QueryShield: A Platform to Mitigate Enterprise Data Leakage in Queries to External LLMs

Nitin Ramrakhiyani, Delton Myalil, Sachin Pawar, Manoj Apte
Rajan M A, Divyesh Saglani, Imtiyazuddin Shaik
TCS Research, Tata Consultancy Services Limited, India.
`{nitin.ramrakhiyani, delton.m, sachin7.p, manoj.apte}@tcs.com`
`{rajan.ma, divyesh.saglani, imtiyazuddin.shaik}@tcs.com`

Abstract

Unrestricted access to external Large Language Models (LLM) based services like ChatGPT and Gemini can lead to potential data leakages, especially for large enterprises providing products and services to clients that require legal confidentiality guarantees. However, a blanket restriction on such services is not ideal as these LLMs boost employee productivity. Our goal is to build a solution that enables enterprise employees to query such external LLMs, without leaking confidential internal and client information. In this paper, we propose QueryShield - a platform that enterprises can use to interact with external LLMs without leaking data through queries. It detects if a query leaks data, and rephrases it to minimize data leakage while limiting the impact to its semantics. We construct a dataset of 1500 queries and manually annotate them for their sensitivity labels and their low sensitivity rephrased versions. We fine-tune a set of lightweight model candidates using this dataset and evaluate them using multiple metrics including one we propose specific to this problem.

1 Introduction

The rapid advancement of Generative AI (Gen-AI), especially Large Language Models (LLMs), has significantly improved productivity across various industries. These models, capable of understanding and generating human-like text, save considerable time in tasks that traditionally required extensive human effort (Brown et al., 2020b; Radford et al., 2019). This efficiency allows businesses to enhance throughput without sacrificing output quality. AI is emerging as a tool that augments human capabilities, and by integrating AI, businesses can maintain a competitive edge (Brynjolfsson and McAfee, 2014). Companies that adopted AI experienced substantial productivity gains over those who did not (Bughin et al., 2018). This disparity has further expanded with the introduction of Gen-AI.

However, the privacy, security and safety implications of Gen-AI demands special investigation. We have seen sensitive details inadvertently surfacing in model outputs since they are trained on gargantuan datasets (Carlini et al., 2020). The accurate and coherent performance of LLMs emerge from their ability to memorize rare training samples, and this poses significant privacy threats when the datasets used to train them contain sensitive data (Inan et al., 2021). The above works, among others discuss the inevitable leakage of private data ***from* an LLM**. In contrast, there is potential for data leakage ***to* an LLM** through user queries (or prompts) as humans are the weakest link in security and privacy (Schneier, 2015). LLM service providers may use this interaction data for further model training and this may consequently spill the same sensitive data, that was once sent as a query, when attacked (Nasr et al., 2023).

This risk is further exacerbated when employees of companies, in attempts to gain competitive edge, leak confidential company data through their prompts to an external LLM service such as ChatGPT or Google Gemini. Despite the confidentiality guarantees provided by the LLM service providers, there have been unintentional instances where chat data was leaked (Open-AI, 2023). This concern has led some companies to enforce an organizational ban on chat models (Ray, 2023). Such restrictions severely impact the competitive edge of a company, especially if competent in-house alternatives are not provided. There is an increasing need for a privacy preserving prompting solution that not only safeguards against data leakage, but also ensures that the utility provided by powerful external LLMs like GPT-4o is not impacted.

This is an instance of Private Inferencing (PI) problem of neural networks (Gilad-Bachrach et al., 2016), where inferencing is done on encrypted data. Cryptographic methods like Fully Homomorphic Encryption (FHE) (Gentry, 2009) and Secure Multi-

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Industry Track), pages 358–369
April 30, 2025 ©2025 Association for Computational Linguistics

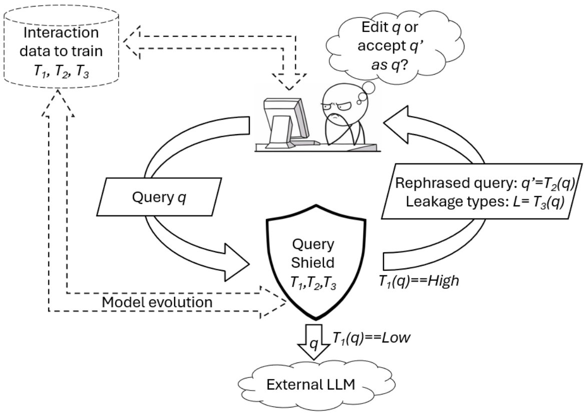
Interaction data to train T_1, T_2, T_3
Edit q or accept q' as q?
Query q
Rephrased query: $q'=T_2(q)$
Leakage types: $L= T_3(q)$
Query Shield T_1,T_2,T_3
$T_1(q)$==High
Model evolution
q
$T_1(q)$==Low
External LLM

Figure 1: QueryShield deployment scenario

Party Computation (MPC) (Ben-Or et al., 1988) also are employed to solve this problem. However, the communication and computation complexities of the above methods make it unrealistic to perform inference on *large* language models. Moreover, cryptographic methods require implementation in the server-side and the client (prompter) side. Execution of server-side code is not entertained by external LLM providers like Open-AI (ChatGPT), rendering such solutions impractical.

