Now suppose we have optimized our main player f_{t+1} that can distinguish p_{data} from p_{θ_t}, within a certain function class F_t, we elaborate how we get parameter θ_{t+1} of the opponent player. Specifically, when presented with two responses y and y' to the same prompt x, f_{t+1} assesses the values $f_{t+1}(x,y)$ and $f_{t+1}(x,y')$. It then infers that the response with the higher value is from the real data distribution p_{data} and the response with lower value is attributed to the LLM p_{θ_t}. Subsequently, the objective of the opponent player is to find a better LLM that generates responses indistinguishable from p_{data} for the main player. This is achieved by maximizing the expected value $_{x\sim q(\cdot), \sim p(\cdot|)}[f_{t+1}(x,y)]$. In addition, to prevent excessive deviation of $p_{\theta_{t+1}}$ from p_{θ_t} and stabilize the self-play, we incorporate a Kullback-Leibler (KL) regularization term. Putting these together gives rise to the following optimization problem:

$$\arg\max_{p} E_{x\sim q(\cdot), y\sim p(\cdot|x)}[f_{t+1}(x,y)] - \lambda_{x\sim q(\cdot)}\text{KL}\big(p(\cdot|x)||p_{\theta_t}(\cdot|x)\big), \tag{6}$$

where $\lambda > 0$ is the regularization parameter. Notably, equation 6 has a closed-form solution $\hat{p}(\cdot|)$:

$$\hat{p}(y|x) \propto p_{\theta_t}(y|x)\exp\big(\lambda^{-1} f_{t+1}(x,y)\big). \tag{7}$$

It is worth noting that $\hat{p}(\cdot|x)$ is not guaranteed to be belong to the LLM space $\{p(\cdot|x)|\theta \in \Theta$. Since we hope that the closed-form solution $\hat{p}$ in the probability space can be realized by an LLM

with parameter , i.e., $p_\theta(y|x) = \hat{p}(y|x)$, solving for $p(y|x) \propto p_{\theta_t}(y|x) \exp\left(\lambda^{-1} f_{t+1}(x,y)\right)$ gives $f_{t+1}(x,y) = \lambda \cdot \log \frac{p_\theta(\cdot|x)}{p_{\theta_t}(\cdot|x)}$.

This suggests the following function class $_t$ for f_{t+1}:

$$F_t = \left\{ \lambda \cdot \log \frac{p_\theta(y|x)}{p_{\theta_t}(y|x)} \middle| \theta \in \Theta \right\}, \tag{8}$$

where Θ is the parameter space of LLMs being considered. Given the choice of F_t in equation 8, optimizing equation 5 gives f_{t+1} parameterized by $_{t+1}$ in the following form:

$$f_{t+1}(x,y) = \lambda \cdot \log \frac{p_{\theta_{t+1}}(y|x)}{p_{\theta_t}(y|x)}. \tag{9}$$

Substituting equation 9 into equation 7 yields $\hat{p}(y|x) = p_{\theta_{t+1}}(y|x)$. In other words, θ_{t+1} learned from equation 5 is exactly the LLM parameter for our ideal opponent selection.

End-to-end Training Objective. We integrate the previously discussed two steps into a single end-to-end training objective with an update rule of $_{t+1}$. Specifically, plugging equation 8 into equation 5 arrives at the update rule $\theta_{t+1} = \arg\min_{\theta \in \Theta} L(\theta, \theta_t)$, where L is the training objective defined as follows

$$L(\theta, \theta_t) = _{x \sim q(\cdot), y \sim p_{\text{data}}(\cdot|x), y' \sim p_{\theta_t}(\cdot|x)} \left[\ell\left(\lambda \log \frac{p_\theta(y|x)}{p_{\theta_t}(y|x)} - \lambda \log \frac{p_\theta(y'|x)}{p_{\theta_t}(y'|x)} \right) \right]. \tag{10}$$

We summarize the iterative self-play process of our method as follows,

$$\ldots \quad \rightarrow \quad \underbrace{p_{\theta_t}(\cdot|x)}_{\text{Opponent Player at } t} \quad \rightarrow \quad \underbrace{\lambda \cdot \log \frac{p_{\theta_{t+1}}(\cdot|x)}{p_{\theta_t}(\cdot|x)}}_{\text{Main Player at } t+1} \quad \rightarrow \quad \underbrace{p_{\theta_{t+1}}(\cdot|)}_{\text{Opponent Player at } t+1} \quad \rightarrow \quad \ldots$$

Namely, the opponent player chosen from the previous iteration t is employed to train the main player at iteration $t+1$, resulting in the LLM parameterized by θ_{t+1}. Then we determine the next opponent player at iteration $t+1$ by directly copying the LLM parameter θ_{t+1}, which is then used in training the main player at iteration $t+2$.

8.2 Curriculum Learning

In deep learning, it has been observed that training models using data samples arranged in a strategically meaningful order can lead to improved performance compared to training on randomly shuffled data. This approach is commonly known as curriculum learning. Initial studies in curriculum learning introduced efficient algorithms that adhere to an 'easy-to-hard' progression . In the field of Natural Language Processing (NLP), criteria such as sentence length and term frequency are commonly utilized .More recent developments include the application of curriculum learning algorithms in multi-modal learning . Our work shares a similar idea to curriculum learning, wherein the training data evolves iteratively—beginning with responses that are easy to distinguish from human-annotated data and gradually progressing to more challenging instances.

