# Automatic Document Selection for Efficient Encoder Pretraining

**Yukun Feng**[1] **Patrick Xia**[1] **Benjamin Van Durme**[1] **João Sedoc**[2]
[1]Johns Hopkins University
[2]New York University
{yfeng55, paxia, vandurme}@jhu.edu, jsedoc@stern.nyu.edu

## Abstract

Building pretrained language models is considered expensive and data-intensive, but must we increase dataset size to achieve better performance? We propose an alternative to larger training sets by automatically identifying smaller yet domain-representative subsets. We extend *Cynical Data Selection*, a statistical sentence scoring method that conditions on a representative target domain corpus. As an example, we treat the OntoNotes corpus as a target domain and pretrain a RoBERTa-like encoder from a cynically selected subset of the Pile. On both perplexity and across several downstream tasks in the target domain, it consistently outperforms random selection with **20x** less data, **3x** fewer training iterations, and **2x** less estimated cloud compute cost, validating the recipe of automatic document selection for LM pretraining.

## 1 Introduction

Large pretrained language models have achieved state-of-the-art performance in NLP tasks (Devlin et al., 2019; Liu et al., 2019, *i.a.*). These studies find that increasing pretraining data size usually leads to better task performance. For many tasks, additional task (in-domain) data helps improve the performance further (Gururangan et al., 2020; Dery et al., 2021; Li et al., 2022). Several studies have found that directly pretraining on task data is more effective : science texts (Beltagy et al., 2019), tweets (Nguyen et al., 2020), legal texts (Chalkidis et al., 2020) or code (Tabassum et al., 2020; Chen et al., 2021). Notably, these domains are known *a priori*, and identifying data sources for curation is straightforward. In other instances where the domain is less clear, like "offensive online content" (Bai et al., 2021), more complicated data sampling is employed to *guess* at the desired data distribution suitable for training a downstream classifier.

To address such scenarios, we propose automatically identifying relevant domain-specific training data for a large corpus and subsequently pretraining a model on the selected data. Specifically, we use Cynical Data Selection (Axelrod, 2017), an approach that advanced Moore-Lewis sampling (Moore and Lewis, 2010), to select data from the Pile dataset (Gao et al., 2021). This automatic selection method can include possibly overlooked yet relevant documents from domains that may not be too close to the target domain. Figure 1 illustrates this method which achieves higher performance on tasks in the target domain by using only 2.5GB (0.5%) of cynically selected data.

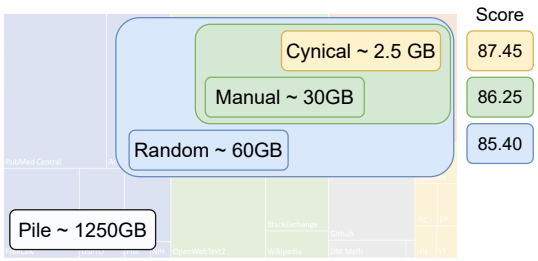


Figure 1: This figure highlights the efficiency of the automatic cynical selection of documents in the target domain. Scores are averaged from 8 Edge Probing tasks. Cynically selected 2.5GB data achieves the best score.

Specifically, we experiment with pretraining encoders with varying amounts of data sampled from the Pile.[1] With our "target corpus" of OntoNotes (Weischedel et al., 2013), we compare language models trained with cynical and random selection at various data levels. We find that the cynically selected encoder achieves consistently lower target corpus perplexity than one trained with random selection. We further finetune the encoders on a suite of tasks, some of which are derived from OntoNotes. Again, we find that models pretrained with cynical selection perform best. We suggest this as a viable method for inexpensively pretraining effective domain-specific encoders.

[1]The Pile consists of 800GB raw text but for this paper, we refer to its "effective" size, which is 1250GB.



*Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9522 - 9530
December 7-11, 2022 

## 2 Cynical Data Selection

Methods for data selection for language-related tasks have been widely studied, usually to select in-domain data (Axelrod et al., 2011; van der Wees et al., 2017; Dai et al., 2020; Killamsetty et al., 2020). One such method is Cynical Data Selection (Axelrod, 2017). The intuition behind cynical selection is greedily ranking sentences from the text corpus, based on its score computed against text *representative* of the target domain, which is based on how much information gained by selecting it.