We propose that client-side **input guardrails**, that do not impact the usefulness of an external LLM, are a necessity to prevent data leakage through queries. A direct solution is data sanitization, where we detect the parts of the text that leak sensitive information (Ren et al., 2016). This approach is limited by the fact that even generic words may leak private information depending on the context in which they are used (Brown et al., 2022). So, *we need a method that analyzes the potential for data leakage from a query as a whole*. Additionally, this analysis should be used to rephrase the query such that data leakage, if any, is minimized, without impacting the semantic integrity of the message that the query aims to convey. This requires a system that can semantically understand the query, while simultaneously understanding the concept of data leakage.

In this paper, we propose *QueryShield*, a platform that lies between the enterprise environment and any external LLM (Figure 1). It detects outgoing queries that leak sensitive data and rephrases them to remove the sensitive contents. Queries that do not leak sensitive data are allowed to pass through to the external LLM. On the other hand, the rephrased versions of high sensitive queries (along with the identified types of leakage (Table 1) as an explanation) are fed back to the user who can optionally edit and re-submit them. The specific contributions of this paper are:

(i) Evaluation of contemporary lightweight language models for the tasks of identifying and rephrasing data leakage found in enterprise queries - especially the multi-task encoder-decoder and decoder-only models that we fine-tuned using curriculum learning (Sections 3.3, 3.5, and 3.4).

(ii) A dataset of 1500 queries[1] which can be fired from an enterprise environment to an external LLM, manually labelled with data leakage sensitivity as well as their corresponding gold-standard human rephrased versions for high sensitivity queries (Section 3.2).

(iii) A novel evaluation metric Cross-Reference ROUGE that evaluates semantic-preserving rephrasing of sensitive queries (Section 4.2).

2 Related Work

Private Inferencing (PI) refers to the process of drawing predictions from a neural network while keeping the input to the neural network private (Gilad-Bachrach et al., 2016). This is traditionally realized using cryptographic methods like MPC (Ben-Or et al., 1988), FHE (Gentry, 2009), and Differential Privacy (DP) (Dwork, 2011). Since MPC and FHE have high communication overheads, hybrid approaches that aim to optimize the solution from both an ML and FHE/MPC perspectives were used to advance PI offerings (Shaik et al., 2021; Jovanovic et al., 2022; Ge et al., 2021). The sheer scale of LLMs made even such optimizations insufficient to achieve PI in real-time. This shifted the focus to other Natural Language Processing methods. The first of such attempts included the usage of Parts of Speech tagging (Zewdu and Yitagesu, 2022), Named Entity Recognition (Ziyadi et al., 2020) and Personally Identifiable Information (PII) detection (Rosado, 2023). DP based methods add noise into private data to guarantee plausible deniability (Dwork, 2011). DP is used in LLM queries at the word, sentence, and document levels (Edemacu and Wu, 2024). Word level implementations like (Feyisetan et al., 2020; Carvalho et al.) where noise is added to word embeddings are limited by context based data leakage. Sentence level DP approaches

---

[1] The data will be made available upon request.

Personally Identifiable Information (PII): Names of any person, contact information like email or address

Business Relationships Information: Names of customers or vendors, their contact information, relationship value, deal information, contract clauses

Proprietary Data: Any kind of internal confidential/private data of an enterprise such as internal data and work artifacts. For an IT company, it would be source code, software requirements, algorithms, implementation details. For a Hospital, it would be treatment details, investigation reports, etc.

Internal Policies: Internal policies and procedures, security protocols, internal audits, project management guidelines/data, governance and compliance guidelines/data.

Strategic Plans: Long term strategy, product/service launch plans, proposed mergers/acquisitions/partnerships, marketing and sales strategies (like detail sales projections, campaign information)

Research and Development: Latest research initiatives, ideas, unpublished intellectual property

Table 1: Types of sensitive data from an organization's perspective

introduce noise in sentence embeddings (Meehan et al., 2022). This captures context based data leakages where words leak data depending on the context in which they are used. Chen et al. (2023) performs word based replacement of the queries and then rectifies the replaced words in the response. Most recently, Shen et al. (2024) propose ProSan which targets individual words using the context from the entire prompt. Our work, in contrast, does a semantic rephrasing of the entire query instead of targeting individual words.

3 Methodology

3.1 Problem Definition

We formulate the problem of preventing input data leakage from queries to an external LLM in the form of the following two tasks:

T_1 Detect whether a given query q contains *sensitive* data leakage or not, i.e., $T_1(q) \in \{HIGH, LOW\}$.

