RepMatch: Quantifying Cross-Instance Similarities in Representation Space

Anonymous ACL submission

Abstract

Advancements in dataset analysis methods have led to the development of ways to analyze and categorize training data instances. These methods categorize the data based on specific features like "difficulty". We propose a framework that categorizes data from a viewpoint of similarity. This framework quantifies the similarities between subsets of training instances by comparing the models trained on them. This approach addresses the limitations of existing methodologies that focus on individual instances and are confined to single-dataset analyses. Our method enables the evaluation of similarities among arbitrary subsets of instances, facilitating both dataset-dataset and instance-dataset analyses. To compare two models efficiently, we leverage the Low-Rank Adaptation (LoRA) method. The effectiveness of our method has been validated across various NLP tasks, datasets, and models. The method can be used to compare datasets, find a smaller subset that outperforms a randomly selected subset of the same size, and successfully uncovers heuristics used in the construction of a challenge dataset.

1 Introduction

Contemporary machine learning models are deeply influenced by the datasets on which they are trained. The characteristics of a dataset, encompassing the quality and diversity of its instances, are critical in shaping a model's learning effectiveness and its capability to generalize. Recent advancements in the field have led to the development of methodologies that facilitate the analysis and categorization of data instances based on specific attributes, notably "difficulty" (Ethayarajh et al., 2022; Siddiqui et al., 2022; Swayamdipta et al., 2020), as well as other attributes, such as noisiness, atypicality, prototypicality, and distributional outliers (Siddiqui et al., 2022). These methodologies typically involve ranking or categorizing training instances according to these attributes, identifying specific types that may require specialized processing or treatment. One intuitive objective would be to identify mislabeled or noisy examples, pruning of which from the training data can lead to more effective training (Mirzasoleiman et al., 2020; Pleiss et al., 2020). Additionally, these studies are instrumental in analyzing and uncovering dataset artifacts (Gardner et al., 2021; Ethayarajh et al., 2022).

However, despite their contributions, these methods are often limited by their focus on individual instances without the capacity to evaluate subsets of data as a whole. Moreover, they are generally confined to analyses within a single dataset and lack the capability to conduct comparisons across different datasets or to perform comprehensive cross-dataset evaluations.

In response to these limitations, we introduce a novel framework that offers a refined lens for the analysis: quantifying the similarities between subsets of training instances from the perspective of the models trained on them. Specifically, we measure the similarity between two subsets, $\mathcal{S} \subseteq \mathcal{D}$ and $\mathcal{S}' \subseteq \mathcal{D}'$ of the training datasets $\mathcal{D}$ and $\mathcal{D}'$ (where $\mathcal{D}$ and $\mathcal{D}'$ could be the same dataset), by comparing the models trained exclusively on each subset. The subsets are deemed similar if the representation space learned by the model trained on $\mathcal{S}$ closely aligns with that learned by the model trained on $\mathcal{S}'$. This re-formulation addresses previous limitations by enabling the analysis and evaluation of similarities among arbitrary subsets of instances, from individual examples to entire datasets, from varied sources. Specifically, it facilitates:

- **Dataset-dataset analysis**: Compare similarities in task and dataset characteristics from a model's perspective, both within and beyond their original domains.
- **Instance-dataset analysis**: Identify the most "informative" instances for the target dataset

(or others), using which a more effective training can be performed. Similarly, identify those with the least information, suggesting out-of-distribution or noisy outliers.

The challenge lies in comparing two models, particularly within the context of modern, heavily parameterized models with expansive weight matrices. To constrain the set of trainable parameters, i.e., the updates in the representation space, we leverage low-rank adaptation Hu et al. (2021, LoRA). LoRA efficiently captures changes in a weight matrix through a low-rank matrix, primarily to expedite the fine-tuning process. Having all the significant training-induced changes captured in a low-rank matrix allows us to quantify the similarity of two models by a direct comparison of the corresponding changes in their low-rank representation spaces.

