Synthetic Query Generation for Privacy-Preserving Deep Retrieval Systems using Differentially Private Language Models

Aldo Gael Carranza
Stanford University
aldogael@stanford.edu

Rezsa Farahani
Google Inc.
farahani@google.com

Natalia Ponomareva
Google Research
nponomareva@google.com

Alex Kurakin
Google DeepMind
kurakin@google.com

Matthew Jagielski
Google DeepMind
jagielski@google.com

Milad Nasr
Google DeepMind
srxzr@google.com

Abstract

We address the challenge of ensuring differential privacy (DP) guarantees in training deep retrieval systems. Training these systems often involves the use of contrastive-style losses, which are typically non-per-example decomposable, making them difficult to directly DP-train with since common techniques require per-example gradients. To address this issue, we propose an approach that prioritizes ensuring query privacy *prior* to training a deep retrieval system. Our method employs DP language models (LMs) to generate private synthetic queries representative of the original data. These synthetic queries can be used in downstream retrieval system training without compromising privacy. Our approach demonstrates a significant enhancement in retrieval quality compared to direct DP-training, all while maintaining query-level privacy guarantees. This work highlights the potential of harnessing LMs to overcome limitations in standard DP-training methods.

1 Introduction

Deep retrieval systems have been widely adopted in many online services, from search to advertising, to match user queries to relevant recommendations (Covington et al., 2016; Huang et al., 2020). In many applications, candidate items for retrieval are often publicly available non-personal information in the sense that they do not contain any specific information related to any single user (e.g., articles, products, movies, ads). However, the input queries to retrieval systems can often contain user personal information. Therefore, training deep retrieval systems on user data may enhance user experience through timely relevance, but it may also unintentionally compromise user privacy since neural network models have been demonstrated to implicitly memorize and leak sensitive user information in the training data (Carlini et al., 2019). This raises privacy sensitivities around each stage of data collection, training, inference, and hosting these systems. In this work, we seek to address the problem of ensuring user query privacy guarantees in deep retrieval systems without significantly hindering their utility.

The standard approach to ensure the privacy of training data in many large-scale machine learning models is to directly introduce *differential privacy* (DP) (Dwork et al., 2014) guarantees during training (Abadi et al., 2016; Ponomareva et al., 2023). These DP-training strategies provide guarantees by limiting the impact each individual data instance has on the overall model. However, some models contain design elements that inherently hinder the ability to limit per-example contributions, and thus are more difficult to directly DP-train. These include models with components that calculate batch statistics such as batch normalization layers (Ponomareva et al., 2023) and models with losses that *cannot* be decomposed into per-example losses such as pairwise and contrastive-style losses (Huai et al., 2020; Xue et al., 2021).

This limits the application of DP-training on deep retrieval systems since these systems typically use *non-per-example decomposable contrastive-style losses* to train semantic neural representations of user queries and candidate items in order to facilitate efficient vector-based retrieval strategies. The injected noise needed to achieve DP guarantees for these losses can scale with the number of candidate items that appear in the example-level loss computations, which can result in excessive retrieval quality degradation. Hence, additional considerations are often needed to adapt DP-training to deep retrieval models for achieving an adequate privacy-performance tradeoff.

In this work, we take an approach that ensures user query privacy *prior* to training a deep retrieval system in order to circumvent the various issues with directly DP-training deep retrieval systems with a non-per-example decomposable contrastive-

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 3920–3930
June 16-21, 2024 ©2024 Association for Computational Linguistics

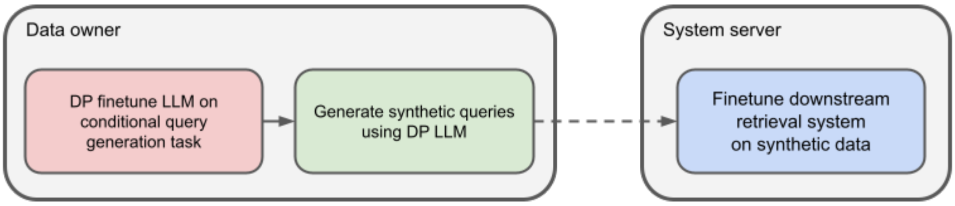
Data owner
DP finetune LLM on conditional query generation task
Generate synthetic queries using DP LLM
System server
Finetune downstream retrieval system on synthetic data

Figure 1: Illustration of approach.

style loss. We build on the framework of *synthetic data generation* using *DP language models* (LMs) (Yue et al., 2022; Mattern et al., 2022) to develop an approach for private query sharing to train any downstream deep retrieval system with *query-level privacy* guarantees with respect to the original training data. We empirically demonstrate considerable improvements in our approach on retrieval quality without compromising privacy guarantees compared to direct DP-training methods. More generally, our work presents an initial study into a nascent opportunity to leverage exciting breakthroughs in LMs to overcome crucial limitations in directly DP-training machine learning systems.