T_2 If a query q contains sensitive data leakage, then rephrase it to another query q' that doesn't leak any sensitive data and ensures that the intent of q is preserved as much as possible in q', i.e., $T_2(q) = q'$.

We define *sensitive* data from an organization's perspective in terms of 6 different types of data leakage which are described in Table 1. Based on these types, we formulate another task T_3 that is used to give feedback to the user for their query. This supplementary task is a more granular version of T_1 and aids in explainability.

T_3 Identify the types of data leakage present in a given query q, i.e., $T_3(q) \subset L$ where L is set of 6 data leakage types identified in Table 1.

In this paper, we evaluate different small language models as part of our *QueryShield* platform for addressing the three tasks described above. We choose the models from the 3 families of language models namely encoder-only models, decoder-only models, and encoder-decoder models.

3.2 Data Collection and Labelling

Here, we describe how we obtained the training examples used for fine-tuning/in-context learning of small language models. On investigating public instruction tuning datasets such as OASST1[2] and ChatAlpaca20K[3], it was evident that these datasets rarely contain information that is sensitive from an organization's perspective. Hence, we decided to create our own dataset, label (and rephrase) it manually, and use it for in-context learning, fine-tuning and evaluation.

3.2.1 Obtaining a collection of queries

We collected a set of 1500 queries by using 3 different strategies.

- A set of 600 queries were created semi-automatically. Multiple associates in our organization recorded an initial set of queries based on their work requirements. Then ChatGPT was used as an assistant to generate similar additional queries by using these human authored queries as seeds.
- A set of 300 queries were again generated by ChatGPT but by specifying a particular data leakage type (Table 1) at a time.
- A set of 600 queries was chosen randomly from a publicly available dataset – `ign_clean`[4].

3.2.2 Obtaining gold-standard labels

Each query in our dataset was manually annotated as follows:

---

[2] https://huggingface.co/datasets/OpenAssistant/oasst1

[3] https://huggingface.co/datasets/robinsmits/ChatAlpaca-20K

[4] https://huggingface.co/datasets/ignmilton/ign_clean_instruct_dataset_500k

Task	Input text	Output text
T_1	*What is the level of data leakage in the following Query with respect to private and confidential information of an organization? Answer as HIGH or LOW.* *Query:*$\langle$*query*$\rangle$	*Data Leakage Level: HIGH/LOW*
T_2	*From an organization's perspective, data leakage can be of following types - Personally Identifiable Information (PII), Business Relationships Information, Proprietary Data, Internal Policies, Strategic Plans, Research and Development.* *Rephrase the following Query by removing the above data leakage types if present in the Query while ensuring that the rephrased Query retains the original meaning as much as possible.* *Query:*$\langle$*query*$\rangle$	*Rephrased Query:* $\langle$*rephrased_query*$\rangle$
T_3	*From an organization's perspective, data leakage can be of following types - Personally Identifiable Information (PII), Business Relationships Information, Proprietary Data, Internal Policies, Strategic Plans, Research and Development.* *Identify the data leakage types present in the following Query.* *Query:*$\langle$*query*$\rangle$	*Data Leakage Types:* $\langle$*comma_separated_types*$\rangle$

Table 2: Input and output text pairs for each task where the input text consists of an instruction followed by a query and the output text consists of an output prefix followed by the expected output.

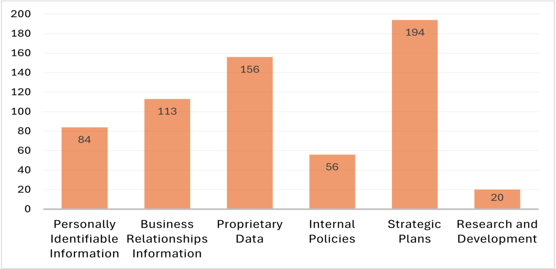

Figure 2: Distribution of various data leakage types in our dataset over 464 HIGH sensitivity queries. Note that a query can simultaneously exhibit multiple leakage types.

- Task T_1: A label (HIGH or LOW) indicating whether the query contains any sensitive data from an organization's point of view.
- Task T_2: When the T_1 label is HIGH, a rephrased version of the query such that it contains no sensitive data and its original semantics are preserved as much as possible.
- Task T_3: When the T_1 label is HIGH, a set of labels indicating the data leakage types (Table 1) mentioned in the query.

For T_1, each query was annotated by two annotators and the inter-annotator agreement in terms of Cohen's Kappa statistic was found to be 0.875. The disagreements were resolved through discussions. 464 queries out of 1500 were identified as HIGH sensitivity queries from a data leakage perspective. The manually rephrased versions of these 464 queries were added back to the dataset with T_1 label as "LOW" (and T_2/T_3 labels as NA), making the final effective dataset size to be of **1964 queries**. Figure 2 shows the distribution of the 6 data leakage types and Table 6 (in Appendix) shows a few examples of these annotations.

