Low-Confidence Gold: Refining Low-Confidence Samples for Efficient Instruction Tuning

Hongyi Cai[1,*], Jie Li[2,*], Mohammad Mahdinur Rahman[1], Wenzhen Dong[3]
[1]Universiti Malaya,
[2]University of Science and Technology Beijing, [3]The Chinese University of Hong Kong
[*]Corresponding author.
Correspondence: xcloudfance@gmail.com

Abstract

The effectiveness of instruction fine-tuning for Large Language Models is fundamentally constrained by the quality and efficiency of training datasets. This work introduces Low-Confidence Gold (LCG), a novel filtering framework that employs centroid-based clustering and confidence-guided selection for identifying valuable instruction pairs. Through a semi-supervised approach using a lightweight classifier trained on representative samples, LCG curates high-quality subsets while preserving data diversity. Experimental evaluation demonstrates that models fine-tuned on LCG-filtered subsets of 6K samples achieve superior performance compared to existing methods, with substantial improvements on MT-bench and consistent gains across comprehensive evaluation metrics. The framework's efficacy while maintaining model performance establishes a promising result for efficient instruction tuning.

1 Introduction

Large Language Models (LLMs) have been trained to follow instructions by specific supervised response data after pre-training stage. Many instruction finetuning (IFT) (Taori et al., 2023) datasets emerge to realize various downstream tasks, for example: mathematic calculation, sentence analysis, haiku writing and etc, aiming to strengthen the ability of LLMs in instruction following. To save vast human costs for data annotation, most of studies introduce other teacher LLMs (e.g. `text-davinci-003` (Brown et al., 2020)) to align the best instructions with corresponding responses.

However, IFT datasets (e.g. `Alpaca_52k` (Taori et al., 2023), `magpie` (Xu et al., 2024)) suffer from misleading content and poor quality, resulting in the bottleneck of post-training performance, even though teacher models replenish the missing parts of context and instruction pairs. This highlights the need for effective data filtering methods that identify high-quality instruction subsets while reducing fine-tuning time and computational costs.

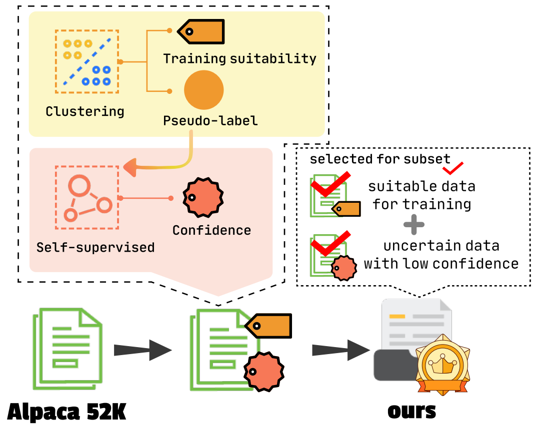

Figure 1: We target to select complex and quality samples confidence ranking for benefiting LLM training.

`Alpagasus` (Chen et al., 2024) proposed a model-based approach that introduces proprietary LLMs to score data quality in multiple facets, replacing human annotation by taking advantage of the automated pipeline. However, this leads to datasets that are likely biased by the preference for redundant and limited responses (Panickssery et al., 2024; Cai et al., 2025), which potentially deteriorates the diversity of the original data. Ge et al., 2024 emphasizes the necessity of diversity and therefore proposed clustering and ranking to select subsets of data. Further, Superfiltering (Li et al., 2024) gains more insights in small open-source LLM that scores the instruction following ability of `Alpaca_52k`. Although the instruction score provides an efficient and simple criterion for data selection, it does not consistently correlate with both the quality and diversity of data. Consequently, improvements in performance may not always be guaranteed.

To address these challenges, we propose a novel data filtering framework, **Low-Confidence Gold (LCG)** for efficient instruction tuning that signif-

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 8233–8240
November 4-9, 2025 ©2025 Association for Computational Linguistics

icantly reduces computational costs while maintaining model performance. Our approach, shown in 1, innovatively seeks to identify high-value instruction data through classification tasks. Specifically, we develop a lightweight classification model trained on centroid subsets that effectively categorizes instruction-response pairs, and leverage low-confidence predictions to curate challenging examples most beneficial for instruction tuning. Another perspective is that, since the common instruction tuning data are lack of annotations and labels, we adopt the manner of semi-supervised learning, to construct pseudo-labels as our training groundtruth, as well as getting inspired quality data from affordable yet effective models.