The efficacy of the proposed method has been validated through a set of experiments across different NLP tasks, datasets, and models. The results demonstrate that LoRA matrices exhibit significant similarities across similar tasks. Additionally, for each model, a compact yet informative subset within any dataset has been identified; models trained on this subset consistently outperform those trained on a comparably sized random subset. In a definitive demonstration of cross-dataset utility, the approach successfully uncovers heuristics used in the automatic construction of a challenge dataset (HANS).

2 Related Work

The process of quantifying the similarity between two distinct datasets is a thoroughly researched topic. The theoretical concept of data similarity can be linked to the traditional KL-divergence (Kullback and Leibler, 1951), a non-symmetric measure that quantifies the dissimilarity between two probability distributions. For 'shallow' datasets, empirical measures such as the Maximal Mean Discrepancy (MMD) (Borgwardt et al., 2006) are often employed. This measure compares the means of samples drawn from two distributions in a high-dimensional feature space.

Tran et al. (2019) utilized an information-theoretic approach to estimate task difficulty, demonstrating a strong correlation between their introduced hardness measure and empirical hardness and transferability. Alvarez-Melis and Fusi (2020) proposed a distance measure to quantify similarity between datasets, thereby assessing transfer learning hardness.

In a more empirical setting, Hwang et al. (2020) presented a method to predict inter-dataset similarity using a set of pre-trained autoencoder. Their approach involves inputting unknown data samples into these pre-trained autoencoders and evaluating the difference between the reconstructed output samples and their original input samples. Our method, while empirical, requires no additional computation beyond regular fine-tuning by LoRA and is robust to the randomness of the training environment.

Our method does not impose constraints on the size of the subsets it compares, thus it can be categorized under data selection research. Swayamdipta et al. (2020) used training dynamics to divide a dataset into subsets of easy-to-learn, hard-to-learn, and ambiguous instances. However, their method has limitations in analyzing individual instances or performing cross-dataset analysis.

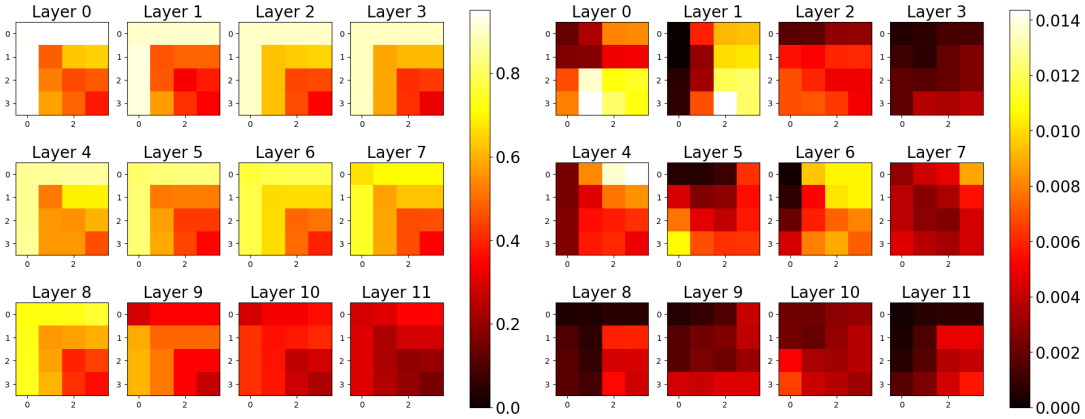
A stream of prior research has aimed to find a subset of training examples that achieves close performance to training on the full dataset by using gradient information (Mirzasoleiman et al., 2020; Wang et al., 2021; Yu et al., 2020; Killamsetty et al., 2021). Recently, Xia et al. (2024) proposed a method to estimate the influence function of a training data point to identify influential data in an instruction tuning setting.

Ethayarajh et al. (2022) presented a metric that quantifies the complexity of individual instances relative to a specific distribution, which is useful for comparing datasets or subsets within a single dataset. Our approach can be extended to facilitate comparisons of instances or segments across different datasets.