2 Related Work

Synthetic Generation using DP LMs Several recent studies (Yue et al., 2022; Mattern et al., 2022; Mireshghallah et al., 2022; Putta et al., 2022) have investigated the utility of private synthetic data from DP finetuned LMs in downstream tasks with pointwise losses, including text classification and semantic parsing. These works find that downstream models trained with private synthetic data outperform directly DP-trained models under the same privacy budget, and non-private synthetic data generation can even improve performance under no privacy constraints. The reason is that DP synthetic data benefits from the injection of additional public information from the pretrained LMs. Our work contributes another exploration of the advantages of private synthetic data generation for downstream training under a different learning paradigm with a non-per-example decomposable loss. In particular, our motivation is to achieve query privacy DP guarantees in deep retrieval systems with high utility.

DP-training under Non-per-example Decomposable Losses Figuring out better ways of DP-training models with non-per-example decomposable losses remains an active research topic. Research in this area has entirely focused on pairwise losses (Huai et al., 2020; Xue et al., 2021; Kang et al., 2021), introducing specialized algorithms under particular conditions like convexity, smoothness, and Lipshitz continuity to maintain a reasonable bound on sensitivity. Our work presents a general-purpose approach without such additional assumptions for achieving some level of privacy for a system trained with a non-per-example decomposable loss.

3 Background

3.1 Deep Retrieval

Deep retrieval systems have emerged as highly effective and scalable information retrieval systems to find candidate items based on their semantic relevance to a specific query (Huang et al., 2020; Ni et al., 2021). These systems typically consist of two neural encoders capable of generating rich, dense representations of queries and items (see Figure 2), which enable efficient *approximate nearest neighbor search* methods (Guo et al., 2020) to retrieve semantically relevant items to a given query. Deep retrieval systems are typically trained on *contrastive-style losses* that make use of two types of data examples: positive examples and negative examples. The positive examples help train the encoders into pulling relevant query-item pair embeddings close together in the embedding space, while negative examples help in preventing embedding space collapse. A popular choice for the loss function in deep retrieval is the *in-batch softmax loss*, which makes memory-efficient use of items already loaded in a mini-batch as randomly sampled soft negatives (Gillick et al., 2019; Karpukhin et al., 2020; Qu et al., 2020). In particular, given a training batch of query-item pairs $\{(q_i, d_i)\}_{i \in \mathcal{B}}$, each d_i is the positive item document for query q_i, and all other item documents $\{d_j\}_{j \neq i}$ within the batch are treated as the negatives. The in-batch softmax loss for each sample in the batch is

$$\mathcal{L}_i = -\log \frac{e^{\text{sim}(q_i,d_i)}}{\sum_{j\in\mathcal{B}} e^{\text{sim}(q_i,d_j)}}, \qquad (1)$$

where $\text{sim}(q_i, d_j)$ is the cosine similarity between the embeddings of q_i and d_j for any $i, j \in \mathcal{B}$. The larger and more diverse the batch, the better it is for representation learning.

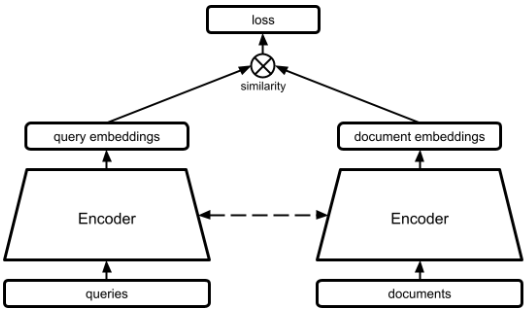

Figure 2: Illustration of deep retrieval dual encoder model. Dashed lines connecting the query and document encoders represent the possibility of sharing the same encoder.

3.1.1 Privacy Risks of Deep Retrieval

The neural encoders in deep retrieval systems are high-capacity models known to implicitly memorize sensitive information present in the training data (Carlini et al., 2019). Such sensitive data can subsequently be extracted from these trained models (Carlini et al., 2021; Lehman et al., 2021). Moreover, retrieval-augmented text generation systems, which utilize deep retrieval systems to aid text generation, have been demonstrated to be more susceptible to leaking private information from their private datastore compared to the language models trained on the private data (Huang et al., 2023; Zeng et al., 2024). This underscores the increase in privacy risks associated with deep retrieval systems.

