TASK-ADAPTIVE PRETRAINED LANGUAGE MODELS VIA CLUSTERED IMPORTANCE SAMPLING

David Grangier, Simin Fan, Skyler Seto, Pierre Ablin
Apple

ABSTRACT

Specialist language models (LMs) focus on a specific task or domain on which they often outperform generalist LMs of the same size. However, the specialist data needed to pretrain these models is only available in limited amount for most tasks. In this work, we build specialist models from large generalist training sets instead. We adjust the training distribution of the generalist data with guidance from the limited domain-specific data. We explore several approaches, with clustered importance sampling standing out. This method clusters the generalist dataset and samples from these clusters based on their frequencies in the smaller specialist dataset. It is scalable, suitable for pretraining and continued pretraining, it works well in multi-task settings. Our findings demonstrate improvements across different domains in terms of language modeling perplexity and accuracy on multiple-choice question tasks. We also present ablation studies that examine the impact of dataset sizes, clustering configurations, and model sizes.

1 INTRODUCTION

Generalist language models (LMs) can address a wide variety of tasks, but this generality comes at a cost (Brown et al., 2020). It necessitates a large training set representative of all prospective tasks, as well as a large model to fit such a comprehensive dataset. Specialist models forgo this generality and fit a model for a limited domain or task. In their narrow specialty, such models can achieve better accuracy at a given model size (Kerner, 2024).

Pretraining a specialist is interesting when two conditions are met: (i) the targeted task justifies the cost of training a dedicated model and (ii) a specialist dataset large enough for pretraining is available. Condition (i) is dependent on the targeted application and its potential economic benefit. Condition (ii) is more limiting since modern LMs are commonly pre-trained on datasets larger than 100B tokens[1], an amount that cannot be commissioned for most applications.

This work considers relaxing condition (ii) and studies methods to train a specialist model when specialized data is scarce. Given a large generalist dataset and a small specialist dataset, we propose to modify the distribution over the generalist dataset guided by the scarce specialist dataset. Training a model on the modified distribution gives a specialist model with better accuracy than a generalist model of the same size.

We study this setting across different specialization tasks including domain-specific language modeling (medical, encyclopedic domains) and end-tasks (scholar exams in science and humanities, reasoning questions). We compare different strategies to manipulate the pretraining distribution. We evaluate strategies based on text classifiers, gradient-alignment and importance sampling (IS). Although IS is rarely used for LM data selection, we build upon on a simple IS recipe based on clustering (Grangier et al., 2024b) and report that the resulting method systematically outperforms alternatives. Our IS recipe clusters the generalist set and computes the cluster histogram over the specialist data. Then, for pretraining, generic data is sampled according to the specialist histogram, see Figure 1. We show the empirical benefit of this method varying model sizes (350m to 7B parameters), the amount of generalist data and the amount of specific data. We assess both perplexity gains for language model adaptation and accuracy improvements for multiple choice question tasks.

[1] 100B tokens $\simeq$ 1m books $\simeq$ 60x the annual publication of the top English language publisher (Lee, 2021).

2 Related Work

Generalist vs Specialist LMs Generalist LMs address tasks which they have not been trained for explicitly (Brown et al., 2020), or provide a good initialization for fine-tuning a dedicated model (Devlin et al., 2019). Nowadays generalists compete with dedicated models on many tasks (Jiang et al., 2024; Dubey et al., 2024). Success however comes at a price: a generalist must be much larger than a specialist for the same accuracy. For instance, on English-to-German translation, the 175-B parameter generalist GPT-3 (Brown et al., 2020) is less accurate than a 136m-parameter specialist (Sennrich et al., 2016a). For neural LMs, the parameter count directly impacts training and inference costs.

Specialist large LMs exist in domains where large amounts of specialized texts are available. Corpora with billions of tokens enable pretraining or *continued pretraining*, a generalist pretraining phase followed by a specialist one (Gururangan et al., 2020; Parmar et al., 2024)). Domains with specialist models include medicine and biology (Lewis et al., 2020; Labrak et al., 2024; Bolton et al., 2024), computer programming and mathematics (Lewkowycz et al., 2022; Rozière et al., 2024; Azerbayev et al., 2024) and finance (Wu et al., 2023; Xie et al., 2023a). When specialist data is available in limited amount, there are methods to train specialist models on generalist data instead.

Task-Adaptive Data-Selection The selection methods over-sample generalist data that aids model generalization in the target domain. For masked LMs, Gururangan et al. (2020) observe that continued pretraining improves the performance on end-tasks when using data with high vocabulary overlap with the targeted task. For machine translation (MT), Aharoni & Goldberg (2020) show that a task-adapted pretraining dataset can be selected from a generalist dataset using the nearest neighbors of a small specialist set. Their nearest neighbor classifier relies on BERT sentence distance (Devlin et al., 2019). Still for MT, other works have used other types of classifiers. In particular, contrasting the scores of two LMs (generalist and specialist) is popular (Moore & Lewis, 2010; Axelrod et al., 2011; Wang et al., 2018; Junczys-Dowmunt, 2018). Other classifiers include logistic regression or fine-tuned BERT (Iter & Grangier, 2021). Outside classification, Xie et al. (2023c) proposed to use importance sampling for continued pretraining. They improve classification tasks by selecting pretraining data with a similar distribution to the targeted domain in terms of hashed-ngrams. Importance sampling is also used in (Grangier et al., 2024b) and we build upon that work which adjusts the frequency of generalist clusters informed by specialist data: we scale the method to millions of clusters, show that it works with larger models, and extend it beyond language modeling tasks.

A third type of methods for task-adaptative selection relies on bilevel optimization and gradient aligment (Pruthi et al., 2020; Xia et al., 2024; Grangier et al., 2023). The pretraining distribution is selected such that the reweighted gradients from the generalist dataset mimics the expected gradient from the small specialist dataset. Gradient-alignment for data selection has also been used for other purposes such as data summarization (Borsos et al., 2024), pretraining acceleration (Xie et al., 2023b; Fan et al., 2024) or auxiliary task weighting (Wang et al., 2020; Raghu et al., 2021). Finally, it is also worth mentioning data selection methods based on reinforcement learning (Liu et al., 2019; Yoon et al., 2020), bayesian optimization (Ruder & Plank, 2017), data models (Ilyas et al., 2022) and influence models (Yu et al., 2024).

Pretraining Data Quality Outside of domain aspects, the quality of pretraining data is also an important topic (Wenzek et al., 2020; Dodge et al., 2021; Penedo et al., 2023; Li et al., 2024). Data quality includes removing data in other languages (Cook & Lui, 2012), text formatting (Xu et al., 2024), favoring long form text Gao et al. (2021); Gunasekar et al. (2023), removing duplicates (Lee et al., 2022). It also involves balancing different sources of data with the goal of reaching a better generic pretraining loss Xie et al. (2023b); Fan et al. (2024); Vo et al. (2024). Recent work also considered filtering Kong et al. (2024), correcting Chen & Mueller (2024) or generating Maini et al. (2024) pretraining data with LMs. These data quality considerations are orthogonal to domain concerns: quality filters are applied alongside domain adaptation decisions (Albalak et al., 2024).

3 Data Selection for Task-Adaptive Pretraining

We consider three methods for task-adaptive pretraining of LMs. Classification and gradient alignment have been evaluated in different contexts before but not for end-tasks like multiple-choice question answering. Clustered-based importance sampling at scale is a contribution of this work, building upon recent work from Grangier et al. (2024b).

3.1 NOTATIONS

D^{g} is the training dataset sampled from the generalist distribution $\mathcal{D}^{\text{g}}$. D^{s} is the specialist dataset representative of the final task, sampled from $\mathcal{D}^{\text{s}} \neq \mathcal{D}^{\text{g}}$. The loss of model θ on D is

$$\mathcal{L}(D;\theta) := \frac{1}{|D|}\sum_{x\in D}\ell(x;\theta) = -\frac{1}{|D|}\sum_{x\in D}\frac{1}{|x|}\sum_{i}\log p(x_i|x_1^{i-1};\theta)$$

where $|D|$ denotes the cardinality of a set D and $|x|$ denotes the length of sequence $x = x_0^{|x|} = (x_1, \ldots, x_{|x|})$. The perplexity of model θ on the dataset D is $\mathcal{P}(D;\theta) := \exp(\mathcal{L}(D;\theta))$.

