Privacy-Preserving Recommender Systems with Synthetic Query Generation using Differentially Private Large Language Models

ALDO GAEL CARRANZA, Stanford University, USA
REZSA FARAHANI, Google Inc., USA
NATALIA PONOMAREVA, Google Research, USA
ALEX KURAKIN, Google Research, USA
MATTHEW JAGIELSKI, Google Research, USA
MILAD NASR, Google Research, USA

We propose a novel approach for developing privacy-preserving large-scale recommender systems using differentially private (DP) large language models (LLMs) which overcomes certain challenges and limitations in DP training these complex systems. Our method is particularly well suited for the emerging area of LLM-based recommender systems, but can be readily employed for any recommender systems that process representations of natural language inputs. Our approach involves using DP training methods to fine-tune a publicly pre-trained LLM on a query generation task. The resulting model can generate private synthetic queries representative of the original queries which can be freely shared for any downstream non-private recommendation training procedures without incurring any additional privacy cost. We evaluate our method on its ability to securely train effective deep retrieval models, and we observe significant improvements in their retrieval quality without compromising query-level privacy guarantees compared to methods where the retrieval models are directly DP trained.

CCS Concepts: • **Information systems → Retrieval models and ranking**; **Query representation**; • **Computing methodologies → Natural language generation**; • **Security and privacy → Privacy-preserving protocols**.

Additional Key Words and Phrases: large language models, differential privacy, synthetic data generation, recommender systems

1 INTRODUCTION

Large-scale multi-stage recommender systems are integral to online platforms, such as search, shopping, content streaming, social media, and online advertising [12, 18, 53]. The ubiquity of these systems has raised privacy concerns regarding the collection of user data for personalization [6, 24, 30, 45]. Moreover, the emerging introduction of pre-trained transformer models in recommender systems, particularly for improving generalization in retrieval [16, 37, 52], presents additional privacy risks, as these high-capacity models can implicitly memorize sensitive user information in the training data [8, 10]. Although these high-quality recommender systems enhance user experience through timely relevance, it is important for this to not come at the cost of user privacy. In many recommendation applications, input queries contain user information, while candidate recommendations (e.g., articles, products, movies, ads) are public information. With this in mind, our goal in this paper is to develop a method for training query-level privacy-preserving recommender systems.

The standard approach to privacy-preserving recommender systems would be to train the models directly using differentially private (DP) training methods [40]. However, there are various issues with direct DP training recommender systems. Firstly, DP training methods typically ensure example-level privacy, which is stricter than our target level of privacy since an example in recommendation datasets corresponds to a query and a number of candidate recommendations. Therefore, standard DP training privacy guarantees may result in greater performance degradation than necessary for preserving query privacy. Further difficulties arise in DP training multi-stage recommender systems, particularly in deep retrieval models that often employ contrastive learning strategies, where representation models are trained to bring relevant query-recommendation pairs closer together and push irrelevant pairs further apart in the embedding space. As we discuss later, many DP training methods are not immediately compatible with the non-per-example decomposable losses of contrastive learning, thus requiring additional considerations to adapt standard DP training to contrastive-style recommender systems.

Rather than dealing with the various issues of directly DP training recommender systems, we take a simpler approach that ensures query privacy prior to training the system. In this work, we build on the framework of synthetic data generation using DP large language models (LLMs) [34, 61] to develop an approach for private text data sharing for training any downstream recommender system with query-level privacy. Our approach involves using DP training methods to fine-tune (or second stage pre-train) a publicly pre-trained generative LLM on a document-conditional query generation task given a private recommendation data set consisting of matched query-document pairs.[1] For DP fine-tuning, we employ differentially private stochastic gradient descent (DP-SGD), which works by clipping per example gradients and injecting calibrated noise to aggregated gradients during backpropagation [1]. The resulting DP-finetuned LLM is used to generate private synthetic queries conditional on documents in the training data. By leveraging the post-processing property of DP [15], one can securely share these private synthetic queries for any downstream non-private training procedures for recommender systems without incurring any additional privacy cost to the original queries. We use this approach to empirically demonstrate considerable improvements in deep retrieval tasks without compromising privacy guarantees compared to direct DP training methods.