Concretely, given representative text from the target domain, cynical selection uses the cross-entropy of the selected text against the representative text and calculates the information gain of each sentence in the general corpus. It then picks the most useful sentence relative to what has already been selected and its similarity to the representative text. This also leads to a bias towards shorter sentences and preferring sentences that contain words with high probability in the representative text.

Our work *extends* the cynical selection to the document level selection. Sentences are still scored at the sentence level, but the average sentence-level gain determines the information gain of a document.[2] We demonstrate its advantages in efficiently selecting related documents to the target domain.

## 3 Experiments and Results

In this work, we set OntoNotes 5.0 (Weischedel et al., 2013) as our target corpus, and we use a smaller sample from the training corpus of the CoNLL 2012 Shared Task (Pradhan et al., 2012) as the representative corpus for data selection. We first train an encoder based on the selected data and use the Edge Probing suite (Tenney et al., 2019b) for the downstream task evaluation, which has previously been used to probe and evaluate language models (Clark et al., 2019; Tenney et al., 2019a; Jiang et al., 2020; Zhang et al., 2021).

### 3.1 Data Selection

**Dataset** We adopt the Pile (Gao et al., 2021) for data selection, which consists of 1250GB text from 22 domains. Cynical selection naturally prefers text data based on the target corpus. To make a more fair comparison, we exclude 100GB data from "DM Mathematics" and "Github" to eliminate the noise of non-text data in random selection.

[2]A formal explanation of Cynical selection and its extension is in Appendix B.

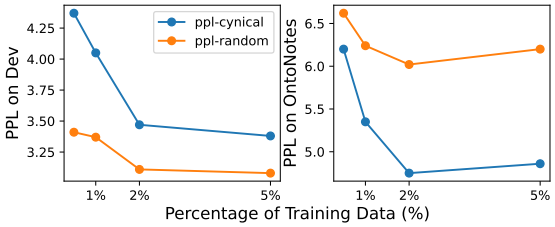


Figure 2: Validation perplexity on held-out set (left), and OntoNotes (right) at 100k training steps.

**Selection Strategy** Encoder pretraining is naturally a document-level task, as context contributes critically to improved representations. Thus, we need to extend the sentence selection into the document selection to achieve a better-contextualized representation at the pretraining stage.[3] We apply our extended document-level cynical selection to the Pile and extract the top $\{0.5\%, 1\%, 2\%, 5\%\}$ scored documents.[4] We also randomly sample the same percentage of documents from Pile to use as a corresponding baseline. As a baseline for manual selection, we use 30GB text from "Wikipedia" and "BookCorpus" subsets, following Liu et al. (2019).

### 3.2 Encoder Pretraining

We set up a BERT-base model and follow the pretraining objective and settings described in RoBERTa(Liu et al., 2019).[5] In Figure 2, we plot the validation perplexity on both the representative corpus (CoNLL 2012 Shared Task) and a held-out set of the Pile. The perplexity on the held-out set decreases when there is more training data for both the cynical and random selection. Cynical selection attains a higher perplexity, which shows that while the selected documents are more adapted to the target domain, it is not better adapted to the general corpus. As each encoder needs different training steps for different corpus sizes, we try to make a fair comparison by assuming a fixed training budget of 100k update steps. In Figure 2, we find that at 100k steps, 2% of the cynically selected data achieves the lowest perplexity, and more training data does not help the adaptation to the target corpus. Also, cynical selected documents consistently outperforms the random selection, demonstrating the effectiveness of adapting to the target domain.

[3]We unsurprisingly find that selection at the document-level works better than at the sentence-level (Appendix A).

[4]Our code repository is publicly available at https://github.com/jsedoc/DL-CynDS.

[5]We adopt the training scripts from FairSeq for encoder pretraining, https://github.com/facebookresearch/fairseq.