3.3 Encoder-only models

We explored encoder-only models only for Tasks T_1 and T_3 which are binary classification and multi-label multi-class classification tasks, respectively. Task T_2 being a text generation task, encoder models are not applicable. We employ **Attn-BERT** (Vaishampayan et al., 2023) which uses attention weighted BERT (Devlin et al., 2019) representations of tokens in a query, concatenated with the [CLS] representation of the query. The concatenated representation is passed through a softmax layer for final prediction. For multi-label classification, each class label has a separate attention head and leads to its specific representation.

3.4 Encoder-Decoder models

We considered encoder-decoder models because they offer text generation capabilities (unlike encoder-only models) as well as they are amenable to full fine-tuning due to their moderate size (unlike larger decoder-only models). We formulate the three tasks as text-to-text transformation tasks and fine-tune a single T5-base model (Raffel et al., 2020) for all the tasks. For each task, a specific instruction is prefixed to a query to construct the input text to the model. Table 2 shows the different instructions used for the tasks T_1, T_2, and T_3. Also, the expected output for each task is different. For T_1, the output text is simply data leakage level of the query which can be either HIGH or LOW. For T_2, the output text is the input query's rephrased version that contains no sensitive data and preserves the original semantics as much as possible. For T_3, the output text is simply a comma-separated list of data leakage types present in the input query. Consideration of the T5-base model enables any organization with limited hardware resources to deploy (and fully fine-tune) it in-house.

Training Strategy: We follow a model training strategy similar to *curriculum learning* (Bengio et al., 2009) where the model is initially trained with instances of an *easier* task followed by instances of *harder* tasks. Task T_1 is easier as compared to task T_3 because T_3 is a more fine-grained version of T_1 where in addition to detecting whether a query contains sensitive data or not, it is expected to specifically identify data leakage types as well. Task T_2 can be considered as the hardest, as it needs to rephrase the input query by lowering the sensitive data leakage level and ensuring that the original meaning is preserved as much as possible. Hence, we train the overall model in the following 3 steps:

1. Train using only T_1 instances for K epochs.
2. Continue training the model with the best validation loss in *Step 1* with instances of T_1 and T_3 for K epochs.
3. Continue training the model with the best validation loss in *Step 2* with instances of all tasks T_1, T_2, and T_3 for K epochs.

The final model trained for $K = 50$ epochs using curriculum learning (CL) for the tasks T_1, T_2, and T_3 is referred to as **T5-base_CL**.

3.5 Decoder-only models

We also explored decoder-only models to solve all the three tasks using few-shot in-context learning (Brown et al., 2020a) as well as fine-tuning.

Few-shot in-context learning: For each task, we designed a prompt which consists of the detailed definition of data leakage in terms of the 6 types followed by an instruction to generate the desired output. For in-context learning, we also added a few demonstrations of the task as few-shot examples. For each query in the test set, we chose 8 most similar queries from the training set to use as few-shot examples. For T_2, we chose only from HIGH sensitivity training queries whereas for T_1 and T_3, we chose 4 HIGH and 4 LOW sensitivity training queries. To identify the most similar queries from the training set, we used cosine similarity between their text embeddings which were obtained using a sentence transformer model[5]. Tables 7 and 8 (in Appendix) show the prompts used for the tasks T_1, T_2, and T_3. We chose one open-source (Mistral-7B-Instruct) and one closed-source model (GPT-4o-mini) for our experiments. Please note that although the GPT-4o model is an external LLM, it is included just for comparison with other models. It is not considered for deployment because the entire purpose of this work is to avoid sending sensitive information to such external LLMs.

Fine-tuning: Considering our limited hardware, we opted for parameter efficient fine-tuning of the 4-bit quantized Mistral-7B-Instruct model using QLoRA (Dettmers et al., 2024). We used the same curriculum learning strategy and the same training instances which are used for fine-tuning the T5-base model as described above. We refer to this fine-tuned model as **Mistral-7B-Instruct_CL**.

4 Experiments

4.1 Dataset

The 1964 queries in our dataset (Section 3.2) were split into train, development and test sets in the proportion (60%, 15%, 25%) respectively, with roughly a similar stratified division of HIGH sensitivity queries entering each split i.e. (280, 74, 110) respectively. We used the development set for tuning the hyperparameters (Appendix A).

4.2 Evaluation Metrics

Task T_1: We report the standard Precision, Recall and F1 score (Manning, 2008) for the HIGH label.

Task T_3: We report the micro and macro averaged F1 scores across the 6 data leakage types.