3.2 CLASSIFICATION

A binary classifier is trained to estimate the probability that a generalist pretraining document belongs to the targeted domain. The classifier ϕ is learned using positive examples from D^{s} and a subset of D^{g} as negative examples. ϕ then builds a domain-specific pretraining set

$$C(D^{\text{g}}, t) := \{x \in D^{\text{g}} \text{ such that } \phi(x) > t\}.$$

which restrict the generic dataset D^{g} to the examples with an estimated probability to be in-domain above threshold t. The threshold is validated as a trade-off between focusing on data close to the domain of interest while keeping $C(D_g, t)$ large enough to train an LM of the targeted capacity. In our case, we rely on a logistic regression classifier trained over sentence BERT (SBERT) text embeddings (Reimers & Gurevych, 2019), an established classification method (Minaee et al., 2021). The SBERT representation is also commonly used in data selection (Albalak et al., 2024; Xie et al., 2023c; Zhang et al., 2024; Su et al., 2023). This representation is also used in the alternative selection strategies we consider. As an ablation, we also evaluate the impact of the choice of SBERT (Section 5.1).

3.3 GRADIENT-ALIGNMENT

Gradient-Alignment (GA) methods are common when the generic pretraining set D^{g} originates from n_{g} different data sources S, i.e. $D^{\text{g}} = \bigcup_{i=1}^{n_{\text{g}}} D_i^{\text{g}}$. These methods select weights for the different sources by considering two functions of θ: the pretraining reweighed loss,

$$\mathcal{L}((w, D^{\text{g}});\theta) := \sum_{i=1}^{n_{\text{g}}} w_i \mathcal{L}(D_i^{\text{g}};\theta),$$

and the targeted loss, i.e. the loss on D^{s} in our case. The weights, on the simplex, can be inferred via a bilevel formulation of the data selection problem (Dagréou et al., 2022): the minimum $\theta^\star(w) = \arg\min_\theta \mathcal{L}((w, D^{\text{g}});\theta)$ depends on w and task-dependent pretraining is interested in weights w that minimize $\mathcal{L}(D^{\text{s}};\theta^\star(w))$ wrt w. This formulation results in algorithms that select weights during pretraining to align the gradients of these two functions wrt θ (Xie et al., 2023b; Grangier et al., 2023). In our case, we rely on the DoGE (Fan et al., 2024) algorithm. Compared to classifiers, GA is harder to scale to large model size. This limitation is commonly addressed by finding the mixture weights with a small model before transferring them to a larger model.

In this work, we consider a generic setting where the pretraining dataset D^{g} is not pre-segmented into few data sources. Instead, we rely on the k-means clustering of the Sentence BERT embeddings to identify data clusters. Clustering based on text embeddings has been used for data selection, both for quality filtering (Kaddour, 2023) and domain adaptation (Grangier et al., 2024a).

3.4 CRISP: CLUSTERED IMPORTANCE SAMPLING FOR PRETRAINING

We sketch our strategy from Figure 1. Initially, we divide the space of text into clusters. We decompose the specialist loss and the generalist loss as a weighted sum of losses over clusters. Then we make an independence assumption that implies that the specialist and generalist loss per cluster are identical. The specialist loss is then computed as the generalist loss with a reweighing of each cluster.

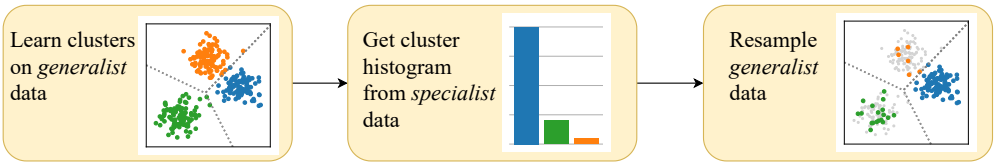


Figure 1: **Clustered-based Importance Sampling with CRISP**.

Specifically, we want to identify a model with a low loss on the specialist distribution $\mathcal{D}^{\text{s}}$,

$$\mathcal{L}(\mathcal{D}^{\text{s}};\theta) = \mathop{\mathbb{E}}_{x\sim\mathcal{D}^{\text{s}}}[\ell(x;\theta)] = \sum_x \ell(x;\theta)P(x|\mathcal{D}^{\text{s}})$$

We marginalize over a discrete latent variable c, the cluster variable, and write

$$\mathcal{L}(\mathcal{D}^{\text{s}};\theta) = \sum_x \sum_c \ell(x;\theta)P(x|c,\mathcal{D}^{\text{s}})P(c|\mathcal{D}^{\text{s}}) \underset{(2)}{=} \sum_x \sum_c \ell(x;\theta)P(x|c)P(c|\mathcal{D}^{\text{s}}) \tag{1}$$

where the second equality $=_{(2)}$ makes the independence assumption $P(x|c,\mathcal{D}^{\text{s}}) = P(x|c)$. If we make a similar assumption for the generalist loss $P(x|c,\mathcal{D}^{\text{g}}) = P(x|c)$, we can write both losses as

$$\mathcal{L}(\mathcal{D}^{\text{s}};\theta) = \mathop{\mathbb{E}}_{c\sim(c|\mathcal{D}^{\text{s}})}[\mathcal{L}(c;\theta)] \quad \text{and} \quad \mathcal{L}(\mathcal{D}^{\text{g}};\theta) = \mathop{\mathbb{E}}_{c\sim(c|\mathcal{D}^{\text{g}})}[\mathcal{L}(c;\theta)] \tag{2}$$

where we define $\mathcal{L}(c;\theta) =: \sum_x \ell(x;\theta)P(x|c)$. We now apply importance sampling to these expectation, defining the importance weights as $w(c) = P(c|\mathcal{D}^{\text{s}})/P(c|\mathcal{D}^{\text{g}})$,

$$\mathcal{L}(\mathcal{D}^{\text{s}};\theta) = \sum_c \mathcal{L}(c;\theta)P(c|\mathcal{D}^{\text{s}}) = \sum_c \mathcal{L}(c;\theta)\frac{P(c|\mathcal{D}^{\text{s}})}{P(c|\mathcal{D}^{\text{g}})}P(c|\mathcal{D}^{\text{g}}) = \mathop{\mathbb{E}}_{c\sim(c|\mathcal{D}^{\text{s}})}[w(c)\mathcal{L}(c;\theta)].$$

In our experiments, we estimate the terms $w(c), \mathcal{L}(c;\theta)$ from the finite training sets $D^{\text{s}} \sim \mathcal{D}^{\text{s}}$ and $D^{\text{g}} \sim \mathcal{D}^{\text{g}}$. We count the number of examples in each cluster to estimate $P(c|D^{\text{s}}), P(c|D^{\text{g}})$. The expected loss over a cluster $\mathcal{L}(c;\theta)$ is estimated as the average loss over the generalist examples in cluster c, $\mathcal{L}(D^{\text{g}} \cap K(c);\theta)$, where $K(c)$ denotes the examples in cluster c. This strategy therefore only estimates $P(c|D^{\text{s}})$ on the small D^{s}. The term $\mathcal{L}(c;\theta)$ is estimated over the large set as $\mathcal{L}(D^{\text{g}} \cap K(c);\theta)$ which has less variance than the estimator $\mathcal{L}(D^{\text{s}} \cap K(c);\theta)$ over the small D^{s}.

We train CRISP models with stochastic optimization (Kingma & Ba, 2015, Adam) and propose Algorithm 1. Here, we do not explicitly reweigh the loss. We instead samples cluster from their importance. This avoids frequently visiting clusters with less weights. This strategy has less variance in its gradient estimates, which can help convergence (Seiffert et al., 2008; An et al., 2021). This algorithm is simple and efficient when one groups the generalist examples by cluster prior to training.

