Using Informative Data Subsets for Efficient Training of Large Language Models: An Initial Study

Abstract

Language Models (LMs) are pretrained on large unlabeled corpora through self-supervision tasks and have become ubiquitous to several NLP applications. Recent trends indicate that the generalization capability of Large LMs (LLMs) improves tremendously with increasing model capacity and size of the pretraining dataset. However, this also results in inefficiencies owing to higher training times, compute requirements and environmental impact. Previous works have mostly addressed the inefficiency concerns with respect to improving sample efficiency, architecture and training loss objective with little focus on data optimization. In this work, we explore if it is possible to use only highly informative subsets of the training data to train LLMs while maintaining their performance. We build upon the work done in informative data subset selection and propose INGENIOUS, a framework that selects highly representative subsets of the training corpus by optimizing a submodular function. We show INGENIOUS can be adopted for the scale of LLM training and empirically demonstrate that the proposed framework achieves $\sim 99\%$ of original BERT performance in about $\sim 35\%$ of the original training time.

1 Introduction

Large pre-trained language models (PLTMs) [Devlin et al., 2019, Radford et al., 2019, Yang et al., 2020, Brown et al., 2020, Raffel et al., 2020] have revolutionized the field of NLP and are the default choice for a wide variety of NLP tasks. However, this versatility of PTLMs comes with serious costs. The ever-growing size of PTLMs and pretraining corpora for improving the generalization ability results in increased consumption of resources and energy along with dire environmental impacts [Sharir et al., 2020]. For instance, it costs an estimated $12 million to train GPT-3 [Brown et al., 2020] with roughly 1.2 million pounds of CO_2 emisssions[1]. Additionally, the sheer amount of resources required and costs incurred for pretraining LMs make them inaccessible to small organizations and universities. Hence, a crucial step towards developing responsible, fair, and GreenAI [Schwartz et al., 2020] involves minimizing inefficiencies and costs of training large LMs.

Most of the previous efforts towards improving the efficiency of PTLMs have primarily focused on enhancements in the model architecture, training pipeline, and the objective function used for loss optimization. Shen et al. [2022] proposed a staged training mechanism where they start with training a relatively smaller model which is used for initialising the full capacity model at a later stage. Sample efficient masking techniques have also been proposed that modify the token masking strategy to enable PTLM to better leverage context leading to an improved convergence [Bitton et al., 2021, Kaur et al., 2022]. Yao et al. [2022] identify relevant samples from pretraining corpus based on their similarity with task-specific dataset to train task-specific PTLM followed by fine-tuning. Such an approach inherently suffers from the limitation of requiring to pretrain a LM for every downstream task. In this work, driven by the observation that the scale of pretraining corpus contributes significantly to training costs of PTLMs, we explore if it is feasible to train PTLMs using highly informative subsets

[1] https://fortune.com/2021/04/21/ai-carbon-footprint-reduce-environmental-impact-of-tech-google-research-study/

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

of the corpus. Recently, informative data subset selection has shown great promise for efficient deep model training in supervised and semi-supervised settings for images [Mirzasoleiman et al., 2020, Killamsetty et al., 2021a,b,c, Pooladzandi et al., 2022]. Consequently, the main question we try to answer in this work is: *Can we efficiently pretrain large language models using informative subsets of the training corpus without compromising performance?*

In particular, we demonstrate that pretraining of LMs can be sped up significantly by training only on informative subsets of data comprising of samples that maximize the representation of the remaining samples from the corpus. The diminishing gains property of submodular functions [Fujishige, 2005] makes it ideal for posing subset selection as a submodular maximization problem. However, applying existing subset selection frameworks at the scale of LLMs is non-trivial as most of such methods rely on per-sample gradients which are computationally expensive to obtain (please refer to Appendix A for a detailed literature survey). Further, previous works have not examined subset selection from enormous datasets such as Wikipedia and Common Crawl, commonly used for pretraining LMs [Devlin et al., 2019, Raffel et al., 2020]. We employ scalable sentence feature encoders to obtain individual data sample features for subset selection, as well as various engineering tricks that allow us to select subsets from large-scale datasets as discussed in Section 2.