3 Methodology

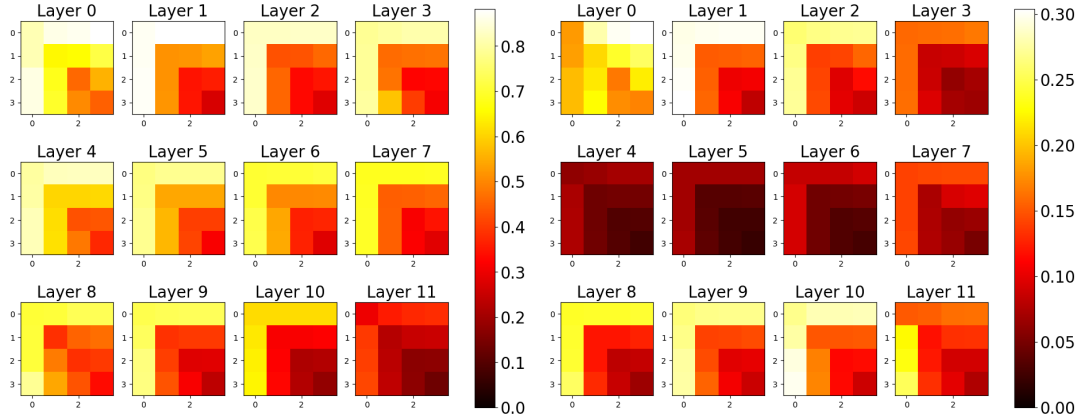
We introduce a method designed to assess the similarity between subsets of data instances, where subsets can be anything from individual instances to entire datasets. We define two subsets, $\mathcal{S}_1$ and $\mathcal{S}_2$, as similar if a model trained on $\mathcal{S}_1$ (denoted as $\mathcal{M}_{\mathcal{S}_1}$) exhibits a representation space akin to that of a model trained on $\mathcal{S}_2$ ($\mathcal{M}_{\mathcal{S}_2}$).

During standard fine-tuning, alterations to a specific weight matrix $\mathcal{W}_i^j$ (the j^{th} weight matrix in layer i) are captured by $\Delta\mathcal{W}_i^j$, also known as the adaptation matrix. After fine-tuning, the updated model weights are then represented as



(a) (b)

Figure 1: Grassmann distance of LoRA matrices for each layer of two BERT$_{base}$ models fine-tuned on SST-2 but (a) with different seeds, and (b) with random baseline (axes are i and j of the corresponding Grassmann distance).



(a) (b)

Figure 2: Grassmann distance of LoRA matrices for each layer of two BERT$_{base}$ models fine-tuned on (a) a single training instance of SST-2 with different seeds, and (b) the instance from part a and another random instance (axis are i and j in Grassmann distance).

$\hat{\mathcal{W}}_i^j = \mathcal{W}_i^j + \Delta\mathcal{W}_i^j$. These adaptation matrices are responsible for extracting task-specific features from the input and incorporating them into the pre-trained weight matrices. Since the pre-trained weights $\mathcal{W}_i^j$ remain constant across both models, comparing the representation spaces of $\mathcal{M}_{\mathcal{S}_1}$ and $\mathcal{M}_{\mathcal{S}_2}$ effectively boils down to analyzing the differences in $\Delta\mathcal{W}_i^j$.

The challenge in comparing models arises from the substantial size and high dimensionality of the weight matrices, particularly in modern language models. To manage this complexity, we propose using the LoRA method to encapsulate the $\Delta\mathcal{W}_i^j$ matrices in a low-rank format. In the following sections, we will provide a brief introduction to the LoRA method and explain how it facilitates the comparison of adaptation matrices between models.

3.1 Background: LoRA

In the Low-rank Adaptation (Hu et al., 2021, LoRA) method, a model $\mathcal{M}$ with a pre-trained weight matrix $\mathcal{W}_i^j$ is efficiently fine-tuned on a specific dataset. the goal in LoRA is to efficiently fine-tune the model on a given dataset. LoRA achieves this by keeping the pre-trained weights ($\mathcal{W}_i^j$) frozen and allowing only the injected low-reank matrices, $\Delta\mathcal{W}_i^j$, to be updated during the fine-tuning process. To ensure parameter efficiency, LoRA restricts these $\Delta\mathcal{W}$ matrices to be low-rank. Specifically, if $\mathcal{W}_i^j$ is a $d \times d$ matrix, instead of updating this full-rank matrix directly, LoRA introduces two low-rank matrices $(\mathcal{A}_i^j)_{d\times r}$ and $(\mathcal{B}_i^j)_{r\times d}$ for each layer j. The product $\mathcal{A}_i^j\mathcal{B}_i^j$ then forms the adaptation matrix $\Delta\mathcal{W}_i^j$. While $\mathcal{W}_i^j$ retains the dimensions $d \times d$, its rank is limited to r, where $r << d$, effectively reducing the number of parameters from d^2 to $2rd$. The authors of LoRA demonstrated that