Algorithm 1 CRISP Training

Parameters: T (number of steps), B (batch size)
Input: D^{s} (specialist set), D^{g} (generalist set)
$h^{\text{s}} \leftarrow \{P(c|D^{\text{s}}), \forall c\}$ ▷ Count cluster frequency on the specialist set D^{s}.
$\theta_0 \leftarrow \text{InitModel}()$ ▷ Initialize the model.
for $t = 1, \ldots, T$ **do**
  for $i = 1, \ldots, B$ **do**
    $c_i \sim \text{Categorical}(h^{\text{s}})$ ▷ Sample a cluster id from the specialist histogram.
    $x_i \sim \text{Uniform}(D^{\text{s}} \cap K(c))$ ▷ Sample a generalist example in the selected cluster.
  end for
  $\theta_t \leftarrow \text{AdamUpdate}(\theta_{t-1}, \{x_0, \ldots, x_B\})$
end for

4 Experiments & Results

We perform experiments with transformer LMs (Vaswani et al., 2017). Most of our experiments use models with 1.3B parameters (trained on 120B tokens) and we conduct ablations with 350m and

7B models (resp. trained on 40B, 350B tokens). Our settings for architectures and optimization are borrowed from Brown et al. (2020), see Appendix C.

Our generalist training set is Redpj2 (Together AI Team, 2023). We select this dataset as it contains only web-crawled data without additional interventions to help evaluation tasks (e.g. adding encyclopedias, books or academic articles). Redpj2 contains over 30T tokens with our 32k byte-pair encoding tokenizer (Sennrich et al., 2016b), see Table 2 in Appendix B. We segment the dataset into non-overlapping 1,024 token windows (the model context limit) and compute SBERT embedding for every window. We cluster the generalist dataset hierarchically with a clustering tree with branching 64 for 4 levels, see Appendix A. The levels therefore have 64, 4,096 ($= 64^2$), 260k ($= 64^3$) and 16.7m ($= 64^4$) clusters with an average of 540B, 8.4B, 130m and 2m tokens per cluster respectively. As an alternative to SBERT embeddings, we also consider Latent Semantic Index (LSI), i.e. singular value decomposition over tf-idf representations (Deerwester et al., 1990; Dumais, 2004).

For our specialist tasks, we consider 3 language modeling tasks (LM) and 3 multiple-choice-question tasks (MCQ). For LM, we use Pile subsets from different domains (Gao et al., 2021): medical (Pubmed Central), programming Q&A (Stackexchange), and encyclopedic (Wikipedia). For MCQ answering, we use AI2 Reasoning Challenge (Clark et al., 2018, ARC), Massive Multitask Language Understanding (Hendrycks et al., 2021, MMLU), and Reward Bench Reasoning (Lambert et al., 2024, RWDB-R). ARC focuses on science questions, MMLU focuses on interdisciplinary knowledge, RWDB-R focuses on correct vs incorrect solutions to math and programming problems. To provide a representative specialist train set $D^{\mathrm{s}} \sim \mathcal{D}^{\mathrm{s}}$, we split the questions into a train and test split, see Table 3 in Appendix B.

Our main results are reported with unified settings. For the classifier, the classification threshold is the main parameter. A threshold accepting 2.5% of D^{g} worked best in for the runs with 1.3B models over 120B tokens. For DoGE, the method is costly to apply over many data sources/clusters and we applied it over 64 clusters, i.e. learning a mixture weight of dimension 64. For importance sampling, the results presented in this section relies on 260k clusters. Later, Section 5 studies ablations and parameter sensitivity. Details on hyperparameters can be found in Appendix C.

4.1 Language Modeling Tasks

We evaluate specialist LMs on three domains from the Pile (Gao et al., 2021): medical (PubMed Central), encyclopedic (Wikipedia) and programming Q&A (StackExchange). We limit specialist training data from 14m tokens to the full Pile subset, up to 26.7B tokens, see Table 2 in Appendix B.

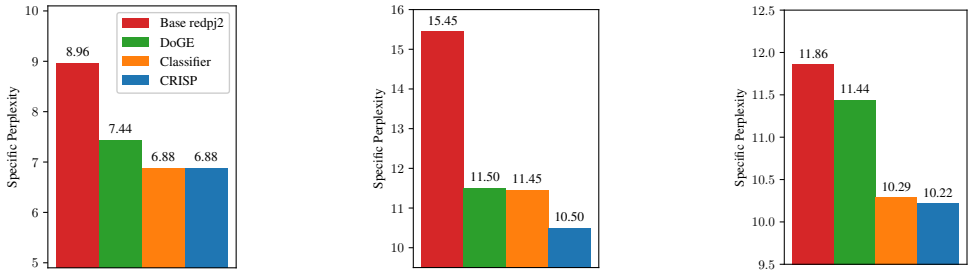
As baselines, we either train only on the in-domain (specialist) data without pretraining or we fine-tune a model pre-trained on Redpj2. We refer to the Redpj2 pretraining distribution as the base distribution. For task-dependent pretraining, we resample the Redpj2 pretraining set using a classifier, DoGE or importance sampling for each domain. The three methods have access to 14m specialist training tokens. In each case, the resampled pretraining set is used to train a 1.3B-parameter transformer model with the same hyperparameters as the Redpj2 baseline.

We report pretraining results in Figure 2, and the fine-tuning results in Figure 3. For each domain, the pretraining results evaluate models trained using the resampled Redpj2 examples. The fine tuning results evaluate models where each model pretrained on (resampled) Redpj2 has been further trained on the in-domain data itself (PubMed, StackExchange, Wikipedia). All experiments consider the same optimization effort and we validate the fraction of steps spent in fine-tuning, from 3%-ft with 14m tokens to 100%-ft with 26.7B tokens.

The pretraining results in Figure 2 show that the in-domain perplexity is better with task-dependent pretraining than with generic pretraining (base Redpj2) for all methods. This gain in perplexity comes as model training focuses on data close to the targeted domain: the model capacity is not used to fit the filtered out training data. Table 8 in Appendix E shows, for instance, that CRISP outperforms base on 97.3% of PubMed but reports worse perplexity on 95.9% of Redpj2.

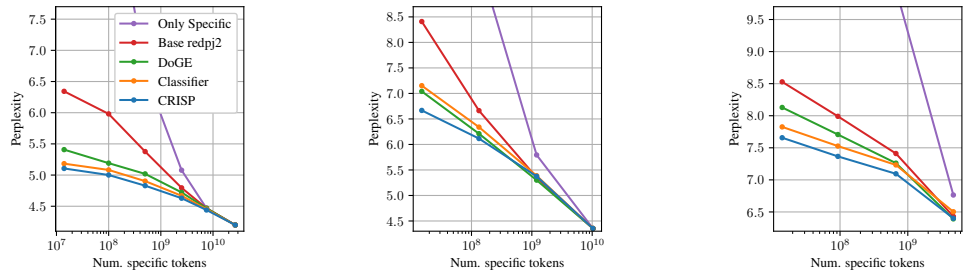
When we fine tune the pretrained models, the advantage of task-dependent pretraining is preserved, in Figure 3. Task-specific pretraining checkpoints are better starting points for fine-tuning than generic ones. This shows the complementarity between task-dependent pretraining and fine-tuning. Figure 3 also shows the necessity of pretraining: below 7B tokens, the "only specific" 1.3B model shows high perplexity. When comparing task-dependent pretraining methods, importance sampling

consistently performs better after fine-tuning, even when the pretraining results are close (e.g. classifier on PubMed, Wikipedia).