8.3 Adversarial Attacks on LLMs

Optimization problem. Attacking LLM $f_\theta(\boldsymbol{x})$ constitutes a combinatorial optimization problem

$$\min_{\tilde{\boldsymbol{x}} \in \mathcal{G}(\boldsymbol{x})} \ell(f_\theta(\tilde{\boldsymbol{x}})) \tag{11}$$

with attack objective ℓ and set of permissible perturbations $\mathcal{G}(\boldsymbol{x})$. While there exist works that approach this optimization problem directly using, e.g., a genetic algorithm, many effective search-based attacks are guided by the gradient towards the one-hot vector representation $\nabla_{\tilde{\boldsymbol{X}}} \ell(f_\theta(\tilde{\boldsymbol{X}}))$ with differentiable objective ℓ.

Jailbreaking. Throughout the paper, we discuss "jailbreak" attacks as main example. For jailbreaking an LLMthe permissible perturbations $\mathcal{G}(\boldsymbol{x})$ allow arbitrarily choosing a substring of $\boldsymbol{x}$.

Specifically, $\tilde{\boldsymbol{x}} = \boldsymbol{x}' \,||\, \hat{\boldsymbol{x}} \,||\, \boldsymbol{y}'$ where $||$ denotes concatenation. $\boldsymbol{x}'$ is a fixed sequence of tokens that may consist of a system prompt and an (inappropriate) user request. $\hat{\boldsymbol{x}}$ is the part of the prompt that the attack may manipulate arbitrarily. We also refer to $\hat{\boldsymbol{x}}$ as the adversarial suffix. The attack objective ℓ is to construct $\hat{\boldsymbol{x}}$ s.t. the harmful response in $\boldsymbol{y}'$ becomes likely given $\boldsymbol{x}' \,||\, \hat{\boldsymbol{x}}$. We instantiate the objective using the cross entropy over the logits belonging to (part of) $\boldsymbol{y}'$.

It has been shown that our PGD is not only effective and flexible, but also efficient. Specifically, our PGD achieves the same effectiveness as the gradient-assisted search GCG with up to one order of magnitude lower time cost. Thus it emphasizes the importance of attacks with lower computational effort for large-scale evaluation or adversarial training. Moreover, using PGD for attacking LLMs may benefit, e.g., from the extensive research on adversarial robustness in other domains.

8.4 Data Subset Selection

Data subset selection or obtaining effective smaller subsets of data finds its use in a variety of applications. Consider deep models for example. While they demonstrate astounding improvements in accuracies on several downstream image, video or text tasks, they pose the following challenges: a) Increased training complexity and computational costs, b) Larger inference time, c) Larger experimental turn around times and difficulty in hyper-parameter tuning, and d) Higher costs and more time for labeling. Some of the ways past work has tried to address one or more of these challenges are through novel network architecture modifications, transfer learning, zero-shot learning , one-shot learning, activelearning, core sets and, in the context of this work, submodular functions, . Yet another application of subset selection is seen in extractive summarization of documents, images or videos wherein a good summary is modeled as an informative, non-redundant and diverse subset of the ground set. Naturally, several past works have leveraged submodular functions and submodular optimization to address document summarization, image collection sumamrization and video summarization.

More recently, it has been proposed to use parameterized submodular information measures to address the problem of *guided subset selection* and *guided summarization*. One application of guided subset selection is *targeted learning*, where the goal is to find subsets with rare classes or rare attributes on which the model is under-performing. Guided data subset selection need not be limited to guiding the subset to be *similar* to a target. Representation based functions attempt to directly model representation, in that they try to find a representative subset of items, akin to centroids and medoids in clustering. In this project,we have used facility location function (FL) is closely related to k-medoid clustering. It is defined as

$$f_{FL}(X) = \sum_{i\in} \max_{j\in X} s_{ij}$$

where i is an element from the ground set and s_{ij} measures the similarity between element i and element j. For each data point i in the ground set , we compute the representative from subset X which is closest to i and add these similarities for all data points. In a more generic setting, the set whose representation is desired (we call it represented set) may be different from the set whose subset is desired (i.e. the ground set). The expression for Facility-Location function in this generic setting then becomes

$$f_{FL}(X) = \sum_{i\in} \max_{j\in X} s_{ij}$$

.

Facility Location is monotone submodular. Note that the Facility Location function requires computing a $O(n^2)$ similarity function.

REFERENCES

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *ArXiv*, abs/2401.01335, 2024. URL https://api.semanticscholar.org/CorpusID:266725672.

Hadi Mohaghegh Dolatabadi, Sarah Monazam Erfani, and Christopher Leckie. Adversarial coreset selection for efficient robust training. *International Journal of Computer Vision*, 131:3307–3331, 2022. URL https://api.semanticscholar.org/CorpusID:252212064.

Simon Geisler, Tom Wollschlager, M. H. I. Abdalla, Johannes Gasteiger, and Stephan Gunnemann. Attacking large language models with projected gradient descent. *ArXiv*, abs/2402.09154, 2024. URL https://api.semanticscholar.org/CorpusID:267657696.

Vishal Kaushal, Ganesh Ramakrishnan, and Rishabh K. Iyer. Submodlib: A submodular optimization library. *ArXiv*, abs/2202.10680, 2022. URL https://api.semanticscholar.org/CorpusID:247025579.

Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. Pre-training a BERT with curriculum learning by increasing block-size of input text. In Ruslan Mitkov and Galia Angelova (eds.), *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pp. 989–996, Held Online, September 2021. INCOMA Ltd. URL https://aclanthology.org/2021.ranlp-1.112.