Task T_2: The evaluation of T_2 is non-trivial because it needs to measure two aspects - *Leakage* prevention and *Intent* preservation. We report **BertScore (BS)** which is generally used to evaluate text generation tasks by comparing the model generated rephrased queries with the gold-standard rephrased queries (Zhang et al., 2019). This metric

---

[5] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Model	**Task T_1**			**Task T_2**			**Task T_3**	
	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F1}$	$\boldsymbol{CRR1_{P/R/F1}}$	$\boldsymbol{P^{LOW}_{T_1(x)}}$	$\boldsymbol{BS_{F1}}$	$\boldsymbol{\mu F1}$	$\boldsymbol{mF1}$
Attn-BERT	0.873	**0.976**	0.921	-	-	-	**0.616**	**0.524**
T5-base_CL	**0.902**	0.946	**0.923**	0.866 / 0.909 / 0.867	0.903	0.875	0.553	0.399
Mistral-7B-instruct (few-shot)	0.509	0.597	0.550	**0.881** / 0.906 / 0.880	**0.924**	0.872	0.413	0.402
GPT-4o-mini (few-shot)	0.599	0.752	0.667	0.869 / 0.921 / 0.880	0.864	0.880	0.500	0.476
Mistral-7B-instruct_CL	0.856	0.973	0.911	0.858 / **0.961** / **0.893**	0.918	**0.892**	0.527	0.408

Table 3: Evaluation results for the Tasks T_1, T_2, and T_3. Evaluation metrics for T_1 are precision, recall, F1-score for HIGH label. Evaluation metrics for T_2 are CRR-1, $P^{LOW}_{T_1(x)}$, and BERTScore. Evaluation metrics for T_3 are micro and macro averaged F1 for all 6 leakage types. All numbers are averaged across 3 independent runs.

Model	**Task T_1**			**Task T_2**			**Task T_3**	
	$\boldsymbol{P}$	$\boldsymbol{R}$	$\boldsymbol{F1}$	$\boldsymbol{CRR1_{P/R/F1}}$	$\boldsymbol{P^{LOW}_{T_1(x)}}$	$\boldsymbol{BS_{F1}}$	$\boldsymbol{\mu F1}$	$\boldsymbol{mF1}$
T5-base_CL (all tasks)	**0.902**	**0.946**	**0.923**	0.866 / 0.909 / **0.867**	0.903	0.875	**0.553**	**0.399**
T5-base (w/o CL, all tasks)	0.849	0.888	0.865	0.866 / **0.910** / 0.866	0.879	**0.881**	0.492	0.340
T5-base_CL (T_1 & T_2)	0.889	0.918	0.903	**0.867** / 0.903 / 0.863	**0.906**	0.877	-	-
T5-base_CL (T_1 & T_3)	0.881	0.964	0.920	-	-	-	0.492	0.385
T5-base (T_1 only)	0.869	0.933	0.899	-	-	-	-	-
T5-base (T_2 only)	-	-	-	0.862 / 0.900 / 0.857	0.876	0.876	-	-

Table 4: Ablation results for T5-base_CL model. All numbers are averaged across 3 independent runs.

measures the *Intent* preservation aspect to some extent. To measure the *Leakage* prevention aspect, we use the Attn-BERT model trained for task T_1 to classify the rephrased queries. The fraction of these queries which are classified as LOW is computed as a new metric - $\boldsymbol{P^{LOW}_{T_1(x)}}$ (*precision of label LOW as per the* T_1 *model*). Higher the value of this metric, better is the rephrasing because the rephrased queries should not ideally contain any sensitive data.

In order to cover both these aspects (*Leakage* and *Intent*) in a single metric, we propose a novel evaluation metric – **Cross-Reference ROUGE (CRR)** which compares the generated text with two references (the original query as well as the gold-standard rephrased query), unlike vanilla ROUGE which uses a single reference. To explain the metric, we discuss its unigram form – CRR1. Let O, G, and R be the sets of unigrams in the original query, the gold-standard rephrased query, and the model-generated rephrased query, respectively.

$$FP_l \;=\; |(O \setminus G) \cap R| \tag{1}$$

$$TP_l \;=\; |R \setminus FP_l| \tag{2}$$

$$CRR1_P = \frac{TP_l}{TP_l + FP_l} \tag{3}$$

$$FN_i \;=\; |(O \cap G) \setminus R| \tag{4}$$

$$TP_i \;=\; |(O \cap G) \setminus FN_i| \tag{5}$$

$$CRR1_R = \frac{TP_i}{TP_i + FN_i} \tag{6}$$

$$CRR1_{F1} = \frac{2 \cdot CRR1_P \cdot CRR1_R}{CRR1_P + CRR1_R} \tag{7}$$

Leakage aspect: $O \setminus G$ captures the *sensitive* contents of the original query and any overlap of R with this sensitive content would indicate *Excess Leakage*. Hence, such overlap is the set of false positives (FP_l) which shouldn't have been there in R (Eq. 1). The remaining terms in R are considered as true positives (Eq. 2) and are used to compute $CRR1_P$ (Eq. 3).

Intent aspect: $O \cap G$ captures the allowable intent of the original query and absence of these terms in R indicates *Intent Loss*. Hence, these missing terms are the false negatives (FN_i) (Eq. 4). The remaining terms in $O \cap G$ are considered as true positives (Eq. 5) and are used to compute $CRR1_R$ (Eq. 6). Finally, the $CRR1_{F1}$ score (Eq. 7) is computed as the final metric.