Our contributions are as follows:

(1) We propose a novel approach to training query-level privacy-preserving recommender systems. Our approach involves generating synthetic query data with DP-finetuned LLMs and using this private synthetic data to train downstream recommender systems without any modification to standard training processes.
(2) In general, our approach presents a novel way of obtaining DP guarantees for models with non-per-example decomposable losses like the contrastive losses ubiquitous in large-scale multi-stage recommender systems.
(3) Through extensive experiments, we empirically confirm our approach generates synthetic data that is indeed query-level privacy-preserving, and we demonstrate that downstream deep retrieval models trained with this private synthetic data achieve excellent performance compared to standard DP training methods.

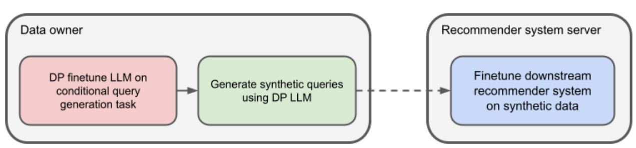

Fig. 1. Illustration of methodology.

2 RELATED WORK

Privacy in Recommender Systems. Previous works on DP for recommender systems have mostly involved matrix factorization classes of models [3, 11, 20, 31, 50, 63]. These methods are challenging to scale, susceptible to *cold start* [47],

[1]Throughout this work, we use the terms *query* and *document* from the traditional information retrieval terminology, where a document is the retrieved item, answer, or response text to an input query.

and generally suffer from higher performance degradation under stricter DP privacy parameters. Training recommender systems in a federated fashion [32, 35, 38, 55, 59] is another approach towards privacy preservation. However, this learning method concerns systems data ownership and processing versus obtaining mathematical privacy guarantees.

Synthetic Generation using DP LLMs. Generating private synthetic tabular data has been extensively explored [40], but private synthetic text data generation is more nascent. DP finetuning LLMs [33, 60] is the prevalent method for generating DP synthetic text, and recent studies have investigated the utility of private synthetic data from DP finetuned LLMs in downstream tasks [34, 36, 42, 61]. These works find that downstream models trained with private synthetic data significantly outperform directly DP trained models under the same privacy budget. Interestingly, there are varied results when comparing to non-DP models. For instance, Yue et al. [61] and Putta et al. [42] report a considerable drop in utility of text classification models trained with synthetic data, even with non-DP finetuned LLM data, suggesting low fidelity of synthetic text. However, Mattern et al. [34] and Mireshghallah et al. [36] report instances where non-DP synthetic data generation improves performance in text classification and semantic parsing tasks. These findings indicate that synthetic data generation can improve performance on downstream tasks, even without DP, and our work explores another potential avenue for its advantages in a distinct learning paradigm. Lastly, to our knowledge, no previous works have investigated synthetic data generation for ensuring privacy only with respect to a portion of the data, such as queries in our case.

3 BACKGROUND

3.1 Deep Retrieval

Deep retrieval systems, also known as dense retrieval systems, have emerged as effective components for retrieval in modern recommender systems [22, 29, 37, 57]. These systems often consist of two deep neural encoders capable of generating rich, dense representations of queries and documents, which enable approximate nearest neighbor search [26] to efficiently retrieve relevant documents for a given query that align with the semantic meaning of the query.

Deep retrieval systems are typically trained on contrastive losses that use two types of examples: positive examples and negative examples. The positive examples pull relevant query-document pair embeddings close together in the embedding space, while negative examples push embeddings of unrelated pairs further apart. Hard negative examples can be challenging to obtain since they require additional mining from the large set of candidate documents [54, 58]. Therefore, a popular choice for the contrastive loss in deep retrieval is the in-batch softmax loss, which makes use of in-batch documents as soft negatives [17, 27, 43]. In particular, given a training batch of query-document pairs $\{(q_i, d_i)\}_{i\in\mathcal{B}}$, each d_i is the positive document for query q_i, and all other 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

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

where $\mathrm{sim}(q_i, d_j)$ 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.