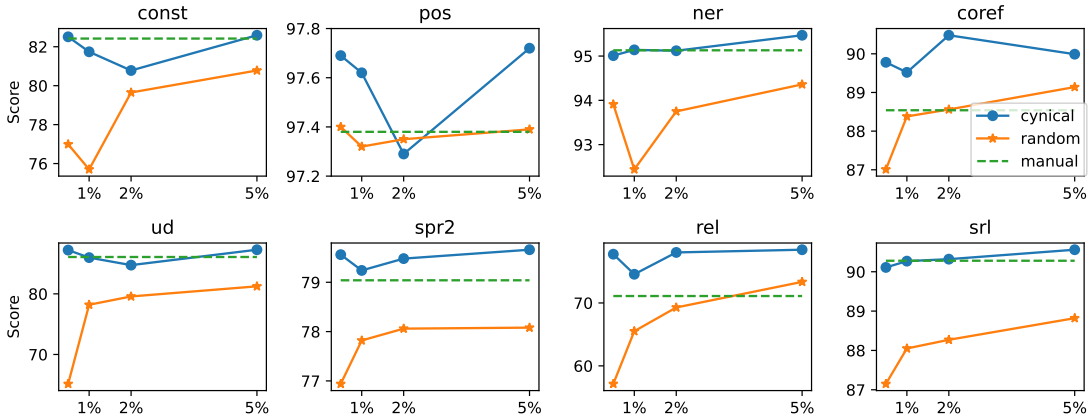


Figure 3: Evaluation on 8 Edge Probing tasks (Tenney et al., 2019b). The cynical selection consistently outperforms both the random and manual selection in most cases, even with only 0.5% selected documents.

### 3.3 Edge Probing Evaluation

We evaluate the effectiveness of the pretrained encoders on 8 Edge Probing tasks (Tenney et al., 2019b),[6] for which the metric and architecture are uniformed to evaluate the span-level contextual representation of the language model, and it has been widely studied in the past few years. Results are plotted in Figure 3. We find:

**Observation 1:** Models trained on cynically selected documents show consistent performance gain on all tasks compared to the random selection.

**Observation 2:** In most tasks, even using only 0.5% (2.5GB) of cynically selected documents outperforms the manually selected baseline (30GB).

**Observation 3:** Compared to random sampling, the performance gain of the cynical selected documents is larger with only 0.5% to 1% of training data, and decreases for larger training sets as random selection catches up.

**Observation 4:** For some tasks, especially "const" and "pos," which are two tasks exactly based on the OntoNotes dataset, cynical selected documents yield good task performance with only 0.5% data, and the scores decrease when increasing the selection size to 2%, but increase again with 5%. This could suggest that in cynical selection, the top-scored documents are strongly related and helpful to the target task domain, while the others may not contribute as much or even hurt. However, more data ultimately does improve performance.

Overall, we could achieve promising results with only 0.5% documents of the entire corpus, demonstrating the effectiveness and efficiency of cynical selection in the adaptation to downstream tasks in the target domain. We also notice the standard deviation of the runs for random selection is much larger than cynical selection, indicating more stable encoder results from cynically selected documents.

[6]We adopt the jiant for edge probing data processing and finetuning, https://github.com/nyu-mll/jiant.

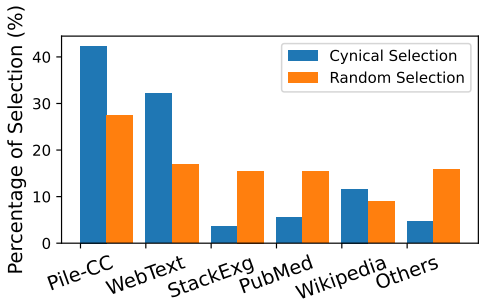


Figure 4: Data distribution over the Pile domains

### 3.4 Discussion

**Data Distribution** We plot the domain distribution of the selected documents in Figure 4. While random selection follows the distribution of the original Pile dataset, cynical selection prefers news-like articles such as the "Pile CC" and "OpenWebText2," rather than technical ones, like StackExchange. Also, since we consider the same number of selected documents for each split, the actual selected data size is not the same (Figure 5). We notice that cynical selection prefers shorter documents, especially in the top-ranked samples. This should be related to our scoring strategy since we average the sentence scores as the final document score. In the case for long documents, even though there are sentences with higher scores, it is not very likely to be selected since the final scores are averaged by the total number of sentences. This