3.2 Conditional Text Generation

Conditional text generation is the task of generating a sequence of text given a specific prompt (Keskar et al., 2019; Schick and Schütze, 2021). Pre-trained generative LMs such as GPT-3 and T5 have been shown to be highly effective at generating high-quality text conditioned on various prompt inputs (Raffel et al., 2020; Brown et al., 2020). Given a context c, the probability distribution of a text sequence $x = (x_1, \dots, x_n)$ is decomposed as $p(x|c) = \prod_{i=1}^{n} p(x_i|x_1, \dots, x_{i-1}, c)$. A neural network p_θ is trained to model the conditional distributions. The model can then be used to generate a new sample $\tilde{x} = (\tilde{x}_1, \dots, \tilde{x}_m)$ conditioned on a given context c by sequentially sampling $p_\theta(\cdot|c), p_\theta(\cdot|\tilde{x}_1, c), \dots, p_\theta(\cdot|\tilde{x}_1, \dots, \tilde{x}_{m-1}, c)$. In this work, we model the distribution of query texts given item documents as contexts with a publicly pre-trained LM.

3.3 Differential Privacy

Differential privacy (DP) has become a gold standard for ensuring data anonymization (Dwork et al., 2014). In this work, we make use of the following relaxed notion of differential privacy known as (ϵ, δ)-differential privacy.

Definition 3.1 (Differential Privacy)**.** A randomized algorithm $\mathcal{M} : \mathcal{D} \to \mathcal{S}$ is (ϵ, δ)-differentially private if for all $S \subset \mathcal{S}$ and for any two neighboring datasets $D, D' \in \mathcal{D}$ that differ exactly by a single data point, we have that $\mathbb{P}[\mathcal{M}(D) \in S] \leq e^{\epsilon}\, \mathbb{P}[\mathcal{M}(D') \in S] + \delta$.

Note that for capturing query-level differential privacy in retrieval datasets under this definition, neighboring datasets are datasets that differ by exactly one query. This definition captures a privacy guarantee based on the indistinguishability of the presence of a single data point in a dataset. The ϵ and δ parameter control the strength of this privacy guarantee, where smaller values correspond to stronger guarantees. A useful property of DP that is crucial to our approach is its *post-processing property* (Dwork et al., 2014) which states that for any deterministic or randomized function f defined over the range of the mechanism $\mathcal{M}$, if $\mathcal{M}$ satisfies (ϵ, δ)-DP, so does the composition $f \circ \mathcal{M}$. The post-processing property ensures that arbitrary computations on the output of a DP mechanism do not incur any additional privacy loss.

3.3.1 Differentially Private Training

In the context of machine learning, DP can be used to protect the privacy of data used to train a model, preventing an adversary from inferring the presence of specific training examples. By far the most practical method of introducing DP to non-convex ML models involves the modification of the training process to limit the impact that each individual data instance has on the overall model, also referred to as *DP-training* (Ponomareva et al., 2023). The most popular methods for DP-training are gradient noise injection methods like differentially private stochastic gradient descent (DP-

SGD) (Abadi et al., 2016). DP-SGD works by clipping per-example gradients to have norm no greater than C and adding isotropic Gaussian noise $\mathcal{N}(0, \sigma^2 C^2 \mathbf{I})$ to the clipped gradients before aggregating and applying the gradient update to model weights. The noise multiplier σ is set based on the privacy parameters ϵ, δ, and it can be determined using a privacy accountant (Abadi et al., 2016).

Clipping is done to bound gradient sensitivity, which captures how much a single example can influence the trained model. The specific value of C does not actually affect the (ϵ, δ)-DP guarantee, since a larger value of C means more noise will be added to compensate. However, the primary challenge in setting the clipping norm is in finding the right balance to maximize utility. If the clipping norm is set too low, it may overly constrain the gradients during training. If the clipping norm is set too high, sensitivity is less controlled, and too much noise is added to the gradients. Both cases hinder the model's ability to learn and worsen utility.

3.3.2 Limitations of Directly Differentially Private Training Retrieval Systems

Our work was primarily motivated by the fact that DP-SGD is *not* immediately compatible with the in-batch softmax loss used to train dual encoders. The primary reason is that per-example gradients of this loss depend on not just the example in consideration but also all other examples in the batch. Therefore, a single example can influence multiple per-example gradient computations which immediately implies an increased sensitivity of the gradient that scales with the batch size. In DP-training, higher sensitivity means that more noise needs to be added to the gradient updates during training to achieve the same level of privacy guarantees, which leads to worse utility.

Moreover, DP-SGD provides guarantees on example-level privacy, and every example in this case contains a query and item. However, in this work, we are interested in achieving query-level privacy which should be easier than protecting both queries and items. Standard DP-SGD is not able to guarantee these less strict levels of privacy.