Our contributions: 1) We propose INGENIOUS, a subset selection framework that selects informative subsets by maximizing a submodular function; 2) we show that pre-training BERT on subsets selected through INGENIOUS leads to close to 99% performance on the GLUE benchmark compared to the original BERT training while taking only $\sim 35\%$ of the original's training time; 3) we perform extensive experimentation (Section 3) to establish the effectiveness of the choice of embedding representation employed for subset selection and show comparisons with various baselines.

2 INGENIOUS Framework

In this section, we introduce INGENIOUS - a data subset selection framework for pre-training language models and the subset selection formulation. We first discuss some preliminary background on submodularity. Let $\mathcal{U} = \{x_j\}_{j=1}^{m}$ denote the pre-training dataset with m data points and $\mathcal{S} \subset \mathcal{U}$ be the subset of the pre-training dataset. A set function $f : 2^{\mathcal{U}} \rightarrow \mathbb{R}$ is **submodular** [Fujishige, 2005] if for $x \in \mathcal{U}$, $f(\mathcal{A} \cup x) - f(\mathcal{A}) \geq f(\mathcal{B} \cup x) - f(\mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{U}$ and $x \notin \mathcal{B}$. In our work, we use **Facility Location** [Kothawade et al., 2022], a submodular function which is defined as

$$f_{FL}(\mathcal{A}) = \sum_{i \in \mathcal{U}} \max_{j \in \mathcal{A}} \boldsymbol{S}_{ij}$$

where $\boldsymbol{S}$ is pair-wise similarity kernel matrix and $\boldsymbol{S}_{ij}$ is similarity between the i^{th} and j^{th} samples.

2.1 Methodology for Subset Selection

INGENIOUS selects an informative subset $\mathcal{S}$ of size k from entire unlabeled dataset $\mathcal{U}$. We pose the data subset selection problem as a submodular maximization problem which can be formulated as:

$$\mathcal{S} = \arg\max_{\mathcal{X} \in \mathcal{U}} f(\mathcal{X}, \mathcal{U}) \tag{1}$$

The optimization problem in equation 1 is an instance of cardinality constrained monotone submodular maximization where an approximate solution can be obtained by expanding the subset through lazy greedy algorithm [Minoux, 1978]. We incrementally add that sample to the subset which increases the value of f_{FL} the maximum. Further, to encourage diversity, we employ an exploration heuristic where we modify the selected subset periodically after every R model update steps. Precisely, for each sample, we obtain an importance score through estimating submodular gain value while creating the subset through lazy greedy algorithm. We normalize the importance scores for all the samples through softmax operation. Finally, to select a subset comprising of k samples after every R optimization steps, we sample k points (without replacement) by using the normalized importance scores as probabilities. Algorithm 1 in Appendix B summarises the training steps of LM through INGENIOUS. The submodular function needs the pairwise similarity kernel (of size $|\mathcal{U}| \times |\mathcal{U}|$) between the data samples in $\mathcal{U}$. In this study, we employ cosine similarity to compute similarity between the feature representation of samples obtained through a feature encoder as described in Section 2.2.

2.2 Feature Encoders for Similarity Computation

To obtain the embedded feature representation of text samples in $\mathcal{U}$, we explore both dense and sparse feature encoders. For extracting dense features, we compute text embeddings by averaging the output embedding corresponding to all text token using the language model itself as it gets trained. However, the type of information varies depending on the layer chosen for feature extraction. For instance, the initial layers of BERT majorly learn syntactic features while the output of later layers contain more semantic information. Further, it is shown that using the outputs from the middle layers of BERT as features is more robust than using the final task-specific layers [Tenney et al., 2019]. We discuss the choice of layer through tuning in Section 3.2. For sparse feature encoder, we consider TF-IDF [Aizawa, 2003] owing to its success at capturing statistical lexical features [Robertson et al., 2009] and compare performance with using dense representations (Section 3.2).