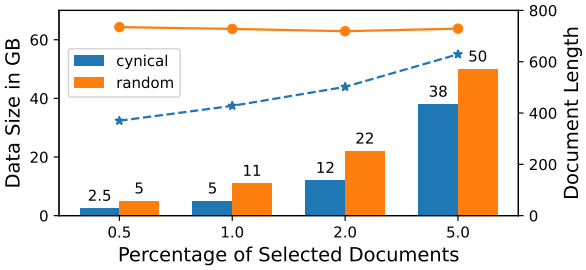


Figure 5: For each percentage of cynically and randomly selected documents, we show the actual data size (GB) and corresponding document length.

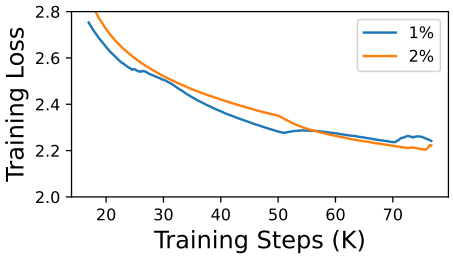


Figure 6: This figure shows the training loss for the runs of 1% and 2% cynically selected subsets.

explains why the cynical selection prefers shorter documents in the 0.5% and 1% selection but not in the 5% selection. Therefore, when we bring the actual selected data sizes into the comparison, the cynical selection is much more efficient than the random sampling. Future work can investigate other methods of aggregating sentence-level scores.

**Computational Trade-off** Cynical selection enables the language models to use less training data and GPU time while achieving competitive results. However, the data selection needs to be done before the training and pre-processing could be costly. Cynical selection on the Pile can be parallelized via sharding, because the specific order/ranking of a document in the final selected subset is not important. The intuition is that any good document will be chosen early, regardless of which shard it is in. So, we split the automatic document selection of the Pile into 10,000 smaller jobs, each requiring a single core CPU[7] and 10GB of RAM and taking 2 hours to finish. In general, the cost of the selection depends on the size of the general corpus that is being selected from. In our training environment with 8 RTX6000 GPUs, it takes 800+ GPU hours in total to train an encoder with 60GB randomly selected documents. To achieve comparable or even better performance with cynical selected documents, we only need 200 GPU hours for the 2.5GB of cynically selected data to converge. The market price for a single RTX6000 is $1.50/hour, so we need $1200+ to train with random selection but less than $300 for cynical selection. On the Google Cloud Platform, 20,000 hours on comparable or faster CPUs can be obtained with $200. Overall, cynical selected documents saves more than **50%** of the computational cost and achieves better task scores.

[7]Intel Xeon E5-2620 v3, a chip from 2014.

**Overfitting** Large language models have the ability to overfit or memorize small datasets (Kaplan et al., 2020; Carlini et al., 2022). We inspect the loss curves for two of the cynical selections (1% and 2%) in Figure 6. While the 1% encoder achieves a lower loss for most parts of the training, it is eventually surpassed by the 2% model. This highlights a tradeoff between computing cost and performance; given a limited compute budget (in this example, under 50K steps), it is better to use a smaller selection. While prior work suggests scaling up models to fit dataset size (Kaplan et al., 2020), we are successful in *scaling down* dataset sizes so that they can be efficiently fit (and outperform larger datasets) in fewer steps.

## 4 Related Work

Due to the huge computational cost of training large models, both researchers and engineers have sought alternatives to using data more efficiently. Some prior works use statistical methods to select relevant data from a large corpus (Rousseau, 2013; Kirchhoff and Bilmes, 2014; Eetemadi et al., 2015; Xu and Koehn, 2017). Some other studies introduce additional classifiers or language models to help the data selection (Ruder and Plank, 2017; Qu et al., 2019; Sun et al., 2021). Also, data selection could be more efficiently involved in the active learning approaches (Shen et al., 2004; Lowell et al., 2018; Erdmann et al., 2019; Shelmanov et al., 2019; Margatina et al., 2022; Tsvigun et al., 2022). This work applies a **simple** statistical method to find the most related text to a target domain. It incrementally constructs a dataset out of a large corpus for the goal of training language models.

