Target-Aware Language Modeling via Granular Data Sampling

Ernie Chang[♠] **Pin-Jie Lin**[♣] **Yang Li**[♦] **Changsheng Zhao**[♠]
Daeil Kim[♠] **Rastislav Rabatin**[♠] **Zechun Liu**[♠] **Yangyang Shi**[♠] **Vikas Chandra**[♠]
[♠]AI at Meta
[♣]Virginia Tech
[♦]Iowa State University
erniecyc@meta.com, pinjie@vt.edu, yangli1@iastate.edu

Abstract

Language model pretraining generally targets a broad range of use cases and incorporates data from diverse sources. However, there are instances where we desire a model that excels in specific areas without markedly compromising performance in other areas. A cost-effective and straightforward approach is sampling with low-dimensional data features, which allows to select large-scale pretraining data for domain-specific use cases. In this work, we revisit importance sampling with n-gram features consisting of multi-granular tokens, which strikes a good balance between sentence compression and representation capabilities. We observed the sampled data to have a high correlation with the target downstream task performance *while preserving its effectiveness on other tasks*. This leads to the proposed data sampling paradigm where language models can be pretrained more efficiently on selected documents. On eight benchmarks we demonstrate with ~1% of the data, pretrained models perform on par with the full RefinedWeb data and outperform randomly selected samples for model sizes ranging from 125M to 1.5B.

1 Introduction

Language model pretraining is the cornerstone of universal language models (LMs), creating general-purpose representations to excel across a variety of NLP downstream tasks (John and Draper, 1975; Murphy, 2012). This process often involves the use of vast amounts of text, sometimes measured in billions or even trillions of tokens from webpages (Abnar et al., 2022; Kaplan et al., 2020). However, there are instances where a model needs to perform well in specific domains while not compromising performance in others. This necessitates the use of data selection methods to determine which potential data points should be included in the training dataset and how to effectively sample from these selected points (Albalak et al., 2024).

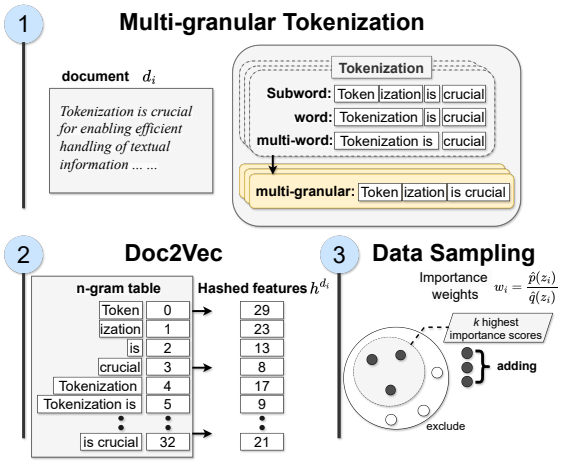

Figure 1: Multi-granular tokenization for more modular feature vectors used in importance sampling. (1) Given a document d_i, it undergoes featurization as a sequence of multi-granular tokens. (2) Subsequently, the document is transformed into a fixed-sized feature representation via hashing N-grams. (3) We measure its significance through the enhancement weight w_i and select a subset of k representative data points from the original target distributions through re-sampling.

One approach to reducing data size is *coreset selection*, which involves selecting a small, representative subset of data (Du et al., 2022). Coresets can significantly decrease computational costs while maintaining robust performance. In this work, we explore the optimization of coresets towards a target data distribution, but relaxing the data sampling process to reduce domain biases.

Here we revisit importance sampling (Rubin, 1988; Xie et al., 2023) by proposing to utilize tokens of different granularities as features, ranging from subword, word, to multi-word (or n-gram) tokens (Shown in Figure 1). We observed empirically that by controlling the granularity of tokens in the tokenizer, we can construct coresets with less domain biases – fine-grained tokens capture more task knowledge while coarse-grained tokens preserve general information. Thus, we experiment with adapting the vocabulary set of pretrained tokenizers