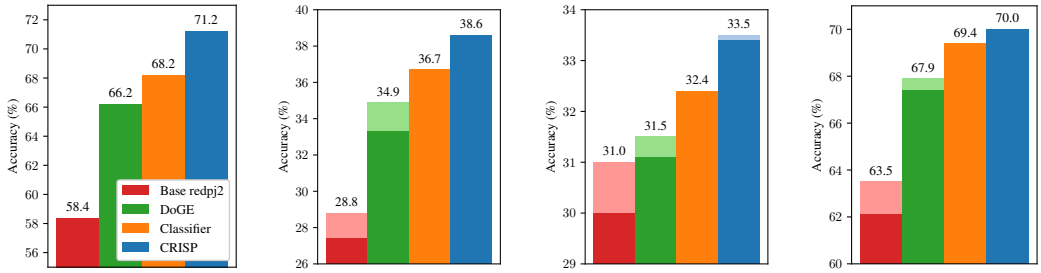
(a) PubMed (b) StackExchange (c) Wikipedia

Figure 2: **Pretraining perplexities for language modeling tasks**



(a) Pubmed Central (b) StackExchange (c) Wikipedia

Figure 3: **Fine-tuned perplexities for language modeling tasks.** Task-dependent pretraining is always better than generic pretraining. The ordering of the methods is unchanged from pretraining.



(a) Arc-E (b) Arc-C (c) MMLU (d) RWDB-R

Figure 4: **Accuracy for multiple choice question tasks.** Light colors indicate fine tuning improvements if any. The ordering of the methods is consistent across all 4 datasets.

4.2 Multiple Choice Question Tasks

Compared to LM, MCQ has much smaller specialist training sets per task, i.e. between 200k and 2m tokens, see Table 3 in Appendix B. The MCQ evaluation is also different: it uses accuracy and not perplexity. For each example, the model scores the concatenation of the question and a possible answer, for each proposed answer. The model is accurate when the correct answer is assigned the highest score (probability or normalized probability, see Appendix D). For MCQ tasks, unlike for LM tasks, the training loss (negative log likelihood) is therefore not closely tied to the test metric.

Despite these differences, we observe a similar benefit for task-dependent pretraining compared to task-agnostic (base) pretraining. Figure 4 displays a similar method ordering and CRISP is consis-

tently the best method. As a difference with LM tasks, we observe limited benefits from fine tuning, see Figure 4. Fine-tuning improves the base method on all datasets except ARC-E, but not enough to outperform task-specific pretraining, see Table 10 in Appendix F.

5 Analysis

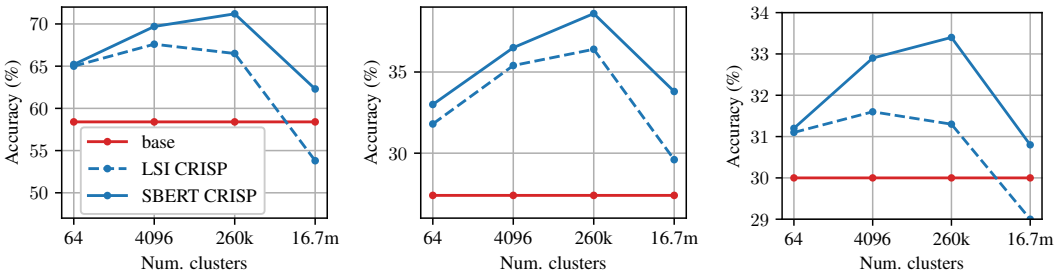
5.1 Clustering

We study the impact of the text representation for clustering and the number of clusters. We consider two representations for clustering, the SBERT embeddings used in all other experiments and LSI embeddings, see Section 4. We report their performance with 64, 4096, 262k and 16.7m clusters.

The representation is important: the examples in a cluster are close in the embedding space. Our independence assumption, Equation 1, assumes that the loss in a cluster c is the same regardless whether its data originates from D^{g} or D^{s}, i.e.

$$\mathcal{L}(D^{\text{g}} \cap K(c); \theta) \simeq \mathcal{L}(D^{\text{s}} \cap K(c); \theta). \tag{3}$$

In practice, it is sufficient that the embedding space reflects the similarity of the loss gradient, i.e. if the gradients of the loss over a generalist cluster $D^{\text{g}} \cap K(c)$ is correlated with the gradient over a specialist cluster $D^{\text{s}} \cap K(c); \theta)$, the model trained on the former improves on the later. Figure 5 shows that the SBERT representation is more adapted and yields better results than LSI for all settings.



(a) Arc-E (b) Arc-C (c) MMLU

Figure 5: **Accuracy for multiple choice question tasks varying the text representation for clustering and the number of clusters.** SBERT is more effective than LSI in all cases.

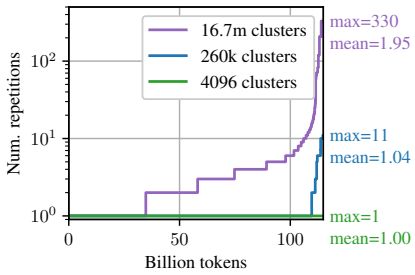


Figure 6: **Number of occurrences of each training example for CRISP on MMLU.** Repeated examples increase with the number of clusters.

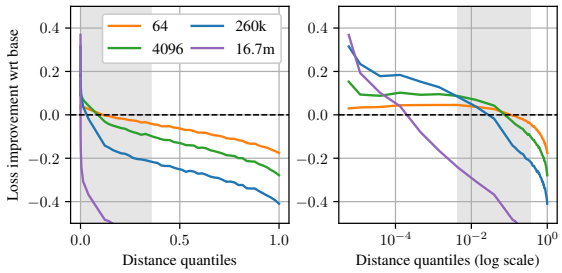


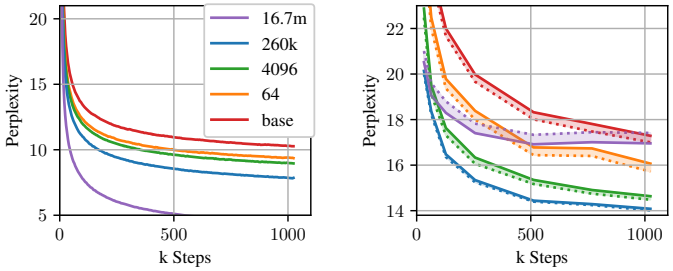
Figure 7: **Loss improvement on Redpj2 (valid) wrt base as a function of the SBERT distance to MMLU train.** Models with a large number of clusters are better than base in a small area near MMLU train. The gray area indicates the 25-75% quantiles for the MMLU test set.

The number of clusters is a trade-off and its optimum is at 260k for most of our experiments. There are multiple factors at play when the number of clusters varies. A smaller number of clusters implies larger clusters: our hypothesis, Equation 3, is then stronger as it assumes loss similarity on large areas of the embedding space. At the limit, with one cluster, this hypothesis assumes that the specialist loss and generalist loss are identical everywhere. Conversely, as the number of clusters

gets larger, the estimation of the cluster density on the small specialist set $P(c|\mathcal{D}^s) \simeq P(c|D^s)$ gets less accurate. The estimator risks overfitting, i.e. favoring clusters frequent in the training set D^s but not as frequent on other samples from $\mathcal{D}^s$. Increasing the number of clusters also risks reducing the effective training set size: the specialist data could be mostly concentrated in a few clusters, corresponding to a small fraction of the overall generalist set D^g.

We explore these aspects on MMLU. We first measure the number of repeated examples when training models with CRISP pretraining for different number of clusters. Figure 6 shows the number of repetitions for each quantile of the training set. Even for 16.7m clusters, only a small minority of training examples are repeated beyond 10 times and the average number of occurrences of the training is examples 1.94, well within commonly recommended values (Muennighoff et al., 2023; Xue et al., 2023). Even if exact repetitions do not account for the poorer performance of the 16.7m cluster setting, its training set might be less diverse and the model might generalize well only in a small neighborhood of its training set. We plot Figure 7 to evaluate if the Redpj2 examples with good perplexity concentrate around D^s, the MMLU training set. This plot shows that the benefit of CRISP over base is indeed correlated with the distance to D^s. As the number of cluster increases to 16.7m, the benefit over base concentrate in an area with very few samples. For comparison, we plot the 2 middle quartiles [0.25, 0.75] where most of the MMLU test data concentrate in gray. We remark that MMLU test data mostly lies in an area where the perplexity of IS 16.7m is low.