2.3 Partitioning based Efficient Subset Selection

As discussed earlier, submodular functions require a similarity kernel of size $|\mathcal{U}| \times |\mathcal{U}|$ and the memory required for storing the similarity kernels can be enormous depending on the size of the pre-training dataset (i.e., $|\mathcal{U}|$). In order to minimize the memory usage, instead of selecting a subset of size k from the entire unlabeled set directly, we first partition the unlabeled set into N_P random blocks of equal sizes (i.e., partition size is $\frac{|\mathcal{U}|}{N_P}$) and select a subset of size k/N_P from each partition by submodular function maximization over the partition. Assuming the number of blocks in the partition to be N_P and unlabeled set partition as $\text{partition}\,(\mathcal{U}, N_P) = \{\mathcal{U}_i^p \,:\, |\mathcal{U}_i^p| = \frac{|\mathcal{U}|}{N_P}\}_{i=1}^{N_P}$, we can write the subset selection optimization problem with partitioning as follows:

$$\mathcal{S}_t = \bigcup_{i=1}^{N_P} \underset{\mathcal{S} \in \mathcal{U}_i^p}{\arg\max}\, f(\mathcal{S}, \mathcal{U}_i^p) \tag{2}$$

The partitioning of unlabeled set allows us to get away with constructing similarity kernels of size $\frac{|\mathcal{U}|}{N_P} \times \frac{|\mathcal{U}|}{N_P}$, thereby reducing the similarity kernel memory usage by around ${N_P}^2$ times. As the number of partitions increases, the approximation of the original optimization objective becomes less accurate, i.e., the final subsets returned will be less optimal. As a result, partitions result in a trade-off between performance and memory efficiency. Please refer to Appendix C for further details.

3 Experiments and Results

We take BERT as the underlying LM and use English Wikipedia with BooksCorpus as the pre-training corpora and use MLM and NSP tasks for pretraining following details as mentioned in the work of Devlin et al. [2019]. We perform training using a batch size of 1024 for 1,000,000 steps in case of vanilla-BERT and for 250,000 steps (25%) in case of pretraining through INGENIOUS. We fix the subset size to 25% of corpora size. We use Adam optimizer [Kingma and Ba, 2014] with learning rate of 1e-4, $\beta_1 = 0.9$, $\beta_2 = 0.99$, L2 weight decay of 0.01. We warmstart the model with full data training for first 2 epochs and subsequently train only on selected subsets. Training of all the models is performed on 8 NVIDIA A100-SXM4-40GB GPUs.

3.1 Efficacy of Subset Selection in making LM Training Efficient

We evaluate the performance of BERT pretrained on subsets selected through INGENIOUS and compare it with fully pretrained vanilla BERT. We employ the commonly used GLUE benchmark [Wang et al., 2019] and report the accuracy for each task along with the mean on the dev set in Table 1 (Appendix D compares GLUE performance at different checkpoints during pretraining). We compare INGENIOUS with two baselines - **1) Early Stopping:** BERT pretraining stopped at 250K steps and checkpoint is used for evaluation; and **2) Random Online:** a subset of same size as selected by INGENIOUS is obtained for pretraining by randomly selecting samples from $\mathcal{U}$. Observe that despite using only informative subsets and trained only for 250K steps, INGENIOUS achieves 98.5% performance of vanilla BERT in only $\sim 35\%$ of the latter's training time. Compared to baselines, slightly higher time due to subset selection overheads leads to higher accuracy as well. Most importantly, on complex tasks such as COLA, INGENIOUS achieves performance closest to full BERT training.

Table 1: Comparison of pretraining time and GLUE performance (averaged over 20 runs). INGENIOUS achieves 98.5% of fully pretrained BERT performance, reducing pretraining time to $\sim 35\%$.