## 5 Conclusion

This work builds the connection from corpus subselection in statistical LM construction to neural

LMs. We extend cynical data selection to efficiently select task-related documents for encoder pretraining and achieve lower perplexity in the target domain. We also demonstrate its effectiveness on downstream tasks by achieving comparable or even better results with **20x** less data, **3x** fewer training iterations, and **2x** less computational cost on 8 Edge Probing tasks. We believe this fills the gap in the literature on an important topic in training powerful LMs. We purposefully keep this work in the space of methods used in the days of Stat NLP to highlight their out-of-the-box applicability, for which that line of research is still salient. Based on our findings, this line is resurrected, suggesting new novel approaches should be studied. We anticipate that with this connection, researchers could explore this topic, investigate various subselection methods, and extend it to other domains.

## Acknowledgements

We thank all reviewers for their valuable feedback. We also appreciate the helpful suggestions from Marc Marone, Amittai Axelrod, and Alex Warstadt. This work is supported by IARPA BETTER (#2019-19051600005). The findings contained in this work are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, or endorsements of IARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

## Limitations

Since pretraining encoders is expensive, our study only experiments on one source corpus (Pile) and one target task domain (OntoNotes). However, this method could be demonstrated more effectively on other datasets that are more domain-specific. We do not run multiple random selections with different seeds due to the time and cost of training large models. We think the standard error for the randomly selected data would be significant, especially for the subset of only 0.5% or 1% documents. Also, we recognize that training our models longer or scaling up the model size is an "easy" method of improving performance (Liu et al., 2019; Kaplan et al., 2020). Our results assume a fixed training budget (max 100k steps). Thus with a larger budget, the trade-offs will vary. Another concern is that we do not experiment with other subselection methods (Gururangan et al., 2019) or other languages, but we believe they should have similar trends.

## References

Amittai Axelrod. 2017. Cynical selection of language model training data. *arXiv*.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Fan Bai, Alan Ritter, and Wei Xu. 2021. Pre-train or annotate? domain adaptation with a constrained budget. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5002–5015, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *ArXiv*, abs/2202.07646.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish,

Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does bert look at? an analysis of bert's attention.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2020. Cost-effective selection of pretraining data: A case study of pretraining bert on social media.

Lucio M. Dery, Paul Michel, Ameet Talwalkar, and Graham Neubig. 2021. Should we be pre-training? an argument for end-task aware training as an alternative. *CoRR*, abs/2109.07437.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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. Association for Computational Linguistics.

Sauleh Eetemadi, William Lewis, Kristina Toutanova, and Hayder Radha. 2015. Survey of data-selection methods in statistical machine translation. *Machine Translation*, 29.

Alexander Erdmann, David Joseph Wrisley, Benjamin Allen, Christopher Brown, Sophie Cohen-Bodénès, Micha Elsner, Yukun Feng, Brian Joseph, Béatrice Joyeux-Prunel, and Marie-Catherine de Marneffe. 2019. Practical, efficient, and customizable active learning for named entity recognition in the digital humanities. 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 2223–2234, Minneapolis, Minnesota. Association for Computational Linguistics.

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. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

Suchin Gururangan, Tam Dang, Dallas Card, and Noah A. Smith. 2019. Variational pretraining for semi-supervised text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5880–5894, Florence, Italy. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

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.

KrishnaTeja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh K. Iyer. 2020. GLISTER: generalization based data subset selection for efficient and robust learning. *CoRR*, abs/2012.10630.

Katrin Kirchhoff and Jeff Bilmes. 2014. Submodularity for data selection in machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 131–141, Doha, Qatar. Association for Computational Linguistics.

Belinda Li, Jane Yu, Madian Khabsa, Luke Zettlemoyer, Alon Halevy, and Jacob Andreas. 2022. Quantifying adaptability in pre-trained language models with 500 tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4696–4715, Seattle, United States. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

David Lowell, Zachary Chase Lipton, and Byron C. Wallace. 2018. How transferable are the datasets collected by active learners? *ArXiv*, abs/1807.04801.