Figure 8 shows the perplexity during training for CRISP runs on MMLU. On Figure 8a, the perplexity is computed from the reweighed loss on Redpj2. This is the loss optimized during pretraining. It shows that when the number of cluster increases the sampled training set is less diverse and corresponds to an easier learning problem ($<$ 5 PPL). On Figure 8b, the perplexity is computed on the MMLU data itself, on the training set (plain) and on the test set (dotted). The scale of both plot is different: the resampled perplexities on Redpj2 are therefore not a good approximation of the MMLU perplexities. This quantifies the error resulting from our assumption, Equation 3. We also remark overfitting for 16.7m clusters, the only case with better MMLU perplexity for train than for test. Finally, we notice that the gray area in Figure 7 fails to show that 260k cluster would have the best perplexity, which highlights that SBERT distance to the training data is not the only factor explaining model performance.



(a) Perplexity on reweighted Redpj2

(b) Perplexity on MMLU train (plain) and test (dotted) sets.

Figure 8: **Perplexity for CRISP on MMLU with different number of clusters.** Y-scales on (a) and (b) are different.

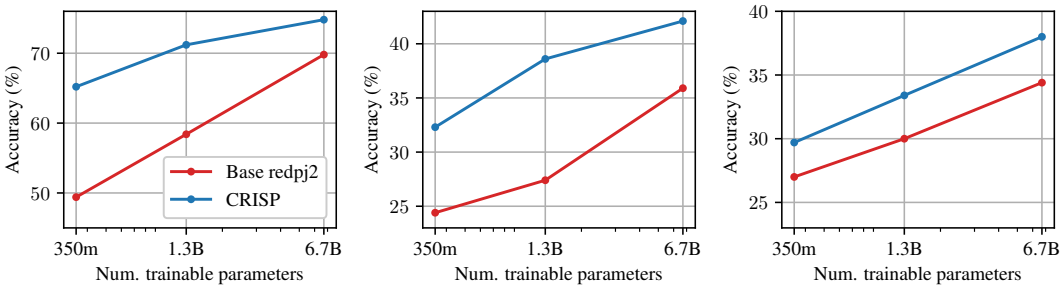
5.2 Model Size

This section compares CRISP and base at 3 model sizes. The benefit of task-dependent training is consistent across model sizes, see Figure 9. We consolidate results across sizes to report the training cost in GPUh vs accuracy in Figure 10. GPUh are measured in training hours per graphic processor (NVIDIA H100). We evaluate multiple checkpoints across model sizes and sort the checkpoints by training cost. The big dots mark transitions between model sizes: they show that the the 1.3B I.S. model outperforms the 6.7B base model on ARC. This shows substantial training speedups (∼30x). Of course, a smaller model is also beneficial at inference.

5.3 Different Amount of Training Data

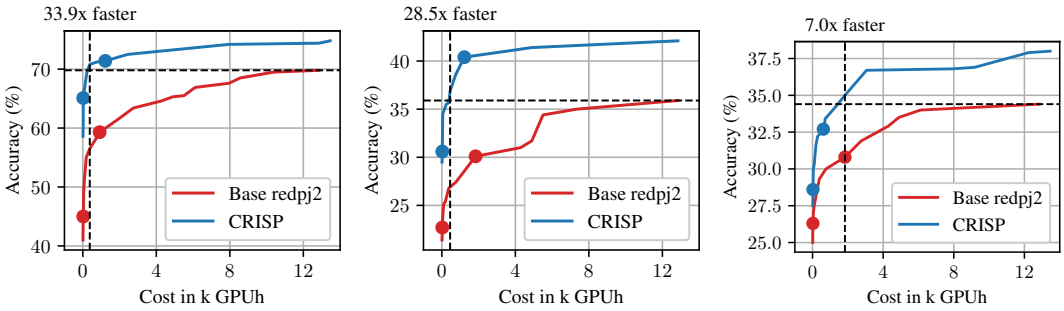
This section varies both (i) the amount of generalist data available to sample the CRISP dataset from and (ii) the amount of specialist data for inferring the CRISP weights.

When specialist data concentrates on a few clusters, CRISP often samples generalist data from the same clusters, which can be problematic when the generalist set is small. We restrict the pretraining



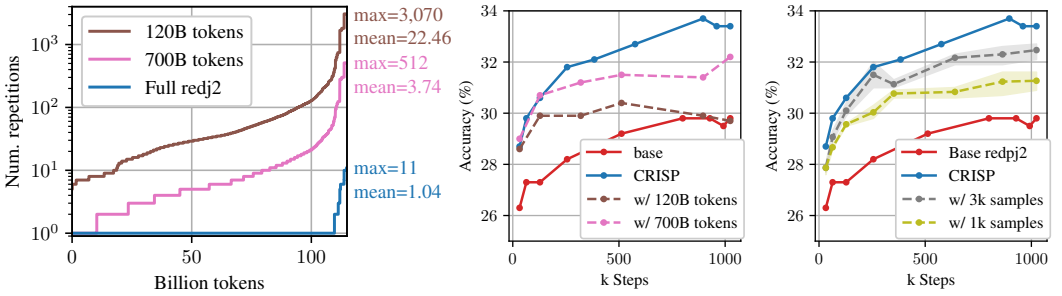
(a) Arc-E (b) Arc-C (c) MMLU

Figure 9: **Accuracy for multiple choice question tasks across model sizes.**



(a) Arc-E (b) Arc-C (c) MMLU

Figure 10: **Accuracy for multiple choice question tasks as a function of training cost.** The large dots mark the transition between model sizes (350m → 1.3B → 6.7B).



(a) Repetitions for less generic data (b) Acc. for less generic data (c) Acc. for less specific data

Figure 11: **MMLU with less training data.** When the generalist set D^g is small (a,b), the importance sampling method will up-sample a small part of D^g and this part will be seen multiple times during training. When this part is too small, the benefit of data selection vanishes. When the specialist set D^s is small (c), the importance sampling weights are poorly estimated and the importance sampled data might not be representative of the targeted task.

set to 700B and 120B tokens (downsampling Redpj2 by resp. $\sim$ 50x and $\sim$300x). Our pretraining runs use 120B tokens, so a base run never repeats in all settings. When CRISP is applied, some tokens are repeated. Figure 11a shows that, when restricting to 120B tokens, the number of repetition becomes high (22.5 on average) and CRISP is ineffective after 256k steps.

When the specialist dataset is smaller, Figure 11c shows that the errors in estimating cluster frequencies $P(c|\mathcal{D}^s)$ negatively impact end task accuracy. This suggests future work to improve this estimation for tasks with small D^s: e.g. specific set augmentations or task grouping.

Table 1: **Accuracy (%) for Task Transfer and Multitasking.** Importance Sampling on MMLU and on multitask improves all tasks compared to baseline.

Model		Evaluation Tasks				
		ARC-E	ARC-C	MMLU	RWDB-R	Multi
Base	Redpj2	58.4	27.5	30.1	62.2	45.1
CRISP	ARC	**71.3**	**38.6**	28.9	60.9	48.2
	MMLU	63.4	28.7	**33.4**	65.2	48.2
	RWDB-R	42.4	23.4	26.4	70.1	43.1
CRISP	Multi	68.6	34.1	31.1	**70.9**	**51.1**

5.4 Task-Transfer and Multitasking

We perform cross-task evaluation, i.e. targeting a task A and evaluating on a task B, we also pretrain a multitask models with CRISP averaged weights from multiple tasks. Our results for the 1.3B models are in Table 1, we also report cross-task evaluation results for different model sizes in Appendix H. Cross-task evaluations show that, perhaps unsurprisingly, the best results on a task A are obtained when pretraining for task A. Transfer differs across tasks: CRISP targeting MMLU gives better results than base for all tasks, which is not the case for CRISP targeting ARC or RWDB-R. The multi-task result which mixes the histograms with the same weight (1/3 for ARC, MMLU and RWDB-R) gives the best result on averaged multitask accuracy. Surprisingly, on RWDB-R, this setting slightly outperforms targeting RWDB-R itself.