Dataset	SST-2	SST-5	IMDB	MNLI	SNLI	SQuAD
Random	68.1	35.3	69.4	56.2	55.9	24.3
RepMatch	83.4	39.5	83.9	60.5	60.7	25.1

Table 1: The accuracy of BERT on different datasets when fine-tuned on a subset of 100 instances selected randomly (Random) or based on the highest RepMatch scores (RepMatch). For SQuAD, F1 score have been used since only F1 and EM (Exact Match) are used as a metric for this task.

setting r to be significantly smaller than d does not generally result in substantial performance degradation across most NLP tasks *(interestingly, they observed that in some cases, the performance of the model actually improved).

3.2 Constraining Model Updates using LoRA

While maintaining the pre-trained weights frozen, we follow Hu et al. (2021) and apply LoRA matrices specifically to attention matrix updates. Here, $\Delta\mathcal{W}_i^j$ is formed by the product of $\mathcal{A}_i^j$ and $\mathcal{B}_i^j$–the LoRA matrices. Given the multiplicity of possible $\mathcal{A}$ and $\mathcal{B}$ combinations that can yield the same $\Delta\mathcal{W}_i^j$, our focus remains solely on their resultant product rather than the individual matrices.

The low-rank nature of the $\Delta\mathcal{W}_i^j$ matrices in LoRA facilitates the efficient comparison of models. Since models trained on similar tasks are expected to extract analogous features, the LoRA matrices associated with a consistent pre-trained model should display similarities across comparable tasks and datasets. This insight drives our proposal to use these task-specific features, as identified by LoRA, to analyze both datasets and individual data instances.

Models $\mathcal{M}_{\mathcal{S}_1}$ and $\mathcal{M}_{\mathcal{S}_2}$ are considered representationally similar if their corresponding LoRA matrices exhibit resemblance. Specifically, we compare the changes in the weight matrices, $\Delta\mathcal{W}_i^j(\mathcal{M}_{\mathcal{S}_1})$ and $\Delta\mathcal{W}_i^j(\mathcal{M}_{\mathcal{S}_2})$, across each layer i and for each weight matrix type j within the set query, key, value, output. This method allows us to assess the similarity in their representation spaces by examining the modifications captured in these matrices.

3.3 Computing RepMatch

To quantify the similarity of the subspaces formed by the two corresponding matrices from $\mathcal{M}_{\mathcal{S}_1}$ and $\mathcal{M}_{\mathcal{S}_2}$, we adopt the Grassmann distance. Hu et al. (2021) used the distance to discern subspace similarities across different ranks within the same dataset, in order to verify the efficacy of low-rank matrices. In contrast, we leverage the distance to measure similarities across varying datasets and tasks. The overall similarity score, i.e., RepMatch, is computed as the average similarity across all matrices.

3.3.1 Grassmann Similarity

Given two matrices $\mathcal{W}_r$ and $\mathcal{W}'_r$, the Grassmann distance computes the similarity (distance) between the subspaces they form as follows:

$$\phi(\mathcal{W}_r, \mathcal{W}_{r'}, i, j) = \frac{||U^{i\top}_{\mathcal{W}_r} U^j_{\mathcal{W}_{r'}}||^2_F}{\min(i,j)} \in [0,1] \quad (1)$$

where both $\mathcal{W}_r$ and $\mathcal{W}_{r'}$ are $d \times d$ matrices. The matrix U is usually taken as the right singular unitary matrix, although the same can be achieved with left unitary matrices.