4.3 Results and Analysis

Table 3 shows the overall evaluation results for all the tasks in terms of all the metrics. For T_1, T5-base_CL is the best performing model, closely followed by Attn-BERT. Decoder-only models do not perform well for T_1 in few-shot setting. For T_3, Attn-BERT is the best model in terms of both micro and macro-F1. For T_2, Mistral-7B-instruct (few-shot as well as fine-tuned) performs the best in terms of $CRR1_{F1}$ as well as $P^{LOW}_{T_1(x)}$ which are the two most important metrics for T_2. We highlight a few examples of the rephrasing in Table 9. Overall, `T5-base_CL` is the best model in practice across the three tasks, because it is either the best or performs comparably in terms of most metrics. Moreover, its inference time and hardware requirements are lower compared to Mistral. Also, we observed that T_1 performance of T5-base_CL is uniformly high across all the 6 data leakage types (Table 5).

Data Leakage Type	Recall
Personally Identifiable Information (PII)	0.944
Business Relationships Information	0.952
Proprietary Data	0.949
Internal Policies	0.923
Strategic Plans	0.941
Research and Development	0.889

Table 5: Recall for T5-base_CL across multiple data leakage types

Ablation analysis: We carry out a detailed ablation analysis for T5-base_CL to gauge two design choices – curriculum learning and multi-task learning (Table 4). It can be observed that the performance of T_1 and T_3 gets affected significantly without curriculum learning as well as multi-task learning. For T_2, the benefit of these two design choices is not very conclusive, especially multi-task learning. However, it can be observed that the model trained only for T_2 lags behind T5-base_CL in terms of $CRR1_{F1}$ and $P^{LOW}_{T_1(x)}$ both.

4.4 Deployment Scenario

QueryShield contains all three models, i.e., Attn-BERT, T5-base_CL, and Mistral-7B-Instruct_CL, configured by the system administrator considering – (i) accuracy, (ii) inference time per query (Mistral-7B-Instruct_CL: 1.4 sec vs T5-base_CL: 0.3 sec), (iii) and fine-tuning capability where a model can be fine-tuned using incremental training data from user feedback. Default recommendations for the best end-to-end accuracy would be using T5-base_CL for T_1, Mistral-7B-Instruct_CL for T_2 and Attn-BERT for T_3.

Long queries: One advantage that Mistral has over T5 is its longer context window. Hence, for a query longer than 512 tokens, Mistral model is preferred for rephrasing. For T_1/T_3 using T5-base_CL and Attn-BERT, if any longer query is encountered, it is first split into multiple chunks and inference is run separately for each chunk. If any of these chunks is found to be sensitive, then T_1 predicts HIGH for overall query whereas T_3 predicts union of leakage types predicted for all the chunks.

Potentially incorrect rephrasing: For any input query q which is detected by T_1 to be sensitive, QueryShield suggests the revised query q' to the user (Figure 1). If q' is obtained using T5-base_CL and its sensitivity is still found to be HIGH as per T_1, then we use Mistral to generate q'' as an alternative to q'. If this alternative q'' is also found to be HIGH as per T_1, then the user is asked to rephrase manually. User interactions, including manual rephrasings are logged for further fine-tuning.

5 Conclusion and Future Work

To balance between access to external LLMs and the potential risk of enterprise data leakage, we proposed the *QueryShield* platform. It lies between any external LLM and the enterprise environment and detects sensitive data leakage in the queries as well as rephrases the original queries to remove any potential data leakage. We explored multiple lightweight language models as part of *QueryShield* so that they can be hosted in-house with limited hardware resources. We evaluated these models for the tasks of detecting sensitive data leakage, rephrasing sensitive queries, and identifying data leakage types, using a manually annotated dataset of 1500 queries.

In future, we would incorporate human feedback once the model is deployed, so that the deployed models can be further fine-tuned periodically. We are also extending the platform to handle data leakage from the context of sequential queries.

References

Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. 1988. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, page 1–10, New York, NY, USA. Association for Computing Machinery.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Hannah Brown, Katherine Lee, Fatemehsadat Mireshghallah, Reza Shokri, and Florian Tramèr. 2022. What does it mean for a language model to preserve privacy? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 2280–2292, New York, NY, USA. Association for Computing Machinery.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. *CoRR*, abs/2005.14165.

E. Brynjolfsson and A. McAfee. 2014. *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. W. W. Norton.

Jacques Bughin, Eric Hazan, Sree Ramaswamy, Michael Chui, Tera Allas, Peter Dahlström, and et al. 2018. Artificial intelligence: The next digital frontier?

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2020. Extracting training data from large language models. *CoRR*, abs/2012.07805.

Ricardo Silva Carvalho, Theodore Vasiloudis, Oluwaseyi Feyisetan, and Ke Wang. *TEM: High Utility Metric Differential Privacy on Text*, pages 883–890.

