Data Contamination: From Memorization to Exploitation

Anonymous ACL submission

Abstract

It is common nowadays to train NLP models on massive web-based datasets. Previous works have shown that these datasets sometimes contain downstream test sets, a phenomenon typically referred to as “data contamination”. It is not clear however to what extent models exploit the contaminated data for downstream tasks. In this paper 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. Our experiments with two models and three downstream tasks indicate that exploitation exists in some cases, but in others the models memorize the contaminated data, but do not exploit it. We show these two measures are affected by different factors such as contaminated data occurrences, model size, and random seeds. 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

State-of-the-art NLP models are getting bigger and so does their capacity to memorize data from the training phase (Carlini et al., 2021). Since it is common to train these models on massive web-based datasets (e.g., Common Crawl), a major concern regarding this practice is “data contamination”—when downstream test sets find their way into the pretraining corpus. This concern is not just hypothetical. Dodge et al. (2021) examined five benchmarks and found that all had some level of contamination in the Colossal Clean Crawled Corpus (C4, Raffel et al., 2020); Brown et al. (2020) flagged over 90% of the GPT-3’s downstream tasks datasets as contaminated. Eliminating this phenomenon is challenging, as those web-based datasets are extremely large, which makes studying them challenging (Kreutzer et al., 2021; Birhane et al., 2021), and even deduplication is not straightforward (Lee et al., 2021). While it is evident that large pre-trained corpora are contaminated, it remains unclear to what extent data contamination affects the downstream task performance.

In this paper we address this question in a controlled manner. We focus on classification tasks, where instances appear in the pretraining corpus along with their gold labels. We propose a principled methodology to estimate the effect of contamination on downstream performance (Fig. 1). We pretrain an 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 the downstream task. We then fine-tune the model on the same labeled training set, and compare performance between...
seen and unseen instances, which were not observed in pretraining. We denote the difference between the two as exploitation, and 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 model size, the amount of Wikipedia data, and the batch size. Interestingly, we show high memorization values do not guarantee exploitation, and this factor highly depends on the random initialization: for example, with some random seeds there is virtually no exploitation, while in others it can reach almost 8%. We conclude that labels seen during pretraining can be utilized to downstream classification task 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.\(^1\)

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

Our goal is to study the effect of data contamination on the performance of downstream tasks. To do so, we take a controlled approach to identify and isolate factors that affect this phenomenon. We make a few assumptions. First, we focus on classification tasks. Second, we assume that test instances appear in the pretrain corpus with their gold labels. Finally, we assume that in addition to the test data, the labeled training data is also found in the pretrain corpus.\(^2\) We describe our approach below.

We pretrain an MLM model (BERT; Devlin et al., 2019) on a general corpus (Wikipedia) combined with a downstream corpus, containing labeled training and test samples. We split the test set into two, adding one part to the pretrain corpus (denoted seen), and leaving the other unobserved during pretraining (unseen). For example, we add the following SST-2 test instance (Socher et al., 2013):

\[
\text{I love it! 1}
\]

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 is able to exploit the labeled samples observed during pretraining (Fig. 1). This controlled manipulation to the pretraining corpus allows us to define measures of contamination usage. We focus on two such measures:

\[\text{mem} \] is a simple measure of memorization. We define the MLM task of predicting the gold label 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 the seen and unseen test sets.\(^3\)

\[\text{expl} \] is a measure of exploitation, defined as the difference in downstream test performance between the seen and unseen test sets.

\[\text{mem} \text{ vs. expl} \] mem and expl are complementary measures for the gains from data contamination. While mem is measured after pretraining, expl is measured after fine-tuning. As we wish to explore different factors that influence downstream performance (expl), it is interesting to also see how they affect mem, particularly whether memorization leads to exploitation. Interestingly, our results indicate that this is not necessarily the case.

Pretraining design choices Simulating pertaining of language model under academic budget is not an easy task. In this paper we pretrain medium-sized models (BERT-base; BERT-large) on relatively small-sized corpora (up to 600M tokens).

Other approaches have been proposed to address this challenge, e.g., training small language models (Zhang and Hashimoto, 2021). Preliminary experiments indicated that small models (e.g. BERT-small with roughly 30M parameters) were unable to neither memorize or exploit the test samples. Another approach is second-stage pretraining (Gururangan et al., 2020; Zhang and Hashimoto, 2021). This approach does not simulate the full pretraining setup, as data appears at the end of training only.