Through extensive experiments on the `Alpaca_52K dataset`, we demonstrate that our filtered subsets achieve comparable or better performance when fine-tuning various open-source language models, while requiring only a fraction of the original data. Our main contributions are threefold:

1. A novel and efficient data filtering paradigm for instruction tuning that combines nearest neighbor classification with confidence-based selection.
2. We train a small classifier model that enables selection for the whole set of instruction fine-tuning data.
3. Experiments and evaluations are conducted that demonstrate the outstanding effectiveness of our filtered datasets working on multiple open-source LLMs. We reach **states-of-the-arts performance** in MT-Bench and HuggingFace OpenLLM Leaderboard benchmarks.

2 Preliminaries

2.1 K-means Clustering

Given the Alpaca_52k dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ where $N = 52,000$, we first cluster instructions into K semantic groups using K-means. Let $\phi(x_i) \in \mathbb{R}^d$ denote the embedding vector of instruction x_i. The clustering objective minimizes:

$$\min_{\{C_k\}_{k=1}^K} \sum_{k=1}^{K} \sum_{x_i \in C_k} \|\phi(x_i) - \mu_k\|^2 \tag{1}$$

where $\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} \phi(x_i)$ is the centroid of cluster C_k. This partitions $\mathcal{D}$ into K disjoint subsets $\{C_1, ..., C_K\}$ based on instruction similarity.

2.2 Problem Setting

Our filtering framework, LCG aims to select a subset $\mathcal{D}_{\text{filtered}} \subseteq \mathcal{D}$ that satisfies:

$$\mathcal{D}_{\text{filtered}} = \bigcup_{k=1}^{K} \{(x_j, y_j) \in C_k \mid \mathcal{F}(x_j, y_j) < \tau_k\} \tag{2}$$

where $\mathcal{F} : \mathcal{D} \to [0, 1]$ is a discriminative confidence scorer and τ_k is an adaptive threshold for cluster C_k. The scorer $\mathcal{F}$ evaluates how "hard" a sample is to be trivially categorized, with higher values indicating the simplicity of data which is easily determined and differentiated. The training efficiency therefore increases since only a small subset of instructions are curated.

3 Methodology

3.1 Motivation

Instruction filtering demands a dual-focus mechanism that intrinsically balances data quality and diversity. Traditional supervised methods face inherent scalability limitations as manual annotation becomes prohibitively expensive for large-scale instruction datasets (Liu et al., 2022; Longpre et al., 2023; Liu et al., 2023). Meanwhile, it is difficult to identify suitable and challenging data for LLMs training without introducing proprietary LLMs or labors. Our semi-supervised framework addresses these limitations through pseudo-label refinement and early-stopped confidence detection, creating dynamic selection boundaries aligned with language model learning dynamics.

Cluster-centric pseudo-labeling addresses data distribution challenges in instruction tuning. Traditional sampling methods often struggle to balance between common and rare instruction patterns, leading to either over-representation of frequent cases or loss of valuable rare examples. We create semantic clustering anchors that naturally preserve the diversity of instruction patterns. By sampling 3% of data points nearest to cluster centroids, we ensure each semantic category contributes meaningful examples while maintaining the inherent data distribution characteristics.

Early-stopped classifier training induces uncertainty to identify high-quality samples. Limiting the classifier to 3 epochs creates deliberate