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 12927–12935
November 12-16, 2024

to data from specific tasks, and modulating token granularity in the vocabulary to maintain generality. To demonstrate the efficacy of multi-granular sampling, we use eight downstream tasks as target tasks with Llama-3's tokenizer being the base tokenizer, where its vocabulary set is adapted to task data. This process creates a domain-specific vocabulary set which we use to featurize text documents[1] for target-aware data sampling with higher quality sampled data. Specifically, we demonstrate that these pretrained models, using only about 1% of the data, perform on par with the full RefinedWeb data and outperform randomly selected samples across eight benchmarks (§4). This contributes to the ongoing discourse on optimizing pretraining data to improve computational efficiency and model performance. Unlike past approaches which can be computationally-intensive (Wenzek et al., 2020; Wettig et al., 2024; Muennighoff et al., 2024) or simple but easily biased towards target data distribution (Xie et al., 2023), our approach is unique in two significant ways:

1. We proposed an algorithm to merge a pretrained tokenizer with multi-granular tokens and empirically showed that it yields highly efficient n-gram features that has high correlation with downstream task performances.

2. Leveraging our findings, we improve upon the importance-based data sampling technique by adapting a general vocabulary set to the target vocabulary. This creates a better representation of data that enhances model performances in target tasks, while maintaining decent performance in non-target tasks.

2 The Approach

Selecting samples from large-scale datasets such as RefinedWeb (Penedo et al., 2023) is slow and expensive. A tractable solution is to encode each document as a vector using n-gram features that can be computed easily. Here we assume a small number of target text examples D_{task} from a target distribution p and a large raw dataset D_{raw} from a distribution q with N examples. We aim to select k examples ($k \ll N$) from the raw dataset that are similar to the target.

We adopted the importance sampling technique as in Xie et al. (2023) which selects examples that align with target distribution. The technique provides a tractable importance estimate of each text and applies importance sampling on a feature space $\mathbb{Z}$ that provides the necessary structure. The feature extractor $h : \mathbb{X} \rightarrow \mathbb{Z}$ is used to transform the input x into features $z = h(x)$. The resulting raw and target feature distributions are q_{feat} and p_{feat}, respectively. Our objective is to select examples whose features align with the target feature distribution p_{feat}. To do so, features q_{feat} and p_{feat} are extracted (Figure 1) using n-grams extracted from each tokenized document using an adapted tokenizer. Each n-gram is mapped to a key in the hash table where the ids of the table define a fixed-size embedding, and each key maps to the n-gram count. Then, the importance weights are computed for each featurized example $z_i = h(x_i)$ from the N raw examples, with the weight $w_i = \frac{\hat{p}_{\text{feat}}(z_i)}{\hat{q}_{\text{feat}}(z_i)}$. The final step involves sampling, where we select k examples without replacement from a categorical distribution, the probabilities of which are given by $\frac{w_i}{\sum_{i=1}^{N} w_i}$.

Tokenizer Adaptation. Here we adapt the vocabulary to the target data. To derive target vocabulary $V(t)$, we use Llama-3 tokenizer's vocabulary V_{start} as the starting point and merge V_{start} with V_{task} which is learned from task data D_{task}. In constructing V_{task}, we make sure to include multi-granular tokens (i.e. subwords, words and multi-words), where V_{task} is then merged with V_{start} to form $v(t-1)$. Next, we incrementally remove tokens from $v(t-1)$ to obtain $v(t)$, where we minimize the distance from the original vocabulary set such that a less biased document feature can be extracted as n-gram vectors. We first define a metric to measure the quality of vocabulary set on a corpus, following Xu et al. (2021), which proposed to learn optimal vocabulary by maximizing the vocabulary utility metric $\mathcal{H}_v$ (Samuelson, 1937) computed as:

$$\mathcal{H}_v = -\frac{1}{l_v} \sum_{j \in v} P(j) \log P(j), \tag{1}$$

where $P(j)$ is the relative frequency of token j from the target data and l_v is the average length of tokens in vocabulary v. For any vocabulary, its entropy score $\mathcal{H}_v$ can be calculated based on a vocabulary from its previous step. The optimization problem can be formulated as:

$$\underset{v(t-1), v(t)}{\arg\min} \left[\mathcal{H}v(t) - \mathcal{H}v(t-1)\right] \tag{2}$$

[1] We employ n-gram features in the same way as Xie et al. (2023).

Approach	ARC-Easy	ARC-Hard	BOOLQ	PIQA	SIQA	HELLASWAG	OBQA	WINOGRANDE	Avg.
125M params									
RANDOM	45.74	27.64	59.38	66.41	41.02	37.13	34.77	52.64	45.59
N-GRAM	43.59	27.20	57.76	72.17	42.29	45.89	31.05	50.24	46.27
MULTI-GRANULAR	44.24	30.13	57.98	72.71	41.16	46.68	35.74	52.10	**47.59**
350M params									
RANDOM	52.12	29.20	62.38	69.04	42.92	46.21	40.23	54.39	49.56
N-GRAM	49.58	30.52	62.23	75.68	42.38	57.18	40.72	53.91	51.52
MULTI-GRANULAR	49.28	31.74	60.06	76.51	42.09	56.36	41.99	53.88	**51.61**
500M params									
RANDOM	54.17	31.84	59.86	70.85	43.07	49.02	39.55	56.49	50.61
N-GRAM	49.65	31.02	63.09	76.29	42.83	58.17	41.56	54.47	52.13
MULTI-GRANULAR	52.64	30.71	53.86	76.56	43.02	60.40	47.59	54.59	**52.42**
1.5B params									
RANDOM	58.89	32.23	51.56	72.07	42.58	55.05	41.80	57.71	51.49
N-GRAM	53.91	34.28	60.57	79.49	44.48	66.42	40.43	54.49	54.26
MULTI-GRANULAR	55.47	34.28	59.11	78.22	43.02	69.45	39.84	56.88	**54.53**

Table 1: Results over all downstream tasks selecting based on all task validation sets, each in terms of its respective metric. Here we compare **Random**, **N-gram**, and **Multi-granular** data selection techniques sampling for 1% data of RefinedWeb (Penedo et al., 2023) and pretrained with ~700 million tokens. We observe two major trends: (1) performance improves with increase in number of parameters. However, the improvement begins to plateau as model becomes larger. (2) **Multi-granular** has the best overall performance across all benchmarks, despite being worse on some individual tasks.

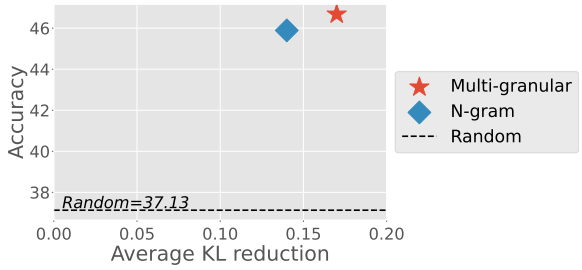

Figure 2: The plot of average KL reduction and the performance on HellaSwag. We measure how the granularity of tokens used for coreset selection reduces KL divergence to the target distribution. This reduction is compared to random sampling from The RefinedWeb, suggesting a strong correlation between KL reduction and downstream performance (Pearson r = 0.82).

where $v(t)$ and $v(t-1)$ are two sets containing all vocabularies with upper bound of size $|v(t)|$ and $|v(t-1)|$ respectively. In our implementations, we set $|v(t)| = 10k$, where $t = 10$; and $|v(0)|$ is the default Llama-3 tokenizer's vocabulary size. Here $\arg\min$ aims to find the vocabulary from V_{start} with the minimum entropy difference. In this optimization process, we include varying granularity of tokens in the vocabulary ranging from n-gram to multi-granular tokens. To evaluate the effectiveness of this approach, we compare the overall Kullback–Leibler (KL) divergence reduction and downstream task performances with different granularities. We observed that a mix of all granularities yields the best results overall on the downstream task (See Figure 2) – where there is a clear trend of increasing performance with mixed granularities. However, a finer granularity also decreases the representation power of the features, as seen from the degradation in using subword tokens alone. Empirically, we found that the proposed tokenizer adaptation technique yields significant advantage over naive merging of two vocabulary sets (See Appendix A.2).

3 Experimental Setup

Network, Training Details and Evaluation. We pretrain the decoder-only transformer using causal language modeling objectives on selected datasets, averaging over three initialization runs for each configuration, where model weights were randomly initialized. The language models varied in size, with 125M, 350M, 500M, and 1.5B parameters. This range allowed us to explore how model complexity impacts the final results. Pretraining was conducted on a distributed computing setup with 32 GPUs across 4 nodes, each equipped with an H100 graphics card.

Coreset Selection. We evaluated our proposed **Multi-granular** selection approach against random selection (**Random**) and compared it with the same sampling algorithm using word-based **N-gram** features. Importance sampling (Xie et al., 2023) was employed for all feature types.

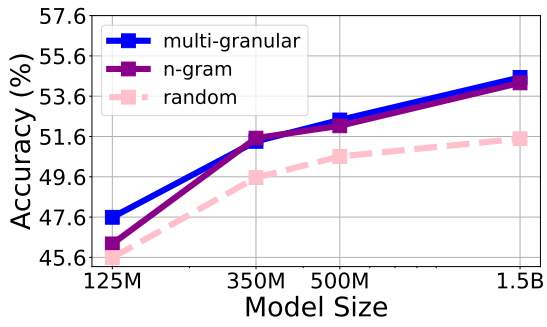

Figure 3: Zero-shot performances averaged over eight tasks computed across all model sizes, where emergent characteristic can be observed at the model of size 350M parameters.

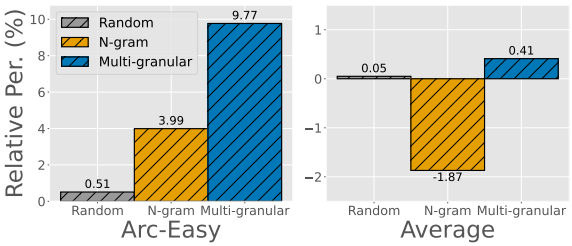

Figure 4: Comparison of **Multi-granular** n-grams with **N-gram** and **Random** baseline across eight tasks using 125M models, trained solely on data selected with ARC-Easy data as the target. Relative performance is used. We observe that multi-granular features enable the model to consistently outperform the baseline despite task-specific biases in the data. We performed similar experiments for all other benchmarks in the appendix, where for all eight tasks, the same pattern is observed that multi-granular n-grams yield almost no degradation across benchmarks.

Datasets. We evaluate the models on eight common sense reasoning tasks in a zero-shot fashion, including ARC-easy, ARC-challenge (Clark et al., 2018), BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), OBQA (Mihaylov et al., 2018), and WinoGrande (Sakaguchi et al., 2021).

4 Main Results

Overall, language models varied in four sizes display marked improvements when trained on sampled coresets selected using multi-granular features, achieving a 6.94% improvement in average benchmark scores (AVG.), as shown in Table 1. The proposed method with multi-granularity consistently outperforms importance sampling that uses only n-gram features on the average of all benchmarks. Moreover, our result also shows that despite sampling based upon a target dataset, the model performance does not degrade on non-target benchmarks (See Figure 4). Figure 3 shows the sharp metric improvements of averaged performance on the eight tasks, starting at a model of size 125M to 1.5B, which indicates the potential of the technique to scale up the capabilities of small language models.

5 Further Discussion

Finer-grained Features Reduce Task Biases. Based on our ablation, we observe marked improvement by simply using subword n-grams. Moreover, we show in Figure 4 that selecting from a single task introduces task data biases that degrades the performance. This is mitigated through the use of finer-grained n-gram features where we introduce multi-granular tokens containing subwords and multi-words, which gives an additional 5.78% improvement over word-based n-grams. We postulate that this improvement has to do with the reduction of hash collisions in the hashed n-gram features, where the joint use of subword and multi-word capture beyond the boundaries of a word while preserving parts of a word in tokens so that during collisions the whole word is not entirely discarded and not represented (More analysis in Appendix A.1).

Impact of Multi-granularity. We ablate the percentages of subword, word, and multi-word tokens on a subset of RefinedWeb. These token types are bounded by the vocabulary size, and we generally find that multi-granular tokens outperform single-granular tokens. This results in a 3.23% margin in zero-shot performance for a 125M model. Further, we conduct the experiments to vary the distribution of all three granularities while maintaining a fixed vocabulary size. We found that a higher percentage of subword (60%) is the most viable, along with mixing with some word (30%) and multi-word (10%) tokens. Subword tokens capture finer-grained details so they are more representative, however, they also make the sampling process slow as they make the sequence length longer. In contrast, word and multi-word tokens compress the sequence length, thereby reducing the impact of subword tokens on sampling speed.

6 Conclusion and Future Works

In this study, we revisited importance sampling of text corpus in language modeling by exploring multi-granular n-grams as features. This led us to explore a pretraining paradigm where we can obtain more targeted data for more efficient lan-

guage modeling. Our findings, validated across eight benchmarks, allow us to put forward multi-granular n-grams features as viable document representations used in importance sampling. For future work, we will aim to extend this approach to larger language models and datasets.

Limitations

While the method of targeted data sampling using low-dimensional features is efficient and enhances specific performance areas, it is not without its challenges. Further exploration is needed to refine the process of selecting optimal data features that balance domain specificity with general applicability. Importantly, we have not taken explicit steps to ensure that the sampled data does not contain biases in the data. Moreover, we believe a more solid conclusion can be drawn when even larger pretraining data is experimented, with and other model-based approaches are also taken into account. All in all, this study highlights the importance of finer-grained data selection for pretraining smaller language models. This technique can prevent overfitting while maintaining robust performance across diverse tasks.

Ethics Statement

The practice of selective data sampling in language model pretraining has shown promising results in enhancing model performance in targeted tasks. Our experiments are conducted using datasets that are widely recognized and utilized within the research community, ensuring the reproducibility and reliability of our results. However, the application of this method to sensitive or private datasets necessitates stringent adherence to ethical standards. Furthermore, the increased efficiency in training specialized smaller models could potentially lead to escalated computational demands, which must be considered when scaling these methods.

References

Samira Abnar, Mostafa Dehghani, Behnam Neyshabur, and Hanie Sedghi. 2022. Exploring the limits of large scale pre-training. In *International Conference on Learning Representations*.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. 2024. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

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

Gautier Dagan, Gabriele Synnaeve, and Baptiste Rozière. 2024. Getting the most out of your tokenizer for pre-training and domain adaptation. *arXiv preprint arXiv:2402.01035*.

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 P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. GLaM: Efficient scaling of language models with mixture-of-experts. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR.

R. C. St. John and N. R. Draper. 1975. D-optimality for regression designs: A review. *Technometrics*, 17(1):15–23.

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. *arXiv preprint arXiv:2001.08361*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. 2024. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36.

Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and

Julien Launay. 2023. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only. In *Advances in Neural Information Processing Systems*, volume 36, pages 79155–79172. Curran Associates, Inc.

Donald B. Rubin. 1988. Using the sir algorithm to simulate posterior distributions. *Bayesian Statistics*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Paul A Samuelson. 1937. A note on measurement of utility. *The review of economic studies*, 4(2):155–161.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. *arXiv preprint arXiv:2402.09739*.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. 2023. Data selection for language models via importance resampling. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021. Vocabulary learning via optimal transport for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7361–7373.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

A Appendix

A.1 Details on Document Sampling

Here we provide additional details regarding the process of feature extraction from documents. Due to the memory constraints on the machines, we split the RefinedWeb data into 16 shards, and sampled a subset from each shard based on the target data. This process takes around 1.5 days on average for all approaches, meaning that the change in tokenizer's vocabulary does not result in noticeable differences in sampling speed, since vocabulary also defines sentence compression ratio.

Analysis of Sampled Data. Further, we analyze the sampled data using various tokenization techniques. Here we provide the statistics over each technique in Table 2.

Following (Dagan et al., 2024), we defined compression using two metrics. The first, normalized sequence length (NSL), evaluates the efficiency of a tokenizer compared to our baseline Llama tokenizer. Formally, NSL $c_{\frac{\lambda}{\beta}}$ is defined as the ratio of the encoded sequence lengths from two tokenizers, T_λ and T_β, across N samples from dataset D.

$$c_{\frac{\lambda}{\beta}} = \frac{\sum_{i=1}^{N} |T_\lambda(D_i)|}{\sum_{i=1}^{N} |T_\beta(D_i)|}$$

Just like the methodology in (Dagan et al., 2024), we employ the Llama tokenizer (Touvron et al., 2023) as our reference tokenizer T_β[2].

Approach	Normalized Sequence Length (NSL) (↓)	Time Taken (hrs)
Multi-granular	0.58	30.13
Multiword-only	0.21	27.64
Target-only	0.32	27.20
Base-only	0.75	27.10
Merge	0.44	29.00

Table 2: Statistics of the selection process and the selected documents. AVG. Sequence Length is computed on randomly sampled 1000 documents from the pretraining set.

A.2 Comparison of Vocabulary Merging Techniques

In terms of vocabulary merging, we also experiment with fixing the proportion of each type of token (*subword*, *word*, and *multi-word*) in $v(t)$ at percentages $p_{\text{subword}} = 0.6$, $p_{\text{word}} = 0.1$, and $p_{\text{multi-word}} = 0.3$, which we found to be the most performant combination. However, fixed ratios do not work as well as the optimized vocabulary with vocabulary utility metric as described in the paper.

We also compare the proposed **multi-granular** sampling with different techniques of merging the target vocabulary set V_{task} to the Llama-3 tokenizer V_{start}. Several techniques are compared: (1) **Merge**: we take the union of V_{task} and V_{start}, but removing the duplicate tokens. (2) **Target-only**: we use the task vocabulary set with the subword tokens. (3) **Base-only**: we use the llama-3's vocabulary set with the subword tokens. (4) **Multiword-only**: we use the acquired multi-word vocabulary that consists of only tokens concatenated by more than one word. We show the results in Table 3.

[2]This means that if T_λ achieves an average NSL of 0.75, it indicates that sequences encoded by T_λ are 25% shorter in terms of token count compared to those encoded by Llama.

Approach	ARC-Easy	ARC-Hard	BOOLQ	PIQA	SIQA	HELLASWAG	OBQA	WINOGRANDE	Avg.
125M params									
Multi-granular	44.24	30.13	57.98	72.71	41.16	46.68	35.74	52.10	**47.59**
Multiword-only	45.74	27.64	59.38	66.41	41.02	37.13	34.77	52.64	45.34
Target-only	43.59	27.20	57.76	72.17	42.29	45.89	31.05	50.24	46.14
Base-only	43.50	27.10	57.66	72.07	42.19	45.79	30.95	50.14	46.05
Merge	44.00	29.00	57.50	72.00	41.00	46.00	35.00	51.50	46.75

Table 3: Results over all downstream tasks based on data sampled with different granular features, each in terms of its respective metric on different granular tokens with a pretrained 125M model.

A.3 Additional Results of the Impact of Domain Biases (1/2)

Here we present the results as an extension for Figure 4, where we present the results for selected data based on the rest of the seven benchmarks. We show the results of multi-granular n-grams with n-gram baseline across eight tasks using 125M models.

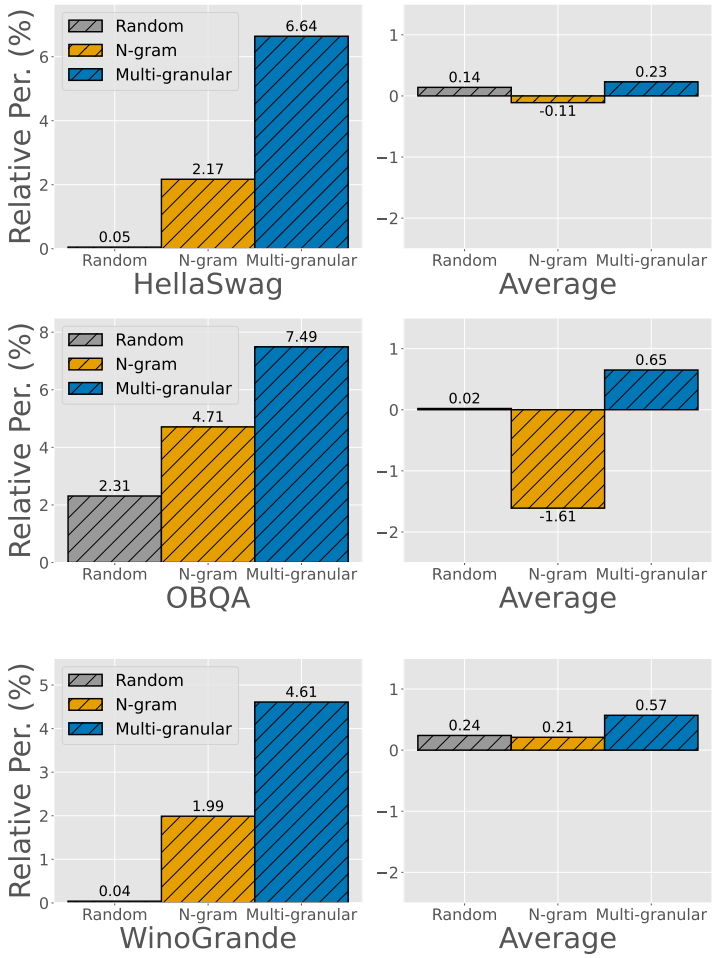

Figure 5: Comparison of **Multi-granular** n-grams with **N-gram** and **Random** baseline using 125M models, trained solely on data selected with HellaSwag, OBQA, WinoGrande data as the target respectively.

A.4 Additional Results of the Impact of Domain Biases (2/2)

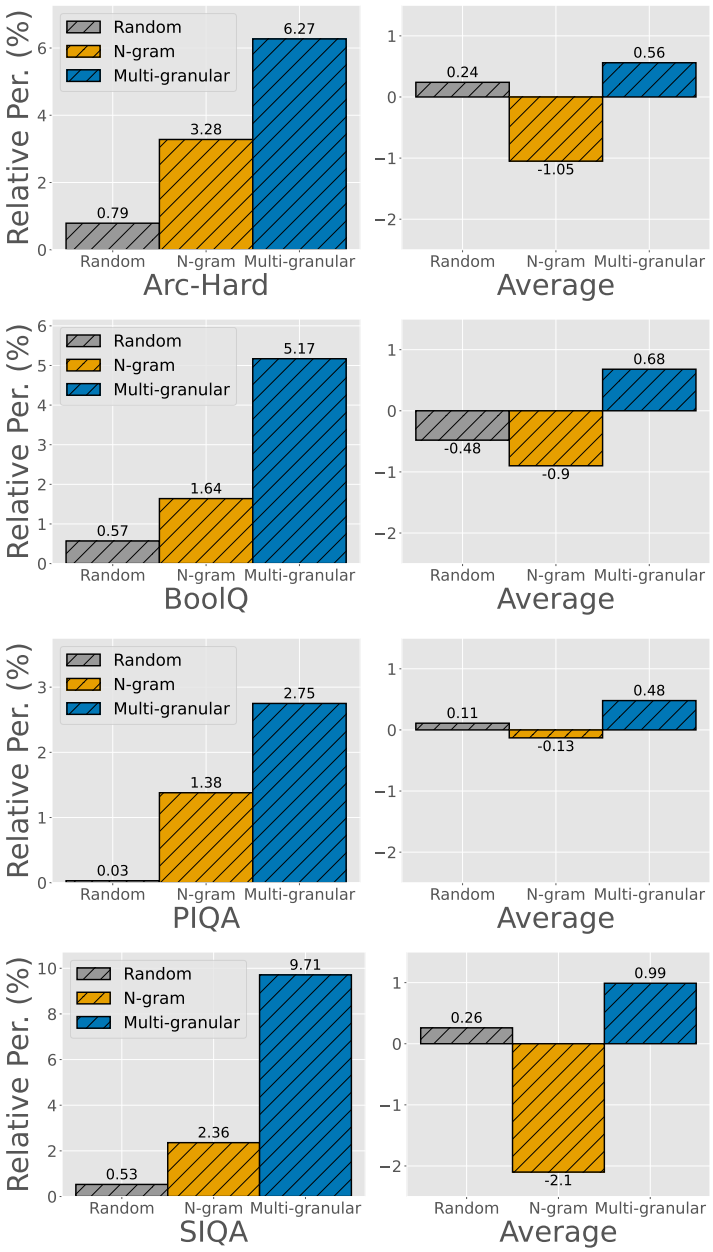

Figure 6: Comparison of **Multi-granular** n-grams with **N-gram** and **Random** baseline using 125M models, trained solely on data selected with Arc-Hard, BoolQ, PIQA, and SIQA data as the target respectively.