5.5 Task-Dependent Continued Pretraining

We have seen the benefit of pretraining a model per task with CRISP in Figure 4. For tasks where pretraining cost is a concern, shorter pretraining runs still provide benefits, see Figure 10. Pretraining a multi-task model is also a cost-effective option, see Table 1. This section evaluates a third cost-effective option when targeting multiple tasks: continued pretraining. In this case, pretraining is divided into a generic pretraining phase and a task-dependent continued pretraining phase using CRISP. The compute cost of the generic pretraining can be shared across multiple tasks. Our results in Figure 12 show that even 10% of CRISP continued pretraining (i.e. generic pretraining for 928 steps out of 1,024) gives an accuracy (32.9%) close to a full CRISP run (33.4%). We also remark that the impact of continued pretraining is stronger than fine tuning a generic model on MMLU (31.0% accuracy), see Figure 4.

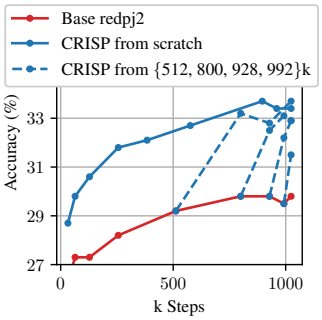


Figure 12: Continued Pretraining on MMLU

6 Conclusions

A specialist LM is interesting since a small specialist LM can outperform a larger generalist LM on its targeted domain while having a lower inference cost. We explore pretraining specialist LMs when little specialization data is available, a common setting that prevents pretraining of dedicated LMs. We evaluate different method that modifies the distribution of a generic training set guided by little specialist data. Our experiments highlight the benefit of clustered importance sampling: i.e. resampling the generic set such that its cluster histogram matches the specialist data. Our findings show that pretraining with this method provides strong models both for LM and question answering tasks. These benefits are studied across model size, training set size and clustering methods. We also explore ways to lower the training cost of specialist models addressing by showing their benefit on shorter training runs, continued pretraining and multitask settings.

Our work shows that a simple, scalable importance sampling method can provide effective specialist LMs, even from little specialization data. Since clustered importance sampling is modality-agnostic, we foresee extensions of this work to other modalities, including vision and audio.

ACKNOWLEDGEMENTS

We thank Matteo Pagliardini and Angelos Karathopoulos for their help throughout the project. We also thank Maartje ter Hoeve, Yizhe Zhang and Navdeep Jaitly for their suggestions and comments.

REFERENCES

Roee Aharoni and Yoav Goldberg. Unsupervised domain clusters in pretrained language models. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7747–7763, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.692. URL https://aclanthology.org/2020.acl-main.692.

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. A survey on data selection for language models, 2024. URL https://arxiv.org/abs/2402.16827.

J An, L Ying, and Y Zhu. Why resampling outperforms reweighting for correcting sampling bias with stochastic gradients. In *International Conference on Learning Representations*, 2021.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In Regina Barzilay and Mark Johnson (eds.), *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 355–362, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://aclanthology.org/D11-1033.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics, 2024. URL https://arxiv.org/abs/2310.10631.

Elliot Bolton, Abhinav Venigalla, Michihiro Yasunaga, David Hall, Betty Xiong, Tony Lee, Roxana Daneshjou, Jonathan Frankle, Percy Liang, Michael Carbin, and Christopher D. Manning. Biomedlm: A 2.7b parameter language model trained on biomedical text, 2024. URL https://arxiv.org/abs/2403.18421.

Zalán Borsos, MojmÃr MutnÃ½, Marco Tagliasacchi, and Andreas Krause. Data summarization via bilevel optimization. *Journal of Machine Learning Research*, 25(73):1–53, 2024. URL http://jmlr.org/papers/v25/21-1132.html.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Jiuhai Chen and Jonas Mueller. Automated data curation for robust language model fine-tuning, 2024. URL https://arxiv.org/abs/2403.12776.

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

Paul Cook and Marco Lui. langid.py for better language modelling. In Paul Cook and Scott Nowson (eds.), *Proceedings of the Australasian Language Technology Association Workshop 2012*, pp. 107–112, Dunedin, New Zealand, December 2012. URL https://aclanthology.org/U12-1014.

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. *Advances in Neural Information Processing Systems*, 35:26698–26710, 2022.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41 (6):391–407, 1990.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1286–1305, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.98. URL https://aclanthology.org/2021.emnlp-main.98.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, ..., Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Susan T Dumais. Latent semantic analysis. *Annual Review of Information Science and Technology (ARIST)*, 38:189–230, 2004.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. Doge: Domain reweighting with generalization estimation, 2024. URL https://arxiv.org/abs/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. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027, 2021.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

David Grangier, Pierre Ablin, and Awni Hannun. Adaptive training distributions with scalable online bilevel optimization, 2023. URL https://arxiv.org/abs/2311.11973.

David Grangier, Angelos Katharopoulos, Pierre Ablin, and Awni Hannun. Projected language models: A large model pre-segmented into smaller ones. In *ICML Workshop on Foundation Models in the Wild*, 2024a.

David Grangier, Angelos Katharopoulos, Pierre Ablin, and Awni Hannun. Specialized language models with cheap inference from limited domain data, 2024b. URL https://arxiv.org/abs/2402.01093.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023. URL https://arxiv.org/abs/2306.11644.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740.

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

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. *arXiv preprint arXiv:2202.00622*, 2022.

Dan Iter and David Grangier. On the complementarity of data selection and fine tuning for domain adaptation, 2021. URL https://arxiv.org/abs/2109.07591.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Marcin Junczys-Dowmunt. Dual conditional cross-entropy filtering of noisy parallel corpora. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor (eds.), *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pp. 888–895, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6478. URL https://aclanthology.org/W18-6478.

Jean Kaddour. The minipile challenge for data-efficient language models, 2023. URL https://arxiv.org/abs/2304.08442.

T Kerner. Domain-specific pretraining of language models: A comparative study in the medical field. *arXiv preprint arXiv:2407.14076*, 2024.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Xiang Kong, Tom Gunter, and Ruoming Pang. Large language model-guided document selection, 2024. URL https://arxiv.org/abs/2406.04638.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. Biomistral: A collection of open-source pretrained large language models for medical domains, 2024. URL https://arxiv.org/abs/2402.10373.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling, 2024. URL https://arxiv.org/abs/2403.13787.

Edmund Lee. What happens when a publisher becomes a megapublisher? *New York Times*, 2021. URL https://www.nytimes.com/2021/02/25/books/penguin-random-house-simon-schuster-publishing.html.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8424–8445, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.577. URL https://aclanthology.org/2022.acl-long.577.

Patrick Lewis, Myle Ott, Jingfei Du, and Veselin Stoyanov. Pretrained language models for biomedical and clinical tasks: Understanding and extending the state-of-the-art. In Anna Rumshisky, Kirk Roberts, Steven Bethard, and Tristan Naumann (eds.), *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pp. 146–157, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.clinicalnlp-1.17. URL https://aclanthology.org/2020.clinicalnlp-1.17.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL https://arxiv.org/abs/2206.14858.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models, 2024. URL https://arxiv.org/abs/2406.11794.

Miaofeng Liu, Yan Song, Hongbin Zou, and Tong Zhang. Reinforced training data selection for domain adaptation. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1957–1968, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1189. URL https://aclanthology.org/P19-1189.

Pratyush Maini, Skyler Seto, Richard Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. Rephrasing the web: A recipe for compute and data-efficient language modeling. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14044–14072, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.757.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. Deep learning–based text classification: a comprehensive review. *ACM computing surveys (CSUR)*, 54(3):1–40, 2021.

Robert C. Moore and William Lewis. Intelligent selection of language model training data. In Jan Hajič, Sandra Carberry, Stephen Clark, and Joakim Nivre (eds.), *Proceedings of the ACL 2010 Conference Short Papers*, pp. 220–224, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/P10-2041.

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

Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Reuse, don't retrain: A recipe for continued pretraining of language models, 2024. URL https://arxiv.org/abs/2407.07263.

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

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e6385d39ec9394f2f3a354d9d2b88eec-Paper.pdf.