A high similarity implies that the subspace formed by the matrix of rank r should predominantly reside within the subspace formed by $\mathcal{W}_{r'}$. The matrices denoted by U can be interpreted as facilitating a change of basis. When these subspaces are in close proximity, the product of their corresponding U matrices tends toward unity, indicating a high degree of similarity between the subspaces. This proximity of subspaces is quantitatively expressed by the Grassmann distance, which approaches zero as the alignment between the subspaces decrease.

4 Analysis Possibilities using RepMatch

The RepMatch similarity metric is unconstrained by the size or origin of the subsets, thus facilitating its application in a multitude of scenarios. For instance, it enables comparisons between individual instances and an entire dataset, or between subsets from distinct datasets. In the following sections, we demonstrate the reliability of this method for both dataset-level and instance-level analyses. To establish this, it is necessary to show that RepMatch is robust against the stochastic nature of the

training environment. Specifically, alterations in the training seed should not significantly affect the similarity score.

4.1 Dataset-level Analysis

For dataset-level analysis, we consider the scenario where two identical models are fine-tuned on the same dataset under the same conditions, with the only difference being the random seed. We would expect these models to exhibit very similar characteristics. Figure 1a shows the Grassmann distance between the changes in the value matrix, $\Delta\mathcal{W}_i^{value}$, for each layer i of two BERT$_{base}$ models fine-tuned on the SST-2 sentiment analysis dataset using different seeds. Notably, there exists at least one vector in the corresponding matrix of each model that closely resembles its counterpart. We set the rank of LoRA to 4 and applied it specifically to the query and value matrices, which also demonstrated similar patterns.

To demonstrate that the observed similarity is not due to chance, Figure 1b presents a random baseline for comparison. This figure compares the LoRA matrices of the fine-tuned model with those of the same model, but with 10% of its entries shuffled. This alteration creates a matrix that, while not drastically different, is distinct from one generated through a standard fine-tuning process. The analysis shown in the figure reveals that the highest similarity score across different seeds exceeds 0.8. In contrast, for the baseline, the similarity score falls below 0.02, highlighting a significant difference.

The Grassmann distance yields an $r \times r$ table delineating the similarity between any subspaces of the two matrices of rank r. To make the RepMatch produce a ranking, we only need one number, selecting the maximum as a representative of the utmost similarity.

Figure 1 also indicates that the greatest similarity typically manifests in a single vector within each matrix. Consequently, setting the rank of LoRA matrices to one incurs minimal data loss. This is supported by the findings of Hu et al. (2021), which suggest that employing LoRA at a rank of one negligibly affects the model's efficacy across many NLP tasks. For these reasons, and to efficiently compute the Grassmann distance, we opted for a rank of one in our experiments detailed in Section 5.

4.2 Instance-level Analysis

We have established that subset size imposes no limitations, thereby enabling the identification of particularly informative instances. We call an instance more *informative* if the RepMatch between that single instance and the whole dataset is higher than another instance. Figure 2 affirms the method's reliability at the instance-level. Figure 2a displays the Grassmann distance for two BERT$_{base}$ models fine-tuned on a randomly selected instance from SST-2 with two different seeds. Moreover, Figure 2b depicts the Grassmann distance between two BERT$_{base}$models: one fine-tuned on the aforementioned instance and the other on a different random instance from the dataset (selected 10 random instances from the dataset, the figure highlights the most analogous one).

It is evident that the Grassmann distance for two models trained on the same instance but with different seeds is above 0.8, while for disparate instances, it hovers around 0.2. Therefore, RepMatch can be confidently employed in various contexts.

Section 5 offers empirical evidence supporting our proposed method. Our findings reveal that datasets related to specific tasks exhibit LoRA matrices with significant similarities, which are distinctly different from those associated with unrelated tasks. Furthermore, our methodology effectively isolates a compact subset of *informative* instances with the highest RepMatch scores within a dataset. Notably, a model trained on this curated subset consistently outperforms one trained on a randomly selected subset of the same size. Additionally, our approach proves versatile, capable of being applied across various datasets to identify heuristic patterns.

5 Experiments

5.1 Experimental Setup

Datasets. To demonstrates the adaptability of our methodology to various scenarios, we experimented with five dataset across two tasks: sentiment analysis (SST-2, SST-5, and IMDB) and textual entailment (MNLI and SNLI). The method's efficacy was further assessed on the SQuAD v1 dataset for question answering.