Lastly, a systems-level issue of DP-training on the in-batch softmax loss is that in order to take advantage of vectorization and parallelization strategies for computing per-example gradients more quickly (Subramani et al., 2021), each query-item example in a batch must be duplicated to be contained in every de facto example in the batch, leading to a quadratic increase in memory requirements. Given fixed memory resources, this necessitates significantly smaller batch sizes, which has an additional deleterious effect beyond gradient clipping and noising since effective representation learning under the in-batch softmax loss highly depends on the amount and diversity of in-batch examples. Our approach of training with private synthetic queries precludes the above limitations when training a downstream dual encoder deep retrieval model.

4 Approach

We describe our general-purpose approach to obtain DP synthetic data for training a downstream deep retrieval system while ensuring query-level privacy on the original training data.

1) DP-training LM on Conditional Query Generation Task First, we obtain a suitable publicly pre-trained LM that has not been pre-trained on the queries in the private training data. We use DP-Adafactor to DP fine-tune the chosen LM with a conditional query generation task. DP-Adafactor is merely an Adafactor optimizer (Shazeer and Stern, 2018) that receives clipped and noised gradients as per the DP-SGD algorithm (Abadi et al., 2016). The conditional query generation task is the following: given a query-item document pair (q, d) in the training data, the LM is fine-tuned to generate the target text "q" given input text "d". Note that for larger LMs with billions of parameters, it is possible to leverage more parameter-efficient finetuning techniques (Lester et al., 2021; Hu et al., 2021) in order to overcome the high cost of training such large models. The effect of parameter-efficient DP-finetuning on the quality of synthetically generated retrieval data is a subject of further study.

2) Synthetic Query Generation using DP LM Then, the DP fine-tuned LM is capable of generating synthetic queries that are representative of the real queries and relevant to the items. For each item document d, we generate a matching synthetic query $\tilde{q}$ by providing the input "generate_query: d" to the model. This method allows for generating multiple synthetic queries from each document. A synthetic training dataset is then constructed to be the set of original documents matched with their corresponding synthetic queries.

3) Training Dual Encoder with DP Synthetic Data Lastly, the synthetic data can then be shared securely for any subsequent training tasks without taking on any additional DP losses on the original queries, as guaranteed by the post-processing property of DP (see Section 3.3). In particular, we can train a dual encoder model with the in-batch softmax loss (see Equation 1) on the synthetic training data using standard non-private training methods, while still guaranteeing DP protection of the original queries.

5 Experimental Setup

5.1 Datasets

We use publicly available datasets for information retrieval tasks. For finetuning and evaluation, we consider the MSMARCO dataset (Bajaj et al., 2016), which consists of nearly 533,000 query-document pairs of search data sampled from Bing search logs covering a broad range of domains and concepts. Additionally, we consider datasets in the BEIR benchmark suite (Thakur et al., 2021), which contains information retrieval datasets across a variety of domains, for zero-shot evaluation.

5.2 Synthetic Data Generation

5.2.1 Implementation Details

Model Training For synthetic data generation, we trained various T5 LMs (Raffel et al., 2020) with different sizes $\{$Small, Base, Large, XL$\}$ and privacy guarantees $\epsilon \in \{3, 8, 16, \infty\}$ to generate synthetic queries given corresponding input documents from the MSMARCO dataset. The T5 Small, Base, Large, XL models have around 60 million, 220 million, 770 million, 3 billion parameters, respectively. All experiments were performed on a TPU v4 chip.

Hyperparameters We trained each LM over 30 epochs with batch size 1024 and set the maximum token length to be 384 for input documents and 128 for target queries. We used the DP-Adam optimizer with a learning rate of 0.001 and clip norm of 0.1. Following (Li et al., 2021), we set the privacy parameter $\delta = 1/2n$ where n is the training dataset size. For sampling, we used a nucleus sampling strategy (Holtzman et al., 2019) with $p = 0.8$.

The hyperparameters above were chosen from a hyperparameter search to identify the optimal hyperparameters for the T5-Small model, DP-finetuned on the MSMARCO training dataset. The optimal criteria were the highest BLEU scores achieved on a validation dataset. We found that learning rate of 0.001, clipping norm 0.1, batch size 1024, and epochs 30 mostly resulted in the best model. We used these hyperparameters in all other T5 models. See Table 1 for the hyperparameter grid.

Table 1: Hyperparameter grid.