Aniruddh Raghu, Jonathan Lorraine, Simon Kornblith, Matthew McDermott, and David K Duvenaud. Meta-learning to improve pre-training. *Advances in Neural Information Processing Systems*, 34:23231–23244, 2021.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2024. URL https://arxiv.org/abs/2308.12950.

Sebastian Ruder and Barbara Plank. Learning to select data for transfer learning with Bayesian optimization. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 372–382, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1038. URL https://aclanthology.org/D17-1038.

Chris Seiffert, Taghi M Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Resampling or reweighting: A comparison of boosting implementations. In *2008 20th IEEE international conference on tools with artificial intelligence*, volume 1, pp. 445–451. IEEE, 2008.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh neural machine translation systems for WMT 16. In Ondřej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Liane Guillou, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Pavel Pecina, Martin Popel, Philipp Koehn, Christof Monz, Matteo Negri, Matt Post, Lucia Specia, Karin Verspoor, Jörg Tiedemann, and Marco Turchi (eds.), *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pp. 371–376, Berlin, Germany, August 2016a. Association for Computational Linguistics. doi: 10.18653/v1/W16-2323. URL https://aclanthology.org/W16-2323.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016b. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Selective annotation makes language models better few-shot learners. In *International Conference on Learning Representations (ICLR)*, 2023.

Together AI Team. Redpajama-data-v2: An open dataset with 30 trillion tokens for training large language models, October 2023. URL https://www.together.ai/blog/redpajama-data-v2.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Huy V. Vo, Vasil Khalidov, Timothée Darcet, Théo Moutakanni, Nikita Smetanin, Marc Szafraniec, Hugo Touvron, Camille Couprie, Maxime Oquab, Armand Joulin, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Automatic data curation for self-supervised learning: A clustering-based approach, 2024. URL https://arxiv.org/abs/2405.15613.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. Denoising neural machine translation training with trusted data and online data selection. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor (eds.), *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 133–143, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6314. URL https://aclanthology.org/W18-6314.

Zirui Wang, Yulia Tsvetkov, Orhan Firat, and Yuan Cao. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models, 2020. URL https://arxiv.org/abs/2010.05874.

G. Wenzek, M. A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, and E. Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pp. 4003–4012, 2020.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance, 2023. URL https://arxiv.org/abs/2303.17564.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning (ICML)*, 2024.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. Pixiu: A large language model, instruction data and evaluation benchmark for finance, 2023a. URL https://arxiv.org/abs/2306.05443.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 69798–69818. Curran Associates, Inc., 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/dcba6be91359358c2355cd920da3fcbd-Paper-Conference.pdf.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. *CoRR*, abs/2302.03169, 2023c. doi: 10.48550/ARXIV.2302.03169. URL https://doi.org/10.48550/arXiv.2302.03169.

Zhipeng Xu, Zhenghao Liu, Yukun Yan, Zhiyuan Liu, Ge Yu, and Chenyan Xiong. Cleaner pretraining corpus curation with neural web scraping, 2024. URL https://arxiv.org/abs/2402.14652.

Fuzhao Xue, Yao Fu, Wangchunshu Zhou, Zangwei Zheng, and Yang You. To repeat or not to repeat: Insights from scaling llm under token-crisis. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural*

Information Processing Systems, volume 36, pp. 59304–59322. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/b9e472cd579c83e2f6aa3459f46aac28-Paper-Conference.pdf.

Jinsung Yoon, Sercan Arik, and Tomas Pfister. Data valuation using reinforcement learning. In *International Conference on Machine Learning*, pp. 10842–10851. PMLR, 2020.

Zichun Yu, Spandan Das, and Chenyan Xiong. Mates: Model-aware data selection for efficient pretraining with data influence models, 2024. URL https://arxiv.org/abs/2406.06046.

Shaokun Zhang, Xiaobo Xia, Zhaoqing Wang, Ling-Hao Chen, Jiale Liu, Qingyun Wu, and Tongliang Liu. IDEAL: Influence-driven selective annotations empower in-context learners in large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

APPENDIX

A SCALABLE CLUSTERING

We cluster the Redpj2 dataset with hierarchical clustering. We build a clustering tree. Each node in the tree is associated with a cluster centroid. The examples traverse the tree from top to bottom, selecting the node corresponding to the closest centroids among the current node's children.

The training of the tree proceed from root to leaves. Iteratively, a new level is built by applying k-means to a subset of the examples belonging to each node. We built a tree of depth up to 4, always splitting nodes in 64 clusters. For k-means, we normalize the Euclidean norm of the vectors prior to clustering. We train the model via Expectation Maximization using k-means++ initialization (Arthur & Vassilvitskii, 2006). At each step, we sample 6,400 new examples. With 20 steps, we visit 128k examples. To ensure a cluster distribution close to uniform, we monitor the cluster sizes at each assignment steps. If a cluster is larger than our balancing limit $(0.022 \simeq 1.5 * 1/64)$, we split evenly at random its assignments with the smallest cluster, as suggested by Jegou et al. (2010). The clustering hyper-parameters can be found in Table 6.

B DATASET STATISTICS

Table 2: LM Datasets.

		Redpj2	PubMed	StackExchange	Wikipedia
Dataset role		*generalist*	*specialist*	*specialist*	*specialist*
Train	Num. tokens	34.6T	26.7B	10.3B	4.68B
	Num. documents	24.0B	2.94m	15.4m	5.79m
Test	Num. tokens	359m	52.4m	20.1m	14.1m
	Num. documents	248k	5.82k	29.9k	17.4k

Table 3: MCQ Datasets.

		ARC-E	ARC-C	MMLU	RWDB-R
Train	Num. tokens	143k	79.6k	2.05m	426k
	Num. questions	1.18k	578	6.95k	736
	Avg. tokens per choice	30.3	34.5	73.5	289
Test	Num. tokens	144k	87.7k	2.09m	408k
	Num. questions	1.19k	593	7.09k	695
	Avg. tokens per choice	30.2	37.0	73.6	293
Num. choices per question		4	4	4	2

Our generic pretraining set is Redpj2 (Together AI Team, 2023). We use the head+middle English version of the dataset, i.e. web-documents with a high density of English text. Our specialization datasets for language modeling are much smaller, see Table 2. Compared the LM tasks, the multiple choice question tasks have even smaller specialization training set, i.e. between 200k and 2m tokens, see Table 3. For the LM data, we rely on the train split provided by Pile Gao et al. (2021). For the MCQ data, we split each evaluation set into an equal sized train and test set uniformly at random. This provides a representative specialist train set $D^{\text{s}} \sim \mathcal{D}^{\text{s}}$. This also avoids cross-contamination between tasks, e.g. the official training set of MMLU contains ARC which would prevent the task transfer experiments in Section 5.4.

C ARCHITECTURES & HYPERPARAMETERS

Our architecture configurations are borrowed from Brown et al. (2020) and described in Table 4. We report the data selection hyperparameters in Table 5 and the clustering hyper-parameters in Table 6.

Table 4: Model Hyperparameters

Num. parameters	350m	1.3m	6.7B
Architecture			
Embedding dim.	1,024	2,048	4,096
Latent dim.	4,096	8,192	16,384
Num. heads	16	16	32
Depth	24	24	32
Context limit	1,024	1,024	1,024
Optimization			
Batch size	96k	115k	1.04m
Learning rate	1e-4	1e-4	3e-4
Grad clipping	5.0	5.0	0.1
Steps	400k	1m	340k
Num. train tokens	40B	120B	350B

Table 5: Data-Selection Hyperparameters

Method	Parameter	Range
Classifier	Regularization strength	{None, 1000, 100, 10, 1, 0.1, 0.01, 0.001}
	Threshold quantiles	{0.5, 0.6, 0.7, 0.75, 0.8, 0.9, 0.95, … …, 0.975, 0.98, 0.9875, 0.99, 0.995, 0.9975}
DoGE	Num. clusters	64
	Proxy model size	Transformer base, 110m parameters
	Proxy model optimization	32k batch size, 1e-4 learning rate, 100k steps
	Bregman coefficient μ	5e-4
	Transferred weights	{run average, last 20 step average}
Importance S.	Num. clusters	{64, 4096, 262k, 16.7m}

D MCQ EVALUATION

For the multiple-choice-question tasks, we use the LM eval harness (Gao et al., 2024). For each task, the evaluated model estimates the (log) probablity of each answer a given the context c, i.e. $\log P(a|c)$. The question contains the task prompt concatenated with the current question, while the answer contains the answer text. With this strategy, the model does not have access to all proposed answer choices in the prompt. Table 7 reports our prompt. For all evaluation, we use the above prompt without example questions, i.e. a zero-shot evaluation (Brown et al., 2020). Accuracy is computed by verifying if the highest score is assigned to the correct answer. The scores correspond to log probabilities for ARC-E and RWDB-R, while ARC-C, MMLU uses normalized scores, i.e. log probabilities divided by the number of characters in the answer.

E SUPPLEMENTARY RESULTS FOR LANGUAGE MODELING TASKS

We measure the fraction of examples where the pretrained model is better than the base model. We measure this rate both on held-out data from $\mathcal{D}^{\text{g}}$ (measured on the 360m tokens from Redpj2 valid) and on held-out data from $\mathcal{D}^{\text{g}}$ (measured on the full Pile validation set). The results in Table 8 shows that the model trained with importance sampling improves perplexity on most specialist documents (right column). Its training on the importance sampled distribution utilize model capacity mostly on data close to the domain of interest, this relieves the model from fitting well most of the generic data, hence most generic documents have higher perplexity with CRISP (left column).

For completeness, we also report the perplexity numbers from Figure 3 in Table 9.

Table 6: Hierarchical Clustering Hyperparameters

Parameter	Range
Tree depth	4
Tree arity	64
Balancing limit	0.022
Number of samples per step	6,400
Number of steps	20
SBERT model	`MiniLM-L6-v2`
SBERT emb. dim.	384
LSI dim.	256

Table 7: Task prompts (non-bold) for the multiple-choice-question tasks.

AI2 Reasoning Challenge (ARC) Easy and Challenge
Question: <question>\n Answer: **<answer>**
Massive Multitask Language Understanding (MMLU)
The following are multiple choice questions (with answers) about <topic>.\n Question: <question>\n Answer: **<answer>**
Rewardbench Reasoning (RWB-R)
Follow the instructions below.\n Instructions: <question>\n Answer: **<answer>**

F Supplementary Results for Multiple Choice Questions

Table 10 reports the MCQ results before and after fine-tuning, i.e the accuracy numbers from Figure 4. Fine-tuning on the small MCQ train sets optimizing log-likelihood does not always benefit end-task accuracy.

G Comparing the results of DoGE and Importance Sampling

We observe in Table 11 that the pretraining results of DoGE and importance sampling on 64 clusters are close. Both methods pretrain models by sampling the clustered generalist data according to the cluster weights. If both methods would infer the same cluster weights, their pretraining runs would be identical. We therefore ask if the similar results are due to similar cluster weights. Figure 13 compares the cluster weights for both methods. The top clusters for both methods are similar, but their histograms are not identical. This shows that similar pretraining results can be obtained with different weights.

Table 8: **Fraction of examples with lower perplexity with importance sampling than with base.** Compared to base, CRISP models specialize: they performs better on most specialist examples and worse on most generic examples.

	Generalist $\mathcal{D}^g$ (Redpj2)	Specialist $\mathcal{D}^s$ (Pile subset)
PubMed	6.1%	97.3%
StackExchange	2.9%	92.6%
Wikipedia	12.4%	86.7%

Table 9: **Perplexity on language modeling tasks after fine-tuning.** These tables reports the perplexity numbers from Figure 3.

(a) PubMed

Specific tokens	14m	100m	500m	2.5B	7.5B	26.7B
Only Specific	25.73	10.09	6.64	5.08	4.47	4.20
Base redpj2	6.34	5.98	5.38	4.79	4.47	4.20
DoGE	5.41	5.19	5.02	4.72	4.47	4.20
Classifier	5.18	5.08	4.90	4.67	4.45	4.20
CRISP	5.11	5.00	4.83	4.63	4.44	4.20

(b) StackExchange

Specific tokens	15m	133m	1.2B	10.3B
Only Specific	23.93	9.60	5.79	4.35
Base redpj2	8.41	6.66	5.35	4.35
DoGE	7.04	6.21	5.30	4.35
Classifier	7.15	6.34	5.39	4.35
CRISP	6.67	6.12	5.38	4.35

(c) Wikipedia

Specific tokens	14m	93m	668m	4.7B
Only Specific	57.13	18.22	9.97	6.76
Base redpj2	8.53	7.99	7.41	6.43
DoGE	8.13	7.71	7.26	6.39
Classifier	7.83	7.53	7.23	6.50
CRISP	7.66	7.37	7.09	6.40

Table 10: **MCQ Accurary.** Fine tuning results are dashed when not improved from pretraining. This table reports the accuracy numbers from Figure 4.

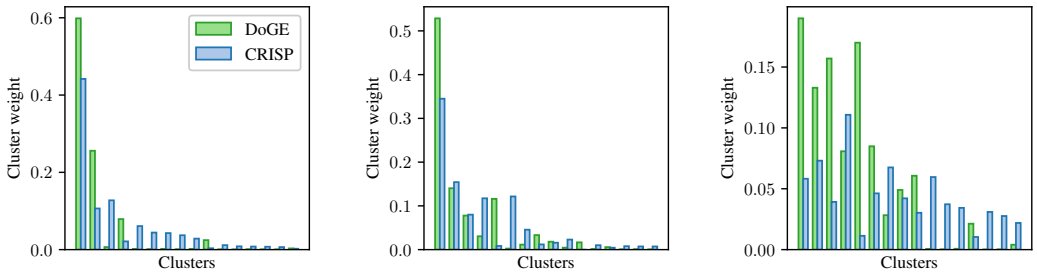
	ARC-E		ARC-C		MMLU		RWDB-R	
	Pretr.	+ft	Pretr.	+ft	Pretr.	+ft	Pretr.	+ft
Base redpjv2	58.4	–	27.4	28.8	30.0	31.0	62.1	63.5
DoGE	66.2	–	33.3	34.9	31.0	31.5	67.4	67.9
Classifier	68.2	–	36.7	–	32.4	–	69.4	–
CRISP	71.2	–	38.6	–	33.4	33.5	70.0	–

H TASK TRANSFER FOR 350M, 1.3B AND 7B MODELS

Table 12 complements the task-transfer results from Table 1 in Section 5.4 with the results across different model sizes. The importance sampling models trained with MMLU histograms outperform the base models on all tasks for all model sizes.

Table 11: DoGE & CRISP on 64 Clusters

	LM PPL↓	MCQ Acc (%) ↑		
	PubMed	ARC-E	ARC-C	MMLU
DoGE	7.44	66.2	33.3	31.0
CRISP	7.28	65.2	33.0	31.2



(a) PubMed (b) ARC (c) MMLU

Figure 13: DoGE vs CRISP weights with 64 clusters. We report the top-16 clusters sorted by mean weight across methods.

Table 12: Accuracy (%) for Task Transfer on 350m, 1B and 7B models.

Model		Evaluation Tasks				
		ARC-E	ARC-C	MMLU	RWDB-R	Multi
350m	Base	49.5	24.5	27.0	57.6	40.5
	CRISP ARC	65.3	31.5	27.4	58.4	44.7
	CRISP MMLU	55.6	26.3	29.8	61.3	44.0
1B	Base	58.4	27.5	30.1	62.2	45.1
	CRISP ARC	71.3	38.6	28.9	60.9	48.2
	CRISP MMLU	63.4	28.7	33.4	65.2	48.2
7B	Base	69.9	35.9	34.4	64.9	50.7
	CRISP ARC	74.5	42.2	32.6	62.4	51.1
	CRISP MMLU	70.0	37.6	38.0	67.5	53.1