Models and hyperparameters. While the majority of experiments were conducted on BERT$_{base}$, with additional trials on LLaMA2-7B to verify that

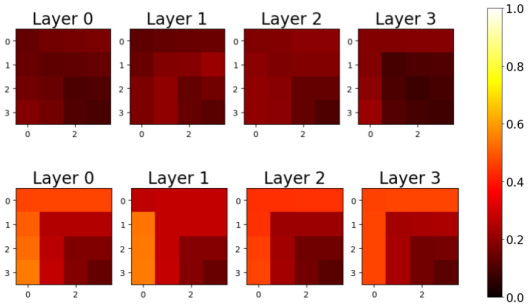


Figure 3: Grassmann distance of LoRA matrices for two BERT$_{base}$ models. The first row compares the first four layers of a model trained on SST-2 and the other trained on IMDB. The second row, compares SST-2 and SST-5. The SST datasets are more similar as expected (axis are i and j in Grassmann distance).

our findings are robust across different models.[1] Both of these models were sourced from Hugging Face. Unless specified otherwise, our default fine-tuning setup involves integrating LoRA modules exclusively to the query and value matrices, while keeping all other model weights frozen. We employed a batch size of 40, conducting 10 epochs for sentiment analysis tasks and 5 epochs for other tasks. The rank of the LoRA matrices was set at one. For dataset-level analysis, we used a learning rate of 10^{-5}, while instance-level experiments were conducted with a learning rate of 10^{-3} for speedup. Due to limited resources, no hyperparameter tuning was done for any of the settings.

5.2 Dataset-level Similarity

In Section 3, we presented heatmaps to illustrate the similarities between subspaces created by the value matrix of a model trained on the same dataset but with differing training seeds. Also, we argued that tasks of a similar nature should exhibit comparable LoRA matrices. To substantiate this claim, experiments were conducted demonstrating that representation similarities between two datasets from the same task are greater than those from different tasks.

As depicted in Figure 3, the similarity between the SST-2 and IMDB datasets is quantified by a RepMatch score of approximately 0.3 across each layer. While additional heatmaps for other datasets are included in the appendix, only their RepMatch scores are reported here. The RepMatch between SST-2 and SST-5 is roughly 0.45 at each layer, aligning with expectations of higher similarity compared to the IMDB dataset. In contrast, the RepMatch between SST-2 and MNLI is around 0.1, indicative of their distinct task natures. Notably, this score is still significantly higher than that of a random matrix, which has a RepMatch of 0.02 as detailed in Section 3. Finally, the RepMatch between the SNLI and MNLI datasets stands at about 0.2, suggesting a closer relationship than with SST-2, yet highlighting considerable differences.

The argument presented is that the notably low random baseline can be attributed to the high dimensionality of the matrices. Where even a minimal random shuffling, such as 10%, could drastically alter the space, resulting in almost no similarities. Consequently, it is concluded that these low-rank matrices encode valuable task-related features, which facilitate the comparison of subsets of instances.

5.3 Instance-level Similarity

We propose utilizing RepMatch for instance-level analysis , where RepMatch is calculated between an instance $x \in \mathcal{X}$ and the entire training set $\mathcal{X}$ to identify instances with representations closely resembling the final model.

In this experiment, we selected 100 samples with the highest RepMatch scores to fine-tune a model, comparing its performance against another model fine-tuned on 100 randomly selected samples. The results demonstrated that the model trained on the 100 most representative samples consistently outperformed the randomly selected sample model, as detailed in Table 1.

The process for calculating individual RepMatch scores involves running a pre-trained model with a batch size of one to update the LoRA matrices. This model is then compared to a model previously fine-tuned on the entire dataset. To ensure a fair comparison, the model is reset to its original pre-trained state before processing each subsequent instance. It's noted that fine-tuning on 100 instances was performed without LoRA, as detailed in Table 1. However, the experiment was also replicated with LoRA, resulting in a performance decline of 3 to 5 percent for both Random and RepMatch groups, yet the gap between them largely remained the same.