Hyperparameter	Values
Token Lengths	Input: 384, Target: 128
Learning Rate	$0.001 \cdot 2^{-k}$ for $k \in \{0, 1, 2, 3\}$
Clipping Norm	$\{0.1, 0.25, 0.5, 1\}$
Batch Size	$\{128, 256, 512, 1024\}$
Epochs	$\{10, 20, 30\}$

5.2.2 Data Synthesis

We used each DP-finetuned T5 LM to generate synthetic queries given documents from the original training data. These pairs of synthetic queries and original documents constitute a new synthetic dataset. For qualitative comparison, we provide an example in Table 2 of an original query-document pairs and the synthetic queries generated under various model configurations and privacy levels.

5.2.3 Pretraining and Training Data Overlap

We note the importance that the pretrained LMs used to generate the synthetic data were not markedly trained on the original query data we seek to make private. Otherwise, the privacy guarantees would be undermined since the models would have already seen the data. To address this matter in our experiments, we conducted an analysis to determine the extent of overlap of the MSMARCO dataset on the pre-training data of T5 models, the C4 common crawl dataset (Raffel et al., 2020). We conducted multiple runs of selecting random subsets of 10,000 query and text pairs to determine if there was an exact match in the C4 dataset.

Our analysis determined that while a significant percentage of MSMARCO documents (~22%) were exactly found in C4 on average, a negligible percentage (<1.9%) of MSMARCO queries were exactly found in C4 on average. Moreover, the queries that were found tended to be generic search terms which could be considered public knowledge. Since we are interested in query-level privacy, we consider this level of dataset overlap acceptable to give reasonable guarantees of privacy. In Section 6.3, we provide a more extensive study of the empirical privacy guarantees of our training procedure.

Table 2: Synthetic query example.

Source		Text
Document		The main cast of the show: Mickey Mouse, Minnie Mouse, Donald Duck, Daisy Duck, Goofy, and Pluto star in the series, which focuses on interacting with the viewer to stimulate problem solving.
Original Query		characters from the Mickey Mouse clubhouse show
T5-Base	$\epsilon = \infty$	the Mickey Mouse show cast
	$\epsilon = 16$	what is the most important characters in the series
	$\epsilon = 8$	what is in this series?
	$\epsilon = 3$	what is in this code for dfr1
T5-XL	$\epsilon = 3$	issue with show by mickey mouse
T5-Large	$\epsilon = 3$	what is the most animated characters on disney cartoon show
T5-Small	$\epsilon = 3$	what is isn't a character will do a story

5.3 Downstream Retrieval System

5.3.1 Implementation Details

Model Training For each data source (i.e., original MSMARCO data and synthetic datasets for various ϵ and model sizes), we train a separate dual encoder model on the in-batch softmax loss. We utilize a separate pre-trained T5-Base encoder for both the query and document encoders, sharing parameters between them. Similar to data synthesis, we use this kind of encoder to ensure that it is not significantly pretrained on the original queries. We emphasize that the encoders of the retrieval models are distinct from the T5 models used to generate synthetic data.

Hyperparameters For the hyperparameters in dual encoder model training, we used learning rate 0.001, batch size 32, epochs 5, the maximum token length 384 for documents and 128 for queries. For the directly DP finetuning experiments, we used a clipping norm of 0.1.

5.3.2 Baseline Approach

A baseline comparison of our approach will be to compare against a deep retrieval system that is directly DP-trained on the original data. For direct DP-training, we used the same hyperparameters as above, but given the memory constraints discussed in Section 3.3.2, the batch size for DP-training a dual encoder model had to be significantly decreased to 32. We do not experiment with different downstream deep retrieval models since our intent is to compare general-purpose methods for achieving DP guarantees in retrieval systems.

6 Evaluation

6.1 Evaluation on Retrieval Tasks

We evaluate the retrieval models on the MS-MARCO test data set and various other BEIR retrieval data sets for zero-shot evaluation. We evaluate on the normalized discounted cumulative gain score over the top 10 predictions (NDCG@10) which measures the relevance and ranking quality of items in a recommendation list, considering both quality and position. We also report the recall score over the top 10 predictions (Recall@10), which measures the percentage of times the ground truth recommendation appears in the top 10 predictions. We report the evaluation results of a single training run.

6.1.1 Search & Retrieval Procedure

Evaluation of the dual encoder retrieval models requires a query-document nearest neighbor search implementation for the inference stage. For our experiments, we used the Scalable Nearest Neighbors (ScaNN) library, an open-source library that provides a fast and scalable approximate nearest neighbor search procedure (Guo et al., 2020). The procedure was executed using ScaNN's brute force scoring and the inner product distance settings.