3.2 Conditional Text Generation

Conditional text generation is the task of generating a sequence of text given a specific context or condition [28, 48]. Pre-trained generative LLMs such as GPT-3 and T5 have been shown to be highly effective at generating high-quality

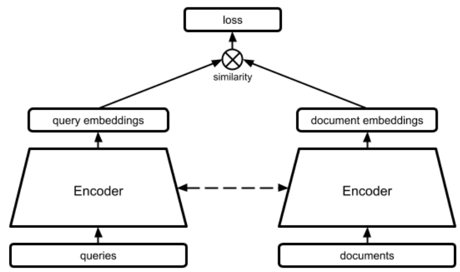

Fig. 2. Illustration of deep retrieval dual encoder model. The dashed lines connecting the encoders represent a shared encoder.

text conditioned on various prompt inputs [5, 44]. Given a context c, the probability distribution of a text sequence $x = (x_1, \ldots, x_n)$ is decomposed as $p(x|c) = \prod_{i=1}^{N} p(x_i|x_1, \ldots, x_{i-1}, c)$. A neural network p_θ is trained to model the conditional distributions. The model can then be used to generate a new sample $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_m)$ conditioned a given context c by sequentially sampling $p_\theta(\cdot|c), p_\theta(\cdot|\tilde{x}_1, c), \ldots, p_\theta(\cdot|\tilde{x}_1, \ldots, \tilde{x}_{m-1}, c)$. In this paper, we model the distribution of query texts given document texts with an encoder-decoder T5 LLM.

3.3 Differential Privacy

Differential privacy (DP) is a gold standard for ensuring data anonymization [15]. We make use of a 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:

$$\mathbb{P}[\mathcal{M}(D) \in S] \leq e^{\epsilon}\, \mathbb{P}[\mathcal{M}(D') \in S] + \delta.$$

This definition provides a privacy guarantee 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 property of DP that is crucial to our approach is the post-processing property [15] 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.

3.4 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, often referred to as DP-Training, with gradient noise injection methods like DP-SGD being the most popular [1, 40]. The DP-SGD method works by clipping per-example gradients to limit sensitivity and adding random noise to the aggregated clipped gradients before applying the gradient update to model weights. The amount of noise added is calibrated to satisfy a given level of (ϵ, δ)-DP.

We note that although DP-SGD can be used seamlessly on losses defined per training data example (e.g., cross-entropy loss), it is not as compatible on contrastive losses that use multiple training instances to compute the loss value (e.g., in-batch softmax loss) [40]. The reason is that per-example gradients also depend on multiple examples, and so the sensitivities can scale with the number of examples used in the computation. Therefore, ensuring privacy requires scaling the sensitivity in gradient clipping by the number of examples used to compute the contrastive loss, which can greatly amplify the noise. Figuring out better ways of DP-Training models with non-per-example decomposable losses remains an active research topic, with various works introducing specialized algorithms under particular conditions like convexity, smoothness, and Lipshitz continuity [21, 56] to maintain a reasonable bound on sensitivity. We emphasize that while it is clear that direct application of DP-training to contrastive losses is crude since it introduces too much noise, obtaining better algorithms is a hard task on its own.

4 APPROACH

We describe our general approach to obtain DP synthetic data for training a downstream recommender system while ensuring query-level privacy. We are specifically interested in training a dual encoder retrieval system in this way. We will also discuss an alternative approach of directly DP training a dual encoder on the original data for comparison.

4.1 Training using Synthetic Data obtained from a DP LLM

DP-Training LLM on Conditional Query Generation Task. The dataset consists of query-document pairs. We obtain a suitable publicly pre-trained LLM that has not been pre-trained on the queries in the training data. We use the encoder-decoder family of T5 language models which are capable of creating responses to prompts by conditioning on the input text and fine-tuning for the specific generative task [44]. We use DP-Adafactor[2] to fine-tune the LLM with the following conditional query generation task. Given a query-document pair (q, d) in the training data, the T5 model is trained using teacher forcing with the input "generate_query: d" and target "q".

Synthetic Query Generation using DP LLM. Then, the DP finetuned LLM is capable of generating synthetic queries that are representative of the real queries and relevant to the documents and provide protection against privacy leakage for the original queries. For each document d, we generate a matching synthetic query $\tilde{q}$ by providing the input "generate_query: d" to the model. For sampling, we employ the nucleus sampling strategy [19]. A synthetic training dataset is then constructed to be the set of original documents matched with their corresponding synthetic queries.

Training Dual Encoder with DP Synthetic Data. The synthetic data can then be shared securely for any downstream training procedures without incurring any additional DP losses on the original queries, as guaranteed by the post-processing property of DP [15]. In particular, we can train a dual encoder model with the in-batch softmax loss (in Equation 1) on the synthetic training data using standard SGD methods.