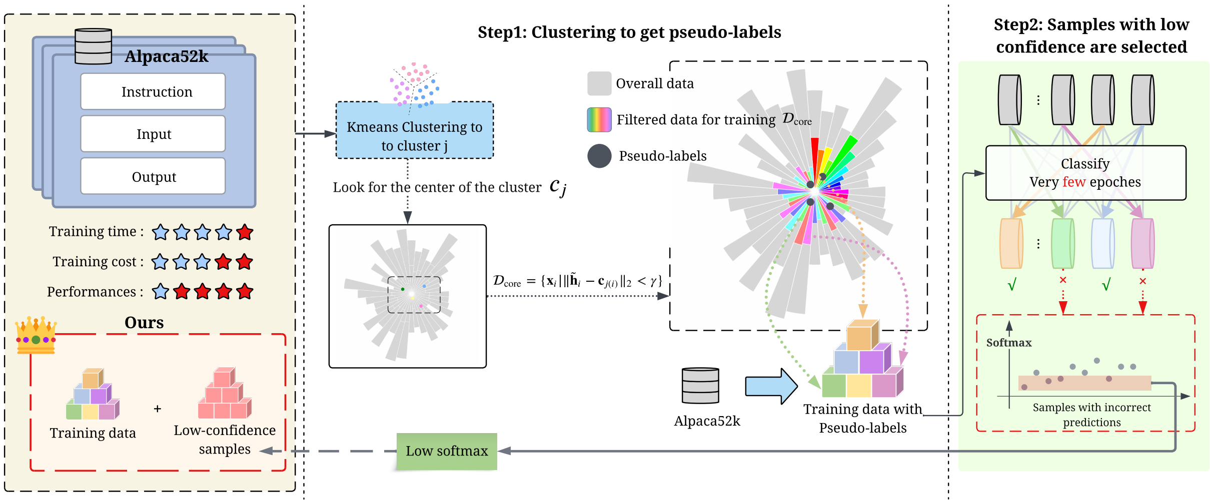

Figure 2: The overall pipeline of Low-Confidence Gold. We split our pipeline into two main steps: 1) Clustering to get pseudo-labels and centroid data to collect the initial diversity of data. 2) We feed annotated data into a tiny yet effective classifier to rank the confidences for the rest of the distant data to implement subset selection.

underfitting - the model develops basic pattern recognition without over-specializing to pseudo-labels. When applied to non-centroid samples, this partially-trained classifier's low-confidence predictions signal instructions containing non-trivial semantic constructs. These samples challenge the classifier's emerging decision boundaries precisely because they contain valuable complexity that language models should master, not avoid.

3.2 Centroid Coreset Selection for Pseudo-labels

In the initial step of our approach, we select a coreset from the whole corpus to identify pseudo-labels by the K-means algorithm, which effectively determine each semantic clusters. Given a dataset of instruction pairs $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, we first encode each instruction $\mathbf{x}_i$ into a dense vector representation using MiniLM (Wang et al., 2022):

$$\mathbf{h}_i = \text{AvgPool}(\text{MiniLM}(\mathbf{x}_i)) \in \mathbb{R}^{384} \quad (3)$$

This geometric progression ensures proportional coverage of both frequent and rare instruction patterns. Cluster centroids $\{\mathbf{c}_j\}_{j=1}^k$ are computed via:

$$\mathbf{c}_j = \frac{1}{|\mathcal{C}_j|} \sum_{\mathbf{x}_i \in \mathcal{C}_j} \tilde{\mathbf{h}}_i \quad (4)$$

where $\mathcal{C}_j$ denotes the set of samples assigned to cluster j. Centroid-proximal samples are selected as high-confidence candidates:

$$\mathcal{D}_{\text{core}} = \{\mathbf{x}_i \mid \|\tilde{\mathbf{h}}_i - \mathbf{c}_{j(i)}\|_2 < \gamma\} \quad (5)$$

where γ is the 90th percentile distance within each cluster.

3.3 Low-Confidence Gold: Calibrating with Low-confidence samples to select data

After determining pseudo-labels based on clusters, those annotations can be served for classification training. Specifically, we train a multi-class classifier on the core samples $\mathcal{D}_{\text{core}}$. The model architecture consists of:

$$f_\theta(\mathbf{x}) = \text{Softmax}(\mathbf{W}_2 \cdot \text{GELU}(\mathbf{W}_1 \mathbf{h}_i + \mathbf{b}_1) + \mathbf{b}_2) \quad (6)$$

where $\mathbf{W}_1 \in \mathbb{R}^{384\times768}$, $\mathbf{W}_2 \in \mathbb{R}^{768}$ are learnable parameters, and GELU denotes the Gaussian Error Linear Unit activation. The model optimizes cross-entropy loss:

$$\mathcal{L}(\theta) = -\frac{1}{|\mathcal{D}_{\text{core}}|} \sum_{(\mathbf{x}_i, y_i)} \sum_{j=1}^{k} \mathbb{I}(y_i = j) \cdot \log p_\theta(y = j|\mathbf{x}_i) \quad (7)$$

Training terminates at epoch $T = 3$ since we aim to keep the model in an early-stopped stage so that they would not overfit to the centroid subset data. After training, we rank the confidence distribution calculated from *softmax* function and select the top K most uncertain data in each cluster.