6.1.2 MSMARCO Evaluation

Table 3 shows the evaluation results on the MS-MARCO test set for deep retrieval models regularly trained on the synthetically generated data under varying generative model configurations and varying privacy levels. For our benchmark comparison, in Table 4 we display the evaluation results on for the deep retrieval models directly DP-trained on the original data under varying privacy levels. In the top rows of both tables, we also provide another baseline reference evaluation of a dual encoder model regularly finetuned on the original data without any DP guarantees (i.e., $\epsilon = \infty$).

We observe that the retrieval model trained on synthetic data with DP significantly outperform retrieval trained with DP on original data. As discussed in Section 3.3.2, there are a number of challenges associated with training DP models with contrastive-style losses. Our naive approach of implementing DP-training on contrastive loss likely explains poor utility of DP-training on original data. Additionally, our DP synthetic data essentially introduces additional public knowledge into the process, since we utilize a publicly pretrained LM.

Moreover, we found that the retrieval model trained with non-DP synthetic data outperformed the retrieval model trained on the original data. This suggests that synthetic data generation indeed augments the original data and to some extent improves generalization, whether it be through introducing additional public information or data cleaning. In fact, data augmentation via synthetic data generation using language models for deep retrieval is an area of research that has gained significant interest in recent years (Dai et al., 2022; Bonifacio et al., 2022). We also observe that performance increases with increasing model size. This is consistent with similar prior results that demonstrate DP-SGD on over-parameterized models can perform significantly better than previously thought (De et al., 2022; Li et al., 2021). Overall, we show that training with synthetic data from DP LMs is viable for achieving DP guarantees and efficiency in retrieval models.

Table 3: Evaluation of retrieval models. Top: Trained on DP synthetic data with varying ϵ and fixed model size T5-Base. Bottom: Trained on DP synthetic data with varying model size and fixed $\epsilon = 3$.

Source	ϵ	NDCG@10	Recall@10
Original	∞	0.2525	0.4098
T5-Base	∞	0.2590	0.4192
T5-Base	16	0.2027	0.3342
T5-Base	8	0.1912	0.3196
T5-Base	3	0.1830	0.3108
T5-XL	3	0.1854	0.3098
T5-Large	3	0.1833	0.3094
T5-Base	3	0.1830	0.3108
T5-Small	3	0.1346	0.2272

Table 4: Evaluation of retrieval models DP-trained directly on original data with varying ϵ.

Source	ϵ	NDCG@10	Recall@10
Original	∞	0.2525	0.4098
Original	16	0.0523	0.0862
Original	8	0.0386	0.0649
Original	3	0.0234	0.0388

6.1.3 Zero-shot Evaluation

We also evaluate the zero-shot generalization capabilities of a retrieval model trained on synthetic data. We compare against a retrieval model trained on the original data with no DP (i.e., $\epsilon = \infty$) and with $\epsilon = 16$. See Table 5 for the results. Again, our results demonstrate significant advantage of DP synthetic data compared to DP-training on original data, nearly matching and in some cases outperforming the non-DP results. This suggests that the benefits of synthetic data generation can outweigh the utility degradation of DP-training with reasonable levels of privacy, at least in zero-shot generalization tasks.

6.2 Similarity between Synthetic and Original Datasets

We compute measures of similarity between the synthetic data generated by the DP-trained T5 models against the original data. Since the synthetic data is one-to-one generated from the original data, we can compute BLEU scores to evaluate similarity (Post, 2018). We also compute the MAUVE scores, shown to be more capable of comparing similarity of text distributions (Pillutla et al., 2021). See Table 6 for the scores. We observe that the non-DP finetune model generates synthetic data that is as similar as one could expect under these metrics, and there is a significant drop with finite ϵ, with increasing similarity with higher ϵ and increasing model size. By comparing the similarity scores with the retrieval evaluation results, we observe that while larger models lead to drastic improvements in the synthetic data similarity, the downstream retrieval performance sees comparatively more modest gains with increasing model size.

6.3 Empirical Privacy

The provable privacy provided by DP decays significantly as ϵ grows, but prior work has shown that even these large values can provide strong protection against state of the art privacy attacks (Carlini et al., 2019, 2022; Ponomareva et al., 2023). To verify our training technique still follows this tendency, we evaluate here the empirical privacy leakage of DP-trained language models, using the *canary exposure* metric introduced in (Carlini et al., 2019). This technique is frequently used to evaluate empirical privacy (Zanella-Béguelin et al., 2020; Ramaswamy et al., 2020; Jagielski et al., 2022). To perform this test, we construct examples with private information, referred to as canaries, and

Table 5: Zero-shot evaluation of retrieval models trained on DP synthetic data vs. directly DP-trained retrieval models with $\epsilon = 16$. Top table: NDCG@10. Bottom table: Recall@10.

Source	ϵ	NDCG@10										
		arguana	cqadup	dbpedia	fiqa	hotpot	nfcorpus	quora	scidocs	scifact	covid	touche
Original	∞	0.2653	0.2659	0.3905	0.2257	0.5232	0.2974	0.8126	0.1857	0.4527	0.4971	0.2764
Original	16	0.2132	0.0990	0.1272	0.0870	0.1422	0.1331	0.6856	0.0792	0.2051	0.3133	0.1185
T5-Base	16	0.2757	0.2474	0.3728	0.2140	0.5122	0.2971	0.7850	0.1750	0.4645	0.4351	0.2547

Source	ϵ	Recall@10										
		arguana	cqadup	dbpedia	fiqa	hotpot	nfcorpus	quora	scidocs	scifact	covid	touche
Original	∞	0.5569	0.3368	0.1479	0.2440	0.3706	0.1013	0.8989	0.1071	0.5801	0.0116	0.0530
Original	16	0.4388	0.1325	0.0313	0.0906	0.1108	0.0365	0.7762	0.0503	0.2969	0.0063	0.0149
T5-Base	16	0.5768	0.3165	0.1217	0.2261	0.3398	0.1110	0.8763	0.1066	0.5848	0.0101	0.0489

Table 6: Similarity scores of generated synthetic data. Top: Varying ϵ and fixed model size. Bottom: Varying model size with fixed $\epsilon = 3$.

Model	ϵ	BLEU	MAUVE
T5-Base	∞	0.2939	0.9763
T5-Base	16	0.0984	0.3715
T5-Base	8	0.0940	0.3431
T5-Base	3	0.0856	0.2974
T5-XL	3	0.1021	0.7117
T5-Large	3	0.1096	0.6359
T5-Base	3	0.0940	0.2974
T5-Small	3	0.0436	0.2296

introduce a subset of them into the original training data, and measure how likely the model is to output the injected canaries. In general, canary generation is a domain-dependent decision, so we design canaries for our retrieval application using the three following types of query-document pairs: (random query, random 10-digit string), (random query, corresponding document + random 10-digit string), (random query, random document + random 10-digit string). The secret part of each canary is the random 10-digit string.

We train the language model on this modified dataset using different DP guarantees, generate synthetic datasets, and assess canary exposure. We conduct this experiment multiple times with different canaries and DP guarantees, averaging the metrics and reporting the results in Table 7. As anticipated, training without DP leads to significant leakage. Canaries repeated 10 times are frequently extractable, and those repeated 100 times are always extractable. However, our approach with a large ϵ of 16 prevents the model from leaking secrets and significantly increases the rank. Recent techniques for converting attack success rates to lower bounds on the ϵ parameter (Stock et al., 2022) allow us to interpret these ranks as a lower bound of roughly 0.015 on ϵ. This large gap is consistent with prior findings on the empirical privacy of DP-SGD on language models (Carlini et al., 2019; Ponomareva et al., 2023).

Table 7: Privacy leakage.

Model	ϵ	Repetition = 10		Repetition = 100	
		Rank	Leaked	Rank	Leaked
T5-Base	∞	1/100	67%	1/100	100%
T5-Base	16	43/100	0%	32/100	0%

7 Conclusion

Our work focused on ensuring DP guarantees in training deep retrieval systems. We discussed the limitations of DP-training such systems with the often used in-batch softmax loss function, which is non-per-example decomposable. We introduce an approach of using DP LMs to generate private synthetic queries for downstream deep retrieval training. This approach ensures theoretical guarantees on query-level privacy prior to downstream training, thereby bypassing some of the limitations of DP-training. Furthermore, we empirically demonstrate that our approach improves retrieval quality compared to direct DP-training, without compromising query privacy. Our work highlights the potential of LMs to overcome crucial limitations in DP-training ML systems.

Limitations There are a few limitations to our approach. Firstly, while we observed that larger LMs generate higher quality synthetic queries, it is worth noting that training such large models may too computationally expensive. Exploring more parameter-efficient finetuning methods tailored to DP-training could mitigate the computational burden associated with training such larger models. Secondly, it is necessary for the publicly pretrained LM utilized for generating synthetic queries to not have been pretrained on the original queries we aim to privatize. This imposes a constraint on the choice of pretrained LMs suitable for generating private synthetic queries. Next, as observed in the synthetic example in Table 2, the DP-finetuned LMs can sometimes generate incoherent queries, which can limit the relevance and interpretability of the data. Lastly, it is essential to recognize that our approach exclusively ensures query-level privacy. For achieving more general example-level privacy, it may be necessary to resort to other approaches or more conventional DP-training methods.