Method	Runtime (s)	Mean Accuracy	CoLA	MRPC	RTE	STS-B	SST-2	MNLI (matched)	MNLI (mismatched)	QNLI	QQP
Vanilla BERT, 1M steps	282,924	82.76	55.98	90.54	69.31	89.25	92.41	83.86	84.31	91.37	87.85
Vanilla BERT Early stopping, 250K steps	**70,861 (-74.95)**	81.27 (-1.49)	50.93	90.57	66.26	88.89	91.21	82.73	83.15	90.17	87.56
Random Online, 250K steps	74,146 (-73.79)	81.04 (-1.72)	50.67	88.93	67.42	88.11	91.33	82.37	82.96	90.04	87.57
INGENIOUS, 250K steps	96,338 (-65.95)	**81.6** (-1.16)	54.61	89.68	67.16	88.94	91.13	82.07	82.81	90.4	87.57

Table 2: Analysing GLUE performance (averaged over 20 runs) by varying feature representations used for subset selection.

Method	Mean Accuracy	CoLA	MRPC	RTE	STS-B	SST-2	MNLI (matched)	MNLI (mismatched)	QNLI	QQP
INGENIOUS, BERT Layer-3, 250K steps	81.05 (-1.71)	51.67	89.99	65.78	88.09	90.85	82.41	83.05	89.93	87.66
INGENIOUS, BERT Layer-6, 250K steps	80.9 (-1.86)	50.68	89.36	65.99	88.21	91.22	82.2	82.84	90.01	87.55
INGENIOUS, BERT Layer-9, 250K steps	**81.6** (-1.16)	54.61	89.68	67.16	88.94	91.13	82.07	82.81	90.4	87.57
INGENIOUS, BERT Layer-12, 250K steps	80.94 (-1.82)	50.97	89.34	66.62	87.82	90.5	82.35	82.84	90.46	87.56
INGENIOUS, TF-IDF, 250K steps	81.12 (-1.64)	52.07	90.19	66.91	88.19	90.64	82.05	82.64	89.87	87.48

3.2 Role of Feature Embeddings for Subset Selection

Different BERT layers capture different information - while the lower layers capture word order [Rogers et al., 2020], the middle layers capture syntactic information [Hewitt and Manning, 2019, Jawahar et al., 2019] and the later layers capture task-specific information [Kovaleva et al., 2019, Hao et al., 2019]. Since it is ambiguous to identify most suitable layer, we experiment by varying layer used to obtain representations for subset selection. We choose layers 3, 6, 9 and 12 as the representatives of the early, middle and later layers and report the performance on dev set of GLUE benchmark in Table 2 where it can be observed that obtaining feature embeddings using layer 9 provides best results (mean acc. of 81.6%). This can be due to the fact that it captures both syntactic and semantic information. Further, we compare using TF-IDF features as sample representations against dense embedded features (Ingenious TF-IDF vs. BERT Layer-9 in Table 2). We observe that dense embeddings performs better (0.6% greater) than shallow syntactic TF-IDF features.

4 Conclusion

In this work, we discuss the inefficiencies pertaining to high pretraining times of LLMs. Different from architecture and loss enhancements done in previous works, we explore optimisation from data perspective to make pretraining efficient. We propose INGENIOUS framework that selects informative data subset representative of remaining samples in corpus by formulating it as a submodular maximisation problem. The value of submodular function is determined based on similarity between sample embeddings learned by the LM. The LLM is trained only on the subset at a given time and the subset is updated periodically. Our experiments on BERT show that its pretraining time can be reduced to $\sim 35\%$ with $\sim 99\%$ retention in performance. As future work, other LLMs such as GPT-2, T5 etc. can be optimised through INGENIOUS. Also, structured data sources like knowledge graphs can be used to guide subset selection for minimising redundancies in the pretraining data.

References

A. Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373, 2013. ISSN 1935-8237. doi: 10.1561/2200000039. URL http://dx.doi.org/10.1561/2200000039.

F. Bach. Submodular functions: from discrete to continuous domains. *Mathematical Programming*, 175(1):419–459, 2019.

Y. Bitton, M. Elhadad, G. Stanovsky, and R. Schwartz. Data efficient masked language modeling for vision and language. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3013–3028, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.259. URL https://aclanthology.org/2021.findings-emnlp.259.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020.

T. Campbell and T. Broderick. Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*, pages 698–706, 2018.

K. L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):1–30, 2010.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

D. Feldman. *Core-Sets: Updated Survey*, pages 23–44. Springer International Publishing, Cham, 2020. ISBN 978-3-030-29349-9. doi: 10.1007/978-3-030-29349-9_2. URL https://doi.org/10.1007/978-3-030-29349-9_2.

S. Fujishige. *Submodular functions and optimization*. Elsevier, 2005.

Y. Hao, L. Dong, F. Wei, and K. Xu. Visualizing and understanding the effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4143–4152, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1424. URL https://aclanthology.org/D19-1424.

S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 291–300, 2004.

J. Hewitt and C. D. Manning. A structural probe for finding syntax in word representations. In *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)*, pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419.

R. Iyer, N. Khargoankar, J. Bilmes, and H. Asanani. Submodular combinatorial information measures with applications in machine learning. In *Algorithmic Learning Theory*, pages 722–754. PMLR, 2021.

R. K. Iyer. *Submodular optimization and machine learning: Theoretical results, unifying and scalable algorithms, and applications*. PhD thesis, 2015.

G. Jawahar, B. Sagot, and D. Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1356. URL https://aclanthology.org/P19-1356.

J. Kaur, S. Bhatia, M. Aggarwal, R. Bansal, and B. Krishnamurthy. LM-CORE: Language models with contextually relevant external knowledge. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 750–769, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.57. URL https://aclanthology.org/2022.findings-naacl.57.

V. Kaushal, R. Iyer, S. Kothawade, R. Mahadev, K. Doctor, and G. Ramakrishnan. Learning from less data: A unified data subset selection and active learning framework for computer vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1289–1299. IEEE, 2019.

K. Killamsetty, D. S, G. Ramakrishnan, A. De, and R. Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5464–5474. PMLR, 18–24 Jul 2021a. URL https://proceedings.mlr.press/v139/killamsetty21a.html.

K. Killamsetty, D. Sivasubramanian, G. Ramakrishnan, and R. Iyer. Glister: Generalization based data subset selection for efficient and robust learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9):8110–8118, May 2021b. doi: 10.1609/aaai.v35i9.16988. URL https://ojs.aaai.org/index.php/AAAI/article/view/16988.

K. Killamsetty, X. Zhao, F. Chen, and R. K. Iyer. RETRIEVE: Coreset selection for efficient and robust semi-supervised learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021c. URL https://openreview.net/forum?id=jSz59N8NvUP.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

K. Kirchhoff and J. Bilmes. Submodularity for data selection in machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 131–141, 2014.

S. Kothawade, J. Girdhar, C. Lavania, and R. Iyer. Deep submodular networks for extractive data summarization. *arXiv preprint arXiv:2010.08593*, 2020.

S. Kothawade, V. Kaushal, G. Ramakrishnan, J. Bilmes, and R. Iyer. Prism: A rich class of parameterized submodular information measures for guided data subset selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(9):10238–10246, Jun. 2022. doi: 10.1609/aaai.v36i9.21264. URL https://ojs.aaai.org/index.php/AAAI/article/view/21264.

O. Kovaleva, A. Romanov, A. Rogers, and A. Rumshisky. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1445. URL https://aclanthology.org/D19-1445.

M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization techniques*, pages 234–243. Springer, 1978.

B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause. Lazier than lazy greedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

B. Mirzasoleiman, J. Bilmes, and J. Leskovec. Coresets for data-efficient training of machine learning models. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6950–6960. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/mirzasoleiman20a.html.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

O. Pooladzandi, D. Davini, and B. Mirzasoleiman. Adaptive second order coresets for data-efficient machine learning. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 17848–17869. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/pooladzandi22a.html.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

S. Robertson, H. Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

A. Rogers, O. Kovaleva, and A. Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020. doi: 10.1162/tacl_a_00349. URL https://aclanthology.org/2020.tacl-1.54.

R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni. Green ai. *Communications of the ACM*, 63(12): 54–63, 2020.

O. Sharir, B. Peleg, and Y. Shoham. The cost of training nlp models: A concise overview. *arXiv preprint arXiv:2004.08900*, 2020.

S. Shen, P. Walsh, K. Keutzer, J. Dodge, M. E. Peters, and I. Beltagy. Staged training for transformer language models. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 19893–19908. PMLR, 2022. URL https://proceedings.mlr.press/v162/shen22f.html.

I. Tenney, D. Das, and E. Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452.

E. Tohidi, R. Amiri, M. Coutino, D. Gesbert, G. Leus, and A. Karbasi. Submodularity in action: From machine learning to signal processing applications. *IEEE Signal Processing Magazine*, 37 (5):120–133, 2020.

A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

K. Wei, Y. Liu, K. Kirchhoff, C. Bartels, and J. Bilmes. Submodular subset selection for large-scale speech training data. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3311–3315. IEEE, 2014a.

K. Wei, Y. Liu, K. Kirchhoff, and J. Bilmes. Unsupervised submodular subset selection for speech data. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4107–4111. IEEE, 2014b.

K. Wei, R. Iyer, and J. Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pages 1954–1963. PMLR, 2015.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020.

X. Yao, Y. Zheng, X. Yang, and Z. Yang. NLP from scratch without large-scale pretraining: A simple and efficient framework. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 25438–25451. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/yao22c.html.

A Related Work

A.1 Submodular Functions

Let $\mathcal{U}$ denote the *unlabeled* set of n data points $\mathcal{U} = \{1, 2, 3, ..., n\}$ and a set function $f : 2^{\mathcal{U}} \rightarrow \mathbb{R}$. Formally, a function f is submodular [Fujishige, 2005] if for $x \in \mathcal{U}$, $f(\mathcal{A} \cup x) - f(\mathcal{A}) \geq f(\mathcal{B} \cup x) - f(\mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{U}$ and $x \notin \mathcal{B}$. For a set $\mathcal{A} \subseteq \mathcal{U}$, $f(\mathcal{A})$ provides a real-valued score for $\mathcal{A}$. A function f is said to be monotone if $f(\mathcal{A}) \leq f(\mathcal{B})$ whenever $\mathcal{A} \subseteq \mathcal{B}$. Further, f is *supermodular* if $-f$ is *submodular*, modular if it is both, and *normalized* if $f(\phi) = 0$. Many combinatorial functions such as facility location, set cover, log determinant, graph cut *etc.* [Iyer et al., 2021, Kothawade et al., 2020] are naturally submodular. Submodularity is particularly appealing because it naturally occurs in real world applications [Tohidi et al., 2020, Bach, 2013, 2019, Iyer, 2015] and also admits a constant factor $1 - \frac{1}{e}$ [Nemhauser et al., 1978] approximation for cardinality constrained maximization. Since most acquisition functions involve maximizing utility functions, this makes submodularity a good choice [Wei et al., 2015]. Additionally, variants of the greedy algorithm maximize a submodular function in *near-linear time* [Mirzasoleiman et al., 2015].

A.2 Submodular Data Subset Selection

Several recent papers have used submodular functions for data subset selection towards various applications like speech recognition [Wei et al., 2014b,a], machine translation [Kirchhoff and Bilmes, 2014] and computer vision [Kaushal et al., 2019]. Other common approaches for subset selection include the usage of coresets. Coresets are weighted subsets of the data, which approximate certain desirable characteristics of the full data (*e.g.*, the loss function) [Feldman, 2020]. Coreset algorithms have been used for several problems including k-means and k-median clustering [Har-Peled and Mazumdar, 2004], SVMs [Clarkson, 2010] and Bayesian inference [Campbell and Broderick, 2018]. Recent coreset selection-based methods [Mirzasoleiman et al., 2020, Killamsetty et al., 2021b,a,c] have shown great promise for efficient and robust training of deep models. CRAIG [Mirzasoleiman et al., 2020] tries to select a coreset summary of the training data that estimate the full training gradient closely. Whereas GLISTER [Killamsetty et al., 2021b] poses the coreset selection problem as a discrete-continuous bilevel optimization problem that minimizes the validation set loss. Similarly, RETRIEVE [Killamsetty et al., 2021c] also uses a discrete bilevel coreset selection problem to select unlabeled data subsets for efficient semi-supervised learning. Another approach GRAD-MATCH [Killamsetty et al., 2021a] selects coreset summary that approximately matches the full training loss gradient using orthogonal matching pursuit.

B Algorithm for training LM through INGENIOUS

Algorithm 1 describes the steps for training an LM efficiently through our INGENIOUS framework.

C Further Details on Parallel Computation of Subsets from Partitions

To maximize the utilization of available compute, we can select the subsets from each partition in parallel. However, it is to be noted that the memory utilization also increases with number of parallel process. For example, if we try to select N_{PP} subsets from partitions in parallel, the memory usage due to similarity kernel is of the order $\mathcal{O}(N_{PP}\frac{|\mathcal{U}|^2}{N_P^2})$. In our experiments, we run $N_{PP} = 100$ processes in parallel.

D Comparison of GLUE Score at Intermediate Pretraining Checkpoints

In this section, we compare the performance of regular BERT pretraining with training on subsets selected through INGENIOUS at intermediate pretraining stages. Figure 1 demonstrates that intermediate checkpoints obtained through INGENIOUS pretraining enables BERT to consistently achieve better GLUE score than regular pretraining. This trend is further amplified for the CoLA task in the GLUE benchmark as can be observed in Figure 2 indicating better convergence.

Algorithm 1: Training of LM through INGENIOUS

Input: Training dataset: $\mathcal{U}$, Initial model parameters: θ_0, Total no of training steps: T, Training steps interval for subset selection: R, Number of steps for warmstart phase: W, Size of the coreset: k, Reg. Coefficient: λ, Learning rates: $\{\alpha_t\}_{t=0}^{t=T-1}$, Tolerance: ϵ

Set $t = 0$
optimizer = AdamW()
*** Warmstart Phase ***
repeat
  Compute batches $\mathcal{U}_b = ((x_b, y_b); b \in (1 \cdots B))$ from $\mathcal{U}$
  for $b = 1$ *to* B **do**
    if $t \geq W$ **then**
      break
    Compute mask $\mathbf{m}_t$ on $\mathcal{U}_b$
    θ_{t+1} = optimizer.step()
    $t = t + 1$
until *until* $t \geq W$
*** Subset Selection ***
greedyIdxs, gains = $argmax_{|S| \leq |\mathcal{U}|} f_{FL}(S, \mathcal{U}, \theta_t)$
probabilities = Softmax(gains)
$\mathcal{S}_t$ ∼sample(greedyIdxs, probabilities, k)
repeat
  Compute batches $\mathcal{S}_{tb} = ((x_b, y_b); b \in (1 \cdots B))$ from $\mathcal{S}_t$
  for $b = 1$ *to* B **do**
    if $t \geq W$ **then**
      break
    Compute mask $\mathbf{m}_t$ on $\mathcal{S}_{tb}$
    $\theta_{t+1} = optimizer.step()$
    $t = t + 1$
    if $(t\%R == 0)$ **then**
      $\mathcal{S}_{t+1}$ ∼sample(greedyIdxs, probabilities, k)
    else
      $\mathcal{S}_{t+1} = \mathcal{S}_t$
until *until* $t \geq T$
*** Evaluate trained model on validation set ***
$eval = \text{evaluate}\ (\theta_T, \mathcal{V})$
return $eval, \boldsymbol{\theta}_T$

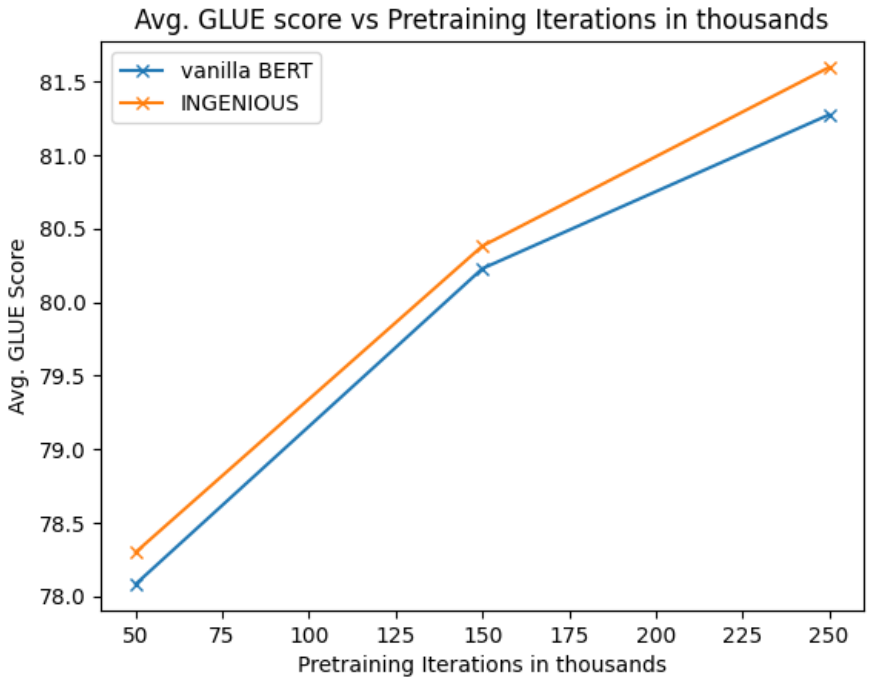

Figure 1: Comparison of INGENIOUS with regular BERT pretraining at different stages of pretraining on **GLUE** benchmark. It can be observed that pretraining on subsets selected through INGENIOUS enables BERT to achieve better performance at intermediate pretraining checkpoints.

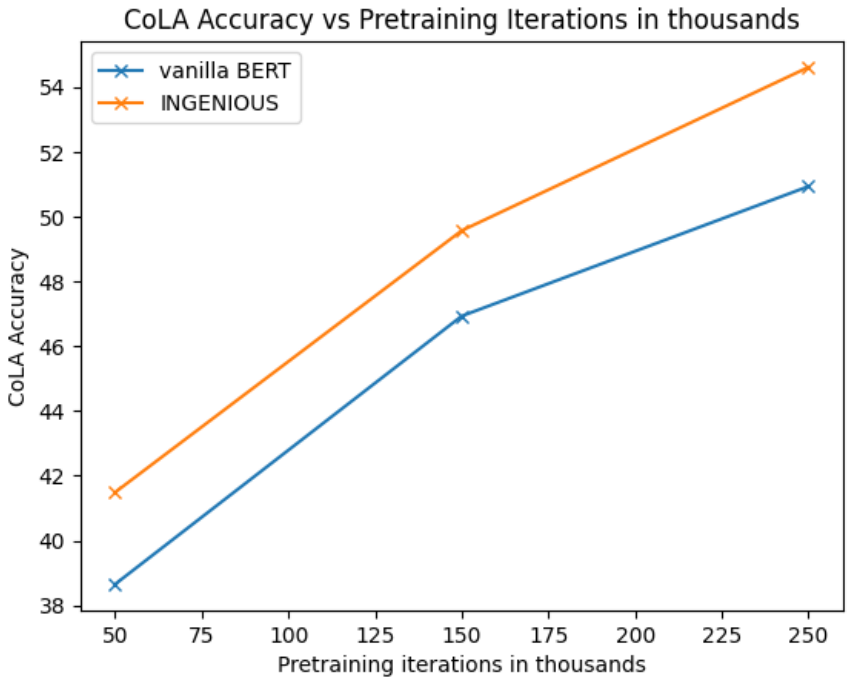

Figure 2: Comparison of INGENIOUS with regular BERT pretraining at different stages on the **CoLA** task. It can be observed that pretraining on subsets selected through INGENIOUS enables BERT to achieve better performance at intermediate pretraining checkpoints indicating better convergence.