ToReMi: Topic-Aware Data Reweighting for Dynamic Pre-Training Data Selection

Anonymous ACL submission

Abstract

Pre-training large language models (LLMs) necessitates enormous diverse textual corpora, making effective data selection a key challenge for balancing computational resources and model performance. Current methodologies primarily emphasize data quality metrics and mixing proportions, yet they fail to adequately capture the underlying semantic connections between training samples and quality disparities within individual domains. We introduce ToReMi (Topic-based Reweighting for Model improvement), a novel two-stage framework that dynamically adjusts training sample weights according to their topical associations and observed learning patterns. Our comprehensive experiments reveal that ToReMi variants consistently achieve superior performance over conventional pre-training approaches, demonstrating accelerated perplexity reduction across multiple domains and enhanced capabilities on downstream evaluation tasks.

1 Introduction

Large language models (LLMs) typically undergo pre-training on extensive corpora derived from heterogeneous sources of varying quality (Gao et al., 2020; Soldaini et al., 2024; Penedo et al., 2023). As model parameters and pre-training datasets continue to scale (Kaplan et al., 2020; Hoffmann et al., 2022), the pre-training phase has emerged as the critical determinant of an LLM's foundational knowledge acquisition and reasoning capabilities (Zhou et al., 2023). Consequently, systematic optimization of pre-training data constitutes a fundamental technical challenge in developing high-performance LLMs.

Current pre-training data optimization methodologies primarily address two complementary dimensions: quality assessment and distribution optimization. Both approaches aim to maximize the utility of pre-training data by prioritizing valuable content while mitigating potentially detrimental samples. Conventional pre-processing pipelines incorporate language identification, corpora filtration, deduplication, and noise reduction (Soldaini et al., 2024; Penedo et al., 2023; Albalak et al., 2024). Quality assessment mechanisms predominantly utilize rule-based heuristics and supervised classification models (Raffel et al., 2023; Rae et al., 2022; Longpre et al., 2023), while distribution optimization refines corpus composition through calibrated domain ratio adjustments and strategic sampling techniques (Xie et al., 2023a; Du et al., 2022; Soldaini et al., 2024; Thrush et al., 2024) to enhance model generalization capabilities.

Despite these advancements, constructing opti-

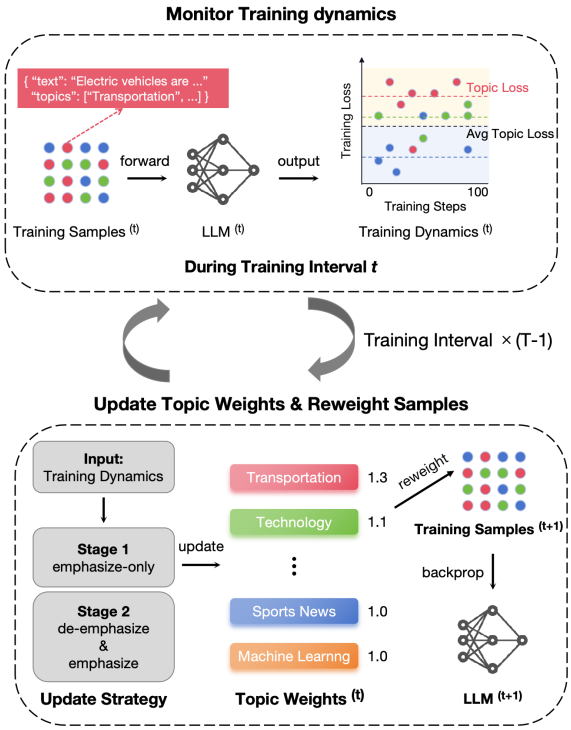

Figure 1: The framework of ToReMi, a two-stage, topic-based reweighting method for dynamic pre-training data selection and model improvement. During each training interval, training samples are reweighted based on their topic labels and previous training dynamics.

mal pre-training datasets presents persistent challenges. Quality assessment approaches based on rules and classifiers remain inherently constrained by subjective annotation biases and limited training samples, effectively filtering only conspicuously low-quality content while failing to discern subtle quality variations (Wenzek et al., 2019; Xie et al., 2023b). Similarly, current distribution optimization techniques employ relatively rudimentary methods, primarily validating effectiveness through proportional adjustments across topical or domain categories without adequately addressing intrinsic semantic relationships or dynamic training requirements (Xie et al., 2023a; Du et al., 2022). These limitations collectively impede improvements in pre-training efficiency and model performance.

To address these limitations, we investigate a fundamental research question: How can pre-training dynamically prioritize high-quality data while accounting for both latent semantic relationships within the corpus and intra-domain quality variations? We propose a two-stage **To**pic-based **Re**weighting framework for **M**odel **i**mprovement (**ToReMi**) in response to these challenges. ToReMi's innovation lies in its collective weight adjustment mechanism operating on topic categories. Rather than optimizing individual sample weights, ToReMi dynamically recalibrates all the samples of the same topic categories based on the aggregate performance of constituent samples during pre-training. The framework operates through two sequential phases: (1) During initial training, the system assigns elevated weights to challenging topic categories, prioritizing the learning of these hard samples; (2) Subsequently, the system progressively attenuates weights for underperforming topic categories (potentially containing higher noise concentrations) to minimize interference effects. Through this topic-level collective adjustment strategy, ToReMi optimizes pre-training data distribution without additional computational overhead while providing interpretable analysis of topic-specific training impact through weight trajectory feedback.

To rigorously evaluate ToReMi's efficacy, we conducted comprehensive experiments using the GPT-2 architecture (Raffel et al., 2023; Rae et al., 2022). The experimental corpus comprised 2.6B tokens of curated Wikipedia content, semantically partitioned into 39 topics through large language model annotation. Experimental results demonstrate that ToReMi consistently outperforms both standard pre-training protocols and enhanced noise-resistant baselines in log perplexity evaluations on the Paloma corpus (Gao et al., 2020). In noise-injection experiments, ToReMi achieved 1.9% average performance improvements on GLUE benchmarks compared to standard pre-training approaches (Longpre et al., 2023). Further robustness analysis confirms that ToReMi maintains performance advantages across varied hyperparameter configurations, demonstrating methodological stability and adaptability.

2 Related Work

2.1 Pretraining Data Filtering

Pre-training data filtering has been extensively studied to enhance model performance and training efficiency (Liu et al., 2024; Albalak et al., 2024). Common steps typically include language filtering (Laurenon et al., 2023; Chowdhery et al., 2022), quality filtering (Raffel et al., 2023; Rae et al., 2022), content filtering (Xu et al., 2021; Longpre et al., 2023), and deduplication (Hernandez et al., 2022; Lee et al., 2022). Filtering methods generally fall into two categories: heuristic-based and classifier-based. Heuristic methods use manually designed rules derived from corpus characteristics (Penedo et al., 2023; Laurenon et al., 2023; Raffel et al., 2023), while classifier-based methods train classifiers to assign quality scores (Brown et al., 2020; Gao et al., 2020; Xie et al., 2023b). Deduplication, on the other hand, typically uses hash-based techniques (Bloom, 1970; Wenzek et al., 2019) for exact matching and model-based methods (Abbas et al., 2023) for approximate matching. While these approaches significantly improve corpus quality, their static nature hinders dynamic adjustments during training, making them prone to discarding valuable data (Muennighoff et al., 2023) and introducing biases (Gururangan et al., 2022; Longpre et al., 2023; Dodge et al., 2021).

2.2 Pretraining Data Mixing

Pre-training datasets are often sourced from diverse domains, making effective data mixing strategies essential for maximizing their utility. Fixed data mixing proportions, commonly used in practice (Gao et al., 2020; Rae et al., 2022; Touvron et al., 2023; Soldaini et al., 2024), often rely on intuition and heuristics, such as upsampling high-quality domains like academic texts. To automate this process, (Xie et al., 2023a) trains a reference

model to guide proxy model training by minimizing worst-case excess loss, while (Fan et al., 2024) learns domain weights that maximize proxy model generalization to target domains. However, the static nature of these methods hinders their adaptability to evolving training dynamics, while the need to train multiple models further reduces their efficiency. To address these limitations, online data mixing strategies have been proposed. ODM (Albalak et al., 2023) dynamically adjusts domain weights at each iteration to prioritize domains that reduce perplexity most effectively. Skill-it (Chen et al., 2023) accelerates skill acquisition by leveraging the inherent order of prerequisite skills in the data. Additionally, (Ye et al., 2024) introduces data mixing laws to predict model performance for different data mixtures. While these methods focus on inter-domain data mixing, intra-domain mixing of diverse data characteristics remains underexplored.

3 Topic-Based Reweighting for Model Improvement (ToReMi)

In this section, we introduce ToReMi (Figure 1), a two-stage topic-based reweighting framework for dynamic pre-training data selection and model improvement, which adjusts sample weights based on their topic labels and model's training dynamics.

3.1 Preliminary

Training dynamics refer to statistical and performance metrics monitored throughout the model's training process, where high loss or prediction uncertainty is often used to identify challenging or noisy samples (Thakkar et al., 2023; Jiang et al., 2019; Swayamdipta et al., 2020). In this work, we track training loss to guide dynamic data reweighting and selection. In specific, pre-training dataset $\mathscr{D}$ consists of $\mathscr{N}$ samples $\{x_1, x_2, \ldots, x_N\}$, where $x_i = \{\text{text}, \mathscr{L}_i\}$ and $\mathscr{L}_i = \{\ell_1, \ell_2, \ldots\}$ denotes topic labels assigned to sample x_i. Let $\mathscr{L} = \bigcup_{i=1}^{N} \mathscr{L}_i$ denote the total set of all unique topic labels in the dataset. For each topic label $\ell_i \in \mathscr{L}$, an associated weight w_{ℓ_i} is assigned, and initially, all weights are uniformly set to 1.

In LLM pre-training, for each sample x_i with ground truth y_i, the training sample loss $L(x_i)$ is computed using the cross-entropy loss between the model's predicted probability distribution and the target ground truth labels, which is calculated as:

$$L(x_i) = -\frac{1}{T}\sum_{t=1}^{T} \log P(y_t \mid x_i, \theta) \tag{1}$$

where T is the sequence length of x_i.

For a specific label $\ell \in \mathscr{L}$, the training label loss L_ℓ is defined as the average loss over all samples containing ℓ, which is calculated as:

$$L_\ell = \frac{1}{|\mathscr{D}_\ell|}\sum_{x_i \in \mathscr{D}_\ell} L(x_i) \tag{2}$$

where $\mathscr{D}_\ell = \{x_i \in \mathscr{D} : \ell \in \mathscr{L}_i\}$ is the subset of samples tagged with ℓ. The average label loss $L_{\mathscr{L}}$ is:

$$L_{\mathscr{L}} = \frac{1}{|\mathscr{L}|}\sum_{\ell \in \mathscr{L}} L_\ell \tag{3}$$

3.2 ToReMi: Topic-Based Reweighting

As a two-stage topic-based reweighting framework, ToReMi aims to prioritize high-quality and impactful data while minimizing the influence of noisy or less relevant data. Reweighting is an effective approach for online data selection, as it dynamically adjusts the influence of individual samples during training, offering nuanced control without the need to exclude data outright. Prior work (Thakkar et al., 2023) computes the squared norm of a sample's gradient, showing that in the early stage of training, samples with higher scores are key contributors to the model's learning, while in later stage, such samples are more likely to represent noise or out-of-domain data. Since samples with higher training loss generally produce larger gradients, ToReMi simplifies the process by monitoring training loss directly.

In the first stage, ToReMi focuses on samples with high training loss, prioritizing their learning to help the model efficiently acquire diverse and foundational knowledge. To incorporate topic-level associations, sample weights are adjusted based on their relative topic weights. Specifically, the entire training process is divided into multiple fixed training intervals $\{t_1, t_2, \ldots, t_T\}$. Over a fixed training interval t, for each topic ℓ trained during t, we compute the training label loss $L_\ell^{(t)}$ for the topic and the average label loss $L_{\mathscr{L}}^{(t)}$ across all topics within the interval. In the subsequent interval $t+1$, the sample loss is adjusted using the weight:

$$w_\ell^{(t)} = \begin{cases} \min(w_\ell^{(t-1)} + \alpha \cdot \Delta L_\ell^{(t)}, \beta) & \text{if } L_\ell^{(t)} > L_{\mathscr{L}}^{(t)} \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

where $\Delta L_\ell^{(t)} = L_\ell^{(t)} - L_{\mathscr{L}}^{(t)}$ is the difference between the topic's loss and the average label loss. α is a scaling factor controlling the adjustment magnitude. β is the upper limit for label weights, preventing excessive upweighting and maintaining training stability. The weighted sample loss is calculated by:

$$L_w^{(t+1)}(x_i) = \min(\prod w_\ell^{(t)}, \beta) \cdot L^{(t+1)}(x_i), \quad \ell \in \mathscr{L}_i \tag{5}$$

The weighted loss is then utilized for backpropagation, enabling the model to dynamically adapt its training focus.

In the second stage, the focus transitions to minimizing the impact of noisy data while further prioritizing high-quality samples. The label weights are adjusted as follows:

$$w_\ell^{(t)} = \begin{cases} \max(w_\ell^{(t-1)} - \alpha \cdot \Delta L_\ell^{(t)}, \gamma) & \text{if } L_\ell^{(t)} > L_{\mathscr{L}}^{(t)} \\ \min(w_\ell^{(t-1)} + \alpha \cdot \Delta L_\ell^{(t)}, \beta) & \text{otherwise} \end{cases} \tag{6}$$

where γ is the lower limit for label weights to ensure sufficient representation of all labels. Then, the weighted sample loss is calculated as described in the first stage and utilized in backpropagation to guide the training process. The complete algorithm is presented in Algorithm 1.

4 Topic Annotation

Pre-training datasets are vast and encompass a wide range of topics and domains. However, the scarcity of datasets with predefined topic labels makes it difficult to directly leverage labeled data for effective training. Thus, we propose two methods for annotating topic labels to each sample within general pre-training corpora.

Given the growing volume of data and the computational costs, clustering algorithms are first applied to group similar samples based on their semantic features. After forming the clusters, the generative capabilities of LLMs are utilized to assign meaningful topic labels. This process involves extracting representative keywords from each cluster, which are then used to generate topic labels through LLMs. Specifically, there are two strategies: one where the LLM generates abstract and customized labels directly from the keywords, and another where it selectes the most relevant labels from a predefined taxonomy of topics. The first strategy, *Cluster&Generate*, enables the creation of customized topic labels, which offers flexibility and makes it particularly useful for datasets that do not align with existing classification systems. In contrast, the second strategy, *Cluster&Select*, maps clusters to an existing taxonomy, ensuring consistency and standardization across diverse datasets.

5 Experiments

5.1 Experimental Setup

Dataset and Model The pre-training dataset is sampled from Dolma-v1_5-sample (Soldaini et al., 2024), a high-quality English-only corpus curated from diverse sources. Input sequences consist of 1024 consecutive tokens randomly sampled from the dataset. Due to computational constraints, we pre-train GPT-2 (Radford et al., 2019) models from scratch, focusing on the 124M parameter variant trained on 2.6B tokens in this work. Additionally, we curate a 30B token subset to support future research, including experiments with the largest GPT-2 models (GPT-XL which contains 1.5B parameters) for compute-sufficient teams. This setup aligns with the Chinchilla-optimal scaling law (Hoffmann et al., 2022), which recommends training tokens to be $\tilde{20}\times$ the number of model parameters.

Topic Annotaion Details For topic annotation, K-means clustering is first applied to group samples based on their embeddings generated by the BGE-M3 model (Chen et al., 2024). Then, 100 representative keywords per cluster are extracted using TF-IDF. These keywords serve as input for Llama3-70B (AI@Meta, 2024), which is utilized to either generate topic labels directly or select the most relevant labels from Wikipedia's main topic classifications. The prompts employed for this purpose are detailed in Fig. 6 and Fig. 7. Fig. 2 presents the topic distribution of the entire 30B-token dataset as categorized according to the Wikipedia taxonomy.

Baselines We compare our two-stage ToReMi framework against two baseline approaches: standard pre-training (referred to as *Standard*) and a partial implementation that applies only Stage 1 of our framework (denoted as *ToReMi+Stage1*). The latter approach consistently prioritizes high-loss samples throughout training, similar to the strategy employed in Focal Loss (Lin et al., 2018), which aims to enhance the model's capacity to learn from challenging samples.

Pre-training Settings During pre-training, training dynamics are monitored at intervals of $t = 100$

Algorithm 1 Topic-Based Reweighting Framework for Model Improvement (ToReMi)

Input: Training dataset $\mathscr{D}$ with samples $\{x_1, x_2, \ldots, x_N\}$, associated topic labels $\mathscr{L} = \{\ell_1, \ell_2, \ldots \ell_L\}$, label weights $\{w_{\ell_1}, w_{\ell_2}, \ldots, w_{\ell_L}\}$, training intervals $\{t_1, t_2, \ldots, t_T\}$, scaling factor α, upper limit β, lower limit γ. Initialize $w_\ell = 1$ for all $\ell \in \mathscr{L}$.
for $t = 1, 2, \ldots, T-1$ **do**
  Compute $L_\ell^{(t)}$ and $L_{\mathscr{L}}^{(t)}$ for all $\ell \in \mathscr{L}^{(t)}$.
  for each $\ell \in \mathscr{L}^{(t)}$ **do**
    if Stage 1 **then**

$$w_\ell^{(t)} \leftarrow \begin{cases} \min(w_\ell^{(t-1)} + \alpha \cdot \Delta L_\ell^{(t)}, \beta), & \text{if } L_\ell^{(t)} > L_{\mathscr{L}}^{(t)} \\ 1, & \text{otherwise} \end{cases}$$

    else if Stage 2 **then**

$$w_\ell^{(t)} \leftarrow \begin{cases} \max(w_\ell^{(t-1)} - \alpha \cdot \Delta L_\ell^{(t)}, \gamma), & \text{if } L_\ell^{(t)} > L_{\mathscr{L}}^{(t)} \\ \min(w_\ell^{(t-1)} + \alpha \cdot \Delta L_\ell^{(t)}, \beta), & \text{otherwise} \end{cases}$$

    end if
  end for
  for each sample $x_i \in \mathscr{D}^{(t+1)}$ **do**
    Compute $L_w^{(t+1)}(x_i) \leftarrow \min(\prod_{\ell \in \mathscr{L}_i} w_\ell^{(t)}, \beta) \cdot L^{(t+1)}(x_i)$.
  end for
  Perform backpropagation using $L_w^{(t+1)}(x_i)$ to update model parameters.
end for

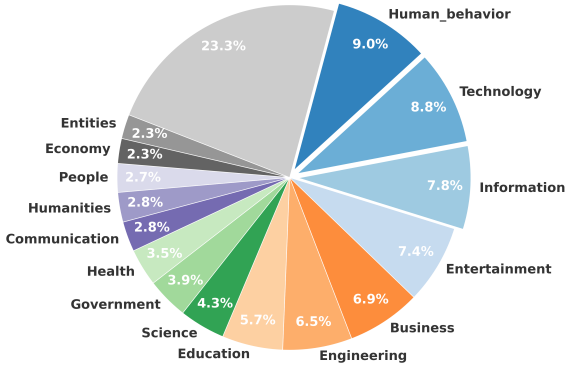

Figure 2: Topic distribution of the 30B-token dataset organized by Wikipedia taxonomy.

steps. The weight adjustment scaling factor α is configured with a default value of 1.0, while the upper and lower limits β and γ are set to 5.0 and 0.1 respectively. The reweighting mechanism transitions from Stage 1 to Stage 2 after completing 4,000 training steps.

Evaluation Settings The evaluation of ToReMi encompasses two primary aspects. For language modeling capabilities, we measure perplexity on the Paloma dataset (Magnusson et al., 2023) to evaluate how well the model fits to language distributions in diverse domains. Specifically, Paloma contains data collected from 12 distinct sources, all of which are held out from the pre-training corpus. For downstream task performance, the GLUE benchmark (Wang et al., 2019) (i.e., CoLA, SST-2, MRPC, QQP, STS-B, MNLI, QNLI, RTE, and WNLI) is utilized, which covers various dimensions of language understanding from grammaticality judgment to natural language inference. Additionally, we also evaluate on PIQA (Bisk et al., 2020) for physical commonsense reasoning and SciQ (Johannes Welbl, 2017) for scientific knowledge assessment. Both tasks are selected according to the Pythia scaling experiment (Biderman et al., 2023), which demonstrates that models with approximately 160M parameters perform meaningfully above chance.

5.2 Overall Performance

The experimental results are presented in Fig. 3 and Tab. 1. As is shown in Fig 3, all ToReMi variants consistently reduce perplexity more rapidly than the standard method in all domains, particularly during steps 1000-5000, indicating faster convergence with the topic-based reweighting mechanism. By final training steps, ToReMi achieves lower perplexity scores than the standard method in most datasets, indicating better overall language modeling capability.

Furthermore, the first section of Tab. 1 reveals that ToReMi's impact on downstream tasks is task-

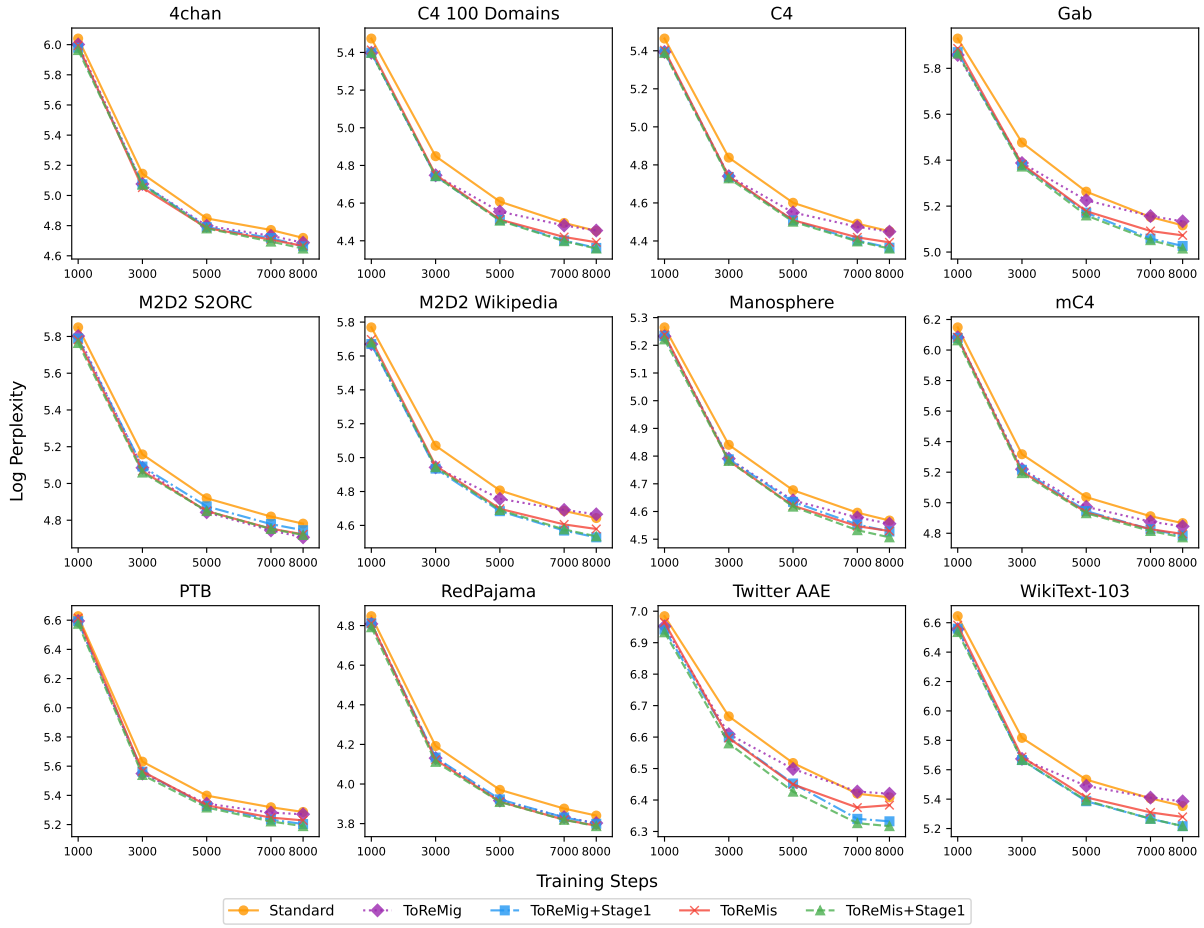

Figure 3: The log perplexity for different methods on the Paloma test dataset across 12 domains. ToReMig refers to ToReMi with directly generated topic labels, and ToReMis refers to ToReMi with topic labels selected from Wikipedia taxonomy.

dependent. For example, $ToReMi_S + Stage1$ improves by 5.78% over standard method on CoLA, and all ToReMi variants show consistent gains on SST-2. However, standard method outperforms on tasks like STS-B and RTE. This pattern indicates that topic-based reweighting has varying effects on different linguistic capabilities. ToReMi excels in tasks requiring broad linguistic patterns across diverse topics, strengthening foundational representations for syntactic understanding and sentiment analysis. Conversely, specialized reasoning tasks benefit from exposure to difficult examples that may be underrepresented after reweighting. The downweighting mechanism, while reducing noise, potentially limits exposure to challenging but informative instances needed for complex reasoning and domain-specific tasks.

5.3 Synthetic Experiment

To further evaluate the effectiveness of ToReMi in dynamically selecting high-quality data during pre-training, a synthetic experiment was conducted by injecting noise into samples associated with a specific topic label. The *Technology* label, which accounts for a significant proportion of the dataset and represents an important domain for evaluation, was selected for this purpose. Noise was introduced by randomly shuffling all characters within each sample to simulate low-quality data. For better reproducibility, ToReMi with the Wikipedia topic classification ($ToReMi_S$) was adopted for all subsequent experiments.

The results are presented in the second section of Tab. 1. Standard pre-training performs poorly on most metrics, indicating that noisy samples significantly impede model learning. ToReMi with Stage1 achieves notable gains in CoLA (5.02%) and RTE (6.5%), demonstrating that prioritizing high-loss labels in early training enhances the model's linguistic understanding, strengthening its grasp of both grammatical structures and semantic relationships. The complete two-stage ToReMi achieves the highest overall score (61.52) with substantial improvements on both MRPC and STS-B com-

Method	CoLA	SST-2	MRPC	STS-B	QQP	MNLI	QNLI	RTE	WNLI	SciQ	PIQA	Overall
Overall Performance of Pre-training the 124M GPT-2 Model Using Different Methods												
Standard	17.49	86.58	75.37	**74.64**	84.42	75.10	82.81	**59.20**	**56.33**	**24.60**	56.31	**62.99**
$ToReMi_S + Stage1$	**23.27**	**88.07**	**77.47**	54.55	**84.75**	75.32	82.26	52.34	43.66	23.50	**57.07**	60.21
$ToReMi_S$	18.18	86.69	75.88	72.80	84.71	**75.46**	82.46	55.23	54.93	24.30	56.53	62.47
$ToReMi_G + Stage1$	15.93	87.27	76.34	72.48	**84.75**	75.24	**82.96**	57.04	43.66	23.50	56.91	61.46
$ToReMi_G$	16.84	87.61	76.36	73.24	84.72	75.26	82.04	54.15	42.25	23.70	56.31	61.13
Pre-training the 124M GPT-2 Model on Synthetic Noise Text												
Standard	17.79	86.35	74.40	71.18	84.09	75.08	81.84	48.01	**54.93**	**24.60**	55.88	61.29
$ToReMi_S + Stage1$	**22.81**	**86.81**	74.43	69.09	**84.47**	75.36	81.69	**54.51**	43.66	23.20	**56.91**	61.18
$ToReMi_S$	21.35	86.69	**76.23**	**73.25**	84.39	**75.61**	**82.15**	51.98	43.66	25.20	56.20	**61.52**
Effect of Stage Transition Point												
$ToReMi_S + 3000step$	20.79	86.81	75.48	68.71	84.12	75.04	81.73	52.70	43.66	**27.40**	**56.80**	61.20
$ToReMi_S + 4000step$	21.35	86.69	76.23	**73.25**	84.39	**75.61**	82.15	51.98	43.66	25.20	56.20	61.52
$ToReMi_S + 5000step$	13.01	**87.50**	74.68	72.78	84.37	75.46	**82.75**	**58.12**	38.02	24.00	**56.80**	60.68
$ToReMi_S + 6000step$	**22.62**	86.35	75.24	68.72	84.52	75.24	82.20	49.45	**53.52**	27.00	56.31	**61.92**
$ToReMi_S + 7000step$	20.13	**87.50**	**77.06**	70.30	**84.54**	75.57	82.39	54.51	42.25	25.40	56.69	61.49
Effect of Reweighting Bounds (γ, β)												
$ToReMi_S + Stage1 + (1.0, 5.0)$	22.81	86.81	74.43	69.09	84.47	75.36	81.69	54.51	43.66	23.20	**56.91**	61.18
$ToReMi_S + Stage1 + (1.0, 10.0)$	**24.11**	**87.72**	76.79	**74.62**	**84.72**	75.29	**83.39**	**59.20**	36.62	23.90	56.26	62.06
$ToReMi_S + Stage1 + (1.0, 20.0)$	19.73	87.38	76.80	71.43	84.66	**75.64**	82.02	51.62	40.84	24.80	56.58	61.05
$ToReMi_S + (0.1, 5.0)$	21.35	86.69	76.23	73.25	84.39	75.61	82.15	51.98	43.66	25.20	56.20	61.52
$ToReMi_S + (0.1, 10.0)$	21.31	86.23	**76.91**	73.06	84.37	75.35	82.04	57.04	**56.33**	**25.80**	56.64	**63.19**
$ToReMi_S + (0.1, 20.0)$	20.68	85.78	75.05	63.26	84.50	75.19	82.31	49.81	**56.33**	24.80	55.98	61.24

Table 1: Model performance using different pre-training methods on downstream tasks. The table presents results for: (1) pre-training with normal data, (2) pre-training with synthetic noise data, (3) effect of various stage transition points, and (4) effect of different reweighting bounds.

pared to the standard method and Stage1-only variant, highlighting how effectively its downweighting strategy mitigates the impact of noisy data.

5.4 Ablation Experiment

Effect of Stage Transition Point To investigate the impact of stage transition point between training phases in ToReMi, we conducted experiments by varying the step at which training switches from weighting (Stage 1) to de-weighting (Stage 2) on the noisy dataset introduced in Sec. 5.3. While the default transition occurs at 4000 steps within a total of 8000 steps, additional experiments were conducted with the transition points at {3000, 5000, 6000, 7000} steps. We primarily focused on delayed transitions, as entering Stage 2 prematurely before model convergence results in downweighting certain topics before adequate learning, decreasing pre-training efficiency.

Results presented in the third section of Tab. 1 indicate that transition timing significantly impacts model performance. The 6000-step transition point achieved the highest overall score (61.92), effectively balancing the initial aggressive learning phase with the subsequent noise-reduction phase. This point provides sufficient time for the model to learn important patterns while still allowing adequate time to downweight noisy samples. In contrast, the 5000-step point produced the lowest performance with a significant drop in CoLA (13.01) despite achieving the highest RTE score (58.12), suggesting that delayed transitions may cause overfitting to noisy samples in certain tasks while benefiting others. The non-linear relationship between transition point and model performance demonstrates that hyperparameter is critical when applying ToReMi to different task settings.

Furthermore, Fig. 4 illustrates the performance difference between standard method and ToReMi with various stage transition points. ToReMi outperforms standard method on most tasks regardless of transition point, with the exception of WNLI. The consistent improvement on various tasks further validates the effectiveness and robustness of ToReMi. The underperformance on WNLI is attributed to its unique characteristics as a natural language inference task with a small dataset (only 634 training examples). WNLI requires understanding of complex pronoun resolution and discourse relationships, which are disproportionately affected by the topic-based reweighting mechanism. The sample reweighting approach inadvertently downweights examples crucial for this particular task during Stage 2, indicating that specialized treatment is necessary for tasks heavily dependent on specific linguistic phenomena.

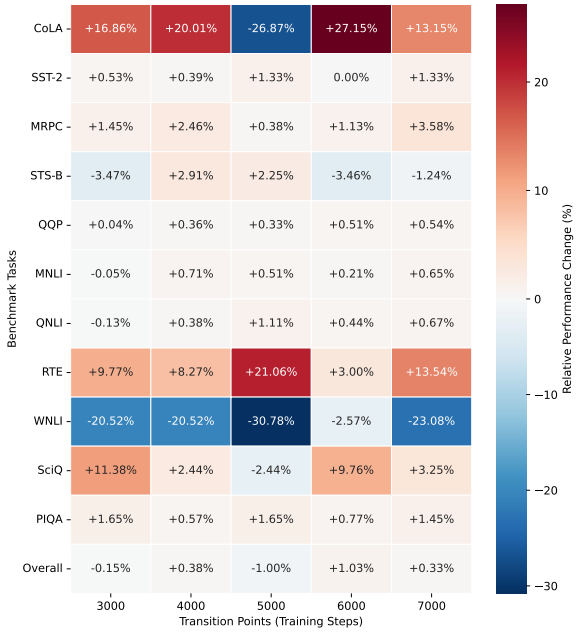

Figure 4: Performance difference between the standard method and ToReMi with various stage transition points. Red indicates performance improvement over the standard model, while blue indicates degradation.

Effect of Reweighting Bounds To investigate the impact of reweighting bounds on model performance, experiments was conducted by varying the weight upper bound β while maintaining a constant downweighting lower bound ($\gamma = 0.1$). The experiment focus on upper bound because excessively high weighting is susceptible to loss overexpansion for certain samples and introduces training instability, while the lower bound has comparatively smaller influence on overall performance. Both the ToReMi+Stage1 variant and the complete ToReMi were evaluated with β values of {5.0, 10.0, 20.0}.

The results are presented in the fourth section of Tab. 1. It is shown that moderate weight (β = 10.0) produces optimal performance for both methods (62.06 for ToReMi+Stage1 and 63.19 for complete ToReMi), while further increasing β to 20.0 causes degradation below even the β = 5.0 configuration. These findings indicate that increased weighting helps the model focus on challenging samples, though excessive upweighting leads to overfitting on particular topics and introduces instability in the training process. Furthermore, when comparing the upweighting-only approach and the complete ToReMi at the same β values, it is observed that the two-stage approach consistently outperforms the upweighting-only variant. The performance gap is particularly pronounced at β = 10.0, where the complete ToReMi achieved 1.13% improvement. The results show the importance of noise reduction

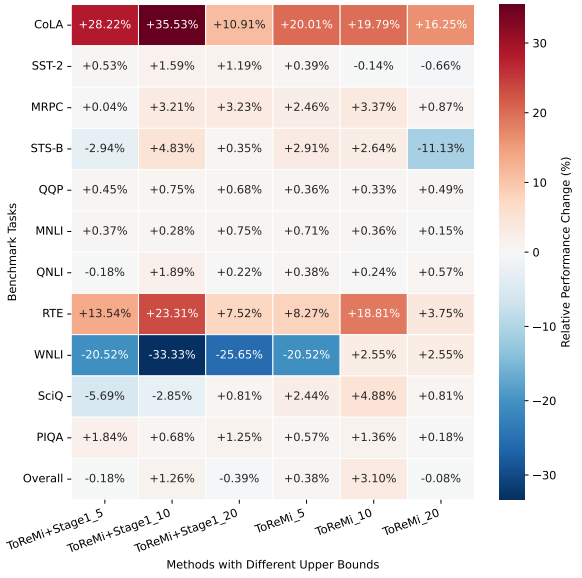

Figure 5: Performance difference between the standard method and ToReMi variants with various weight upper bounds. Red indicates performance improvement over the standard model, while blue indicates degradation.

during later training. Initial upweighting enables model to efficiently learn challenging topic-specific patterns, while subsequent downweighting reduces the influence of noisy samples.

Fig. 5 illustrates the performance difference between standard method and ToReMi variants with different weight upweighting upper bounds. It is shown that both ToReMi and ToReMi+Stage1 outperform standard method on most tasks, demonstrating the effectiveness of our topic reweighting mechanism. Notable improvements appear on CoLA and RTE, where ToReMi+Stage1 with β = 10.0 achieves gains of 35.53% on CoLA and 23.31% on RTE. However, ToReMi+Stage1 underperforms on WNLI, indicating that the sole upweighting leads to overfitting on specific patterns, and the complete ToReMi (particularly with β = 10.0) addresses this limitation through its downweighting strategy in later training.

6 Conclusion

We propose ToReMi, a two-stage framework that dynamically reweights pre-training data by topic. Experiments with GPT-2 on Dolma show ToReMi accelerates perplexity reduction and improves final performance, especially on syntactic and sentiment tasks. Our results highlight topic-aware reweighting as a powerful tool for efficient and effective pre-training. Future work could explore optimal topic characteristics for further gains.

Limitations

One key limitation of ToReMi is its reliance on topic annotation quality, which depends on clustering accuracy and LLM-generated labels. Noisy or imbalanced topic assignments may skew reweighting decisions. Additionally, the framework assumes topic-level homogeneity in sample quality, which may not hold for fine-grained intra-topic variations. Computational overhead for dynamic weight tracking—though modest—scales with topic diversity, potentially limiting applicability to ultra-large corpora. Finally, the two-stage transition requires manual tuning of hyperparameters (e.g., stage switch timing), which may need adaptation across datasets.

Ethical Concerns

ToReMi' s topic-based reweighting could inadvertently amplify biases if certain topics correlate with demographic or cultural groups. We mitigate this by auditing topic distributions for skew and using diverse annotation taxonomies. Data privacy risks are minimal as our method processes existing public corpora without collecting user-generated content. The noise-reduction stage further reduces exposure to potentially harmful text by downweighting low-quality samples. All experiments comply with dataset licenses, and our open-source release enables transparency in reweighting logic.

References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*.

AI@Meta. 2024. Llama 3 model card.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A survey on data selection for language models. *Preprint*, arXiv:2402.16827.

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. 2023. Efficient online data mixing for language model pre-training. *Preprint*, arXiv:2312.02406.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *Preprint*, arXiv:2304.01373.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Burton H Bloom. 1970. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Mayee F. Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher R é . 2023. Skill-it! a data-driven skills framework for understanding and training language models. *Preprint*, arXiv:2307.14430.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *Preprint*, arXiv:2204.02311.

Jesse Dodge, Maarten Sap, Ana Marasovi, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *Preprint*, arXiv:2104.08758.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. Glam: Efficient scaling of language models with mixture-of-experts. *Preprint*, arXiv:2112.06905.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. 2024. Doge: Domain reweighting with generalization estimation. *Preprint*, arXiv:2310.15393.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Suchin Gururangan, Dallas Card, Sarah Dreier, Emily Gade, Leroy Wang, Zeyu Wang, Luke Zettlemoyer, and Noah A. Smith. 2022. Whose language counts as high quality? measuring language ideologies in text data selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2562–2580, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Scott Johnston, Ben Mann, Chris Olah, Catherine Olsson, Dario Amodei, Nicholas Joseph, Jared Kaplan, and Sam McCandlish. 2022. Scaling laws and interpretability of learning from repeated data. *Preprint*, arXiv:2205.10487.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Angela H. Jiang, Daniel L. K. Wong, Giulio Zhou, David G. Andersen, Jeffrey Dean, Gregory R. Ganger, Gauri Joshi, Michael Kaminksy, Michael Kozuch, Zachary C. Lipton, and Padmanabhan Pillai. 2019. Accelerating deep learning by focusing on the biggest losers. *Preprint*, arXiv:1910.00762.

Matt Gardner Johannes Welbl, Nelson F. Liu. 2017. Crowdsourcing multiple choice science questions.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Hugo Laurenon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo Gonz á lez Ponferrada, Huu Nguyen, et al. 2023. The bigscience roots corpus: A 1.6tb composite multilingual dataset. *Preprint*, arXiv:2303.03915.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. *Preprint*, arXiv:2107.06499.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Doll á r. 2018. Focal loss for dense object detection. *Preprint*, arXiv:1708.02002.

Yang Liu, Jiahuan Cao, Chongyu Liu, Kai Ding, and Lianwen Jin. 2024. Datasets for large language models: A comprehensive survey. *Preprint*, arXiv:2402.18041.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2023. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. *Preprint*, arXiv:2305.13169.

Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, A. Jha, Oyvind Tafjord, Dustin Schwenk, Pete Walsh, Yanai Elazar, Kyle Lo, Dirk Groeneveld, Iz Beltagy, Hanna Hajishirzi, Noah A. Smith, Kyle Richardson, and Jesse Dodge. 2023. Paloma: A benchmark for evaluating language model fit. *ArXiv*, abs/2312.10523.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling data-constrained language models. *Preprint*, arXiv:2305.16264.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. *Preprint*, arXiv:2306.01116.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2022. Scaling language models: Methods, analysis & insights from training gopher. *Preprint*, arXiv:2112.11446.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh

Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. *Preprint*, arXiv:2402.00159.

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.

Megh Thakkar, Tolga Bolukbasi, Sriram Ganapathy, Shikhar Vashishth, Sarath Chandar, and Partha Talukdar. 2023. Self-influence guided data reweighting for language model pre-training. *Preprint*, arXiv:2311.00913.

Tristan Thrush, Christopher Potts, and Tatsunori Hashimoto. 2024. Improving pretraining data using perplexity correlations. *Preprint*, arXiv:2409.05816.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timoth é e Lacroix, Baptiste Rozi è re, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzm á n, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *Preprint*, arXiv:1911.00359.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023a. Doremi: Optimizing data mixtures speeds up language model pretraining. *Preprint*, arXiv:2305.10429.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. 2023b. Data selection for language models via importance resampling. *Preprint*, arXiv:2302.03169.

Albert Xu, Eshaan Pathak, Eric Wallace, Suchin Gururangan, Maarten Sap, and Dan Klein. 2021. Detoxifying language models risks marginalizing minority voices. *Preprint*, arXiv:2104.06390.

Jiasheng Ye, Peiju Liu, Tianxiang Sun, Yunhua Zhou, Jun Zhan, and Xipeng Qiu. 2024. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *Preprint*, arXiv:2403.16952.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. *Preprint*, arXiv:2305.11206.

A Prompt for Topic Annotation

```
Task:
Please generate one to three abstract labels with one
word based on the following list of 100 keywords.
The labels should be general and as abstract as
possible, aiming to cover the main topics and categories.

You must response with the following format, and don't
response anything else:
### Labels: Label1, Label2

Examples:
### Labels: Technology, Health

Keyword list:
{}
```

Figure 6: Prompt for generating topic labels for each sample using the provided extracted keywords.

```
Task: Select one to three most related topic labels
based on the given keywords.

[Keywords]
{}

[Topic labels]
{}

Based on the given keywords, please select one to
three most relevant labels from the provided topic labels.
Ensure that the selected labels best capture the primary
concepts and topics represented by the keywords.

Note:
1. The selected labels must be from the given topic
labels!
2. Don't respond any reasoning process or explanations!

You must respond with the following format, and don't
respond anything else:
### Labels: Label1, Label2

Examples:
### Labels: xxxx, xxxx, ...
```

Figure 7: Prompt for assigning topic labels to each sample based on the provided Wikipedia taxonomy.