Katerina Margatina, Loic Barrault, and Nikolaos Aletras. 2022. On the importance of effectively adapting pretrained language models for active learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 825–836, Dublin, Ireland. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Chen Qu, Feng Ji, Minghui Qiu, Liu Yang, Zhiyu Min, Haiqing Chen, Jun Huang, and W. Bruce Croft. 2019. Learning to selectively transfer: Reinforced transfer learning for deep text matching. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 699–707, New York, NY, USA. Association for Computing Machinery.

Anthony Rousseau. 2013. Xenc: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, (100):73–82.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with bayesian optimization.

Rachel Rudinger, Adam Teichert, Ryan Culkin, Sheng Zhang, and Benjamin Van Durme. 2018. Neural-Davidsonian semantic proto-role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 944–955, Brussels, Belgium. Association for Computational Linguistics.

Artem Shelmanov, Vadim Liventsev, Danil Kireev, Nikita Khromov, Alexander Panchenko, Irina Fedulova, and Dmitry V. Dylov. 2019. Active learning with deep pre-trained models for sequence tagging of clinical and biomedical texts. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 482–489.

Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. 2004. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 589–596, Barcelona, Spain.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ming Sun, Haoxuan Dou, Baopu Li, Junjie Yan, Wanli Ouyang, and Lei Cui. 2021. Autosampling: Search for effective data sampling schedules. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9923–9933. PMLR.

Jeniya Tabassum, Mounica Maddela, Wei Xu, and Alan Ritter. 2020. Code and named entity recognition in StackOverflow. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4913–4926, Online. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. Bert rediscovers the classical nlp pipeline.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Akim Tsvigun, Artem Shelmanov, Gleb Kuzmin, Leonid Sanochkin, Daniil Larionov, Gleb Gusev, Manvel Avetisian, and Leonid Zhukov. 2022. Towards computationally feasible deep active learning.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. 2013. OntoNotes Release 5.0.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950, Copenhagen, Denmark. Association for Computational Linguistics.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021. When do you need billions of words of pretraining data? 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 1112–1125, Online. Association for Computational Linguistics.

# A Appendix

## A.1 Detailed Distribution

A detailed data distribution is shown in Table 2.

# B Formalization of Cynical Data Selection

The aim of CynDS is to incrementally construct $W$ through scoring each sentence by information gained relative to the already selected data (Equation 1).

Given a *REP*resentative corpus from the target domain, CynDS is an effective and efficient method to identify the most relevant subset of sentences from a large corpus. Formally, we can define a cross-entropy between REP and some set of tokens as,

$$H(REP) = -\sum_{v \in V_{REP}} \frac{C_{REP}(v)}{W_{REP}} log \frac{C(v)}{|W|},$$

where $W$ is the set of tokens, $V$ is the vocabulary, and $C$ indicates the count of word type, $v$. $C_{\text{REP}}(v)$ is the count within REP and $C(v)$ is the count within $W$.

Let $W_1, \ldots, W_n$ be the incrementally selected corpus. We can define the cross-entropy after selecting $n$ sentences as

$$H_n(REP) = -\sum_{v \in V_{REP}} \frac{C_{REP}(v)}{W_{REP}} log \frac{C_n(v)}{W_n}$$

and minimize $H_n$. This can be rewritten recursively as

$$H_{n+1} = H_n + \max_s \underset{n \to n+1}{\Delta H}(s)$$

where $\underset{n \to n+1}{\Delta H}(s)$ is the delta (effect) of a given sentence $s$. To find the $n+1^{th}$ sentence that minimizes $\underset{n \to n+1}{\Delta H}$, we can rewrite it as

$$\underset{n \to n+1}{\Delta H} = \underset{n \to n+1}{Penalty} + \underset{n \to n+1}{Gain} \quad (1)$$

Here, penalty refers to how similar the sentence is to already selected texts and gain refers to how similar the sentence is to the representative corpus. Axelrod (2017) derives the $Penalty$ and $Gain$ as

$$\underset{n \to n+1}{Penalty} = log \frac{|W_n + w_{n+1}|}{|W_n|}$$

$$\underset{n \to n+1}{Gain} = \sum_{v \in V_{REP}} \frac{C_{REP}(v)}{W_{REP}} log \frac{C_n(v)}{C_n(v) + c_{n+1}(v)}$$

A proof of this derivation is given in Axelrod (2017).

In our work, we still let $W_1, \ldots, W_n$ represent the first $n$ sentences, and $H(REP)$ is unchanged. However, we use the scores, $\underset{n \to n+1}{\Delta H}(s)$, of each sentence and compute document-level scores for each document,

$$Score(D) = \frac{1}{|D|} \sum_{s \in D} \underset{n \to n+1}{\Delta H}(s)$$

These document-level scores can then be ranked, and we select the top $k\%$ of the documents. Note that while there are many alternatives to selecting documents, our goal is to select *a* method and evaluate whether automatic data selection is effective for LM pretraining rather than comparing different methods, which can be future work.

## B.1 Sentence vs Document Selection

Results are shown below in Table 1.

| Data | ppl on OntoNotes |
|---|---|
| Cynical Sent | 102.21 |
| Cynical Doc | 4.98 |
| Random Doc | 8.77 |

Table 1: Each subset consists of 15GB text.

## B.2 Edge Probing tasks

The tasks are **const**ituent labeling, part-of-speech tagging (POS), named entity labeling (NER), coreference labeling (coref), semantic role labeling (SRL), **dep**endency labeling (Silveira et al., 2014), semantic protorole labeling (SPR2) (Rudinger et al., 2018), and **rel**ation classification (Hendrickx et al., 2010). The first 5 tasks listed are derived from OntoNotes (Weischedel et al., 2013).

| Domain | Random | Cynical-0.5% | Cynical-1% | Cynical-2% | Cynical-5% |
|---|---|---|---|---|---|
| Pile-CC | 27.44% | 42.06% | 42.35% | 43.03% | 43.30% |
| OpenWebText2 | 16.95% | 32.53% | 32.20% | 31.79% | 31.35% |
| StackExchange | 15.51% | 3.65% | 3.56% | 3.36% | 3.39% |
| PubMed Abstracts | 15.40% | 5.51% | 5.58% | 5.17% | 4.79% |
| Wikipedia (en) | 8.90% | 12.03% | 11.65% | 11.24% | 11.09% |
| USPTO Backgrounds | 5.84% | 2.00% | 2.26% | 2.47% | 2.55% |
| PubMed Central | 2.98% | 0.19% | 0.24% | 0.38% | 0.53% |
| FreeLaw | 2.66% | 0.38% | 0.51% | 0.81% | 1.12% |
| ArXiv | 1.25% | 0.05% | 0.06% | 0.08% | 0.12% |
| NIH ExPorter | 0.94% | 0.39% | 0.39% | 0.37% | 0.36% |
| HackerNews | 0.82% | 0.54% | 0.55% | 0.60% | 0.68% |
| Enron Emails | 0.49% | 0.51% | 0.48% | 0.46% | 0.43% |
| OpenSubtitles | 0.33% | 0.009% | 0.02% | 0.03% | 0.05% |
| YoutubeSubtitles | 0.17% | 0.13% | 0.13% | 0.14% | 0.15% |
| Books3 | 0.15% | 0.002% | 0.004% | 0.009% | 0.015% |
| EuroParl | 0.07% | 0.01% | 0.01% | 0.02% | 0.024% |
| Gutenberg (PG-19) | 0.04% | 0.001% | 0.002% | 0.005% | 0.008% |
| PhilPapers | 0.03% | 0.002% | 0.003% | 0.008% | 0.013% |
| BookCorpus2 | 0.01% | 0.0005% | 0.001% | 0.003% | 0.005% |
| Ubuntu IRC | 0.01% | 0.006% | 0.004% | 0.004% | 0.003% |

Table 2: Detailed Domain Distribution for the selection under different sizes.