The comparisons between random selection and selection based on the highest score subset of Rep-

[1]Due to limited access to GPUs, we were constrained in our ability to test additional models, datasets, and configurations.

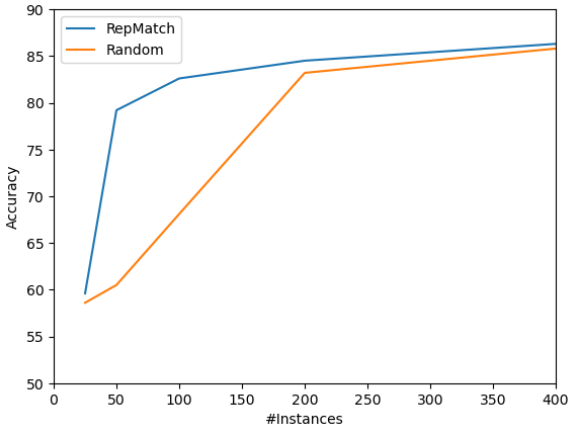


Figure 4: This figure illustrates the performance variation of a BERT$_{base}$ model, fine-tuned on different subset sizes of SST-2. The blue line represents the scenario where the subset is selected using the RepMatch method, while the other line corresponds to a randomly chosen subset. Evidently, the RepMatch method has successfully identified a smaller subset that achieves superior performance compared to a random selection.

Match, have been conducted using 100 instances. Additionally,experiments were carried out with varying subset sizes using the SST-2 and BERT$_{base}$ model. As depicted in Figure 4, a subset smaller than 400, selected using RepMatch, consistently outperforms a randomly selected subset of the same size, although the performance gap decreases.

According to the table, SST-2 and IMDB exhibit the most significant gap. We attribute this to the limited matrix rank, which might be less restricting for simpler tasks. Increasing the matrix rank could potentially enhance this disparity across other datasets, albeit possibly hitting a performance ceiling.

Additionally, the methodology was tested using the LLaMA2-7B model on the SST-5 dataset with LoRA, resulting in performance scores of 30% for the Random group and 34% for the RepMatch group. This test, notably time-intensive due to the model's complexity, was conducted to validate that the effectiveness of the RepMatch method is consistent across different models.

5.3.1 Detecting out-of-distribution instances

The RepMatch method has no limitations on the size or domain of the considered set, thus making it applicable in various analytical contexts. To demonstrate this, an experiment was designed to showcase the cross-dataset capabilities of the method. Specifically, we opted for detecting out-of-distribution instances. Previous studies have identified certain superficial artifacts in widely used textual entailment datasets, such as MNLI and SNLI (Rajaee et al. (2022) inter alia). Models often leverage these artifacts (which usually arise as a result of decisions made during dataset construction) to achieve high performance without truly learning the task. One such artifact in textual entailment datasets is that high overlap between the premise and the hypothesis is likely indicative of an entailment label.

To address this issue, challenge sets like HANS (McCoy et al., 2019) were created to test the models' genuine understanding of the task. This dataset includes examples that counter the heuristics in the NLI datasets. For instance, in the case of overlap bias, a high overlap between the premise and the hypothesis results in a non-entailment label in HANS, contrasting with MNLI and SNLI. Henceforth, we will refer to these two datasets as NLI datasets.

We hypothesize that non-entailment instances in the training set of NLI datasets with high overlaps will be more similar to the HANS dataset than other instances. To validate our hypothesis, we leveraged our instance-level analysis setting. The only difference is that here we measure the similarity across datasets, i.e., between each instance of the NLI datasets and the entire set of HANS instances (rather than to the dataset from which they originate).

To this end, we extracted three sets from each NLI dataset, all with non-entailment labels but varying in the overlap between the premise and hypothesis. All instances in the first set have full overlap, the second set have overlap between 60% and 80%, and the third set have no overlap. We then calculated the RepMatch using BERT$_{base}$ for each instance with respect to a model trained on HANS. Table 2 shows the number of instances in each set. As the number of instances in each set varies, we selected 300 samples randomly (without replacement) from each overlap subset for a fair comparison. We then took the average RepMatch score for the 300 instances in the newly selected subset. We repeated the experiment multiple times for different subsets and reported the average of all experiments in table 3. For a clearer comparison, all scores were multiplied by $\frac{1}{\text{learning rate}}$, which does not affect the comparison since we are comparing the numbers.