Model	MT-bench	Huggingface Open LLM Leaderboard Scores (%)				
	Score	Hellaswag	MMLU	GSM8k	ARC	Avg
First Group - Base Model						
Mistral-7b-v0.3	3.639	60.94	58.96	36.62	48.81	51.33
First Group - Methods						
Alpaca-52k	4.018	61.18	57.73	31.61	53.07	50.90
SuperFiltering-10%	3.963	60.98	59.34	35.71	49.83	51.47
Random-6k	4.314	60.83	58.75	35.03	53.07	51.92
Perplexity-6k	4.352	61.64	58.48	37.00	51.88	52.25
Kmeans-6k	4.283	60.86	58.45	35.10	52.05	51.62
LIMA-6k	4.440	60.58	59.34	37.31	51.11	52.09
LCG-MultinomialNB-6k (Ours)	**5.086**	**62.00**	**59.51**	**40.51**	**52.90**	**53.73**
LCG-DistilBERT-6k (Ours)	4.894	61.99	**59.51**	40.33	52.22	53.51
LCG-DistilBERT-1k (Ours)	4.869	61.94	59.24	38.29	51.62	52.77
Second Group - Base Model						
LLaMa3-8b	3.418	60.17	62.13	50.42	50.26	49.98
Second Group - Methods						
Alpaca-52k	3.718	60.57	61.36	46.10	52.41	55.74
SuperFiltering-10%	3.968	60.38	61.95	50.34	51.54	55.36
Random-6k	3.912	60.83	58.75	35.03	53.07	51.92
Perplexity-6k	4.120	61.14	61.09	50.87	53.50	56.65
Kmeans-6k	3.731	60.86	58.45	35.10	53.07	51.87
LIMA-6k	4.450	60.58	62.13	50.34	51.11	55.82
LCG-MultinomialNB-6k (Ours)	4.815	61.61	62.23	53.75	54.95	58.14
LCG-DistilBERT-6k (Ours)	**4.963**	**61.43**	**62.67**	**54.28**	**54.78**	**58.29**
LCG-DistilBERT-1k (Ours)	4.776	60.95	62.26	52.92	52.82	57.23

Table 1: Performance comparison on standard benchmarks. Results in **bold** indicate best performance within each group, while underlined values represent second-best performance within each group. The table is divided into two groups, each with its base model and various fine-tuning methods. We add the complete results for LCG-MultinomialNB-6k on LLaMa3-8b.

4 Experiments

4.1 Experimental Setup

We utilize LCG to filter the `Alpaca_52k` dataset. Two classifiers are used for the selection process: Multinomial Naive Bayes (MNB) (Pedregosa et al., 2011) and DistilBERT (Sanh et al., 2019). The classifiers are trained for 3 epochs with a learning rate of 1e-5. We then select curated datasets by a confidence threshold of < 0.7. The resulting subsets are used to fine-tune two open-source LLMs: `Mistral-7b-v0.3` (Jiang et al., 2023) and `LLaMa3-8b` (Dubey et al., 2024), using LoRA (Hu et al., 2022) with a learning rate of 2e-5 for 3 epochs.

4.2 Main Results

As presented in Table 1, our proposed LCG method consistently outperforms existing instruction data filtering approaches. When applied to Mistral-7b, LCG with MultinomialNB achieves the highest MT-bench score of 5.086, surpassing the previous best (LIMA-6k (Zhao et al., 2024)) by 14.5%. Similarly, LCG with DistilBERT demonstrates superior performance on LLaMA3-8b, improving the MT-bench score by 11.5% over LIMA-6k. Notably, our method maintains strong performance even with only 1k examples, highlighting its effectiveness. The consistent improvements across diverse metrics (Hellaswag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), GSM8k (Cobbe et al., 2021), and ARC (Clark et al., 2018)) further validate the robustness of our approach.

Our LCG-tuned models show particularly strong performance on MT-Bench. We attribute this success to the fact that low-confidence samples often represent nuanced dialogue scenarios, edge cases, and diverse response patterns. Training on these samples equips the model with superior reasoning

Models (Mistral-7B)	ARC	GSM8k	HellaSwag	MMLU	Avg
WizardLM-50k (Full)	51.54	38.59	62.14	59.36	52.91
WizardLM-Longest	51.08	38.99	62.10	58.99	52.79
WizardLM-Perplexity	51.19	39.77	62.21	59.01	53.05
WizardLM-K-means	51.96	39.95	62.11	59.38	53.35
WizardLM-SuperFiltering	51.54	39.65	62.21	59.22	53.16
LCG-DistilBERT-1k (Ours)	**52.31**	**40.00**	**62.39**	**60.01**	**53.68**
LCG-MNB-6k (Ours)	51.67	38.97	62.28	60.75	53.42

Table 2: Cross-dataset validation on WizardLM with Mistral-7B. Our LCG method demonstrates superior performance against baselines.

capabilities, essential for the sophisticated multi-turn conversations in MT-Bench.

5 Analysis and Discussion

5.1 Cross-Dataset Generalization

To validate the robustness of our approach, we applied LCG to the `WizardLM` dataset, which is larger and more complex than Alpaca. As shown in Table 2, our method continues to outperform other selection strategies, confirming that LCG is not overfitted to a single dataset but offers a generally applicable mechanism for curating high-quality instruction data.

5.2 Ablation Studies

We conducted ablation studies on key hyperparameters to validate our methodological choices.

Impact of Confidence Threshold τ The confidence threshold τ directly controls the size of the filtered subset. As shown in our main results (Table 1), relaxing τ to expand the dataset from 1K to 6K samples consistently improves performance on reasoning tasks like GSM8k. However, further experiments with a looser threshold (yielding 9K samples) resulted in significant performance degradation, indicating that our moderate threshold effectively balances quality and quantity by filtering out noise.

5.3 Synergy with RLHF

To explore modularity, we designed a two-stage pipeline combining LCG with Reinforcement Learning from Human Feedback (RLHF). First, LCG selects 10K candidates. Then, an RLHF preference model refines this set to the top 6K. As shown in Table 3, this combined approach significantly improves performance, especially on reasoning tasks, demonstrating that LCG serves as an effective initial filter for more advanced refinement techniques.

Method	GSM8k	HellaSwag	MMLU	Avg
RLHF-Only-6k	39.85	61.45	60.12	53.84
LCG+RLHF-6k	**42.15**	**62.35**	**60.45**	**54.66**

Table 3: Performance of combining LCG with RLHF. The synergistic approach yields superior results, especially on reasoning.

6 Conclusion

In this paper, we proposed Low-Confidence Gold (LCG), a novel data filtering framework that combines cluster-centric pseudo-labeling with early-stopped classifier training for efficient instruction tuning. Through extensive experiments, we demonstrated the strong performance across multiple benchmarks and base models, validating the effectiveness of our semi-supervised learning paradigm in maintaining both data quality and diversity for instruction tuning.

7 Limitation

Our work introduces a semi-supervised training paradigm to curate a subset of data for instruction tuning based on confidence score. However, there still exist several challenges: 1) Even though classifiers are tiny and spend low computational resources to train, it still takes time and effort to initially select data with annotated pseudo-labels. 2) It is likely to be hindered by the original biases and tasks of the dataset, which might still cause inefficiency after selection.

References

Tom Brown, Benjamin Mann, Nick Ryder, and Subbiah et al. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901.

Hongyi Cai, Yuqian Fu, Hongming Fu, and Bo Zhao. 2025. Mergeit: From selection to merging for efficient instruction tuning. Preprint, arXiv:2503.00034.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024. Alpagasus: Training a better alpaca with fewer data. In The Twelfth International Conference on Learning Representations.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. Preprint, arXiv:1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. Preprint, arXiv:2110.14168.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783.

Yixuan Ge, Yang Liu, Chi Hu, Weijie Meng, Shengxuan Tao, Xiaopu Zhao, Haoran Ma, Liang Zhang, Boxing Chen, Hongfei Yang, Bei Li, Tong Xiao, and Jingbo Zhu. 2024. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 464–478. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In International Conference on Learning Representations.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In International Conference on Learning Representations.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Charlie Bamford, Devendra Singh Chaplot, Diego De Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Leo Raymond Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thomas Lavril, Tao Wang, Thibaut Lacroix, and Wissam El Sayed. 2023. Mistral 7b. arXiv preprint arXiv:2310.06825.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14255–14273, Bangkok, Thailand. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 100–114.