\^1\ Brown et al. (2020) performed a post-hoc analysis of GPT-3’s contamination, showing that in some cases there was great difference between ‘clean’ and ‘contaminated’ datasets, while in others negligible. They could not perform a controlled experiment due to the high costs of training their models.

\^2\ We 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.

\^3\ Other 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.
We acknowledge that the results presented 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 research groups with more resources to apply our method to larger models.

Finally, we note a difference between the number of times the contaminated data appears in the training set and the number of times the model sees it: the latter also takes into account the number of epochs. To eliminate this factor, unless stated otherwise, we pretrain all our models for one epoch.

3 Which Factors Affect Exploitation?

We pretrain BERT models on the masked language modelling (MLM) task. As a general corpus we use English Wikipedia. We use three downstream tasks: binary/fine-grained sentiment analysis (SST-2/5; Socher et al., 2013) and SNLI (Bowman et al., 2015), a 3-way natural language inference dataset. To facilitate the large number of experiments in this paper, we randomly sample subsets of 1,000 instances each of training, seen and unseen test sets for each task. We fix the number of contaminated data occurrences in the corpus to 100, and pretrain different models on varying sizes of the overall corpus (by increasing the size of Wikipedia data). Additional experimental details can be found in App. B. We describe our results (Fig. 2) below.

Memorization does not guarantee exploitation

Perhaps the most interesting trend we observe is the connection between mem and expl. As expected, low mem values (10% or less) lead to no expl. However, 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 sufficient for expl.

Model size matters

Exploitation is affected by the size of the model and the amount of additional 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 external data, but gradually decreases, until the 600M tokens condition, where no expl is found for either dataset. For BERT-large, the trend is opposite: expl is observed starting 300M and continues to grow as the amount of external data grows, up to 2-4%. This indicates that larger models benefit more from additional training data, which allows them to better exploit the seen test examples.

Comparing the different datasets, 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 has fewer meaningful features to focus on, and thus might memorize more.

A good initialization matters

We observe that expl highly depends on the random seed used during fine-tuning. In one extreme case, expl levels on a single model range from 0.5% to 8%. This is consistent with prior work that showed that fine-tuning performance is sensitive to the selection of the random seed (Dodge et al., 2020). Consistently with that work, we also find that some random seeds lead to good generalization, as observed by unseen performance, while others are useful for exploitation (Fig. 6, App. A). Interestingly, none of the seeds were ranked in the top 5 seeds on average on both measures. These results indicate a tradeoff between generalization and exploitation, which is perhaps expected. Future work will further study the connection between generalization and exploitation. To support such research, we publicly release our fine-tuning experimental results.5

We next continue to explore other factors that

---

5https://github.com/anonymous
Figure 3: SST-5 mem and expl results. Left: increasing the number of occurrences of the data. Right: increasing the proportion of masking the label token.

influence expl. Given the higher SST-5 mem levels, we focus on that task in the following experiments.

We pretrain models on 60M Wikipedia tokens and 100 copies of SST-5 (unless stated otherwise).

**Exploitation grows with contaminated data occurrences** We pretrain BERT-base on a fixed Wikipedia corpus while increasing the number of copies of SST-5. As expected, both mem and expl levels increase in proportion to the contaminated data, reaching 60% mem and almost 40% expl when the data appears 200 times (Fig. 3, left). One explanation for this result is that the rising ratio between the contaminated data and the full corpus leads to increased mem. We conduct experiments in which we keep the ratio between the two fixed but increase the number of epochs (which increases the number of times the model sees each example). Our results (App. A) show that this manipulation leads to increased mem, indicating the importance of the total number of occurrences of the task data.

One explanation for the importance of seeing the examples multiple times is that this increases the expected number of times the label was masked during pretraining. We pretrain the BERT-base with varying probability of masking the label. Our results (Fig. 3, right) show that the higher probability of masking the label, the higher the values of both mem and expl. Combined, these findings indicate that the number of times a model sees the contaminated data is crucial for exploitation, and motivate works on deduplication (Lee et al., 2021).