Risks & Ethical Considerations Data privacy is a crucial consideration in the responsible development of personalized machine learning systems. Our work directly offers potential solutions to address issues of data privacy in deep retrieval systems. However, it is important to acknowledge the above limitations on privacy guarantees of our approach to prevent undesired privacy risks.

Acknowledgements

We thank Rishabh Bansal, Manoj Reddy, Andreas Terzis, Sergei Vassilvitskii, Abhradeep Guha Thakurta, Shuang Song, Arthur Asuncion, and Heather Yoon for the helpful discussions. We also thank Jianmo Ni for their assistance in setting up the retrieval training pipeline. Finally, we appreciate the support and encouragement of YouTube Ads leadership Shobha Diwakar, Marija Mikic, and Ashish Gupta throughout this work.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *arXiv preprint arXiv:2202.05144*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, volume 267.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650.

Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, New York, NY, USA.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. 2022. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*.

Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. *arXiv preprint arXiv:1909.10506*.

Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. 2020. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, pages 3887–3896. PMLR.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Mengdi Huai, Di Wang, Chenglin Miao, Jinhui Xu, and Aidong Zhang. 2020. Pairwise learning with differential privacy guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 694–701.

Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2553–2561.

Yangsibo Huang, Samyak Gupta, Zexuan Zhong, Kai Li, and Danqi Chen. 2023. Privacy implications of retrieval-based language models. *arXiv preprint arXiv:2305.14888*.

Matthew Jagielski, Om Thakkar, Florian Tramer, Daphne Ippolito, Katherine Lee, Nicholas Carlini, Eric Wallace, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, et al. 2022. Measuring forgetting of memorized training examples. *arXiv preprint arXiv:2207.00099*.

Yilin Kang, Yong Liu, Jian Li, and Weiping Wang. 2021. Towards sharper utility bounds for differentially private pairwise learning. *arXiv preprint arXiv:2105.03033*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron C Wallace. 2021. Does bert pretrained on clinical notes reveal sensitive data? *arXiv preprint arXiv:2104.07762*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. 2021. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*.

Justus Mattern, Zhijing Jin, Benjamin Weggenmann, Bernhard Schoelkopf, and Mrinmaya Sachan. 2022. Differentially private language models for secure data sharing. *arXiv preprint arXiv:2210.13918*.

Fatemehsadat Mireshghallah, Richard Shin, Yu Su, Tatsunori Hashimoto, and Jason Eisner. 2022. Privacy-preserving domain adaptation of semantic parsers. *arXiv preprint arXiv:2212.10520*.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Natalia Ponomareva, Hussein Hazimeh, Alex Kurakin, Zheng Xu, Carson Denison, H Brendan McMahan, Sergei Vassilvitskii, Steve Chien, and Abhradeep Thakurta. 2023. How to dp-fy ml: A practical guide to machine learning with differential privacy. *arXiv preprint arXiv:2303.00654*.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Pranav Putta, Ander Steele, and Joseph W Ferrara. 2022. Differentially private conditional text generation for synthetic data production.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191*.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Swaroop Ramaswamy, Om Thakkar, Rajiv Mathews, Galen Andrew, H Brendan McMahan, and Françoise Beaufays. 2020. Training production language models without memorizing user data. *arXiv preprint arXiv:2009.10031*.

Timo Schick and Hinrich Schütze. 2021. Few-shot text generation with natural language instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Pierre Stock, Igor Shilov, Ilya Mironov, and Alexandre Sablayrolles. 2022. Defending against reconstruction attacks with rényi differential privacy. *arXiv preprint arXiv:2202.07623*.

Pranav Subramani, Nicholas Vadivelu, and Gautam Kamath. 2021. Enabling fast differentially private sgd via just-in-time compilation and vectorization. *Advances in Neural Information Processing Systems*, 34:26409–26421.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Zhiyu Xue, Shaoyang Yang, Mengdi Huai, and Di Wang. 2021. Differentially private pairwise learning revisited. In *IJCAI*, pages 3242–3248.

Xiang Yue, Huseyin A Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Huan Sun, David Levitan, and Robert Sim. 2022. Synthetic text generation with differential privacy: A simple and practical recipe. *arXiv preprint arXiv:2210.14348*.

Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. 2020. Analyzing information leakage of updates to natural language models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 363–375.

Shenglai Zeng, Jiankun Zhang, Pengfei He, Yue Xing, Yiding Liu, Han Xu, Jie Ren, Shuaiqiang Wang, Dawei Yin, Yi Chang, et al. 2024. The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag). *arXiv preprint arXiv:2402.16893*.