4.2 DP-Training using Original Data

For a comparison baseline, we will also directly DP finetune a deep retrieval system on the original data. However, as discussed in Section 3.4, there are additional considerations to make DP-SGD compatible with contrastive losses like the in-batch softmax loss. DP-SGD requires the sensitivities in per-example gradient clipping to scale by the batch size. This means that the most common way of decreasing the amount of per-example noise to improve the utility of DP trained models by increasing the batch size [40] will not work. Moreover, DP-SGD will guarantee example-level privacy, which in this case an example contains a query and every document in the batch. However, we are interested in achieving query-level privacy which should be easier than protecting both queries and documents. Another issue related to DP training implementation is that in order to take advantage of vectorization and parallelization strategies of computing

[2]DP-Adafactor is merely an Adafactor [49] optimizer that receives clipped and noised gradients as per DP-SGD algorithm.

per-example gradients, each query-document example in a batch must be duplicated to be contained in every 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 on learning beyond gradient clipping and noising since the in-batch softmax loss quality highly depends on the amount and diversity of in-batch examples.

5 EVALUATION

5.1 Experimental Setup

5.1.1 Datasets. We use publicly available datasets for information retrieval tasks. For finetuning and evaluation, we consider the MSMARCO dataset [2], which consists of 532,000 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 [52], which contains information retrieval datasets across a variety of domains, for zero-shot evaluation.

5.1.2 Data synthesis. We trained various T5 models with different sizes {Small, Base, Large, XL} and privacy guarantees $\epsilon \in \{3, 8, 16, \infty\}$ to generate synthetic queries given input document. We consider the MSMARCO dataset as the query-document training data. Then, we use each trained model to generate synthetic queries for the documents from the original training data. These pairs of synthetic query and original document constitute a new synthetic dataset. Refer to Appendix C to see how we ensure the LLMs were not pretrained on the queries in the training data. We trained each model 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 [33], we set the privacy parameter $\delta = 1/2n$ where n is the training dataset size. For sampling, we used nucleus sampling with $p = 0.8$. Given time and and computation constraints, our results were obtained through a small-scale hyperparameter search (specified in Appendix B) using the T5-Small model.

5.1.3 Downstream retrieval task. For each data source (original MSMARCO data and synthetic datasets for various ϵ and model sizes), we train a deep retrieval dual encoder model on the in-batch softmax loss (as described in section 3.1). We use standard SGD training methods for synthetic datasets and the original dataset without DP guarantees, while employing DP-SGD methods for the original dataset with DP guarantees. We utilize a pre-trained T5-Base encoder for both query and document encoders, sharing parameters between them. We trained each dual encoder model over 5 epochs, and we used the same token length and hyperparameters as above. For direct DP training we used the same privacy parameters as above, and given the memory constraints discussed in Section 4.2, the batch size for DP training a dual encoder model had to be significantly decreased to 32. It is important to note that the encoders of the retrieval model are distinct from the T5 models used to generate synthetic data. We chose T5 to evaluate LLM-based recommender systems, which are becoming increasingly adopted [16]. We do not experiment with different deep retrieval models since our goal is to strictly evaluate the performance of the synthesized dataset.

5.2 Evaluation on Retrieval Tasks

We evaluate the retrieval models on the MSMARCO 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). In the Appendix A, we also report additional evaluation results of recall scores.

5.2.1 MSMARCO Evaluation. Table 1 shows evaluation on MSMARCO test set of deep retrieval models trained on original and synthetic data. In each table, we provide a reference evaluation of a dual encoder model trained on the original data without any DP. 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 4.2, there are a number of challenges associated with training DP models with contrastive losses. This 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 LLM. Moreover, we found that the retrieval model trained with non-DP synthetic data outperformed a 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 imbibing it with additional public or data cleaning. In fact, data augmentation via synthetic data generation using large language models for deep retrieval is an area of research that has gained significant interest in recent years [4, 13, 25]. 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 [14, 33]. Overall, we show that synthetic data from LLMs is a viable approach for training retrieval models when privacy guarantees are needed.

Table 1. Evaluation of retrieval models. Left: Trained on DP synthetic data with varying ϵ and fixed model size T5-Base. Middle: Trained on DP synthetic data with varying model size and fixed $\epsilon = 3$. Right: DP-trained directly on original data with varying ϵ.