Dataset	Full	Mid	No
MNLI	1,016	43K	8,600
SNLI	940	53K	900

Table 2: The number of instances in each set. The different sets are extracted from NLI datasets based on the degree of overlap between the premise and the hypothesis. The "Full" set encompasses instances with full overlap, the "Mid" set contains instances where the overlap between the premise and the hypothesis ranges from 60% to 80%, and the "No" set, as the name suggests, includes instances where there is no overlap. All sets have non-entailment label.

Dataset	Full	Mid	No
MNLI	37	13	8
SNLI	24	7	9

Table 3: The average RepMatch score calculated for 10 subsets, each randomly selected and consisting of 300 instances from the corresponding set. For every instance within a set, the RepMatch score is computed using BERT_{base} in relation to the HANS dataset.

As expected, the set containing full overlap instances with non-entailment labels showed the highest average similarity to the HANS dataset, suggesting similarities between the two. This demonstrates that the RepMatch method can be used to find or analyze bias or heuristics with respect to another dataset, which could be useful for out-of-distribution generalization purposes

6 Conclusion

In this study, we approached the problem of dataset analysis from a unique perspective. We proposed a method to identify similarities between subsets of training instances by examining the similarities within the representation space of models trained on different subsets. We overcame the challenges of complexity and heavy parameters of language models by utilizing the LoRA method to constrain changes in the representation space.

Our findings suggest that RepMatch can be employed to compare similar tasks and datasets, conduct instance-level analysis to discover heuristics in a dataset, and perform subset analysis to identify a smaller subset that achieves reasonable performance and outperforms a randomly selected subset of the same size. The experiments demonstrated that the proposed method can be utilized in a variety of situations and is not limited by the size of the subset or its domain.

Limitations

In the instance-level setting, the relationship between instances within a training batch is not taken into account. There exists a possibility that a model might exhibit better performance when trained with two less informative instances in a batch, rather than two highly similar ones. This presents a potential avenue for enhancing the experimental setup. Furthermore, while we demonstrated that the entire dataset and individual instances are robust to the random seed of the training environment, the randomness of training and instances in a batch can have a non-negligible effect.

The majority of our experiments were conducted on BERT_{base}, with one experiment on LLaMA2. Due to GPU limitations, further experiments were not viable. Although our focus was on Transformer models with a textual modality and our evaluations were based on three different classification tasks, we believe this method is applicable to other modalities and settings

References

David Alvarez-Melis and Nicolò Fusi. 2020. Geometric dataset distances via optimal transport.

Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans-Peter Kriegel, Bernhard Scholkopf, and Alex Smola. 2006. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22 14:e49–57.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information.

Matt Gardner, William Merrill, Jesse Dodge, Matthew E. Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. 2021. Competency problems: On finding and removing artifacts in language data.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Inseok Hwang, Jinho Lee, Frank Liu, and Minsik Cho. 2020. Simex: Express prediction of inter-dataset similarity by a fleet of autoencoders.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. 2021. Grad-match: Gradient matching based data subset selection for efficient deep model training.

Solomon Kullback and R. A. Leibler. 1951. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. 2020. Coresets for data-efficient training of machine learning models.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q. Weinberger. 2020. Identifying mislabeled data using the area under the margin ranking. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Sara Rajaee, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. Looking at the overlooked: An analysis on the word-overlap bias in natural language inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10605–10616, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shoaib Ahmed Siddiqui, Nitarshan Rajkumar, Tegan Maharaj, David Krueger, and Sara Hooker. 2022. Metadata archaeology: Unearthing data subsets by leveraging training dynamics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Anh T. Tran, Cuong V. Nguyen, and Tal Hassner. 2019. Transferability and hardness of supervised classification tasks.

Xinyi Wang, Hieu Pham, Paul Michel, Antonios Anastasopoulos, Jaime Carbonell, and Graham Neubig. 2021. Optimizing data usage via differentiable rewards.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning.