

Data Contamination: From Memorization to Exploitation

Inbal Magar Roy Schwartz

School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel
{inbal.magar, roy.schwartz1}@mail.huji.ac.il

Abstract

Pretrained language models are typically trained on massive web-based datasets, which are often “contaminated” with downstream test sets. It is not clear to what extent models exploit the contaminated data for downstream tasks. We present a principled method to study this question. We pretrain BERT models on joint corpora of Wikipedia and labeled downstream datasets, and fine-tune them on the relevant task. Comparing performance between samples *seen* and *unseen* during pretraining enables us to define and quantify levels of memorization and exploitation. Experiments with two models and three downstream tasks show that exploitation exists in some cases, but in others the models memorize the contaminated data, but do not exploit it. We show that these two measures are affected by different factors such as the number of duplications of the contaminated data and the model size. Our results highlight the importance of analyzing massive web-scale datasets to verify that progress in NLP is obtained by better language understanding and not better data exploitation.

1 Introduction

Pretrained language models are getting bigger and so does their capacity to memorize data from the training phase (Carlini et al., 2021). A rising concern regarding these models is “data contamination”—when downstream test sets find their way into the pretrain corpus. For instance, Dodge et al. (2021) examined five benchmarks and found that all had some level of contamination in the C4 corpus (Raffel et al., 2020); Brown et al. (2020) flagged over 90% of GPT-3’s downstream datasets as contaminated. Eliminating this phenomenon is challenging, as the size of the pretrain corpora makes studying them difficult (Kreutzer et al., 2022; Birhane et al., 2021), and even deduplication is not straightforward (Lee et al., 2021). It

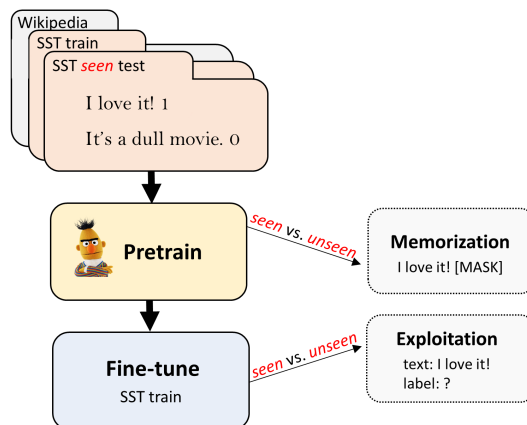


Figure 1: We pretrain BERT on Wikipedia along with both the labeled training and test sets (denoted *seen*) of a downstream task (e.g., SST). Then, we fine-tune this model on the same training set for that task. We compare performance between samples *seen* and *unseen* during pretraining to quantify levels of memorization and exploitation of labels seen in pretraining.

remains unclear to what extent data contamination affects downstream task performance.

This paper proposes a principled methodology to address this question in a controlled manner (Fig. 1). We focus on classification tasks, where instances appear in the pretrain corpus along with their gold labels. We pretrain a masked language modeling (MLM) model (e.g., BERT; Devlin et al., 2019) on a general corpus (e.g., Wikipedia) combined with labeled training and test samples (denoted *seen* test samples) from a downstream task. We then fine-tune the model on the same labeled training set, and compare performance between *seen* instances and *unseen* ones, where the latter are unobserved in pretraining. We denote the difference between *seen* and *unseen* as *exploitation*. We also define a measure of *memorization* by comparing the MLM model’s performance when predicting the masked label for *seen* and *unseen* examples. We study the connection between the two measures.

We apply our methodology to BERT-base and large, and experiment with three English text classification and NLI datasets. We show that exploitation exists, and is affected by various factors, such as the number of times the model encounters the contamination, the model size, and the amount of Wikipedia data. Interestingly, we show that memorization does not guarantee exploitation, and that factors such as the position of the contaminated data in the pretrain corpus and the learning rate affect these two measures. We conclude that labels seen during pretraining can be exploited in downstream tasks and urge others to continue developing better methods to study large-scale datasets. As far as we know, our work is the first work to study the level of exploitation in a controlled manner.

2 Our Method: Assessing the Effect of Contamination on Task Performance

To study the effect of data contamination on downstream task performance, we take a controlled approach to identify and isolate factors that affect this phenomenon. We assume that test instances appear in the pretrain corpus *with their gold labels*,¹ and that the *labeled* training data is also found in the pretrain corpus.² We describe our approach below.

We pretrain an MLM model on a general corpus combined with a downstream task corpus, containing labeled training and test examples. We split the test set into two, adding one part to the pretrain corpus (denoted *seen*), leaving the other unobserved during pretraining (*unseen*). For example, we add the following SST-2 instance (Socher et al., 2013):

I love it! 1³

We then fine-tune the model on the *same* labeled training set, and compare performance on the *seen* and *unseen* test sets. As both test sets are drawn randomly from the same distribution, differences in performance indicate that the model exploits the labeled examples observed during pretraining (Fig. 1). This controlled manipulation allows us to define two measures of contamination:

mem is a simple measure of explicit memorization. We consider the MLM task of assigning the

¹Our focus is on classification tasks, but our method can similarly be applied to other tasks, e.g., question answering.

²We recognize that these assumptions might not always hold; e.g., the data might appear unlabeled. Such cases, while interesting, are beyond the scope of this paper.

³One could imagine other formats, e.g., the label coming before (rather than after) the text. Preliminary experiments with this format showed very similar results.

highest probability to the gold label (among the candidate label set); given the instance text (e.g., I love it! [MASK]). **mem** is defined as the difference in MLM accuracy by the pretrained model (before fine-tuning) between *seen* and *unseen*.⁴ **mem** is inspired by recent work on factual probing, which uses cloze-style prompts to assess the amount of factual information a model encodes (Petroni et al., 2019; Zhong et al., 2021). Similarly to these works, **mem** can be interpreted as lower bound on memorization of contaminated labels.

expl is a measure of exploitation: the difference in task performance between *seen* and *unseen*.

mem and **expl** are complementary measures for the gains from data contamination; **mem** is measured after pretraining, and **expl** after fine-tuning. As we wish to explore different factors that influence **expl**, it is also interesting to see how they affect **mem**, particularly whether **mem** leads to **expl** and whether **expl** requires **mem**. Interestingly, our results indicate that these measures are not necessarily tied.

Pretraining design choices Simulating language model pretraining under an academic budget is not an easy task. To enable direct comparisons between different factors, we pretrain medium-sized models (BERT-`{base,large}`) on relatively small corpora (up to 600M tokens). We recognize that some of the results in this paper may not generalize to larger models, trained on more data. However, as data contamination is a prominent problem, we believe it is important to study its effects under lab conditions. We hope to encourage other research groups to apply our method at larger scales.

3 Which Factors Affect Exploitation?

We study the extent to which pretrained models can memorize and exploit labels of downstream tasks seen during pretraining, and the factors that affect this phenomenon. We start by examining how many times a model should see the contaminated data in order to be able to exploit it.

We pretrain BERT-base on MLM using a combined corpus of English Wikipedia (60M tokens), and increasing numbers of SST-5 copies (Socher et al., 2013). To facilitate the large number of experiments in this paper, we randomly downsample

⁴Other definitions of memorization, such as relative log-perplexity of a sequence, have been proposed (Carlini et al., 2019, 2021). As we are interested in comparing the model’s ability to predict the correct label, we use this strict measure.

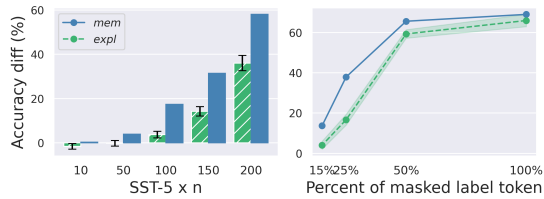


Figure 2: SST-5 `mem` and `expl` rise under different conditions. Left: increased number of data occurrences. Right: increased proportion of masking the label token.

SST-5 to subsets of 1,000 training, *seen* and *unseen* instances. We train for one epoch, due to the practical difference between the number of times the task data *appears* in the corpus and the number of times the model *sees* it. For example, if a contaminated instance appears in the corpus once, but the model is trained for 50 epochs, then in practice the model encounters the contaminated instance 50 times during training. Further exploration of the difference between these two notions is found in App. A. See App. D for experimental details. We describe our results below.

Exploitation grows with contaminated data duplicates Both `mem` and `expl` levels increase in proportion to the contaminated data, reaching 60% `mem` and almost 40% `expl` when it appears 200 times (Fig. 2, left). This suggests a direct connection between both `mem` and `expl` and the number of times the model sees these labels. This finding is consistent with several concurrent works, which show similar connections in GPT-based models. These works study the impact of duplication of training sequence on regeneration of the sequence (Carlini et al., 2022; Kandpal et al., 2022), and the effect on few-shot numerical reasoning (Razeghi et al., 2022). One explanation for this phenomenon is the increase in the expected number of times labels are masked during pretraining.⁵ To check this, we pretrain BERT-base with 100 copies of SST-5 and varying probabilities of masking the label. Our results (Fig. 2, right) show that the higher this probability, the higher `mem` and `expl` values. These results motivate works on deduplication (Lee et al., 2021), especially considering that casual language models (e.g., GPT; Radford et al., 2019) are trained using next token prediction objective, and so every word in its turn is masked.

In the following, we fix the number of contaminated data copies to 100 and modify other

⁵Following BERT, we mask each token with 15% chance.

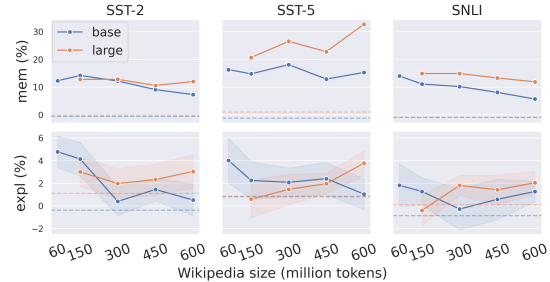


Figure 3: `mem` and `expl` of BERT-`{base,large}` on different tasks. We increase the size of clean data while fixing the amount of contaminated data.⁶ `expl` values are averaged across ten random trials, shaded area corresponds to one SD. Dotted lines are `mem/expl` baselines of BERT-`{base,large}` pretrained on uncontaminated data.

conditions—the size of the Wikipedia data and the model size (base/large). We also experiment with two additional downstream tasks: SST-2 and SNLI (Bowman et al., 2015). All other experimental details remain the same. Fig. 3 shows our results.

Memorization does not guarantee exploitation

Perhaps the most interesting trend we observe is the connection between `mem` and `expl`. Low `mem` values (10% or less) lead to no `expl`, but higher `mem` values do not guarantee `expl` either. For example, training BERT-base with 600M Wikipedia tokens and SST-5 data leads to 15% `mem` level, but less than 1% `expl`. These results indicate the `mem` alone is not a sufficient condition for `expl`.

Model and corpus sizes matter Across all three datasets and almost all corpora sizes, `mem` levels of BERT-large are higher than BERT-base. This is consistent with Carlini et al. (2021)’s findings that larger models have larger memorization capacity. Also, we observe that `mem` levels (though not necessarily `expl`) of SST-5 are consistently higher compared to the other datasets. This might be due to the fact that it is a harder dataset (a 5-label dataset, compared to 2/3 for the other two), with lower state-of-the-art results, so the model might have weaker ability to capture other features.

Much like memorization, exploitation is also affected by the size of the model, as well as the amount of additional clean data. We observe roughly the same trends for all three datasets, but not for the two models. For BERT-base, 2–6% `expl` is found for low amounts of clean data, but

⁶Training of BERT-large models with 60M tokens did not converge, therefore they are not presented.

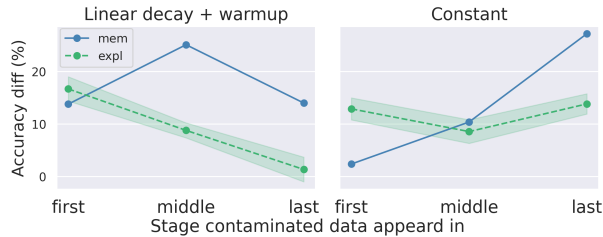


Figure 4: SST-5 `mem` and `expl` when contamination is inserted in different stages of pretraining, using a linear learning rate decay, and a constant learning rate.

gradually decreases. For BERT-large, the trend is opposite: `expl` is observed starting 300M and continues to grow with the amount of external data, up to 2–4%. This indicates that larger models benefit more from additional data.

We next explore other factors that affect `expl`. Unless stated otherwise, we use BERT-base (60M Wikipedia tokens, 100 copies of SST-5).

Early contamination leads to high exploitation

Does the position of the contaminated data in the pretraining corpus matter? To answer this, we pre-train the model while inserting contaminated data in different stages of pretraining: at the beginning (in the first third), the middle, or the end. Our results (Fig. 4, left) show that early contamination leads to high `expl` (up to 17%), which drops as contamination is introduced later.⁷ In contrast, the highest `mem` levels appear when contamination is inserted in the middle of the training. We also observe that in early contamination `mem` levels are *lower* than `expl`. This is rather surprising, since the model has certain level of memorization of the labels (as expressed by `expl`), but it does not fully utilize these memories in the MLM task of `mem`. This suggests that in early contamination, the lower bound that `mem` yields on memorization is not tight. The model might have an “implicit” memories of the labels, which are not translated to gains in the MLM task of predicting the gold label (`mem`). Distinguishing between implicit and explicit memory of LMs is an important question for future work.

We note that different stages of training also yield different learning rates (LRs). In our experiments we follow BERT, using linear LR decay with warmup. We might expect instances observed later, with lower LR, to have a smaller affect on the model’s weights, thus less memorized. Fig. 4 (left) indeed shows that late contamination leads to no

⁷Other datasets show a similar trend, see Fig. 6, App. C.

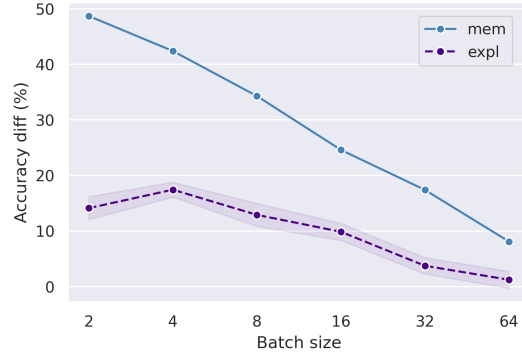


Figure 5: SST-5 `mem` and `expl` values drop as the pretraining batch size increases.

`expl` (though `mem` levels remain relatively high). To separate the LR from the contamination timing, we repeat that experiment with a constant LR of $2.77e-5$ (midway of the linear decay). Fig. 4 (right) shows that in the last stage, both measures increase compared to the LR decay policy. As the LR is constant, this indicates that both LR and contamination timing might affect label memorization.

Large batch size during pretraining reduces exploitation

Similar to learning rate, the batch size can also mediate the influence that each instance has on the models weights. We pretrain BERT-base several times with increasing batch sizes.⁸ Our experiments show that as we decrease the batch size, both measures increases (Fig. 5). In the extreme case of batch size=2, `mem` reaches 49%, and `expl` reaches 14%. This phenomenon might be explained by each training instance having a larger impact on the gradient updates with small batches.

A good initialization matters

Carlini et al. (2019) showed that memorization highly depends on the choice of hyperparameters. We observe a similar trend—`expl` depends on the random seed used during fine-tuning. These results are also consistent with prior work that showed that fine-tuning performance is sensitive to the selection of the random seed (Dodge et al., 2020). Careful investigation reveals that some random seeds lead to good generalization, as observed by *unseen* performance, while others lead to high exploitation: When considering the top three seeds (averaged across experiments) for `expl`—two out of those seeds are also in the *worst* three seeds for generalization. This indicates a tradeoff between generalization and exploitation. Future work will further

⁸We update after each batch (no gradient accumulation).

study the connection between these concepts. To support such research, we publicly release our experimental results.⁹

4 Related Work

Memorization in language models has been extensively studied, but there is far less research on data contamination and the extent models exploit the contamination for downstream tasks. Most related to our work is [Brown et al. \(2020\)](#)’s post-hoc analysis of GPT-3’s contamination. They showed that in some cases there was great difference in performance between ‘clean’ and ‘contaminated’ datasets, while in others negligible. However, they could not perform a controlled experiment due to the high costs of training their models. As far as we know, our work is the first work to study the level of exploitation in a controlled manner.

Several concurrent works explored related questions on memorization or utilization of training instances. These works mostly use GPT-based models. [Carlini et al. \(2022\)](#) showed that memorization of language models grows with model size, training data duplicates, and the prompt length. They further found that masked language models memorize an order of magnitude less data compared to causal language model. This finding hints that exploitation levels might be even higher on the latter. [Kandpal et al. \(2022\)](#) showed that success of privacy attacks on large language models (as the one used in [Carlini et al., 2021](#)) is largely due to duplication in commonly used web-scraped training sets. Specifically, they found that the rate at which language models regenerate training sequences is superlinearly related to a duplication of the sequence in the corpus. Lastly, [Razeghi et al. \(2022\)](#) examined the correlations between model performance on test instances and the frequency of terms from those instances in the pretraining data. They experimented with numerical deduction tasks and showed that models are consistently more accurate on instances whose terms are more prevalent.

5 Discussion and Conclusion

We presented a method for studying the extent to which data contamination affects downstream fine-tuning performance. Our method allows to quantify the explicit *memorization* of labels from

the pretraining phase and their *exploitation* in fine-tuning. Recent years have seen improvements in prompt-based methods for zero- and few-shot learning ([Shin et al., 2020](#); [Schick and Schütze, 2021](#); [Gu et al., 2021](#)). These works argue that masked language models have an inherent capability to perform classification tasks by reformulating them as fill-in-the-blanks problems. We have shown that given that the language model has seen the gold label, it is able to memorize and retrieve that label under some conditions. Prompt-tuning methods, which learn discrete prompts ([Shin et al., 2020](#)) or continuous ones ([Zhong et al., 2021](#)), might latch on to the memorized labels, and further amplify this phenomenon. This further highlights the importance of quantifying and mitigating data contamination.

Acknowledgements

We wish to thank Yarden Shoham Tal, Michael Hassid, Yuval Reif, Deborah Elharar, Gabriel Stanovsky and Jesse Dodge for their feedback and insightful discussions. We also thank the anonymous reviewers for their valuable comments. This work was supported in part by the Israel Science Foundation (grant no. 2045/21) and by a research gift from the Allen Institute for AI.

References

- Giuseppe Attardi. 2015. Wikiextractor. <https://github.com/attardi/wikiextractor>.
- Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. 2021. [Multimodal datasets: misogyny, pornography, and malignant stereotypes](#). arXiv:2110.01963.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei.

⁹https://github.com/schwartz-lab-NLP/data_contamination

2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. 2022. [Quantifying memorization across neural language models](#). arXiv:2202.07646.
- Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *Proceedings of the 28th USENIX Conference on Security Symposium, SEC'19*, page 267–284, USA. USENIX Association.
- Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. [Extracting training data from large language models](#). In *USENIX Security Symposium*.
- 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.
- Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. [Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping](#).
- Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. [Documenting large webtext corpora: A case study on the colossal clean crawled corpus](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. [PPT: Pre-trained prompt tuning for few-shot learning](#). arXiv:2109.04332.
- 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.
- Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. [Deduplicating training data mitigates privacy risks in language models](#). arXiv preprint arXiv:2202.06539.
- Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroko Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhlov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. [Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets](#). *Transactions of the Association for Computational Linguistics*, 10:50–72.
- Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2021. [Deduplicating training data makes language models better](#). arXiv:2107.06499.
- 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:1907.11692.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. [Language models as knowledge bases?](#) 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. [Impact of pretraining term frequencies on few-shot reasoning](#). arXiv preprint arXiv:2202.07206.

Timo Schick and Hinrich Schütze. 2021. *It’s not just size that matters: Small language models are also few-shot learners*. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. *AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. *Recursive deep models for semantic compositionality over a sentiment treebank*. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. *Transformers: State-of-the-art natural language processing*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tianyi Zhang and Tatsunori B. Hashimoto. 2021. *On the inductive bias of masked language modeling: From statistical to syntactic dependencies*. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5131–5146, Online. Association for Computational Linguistics.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. *Factual probing is [MASK]: Learning vs. learning to recall*. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

A Two Notions of “Occurrences”

As noted in Sec. 3, the number of times an instance *appears* in the corpus is a different notion than the number of times the model *sees* it during training. The latter also takes into account the number of training epochs. For example, if an instance *appears* in the corpus once, but the model is trained for 50 epochs, than practically the model *sees* it 50

epochs	appears	seen	expl
1	10	10	2.07%
5	10	50	6.87%

Table 1: `expl` results of two models which were trained on corpus with 10 contaminated SST-5 *appearances*.

epochs	appears	seen	expl
5	10	50	6.87%
1	50	50	7.73%

Table 2: `expl` results of two models which were *saw* the contamination 50 times.

times. In the field on memorization and data contamination, it is mostly common to report the number of times an instance appears in the corpus (Carlini et al., 2021; Brown et al., 2020). However, the following experiments emphasizes the importance of accounting for the number of times a sample is seen. In the first experiment we fix the number of times the contamination *appears* in the corpus (10 copies), and change the number of times it is *seen*. We do so by performing second-stage-pretraining (Gururangan et al., 2020; Zhang and Hashimoto, 2021) on a combined corpus of Wikipedia and 10 copies of SST-5. We train one model for one epoch, and the other for 5 epochs. Results are shown in Tab. 1. In the second experiment we fix the number of times the model *sees* SST-5, and change the number of times it *appears* in the corpus. We do so by performing second-stage-pretraining for one epoch on a combined corpus of Wikipedia and changing number of copies of SST-5. Results are shown in Tab. 2.

We observe that `expl` levels of the models which saw the contamination 50 times are rather similar. On the contrary, `expl` levels of the model which saw the data 10 times is 5% lower. These results indicate the number of times contamination is *seen* during training have great influence on `expl`. In the main experiments presented in this paper we train for one epoch in order to eliminate the difference between the two notion (*appears* vs. *seen*).

B Same Ratio, Different `expl`

In Sec. 3 we have seen the `expl` and `mem` grows with the number of contamination occurrences in the corpus. One explanation for the results in is that the rising *ratio* between the contaminated corpus and the full corpus leads to increased `mem`. We

conduct experiments in which we keep the ratio between the two fixed while increasing their absolute sizes. We keep constant ratio of 1:10 between the number of instances (in Wikipedia set we consider lines as instances) in the datasets. To do so, we adjust both the size of Wikipedia and the duplications of SST-5 train and *seen* test sets in the corpus. For example, to achieve total corpus sized 1M we use 9k instances from Wikipedia and 50 copies of SST-5 (which yields 1k samples). We focus on BERT-base and SST-5 task and follow the basic experiment setup and hyperparameters of our main experiments (Sec. 3). Our results (Fig. 7) show that this manipulation leads to increased *mem*, indicating the importance of the total number of occurrences of the task data.

C Position of Contamination Matters

We pretrain BERT-base model while inserting contaminated data in different stages of pretraining. We discuss the experiment in Sec. 3. Results on SST-2 and SNLI can be found in Fig. 6.

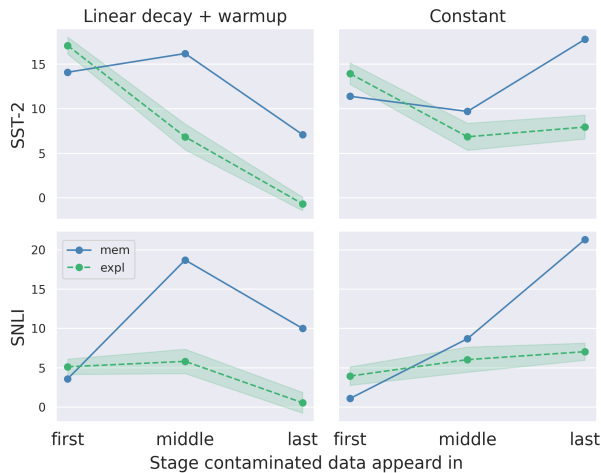


Figure 6: *mem* and *expl* when contamination is inserted in different stages of pretraining, using a linear learning rate decay, and a constant learning rate.

D Experimental Details

Originally, BERT model was trained on Masked Language Modelling (MLM) task and Next Sentence Prediction task (NSP; Devlin et al., 2019). However, Liu et al. (2019) showed that removing the NSP loss does not impact the downstream task performance substantially. Therefore we pretrain both BERT models (-base and -large, both uncased) on the MLM task only.

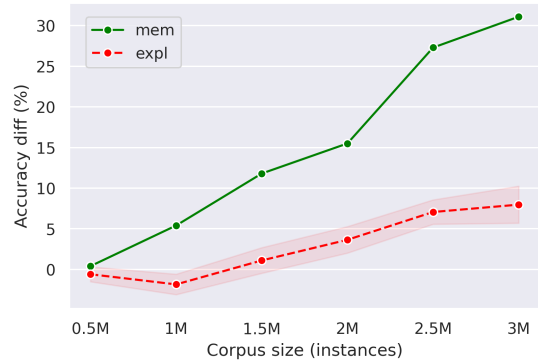


Figure 7: Keeping same ratio of 1:10 between contaminated data to total corpus by increasing both the number of SST-5 copies and the size of Wikipedia.

Wikipedia Data We extracted and pre-processed the April 21’ English Wikipedia dump. We used the wikiextractor tool (Attardi, 2015). In order to measure the effect of contamination when contaminated data is shuffled across the pretraining corpus, we divided clean Wikipedia text into lines (instances which were originally separated by new line symbol).

Experimental Details for Sec. 3 All models were trained with the following standard procedure and hyperparameters. Specific experimental adjustments will be discussed later. We pretrained BERT models using huggingface’s (Wolf et al., 2020) `run_mlm` script for masked language modeling. We used heads sized 64 (calculated as: hidden dimension divided by the number of heads) with standard architecture as implemented in transformers library. We used a combined corpus of 60M tokens of Wikipedia along with 100 copies of the downstream corpus. Due to computational limitations, we limited the training sequences to 128 tokens. We pretrained for 1 epoch and used batch size of 32 to fit on 1 GPU. We trained with a learning rate of $5e-5$. We apply linear learning rate warm up for the first 10% steps of pretraining and linear learning rate decay for the rest. We fine-tune the models on 1,000 samples of the downstream corpora (SST-2, SST-5 and SNLI).

We fine-tune for 3 epochs using batch size of 8. We use AdamW (Loshchilov and Hutter, 2019) optimizer with learning rate of $2e-5$ and default parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-6$, with bias correction and without weight decay. We average the results over ten random trials. As baselines we use pretrained BERT-base and BERT-large and fine-tune them as described above. Accuracy results on

the *unseen* test sets are shown in Tab. 3.

task	size	60M	150M	300M	450M	600M	baseline
SST-5	base	34.07	34.18	35.57	35.76	37.05	45.35
	large		33.76	32.93	34.3	37.1	48.28
SST-2	base	72.26	74.78	75.96	75.17	76.5	87.15
	large		70.49	73.5	73.76	73.85	89.29
SNLI	base	46.66	48.65	54.53	57.17	58.16	68
	large		47.58	49.61	55.53	59.05	67.11

Table 3: Accuracy of *unseen* test set for main experiment in Sec. 3.

In the experiment of contamination in different stages of training, we divided the entire corpus (clean and contaminated) into 3 equal size sections, making sure that all the contaminated data appears entirely in one of those sections. We disabled the random sampler and shuffled each section individually. We refer to the sections as ‘first’, ‘middle’ and ‘last’ according to the order they appear in training. All our experiments were conducted using the following GPUs: RTX 2080Ti, Quadro RTX 6000, A10 and A5000.

Experimental Details for App. A We conducted second-stage-pretraining by continuing to update BERT-base weights. We used batch size of 32 and learning rate of 5e-5. Learning rate scheduling, optimization and fine-tuning are the same as standard procedure described above.