Wenxuan Liu, Weiwen Zeng, Kaiyan He, Yijun Jiang, and Jun He. 2023. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. arXiv preprint arXiv:2312.15685.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. arXiv preprint arXiv:2301.13688.

Athul Panickssery, Samuel R. Bowman, and Shangmin Feng. 2024. Llm evaluators recognize and favor their own generations. arXiv preprint arXiv:2404.13076.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Wenhui Wang, Li Dong, Hao Cheng, Furu Wei, and Ming Zhou. 2022. Minilmv2: Multi-task pre-training for multi-task all-purpose text representations. In Findings of the Association for Computational Linguistics: ACL 2022, pages 2907–2918.

Zhiqing Xu, Fanjia Jiang, Lu Niu, Yiming Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024. Magpie: Alignment data synthesis from scratch by prompting aligned LLMs with nothing. arXiv preprint arXiv:2406.08464.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In Proceedings of the 57th Annual Meeting of the Association for

Computational Linguistics, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2024. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. In International Conference on Machine Learning, Proceedings of Machine Learning Research. PMLR.

A Extended Analysis of Semi-Supervised Model Configurations

A.1 MultinomialNB Implementation

The confidence distribution patterns of our MultinomialNB baseline, as visualized in Fig. 3, reveal fundamentally different characteristics compared to deep learning architectures. The histogram demonstrates remarkable uniformity across confidence intervals (0.0-1.0 with 0.1 increments), showing no significant concentration in specific confidence ranges. This equilibrium phenomenon stems from the model's inherent probabilistic nature and linear decision boundaries, which produce well-calibrated confidence estimates despite its simplicity.

A.2 DistilBERT comparative experiment on learning rate

Our DistilBERT implementation employed a systematic exploration of learning rate hyperparameters 1e-4, 1e-5, 1e-6 within the following experimental framework:

1. Architecture: DistilBERT-base-uncased (66M parameters) with custom classification head.

2. Optimization: Adam optimizer.

3. Training regime: 3-epoch constraint to prevent overfitting in low-data scenarios.

4. Data alignment: Identical train/test splits (stratified sampling) as MultinomialNB for direct comparability.

The empirical results (shown in Fig. 4) demonstrate non-monotonic performance relationships with learning rate scaling. Peak accuracy (62%) emerged at 1e-5, while extreme values at both ends (1e-4: 36%, 1e-6: 28%) showed substantial performance degradation. This U-shaped accuracy curve suggests the existence of optimal learning rate basins in semi-supervised BERT fine-tuning.

The model exhibited distinct confidence distribution characteristics at the 1e-6 learning rate, with predictions predominantly clustered in the low-confidence range (0-0.2). However, as revealed in Figure 2, comparative analysis across learning rates demonstrated minimal performance variation, showing only marginal improvements that correlated with accuracy increments.

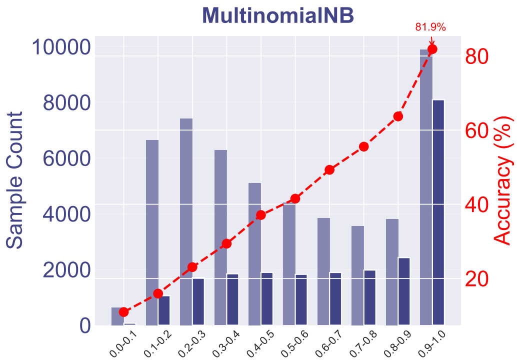

Figure 3: The data distribution of MultinomialNB across different confidence intervals.

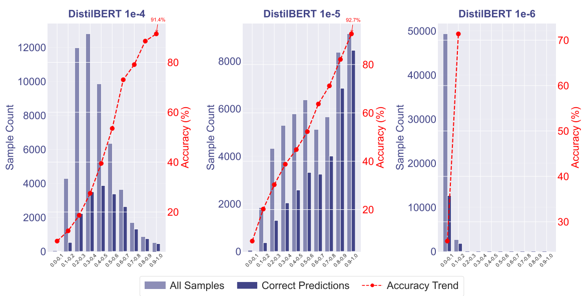

Figure 4: The data distribution of DistilBERT across different confidence intervals under various learning rates.