Source	ϵ	NDCG@10
Original	∞	0.2525
T5-Base	∞	0.2590
T5-Base	16	0.2027
T5-Base	8	0.1912
T5-Base	3	0.1830

Source	ϵ	NDCG@10
Original	∞	0.2525
T5-XL	3	0.1854
T5-Large	3	0.1833
T5-Base	3	0.1830
T5-Small	3	0.1346

Source	ϵ	NDCG@10
Original	∞	0.2525
Original	16	0.0523
Original	8	0.0386
Original	3	0.0234

5.2.2 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 and with $\epsilon = 16$. See Table 2 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. Further studies are needed to understand this observation.

Table 2. Zero-shot evaluation of retrieval models trained on DP synthetic data vs. directly DP-trained retrieval models with $\epsilon = 16$.

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

5.3 Similarity between Synthetic and Original Datasets

In this section we describe our analysis of fidelity of the synthetic data. We compute measures of similarity between the synthetic data generated by the DP-trained T5 models against the original data. In Appendix D, we provide a few example queries for a more qualitative similarity comparison.

5.3.1 Similarity Scores. Since the synthetic data is one-to-one generated from the original data, we can compute BLEU scores to evaluate similarity [41]. We also compute the MAUVE scores, shown to be more capable of comparing similarity of text distributions [39]. See Table 3 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.

Table 3. Similarity scores of generated synthetic data. Left: Varying ϵ and fixed model size. Right: 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

Model	ϵ	BLEU	MAUVE
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

5.4 Empirical Privacy

The provable privacy provided by differential privacy 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 [7, 9, 40]. 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 [9]. This technique is frequently used to evaluate empirical privacy [23, 46, 62]. To perform this test, we construct examples with private information, referred to as canaries, and 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 repeat these type of canaries a varying number of times and include them in the training data. Then, we train the language model on the poisoned training data set with different DP guarantees. Once the model is trained, we generate the synthetic dataset and we check whether the secret part of each canary query is recovered in its output, and compare the injected canary to 100 different candidate sequences with different random 10-digit secret strings, measuring its rank among these secrets. We repeat this entire experiment, generating canaries, training models, and measuring leakage multiple times and average the metrics, reporting our results in Table 4. Here, as expected, we see that training without differential privacy leads to significant leakage. Canaries repeated 10 times are always the highest likelihood secrets, and are frequently extractable, while canaries repeated 100 times are always extractable. However, using our approach even with a large ϵ of 16 prevents the model from leaking the secrets, and increases the rank significantly. Recent techniques for converting attack success rates to lower bounds on the ϵ parameter[51] 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 large language models [9, 40].

Table 4. Privacy leakage.

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

6 CONCLUSION

We presented a novel approach for training downstream retrieval systems in a privacy-preserving manner by generating synthetic queries with DP LLMs. More generally, our approach presents a novel way to obtain DP guarantees for models with non-per-example decomposable losses. Empirically, we have shown that high performance on standard retrieval metrics can be achieved with DP guarantees, and verified the privacy protection of our approach. We also found that larger models can improve performance at the same privacy level. We conclude that private synthetic text generation represents a major step towards the development of privacy-preserving large-scale recommender systems.

ACKNOWLEDGMENTS

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 T5X 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

[1] 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*. 308–318.

[2] 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* (2016).

[3] Arnaud Berlioz, Arik Friedman, Mohamed Ali Kaafar, Roksana Boreli, and Shlomo Berkovsky. 2015. Applying Differential Privacy to Matrix Factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems* (Vienna, Austria) *(RecSys '15)*. Association for Computing Machinery, New York, NY, USA, 107–114. https://doi.org/10.1145/2792838.2800173

[4] 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* (2022).

[5] 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 (2020), 1877–1901.

[6] Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. 2011. "You Might Also Like:" Privacy Risks of Collaborative Filtering. In *2011 IEEE Symposium on Security and Privacy*. 231–246. https://doi.org/10.1109/SP.2011.40

[7] 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)*. IEEE, 1897–1914.

[8] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188* (2023).

[9] 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*, Vol. 267.

[10] Nicholas Carlini, Florian Tramer, Eric Wallace, et al. 2021. Extracting Training Data from Large Language Models.