Yu Chen, Tingxin Li, Huiming Liu, and Yang Yu. 2023. Hide and seek (has): A lightweight framework for prompt privacy protection. *Preprint*, arXiv:2309.03057.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Cynthia Dwork. 2011. *Differential Privacy*, pages 338–340. Springer US.

Kennedy Edemacu and Xintao Wu. 2024. Privacy preserving prompt engineering: A survey. *Preprint*, arXiv:2404.06001.

Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 178–186, New York, NY, USA. Association for Computing Machinery.

Zhengqiang Ge, Zhipeng Zhou, Dong Guo, and Qiang Li. 2021. Practical two-party privacy-preserving neural network based on secret sharing. *CoRR*, abs/2104.04709.

Craig Gentry. 2009. A fully homomorphic encryption scheme.

Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. 2016. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 201–210, New York, New York, USA. PMLR.

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*.

Huseyin A. Inan, Osman Ramadan, Lukas Wutschitz, Daniel Jones, Victor Rühle, James Withers, and Robert Sim. 2021. Privacy analysis in language models via training data leakage report. *CoRR*, abs/2101.05405.

Nikola Jovanovic, Marc Fischer, Samuel Steffen, and Martin Vechev. 2022. Private and reliable neural network inference. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 1663–1677, New York, NY, USA. Association for Computing Machinery.

Christopher D Manning. 2008. *Introduction to information retrieval*. Syngress Publishing,.

Casey Meehan, Khalil Mrini, and Kamalika Chaudhuri. 2022. Sentence-level privacy for document embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3367–3380, Dublin, Ireland. Association for Computational Linguistics.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *Preprint*, arXiv:2311.17035.

Open-AI. 2023. March 20 chatgpt outage: Here's what happened.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Siladitya Ray. 2023. Samsung bans chatgpt among employees after sensitive code leak.

Jingjing Ren, Ashwin Rao, Martina Lindorfer, Arnaud Legout, and David Choffnes. 2016. Recon: Revealing and controlling pii leaks in mobile network traffic. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '16, page 361–374, New York, NY, USA. Association for Computing Machinery.

Eidan Rosado. 2023. Pii-codex: a python library for pii detection, categorization, and severity assessment. *The Journal of Open Source Software*, 8.

Bruce Schneier. 2015. *The Human Factor*, chapter 17. John Wiley & Sons, Ltd.

Imtiyazuddin Shaik, Raj Chaudhari, M. A. Rajan, J. Gubbi, P. Balamuralidhar, and S. Lodha. 2021. Wip: Qos based recommendation system for efficient private inference of cnn using fhe. In *Information Systems Security*, pages 198–211, Cham. Springer International Publishing.

Zhili Shen, Zihang Xi, Ying He, Wei Tong, Jingyu Hua, and Sheng Zhong. 2024. The fire thief is also the keeper: Balancing usability and privacy in prompts. *CoRR*, abs/2406.14318.

Sushodhan Vaishampayan, Nitin Ramrakhiyani, Sachin Pawar, Aditi Pawde, Manoj Apte, and Girish Palshikar. 2023. Audit report coverage assessment using sentence classification. In *Proceedings of the Sixth Workshop on Financial Technology and Natural Language Processing*, pages 31–41.

Alebachew Zewdu and Betselot Yitagesu. 2022. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Morteza Ziyadi, Yuting Sun, Abhishek Goswami, Jade Huang, and Weizhu Chen. 2020. Example-based named entity recognition. *CoRR*, abs/2008.10570.

A Implementation Details

Attn-BERT: We fine-tuned Attn-BERT model with the following hyper-parameters – Batch size = 64, Adam optimizer with learning rate = 0.0001, number of epochs = 10. Also, we only fine-tuned the last encoder layer of BERT, keeping other BERT parameters unchanged. The hyper-parameters were tuned using the development set.
T5-base: We fine tuned the T5-base[6] model with the following hyper-parameters – Batch size $= 64$, Adam optimizer with learning rate $= 0.00005$, number of epochs $(K) = 50$. These hyper-parameters were tuned using the development set.
Mistral-7B-Instruct (few-shot): We used the Mistral-7B-Instruct[7] with temperature setting of 0.3 and maximum number of output tokens as 1000.
Mistral-7B-Instruct (fine-tuned): We considered the Mistral-7B-Instruct model as above and fine-tuned it using QLoRA with the following parameters – quantization: 4-bit, LoRA $r = 64$, LoRA $\alpha = 2$, LoRA dropout $= 0.0$, and no LoRA bias (as suggested in the mistral-finetune library[8]). Further, the target modules for appending LoRA adapters were only the self-attention layers, namely q, k, v, and o (following Hu et al. (2021)). Other training hyper-parameters – Batch size $= 4$, Adam optimizer with learning rate $= 0.0001$, number of epochs $(K) = 5$. During inference, we used a temperature setting of 0.3 and maximum number of output tokens as 1000.