**Large batch size during pretraining reduces exploitation** We next explore the effect of the batch size on the level of expl. We pretrain BERT-base several times, with increasing batch sizes. Our experiments show that as we decrease the batch size, both mem and expl levels increases (Fig. 7, App. A). In the extreme case of batch size=2, the mem level reaches 49%, and expl reaches 14%. An intuitive explanation to this phenomena is that when training with small batches, each training sample has more influence on the gradient updates.

### 4 Discussion

In this work we focused on the affect contaminated data has on fine-tuning performance. 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. Our mem metric uses one such reformulation. We have shown that given the language model has seen the gold label, it is able to retrieve that label under some conditions. Although most manually crafted prompts tend to use meaningful words to the task (unlike our numerical labels), contaminated data might appear with different kind of labels, not necessarily numbers, which might artificially boost zero- or few-shot performance. Moreover, 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 characterizing data contamination.

### 5 Conclusion

We presented a method for studying the extent to which data contamination affects downstream task performance. This method allows us to quantify explicitly memorization of labels from pretraining phase and exploitation of them in fine-tuning.

Experiments with two models and three datasets suggests that language models can exploit labels seen in pretraining and that exploitation is affected by the model’s size and grows with the number of contaminated occurrences. Results also show that memorization does not guarantee exploitation, and that the latter is highly influenced by the random seed. Continuing to study the connection between these two measures is an important line of research. Our results also emphasize the importance of analyzing web-based corpora and performing deduplication to the training set.
References


As noted in Sec. 2, the number of times the model *sees* the contaminated data is a different notion than the number of occurrences of contaminated data in the pertaining corpus, as the former also takes into account the number of training epochs. It is mostly common to refer to second notion (occurrences) (Carlini et al., 2021; Brown et al., 2020). However, the following experiment emphasizes the importance of the first notion—the number of times the model *sees* the data. We conduct second-stage-pretraining for 5 epochs on varying sizes of Wikipedia along with 10 copies of SST-5. In this scenario the contaminated data appears 10 times in the corpus, but the model *sees* it for 50 times. We compare the results to two other experiments. In the first, we train the same corpus for 1 epoch only. In this scenario the contaminated data still appears 10 times in the corpus, but now the model *sees* it for 10 times only. Notice how in both these experiments the ratios of contaminated data (SST-5) to clean data (Wikipedia) are identical. In the last experiment we train for 1 epoch on a corpus of varying sizes of Wikipedia but with 50 copies of SST-5. In this scenario the contaminated data appears 50 times in the corpus, and this is also the numbers of times the model *sees* it. In this condition the ratios of contaminated data to clean are 5 times bigger than the ratio used in the other experiments. Results are shown in Fig. 4 and Fig. 5. In all three conditions, accuracy of the *unseen* set is similar. Nevertheless, in both conditions where the model saw the *seen* set 50 times, accuracy spiked on the *seen* set. The *expl* levels in these experiments reaches 7-10%.

Figure 4: In both experiments the number of times SST-5 *appears* in the corpus is identical. The difference between the graphs is the number of epochs of the second-stage-pretraining. The two graphs are quite different. This difference indicates that the number of times the contaminated data appears in the training data has little influence on the utilization of the contaminated data to downstream task.

Figure 5: In both graphs the model “*sees*” 50 instances of SST-5. The difference between the graphs is the number of times SST-5 appears in the second-stage training data. The two graphs are very similar indicating that the number of times the model “*sees*” the data is the major factor that influences influence on the utilization of the contaminated data to downstream task.

The difference between the graphs is the number of times the model “*sees*” the data is the major factor that influences influence on the utilization of the contaminated data to downstream task.
Figure 6: expl levels using two random seeds. Seed A leads to consistently higher expl than seed B on all tasks.

Figure 7: mem and expl values drop as the batch size increases.

B 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 doesn’t impact the downstream task performance substantially. Therefore we pretrain both BERT models (-base and -large, both uncased) on the MLM task only.

Wikipedia Data We extracted and preprocessed the April 21’ English Wikipedia dump. We used the wikieextractor 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 Section 3 All models were trained with the following standard procedure and hyperparameters. Specific experimental adjustments will be discussed later. We pre-trained BERT models using transformers (Wolf et al., 2020) run_mlm script for masked language modeling. 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 corporea (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.

Experimental Details for Section 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.