[11] Steve Chien, Prateek Jain, Walid Krichene, Steffen Rendle, Shuang Song, Abhradeep Thakurta, and Li Zhang. 2021. Private Alternating Least Squares: Practical Private Matrix Completion with Tighter Rates. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 1877–1887. https://proceedings.mlr.press/v139/chien21a.html

[12] 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.

[13] 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* (2022).

[14] Soham De, Leonard Berrada, Jamie Hayes, Samuel L. Smith, and Borja Balle. 2022. Unlocking High-Accuracy Differentially Private Image Classification through Scale. arXiv:2204.13650 [cs.LG]

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

[16] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt &; Predict Paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems* (Seattle, WA, USA) *(RecSys '22)*. Association for Computing Machinery, New York, NY, USA, 299–315. https://doi.org/10.1145/3523227.3546767

[17] 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* (2019).

[18] Udit Gupta, Samuel Hsia, Vikram Saraph, Xiaodong Wang, Brandon Reagen, Gu-Yeon Wei, Hsien-Hsin S. Lee, David Brooks, and Carole-Jean Wu. 2020. DeepRecSys: A System for Optimizing End-To-End At-Scale Neural Recommendation Inference. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*. 982–995. https://doi.org/10.1109/ISCA45697.2020.00084

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

[20] Jingyu Hua, Chang Xia, and Sheng Zhong. 2015. Differentially Private Matrix Factorization. In *Proceedings of the 24th International Conference on Artificial Intelligence* (Buenos Aires, Argentina) *(IJCAI'15)*. AAAI Press, 1763–1770.

[21] Mengdi Huai, Di Wang, Chenglin Miao, Jinhui Xu, and Aidong Zhang. 2020. Pairwise Learning with Differential Privacy Guarantees. In *AAAI Conference on Artificial Intelligence*.

[22] 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* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 2553–2561. https://doi.org/10.1145/3394486.3403305

[23] 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* (2022).

[24] Arjan J. P. Jeckmans, Michael Beye, Zekeriya Erkin, Pieter Hartel, Reginald L. Lagendijk, and Qiang Tang. 2013. *Privacy in Recommender Systems*. Springer London, London, 263–281. https://doi.org/10.1007/978-1-4471-4555-4_12

[25] Vitor Jeronymo, Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, Roberto Lotufo, Jakub Zavrel, and Rodrigo Nogueira. 2023. InPars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval. *arXiv preprint arXiv:2301.01820* (2023).

[26] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

[27] 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* (2020).

[28] 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* (2019).

[29] Walid Krichene, Nicolas Mayoraz, Steffen Rendle, Li Zhang, Xinyang Yi, Lichan Hong, Ed H. Chi, and John R. Anderson. 2019. Efficient Training on Very Large Corpora via Gramian Estimation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=Hke20iA9Y7

[30] Shyong K. "Tony" Lam, Dan Frankowski, and John Riedl. 2006. Do You Trust Your Recommendations? An Exploration of Security and Privacy Issues in Recommender Systems. In *Emerging Trends in Information and Communication Security*, Günter Müller (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 14–29.

[31] Jianqiang Li, Ji-Jiang Yang, Yu Zhao, Bo Liu, Mengchu Zhou, Jing Bi, and Qing Wang. 2017. Enforcing Differential Privacy for Shared Collaborative Filtering. *IEEE Access* 5 (2017), 35–49. https://doi.org/10.1109/ACCESS.2016.2600258

[32] Tan Li, Linqi Song, and Christina Fragouli. 2020. Federated Recommendation System via Differential Privacy. In *2020 IEEE International Symposium on Information Theory (ISIT)*. 2592–2597. https://doi.org/10.1109/ISIT44484.2020.9174297

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

[34] 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* (2022).

[35] Lorenzo Minto, Moritz Haller, Benjamin Livshits, and Hamed Haddadi. 2021. Stronger Privacy for Federated Collaborative Filtering With Implicit Feedback. In *Proceedings of the 15th ACM Conference on Recommender Systems* (Amsterdam, Netherlands) *(RecSys '21)*. Association for Computing

Machinery, New York, NY, USA, 342–350. https://doi.org/10.1145/3460231.3474262

[36] Fatemehsadat Mireshghallah, Richard Shin, Yu Su, Tatsunori Hashimoto, and Jason Eisner. 2022. Privacy-Preserving Domain Adaptation of Semantic Parsers. *arXiv preprint arXiv:2212.10520* (2022).

[37] 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* (2021).

[38] Lin Ning, Karan Singhal, Ellie X Zhou, and Sushant Prakash. 2021. Learning federated representations and recommendations with limited negatives. *arXiv preprint arXiv:2108.07931* (2021).

[39] 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 (2021), 4816–4828.

[40] 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* (2023).

[41] Matt Post. 2018. A call for clarity in reporting BLEU scores. *arXiv preprint arXiv:1804.08771* (2018).

[42] Pranav Putta, Ander Steele, and Joseph W Ferrara. 2023. Differentially Private Conditional Text Generation For Synthetic Data Production. https://openreview.net/forum?id=LUql3ZOFwFD

[43] 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* (2020).

[44] 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 (2020), 5485–5551.

[45] N. Ramakrishnan, B.J. Keller, B.J. Mirza, A.Y. Grama, and G. Karypis. 2001. Privacy risks in recommender systems. *IEEE Internet Computing* 5, 6 (2001), 54–63. https://doi.org/10.1109/4236.968832

[46] 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* (2020).

[47] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and Metrics for Cold-Start Recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Tampere, Finland) *(SIGIR '02)*. Association for Computing Machinery, New York, NY, USA, 253–260. https://doi.org/10.1145/564376.564421

[48] 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*. 390–402.

[49] Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. *CoRR* abs/1804.04235 (2018). arXiv:1804.04235 http://arxiv.org/abs/1804.04235

[50] Hyejin Shin, Sungwook Kim, Junbum Shin, and Xiaokui Xiao. 2018. Privacy Enhanced Matrix Factorization for Recommendation with Local Differential Privacy. *IEEE Transactions on Knowledge and Data Engineering* 30, 9 (2018), 1770–1782. https://doi.org/10.1109/TKDE.2018.2805356

[51] Pierre Stock, Igor Shilov, Ilya Mironov, and Alexandre Sablayrolles. 2022. Defending against Reconstruction Attacks with R\'enyi Differential Privacy. *arXiv preprint arXiv:2202.07623* (2022).

[52] 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* (2021).

[53] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-Scale Commodity Embedding for E-Commerce Recommendation in Alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &; Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 839–848. https://doi.org/10.1145/3219819.3219869

[54] Jinpeng Wang, Jieming Zhu, and Xiuqiang He. 2021. Cross-batch negative sampling for training two-tower recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1632–1636.

[55] Chuhan Wu, Fangzhao Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2022. Fedcl: Federated contrastive learning for privacy-preserving recommendation. *arXiv preprint arXiv:2204.09850* (2022).

[56] Zhiyu Xue, Shaoyang Yang, Mengdi Huai, and Di Wang. 2021. Differentially Private Pairwise Learning Revisited. In *International Joint Conference on Artificial Intelligence*.

[57] Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. 2021. Efficient passage retrieval with hashing for open-domain question answering. *arXiv preprint arXiv:2106.00882* (2021).

[58] Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H Chi. 2020. Mixed negative sampling for learning two-tower neural networks in recommendations. In *Companion Proceedings of the Web Conference 2020*. 441–447.

[59] Liu Yang, Ben Tan, Vincent W. Zheng, Kai Chen, and Qiang Yang. 2020. *Federated Recommendation Systems*. Springer International Publishing, Cham, 225–239. https://doi.org/10.1007/978-3-030-63076-8_16

[60] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. 2021. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500* (2021).

[61] 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* (2022).

[62] 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. 363–375.

[63] Xue Zhu and Yuqing Sun. 2016. Differential Privacy for Collaborative Filtering Recommender Algorithm. In *Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics* (New Orleans, Louisiana, USA) *(IWSPA '16)*. Association for Computing Machinery, New York, NY, USA, 9–16. https://doi.org/10.1145/2875475.2875483

A ADDITIONAL RESULTS

For all experiments, we provide additional evaluation results of the recall scores. The Recall@10 score is the percentage of times the ground truth recommendation appears in the top 10 predictions. See Table 5 for the evaluation results on MSMARCO test dataset, and see 6 for the zero-shot evaluation results on the BEIR datasets.

Table 5. Recall evaluation of retrieval models. Left: Trained on synthetic data from DP-trained T5-models with varying ϵ and fixed model size T5-Base. Middle: Trained on synthetic data from DP-trained T5-models with varying model size and fixed $\epsilon = 3$. Right: DP-trained directly on original data with varying ϵ.

Source	ϵ	Recall@10
Original	∞	0.4098
T5-Base	∞	0.4192
T5-Base	16	0.3342
T5-Base	8	0.3196
T5-Base	3	0.3108

Source	ϵ	Recall@10
Original	∞	0.4098
T5-XL	3	0.3098
T5-Large	3	0.3094
T5-Base	3	0.3108
T5-Small	3	0.2272

Source	ϵ	Recall@10
Original	∞	0.4098
Original	16	0.0862
Original	8	0.0649
Original	3	0.0388

Table 6. Zero-shot recall evaluation of retrieval models trained on DP synthetic data vs. directly DP-trained retrieval models with $\epsilon = 16$.

Source	ϵ	Recall@10										
		arguana	cqadup	dbpedia	fiqa	hotpot	nfcorpus	quora	scidocs	scifact	covid	touche
Original	∞	0.5569	0.336758	0.1479	0.2440	0.3706	0.1013	0.8989	0.1071	0.5801	0.0116	0.0530
Original	16	0.4388	0.132508	0.0313	0.0906	0.1108	0.0365	0.7762	0.0503	0.2969	0.0063	0.0149
T5-Base	16	0.5768	0.316542	0.1217	0.2261	0.3398	0.1110	0.8763	0.1066	0.5848	0.0101	0.0489

B HYPERPARAMETERS

Due to time and computation constraints, we conducted a small-scale hyperparameter search to find the best T5-Small model DP finetuned on MSMARCO training dataset that resulted in the best BLEU scores 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 7 for the hyperparameter grid.

For the dual encoder model, we did not do hyperparameter search. We used learning rate 0.001, batch size 32, and epochs 5. For the DP finetuning experiments, we used a clipping norm of 0.1.

C C4 AND MSMARCO OVERLAP

A valid concern is if the T5 generative language models used to generate fine have been pre-trained on the query-document, then the privacy guarantees would be undermined since the LLM would have already seen the data. To

Table 7. Hyperparameters for T5 Model.

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	$\{10, 20, 30\}$

address this matter, we conducted an analysis to determine the extent of overlap of the MSMARCO dataset on the pre-training data, the C4 common crawl dataset [44]. 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.

D SYNTHETIC EXAMPLES

For qualitative comparison, in Tables 8 and 9 we provide a few examples of original query-document pairs and the synthetic queries generated from various model configurations.

Table 8. Synthetic query example 1.

Source	Text
Document	Premise. 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
T5-Base, $\epsilon = 16$	what is the most important characters in the series
T5-Base, $\epsilon = 8$	what is in this series?
T5-Base, $\epsilon = 3$	what is in this code for dfr1 vs
T5-Small, $\epsilon = 3$	what is isn't a character will do a story
T5-Large, $\epsilon = 3$	what is the most animated characters on disney cartoon show
T5-XL, $\epsilon = 3$	issue with show by mickey mouse

Table 9. Synthetic query example 2.

Source	Text
Document	The respiratory system provides oxygen to the body's cells while removing carbon dioxide, a waste product that can be lethal if allowed to accumulate. There are 3 major parts of the respiratory system: the airway, the lungs, and the muscles of respiration. The airway, which includes the nose, mouth, pharynx, larynx, trachea, bronchi, and bronchioles, carries air between the lungs and the body's exterior. The lungs... The cells of the human body require a constant stream of oxygen to stay alive.
Original Query	what is the purpose of the human respiratory system
T5-Base, $\epsilon = \infty$	what are three major parts of the respiratory system
T5-Base, $\epsilon = 16$	what are the respiratory system parts
T5-Base, $\epsilon = 8$	what are the respiratory system parts and do the respiratory system provide oxygen
T5-Base, $\epsilon = 3$	what are the respiratory system parts and do the respiratory system provide oxygen
T5-Small, $\epsilon = 3$	what does the respiratory system lungs do?
T5-Large, $\epsilon = 3$	what are the respiratory system parts
T5-XL, $\epsilon = 3$	what are the respiratory organs