All the experiments were performed by making 3 independent runs and then averaging all the metrics across the 3 runs. All our experiments with the Mistral model are performed on an Nvidia Tesla V100 GPU with 32 GB GPU memory. All experiments with the T5-base model with are performed on an Nvidia Tesla A100 GPU with 10 GB GPU memory.

---

[6] `https://huggingface.co/google-t5/t5-base`

[7] `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2`

[8] `https://github.com/mistralai/mistral-finetune`

Query: *What are the latest trends in employee benefits that we can incorporate into our benefits package, considering our current offerings such as health insurance plans, retirement savings programs, tuition reimbursement, and wellness initiatives?*
Data Leakage Level: HIGH (T_1)
Rephrased Query: *What are the latest trends in employee benefits to incorporate into benefits packages?* (T_2)
Data Leakage Types: Internal Policies; Strategic Plans (T_3)

Query: *Our client, XYZ Pharmaceuticals, requires a mobile app to track patient medication adherence for a new experimental drug undergoing FDA approval. Develop a project plan outlining key milestones and deliverables.*
Data Leakage Level: HIGH (T_1)
Rephrased Query: *Develop a project plan for a mobile app that tracks patient medication adherence for a new experimental drug undergoing FDA approval, outlining key milestones and deliverables.* (T_2)
Data Leakage Types: Business relationships Information, Proprietary data (T_3)

Query: *Write an in-depth analysis on the varying effects of long-term exposure to artificial light at night on different human health parameters such as sleep patterns, mental health, hormonal balance, cardiovascular health, and the risk of chronic diseases. Use reliable scientific sources to support your findings and provide actionable solutions to mitigate the negative effects of artificial light on human health.*
Data Leakage Level: LOW (T_1)
Rephrased Query: NA (T_2)
Data Leakage Types: NA (T_3)

Query: *Please create a NodeJS server using Express that provides clients with access to JSON data through RESTful API endpoints. Ensure that the endpoints return data in a clear and concise format, and that appropriate HTTP status codes are used for responses. Additionally, consider implementing error handling to provide users with meaningful feedback in case of any issues with the API requests.*
Data Leakage Level: LOW (T_1)
Rephrased Query: NA (T_2)
Data Leakage Types: NA (T_3)

Query: *What are the latest trends in employee benefits to incorporate into benefits packages?* (manually rephrased version of an original query with HIGH sensitivity (first query in this table) is added back to the dataset)
Data Leakage Level: LOW (T_1)
Rephrased Query: NA (T_2)
Data Leakage Types: NA (T_3)

Table 6: Some examples of manual annotations (shown in blue) for Tasks T_1, T_2, and T_3 from our dataset.

From an organization's perspective, data leakage can be of following types:
1. Personally Identifiable Information (PII): Names of any person, contact information like email or address
2. Business Relationships Information: Names of customers or vendors, their contact information, relationship value, deal information, contract clauses
3. Proprietary Data: Any kind of internal confidential/private data of an enterprise such as internal data and work artifacts. For an IT company, it would be source code, software requirements, algorithms, implementation details. For a Hospital, it would be treatment details, investigation reports, etc.
4. Internal Policies: Internal policies and procedures, security protocols, internal audits, project management guidelines/data, governance and compliance guidelines/data.
5. Strategic Plans: Long term strategy, product/service launch plans, proposed mergers/acquisitions/partnerships, marketing and sales strategies (like detail sales projections, campaign information)
6. Research and Development: Latest research initiatives, ideas, unpublished intellectual property

There may be multiple data leakage types present in a Query sent to an LLM. Rephrase the following Queries by removing applicable data leakage types while ensuring that the rephrased Query retains the original meaning as much as possible.

Query: $\langle training\_query_1 \rangle$
Rephrased Query: $\langle rephrased\_training\_query_1 \rangle$

. . .
Query: $\langle training\_query_8 \rangle$
Rephrased Query: $\langle rephrased\_training\_query_8 \rangle$

Query: $\langle test\_query \rangle$
Rephrased Query: language model to generate its response here...

Table 7: Few-shot in-context learning prompt used for Task T_2 by the decoder-only models (Mistral-7B-Instruct and GPT-4o-mini)

From an organization's perspective, data leakage can be of following types:
1. Personally Identifiable Information (PII): Names of any person, contact information like email or address
2. Business Relationships Information: Names of customers or vendors, their contact information, relationship value, deal information, contract clauses
3. Proprietary Data: Any kind of internal confidential/private data of an enterprise such as internal data and work artifacts. For an IT company, it would be source code, software requirements, algorithms, implementation details. For a Hospital, it would be treatment details, investigation reports, etc.
4. Internal Policies: Internal policies and procedures, security protocols, internal audits, project management guidelines/data, governance and compliance guidelines/data.
5. Strategic Plans: Long term strategy, product/service launch plans, proposed mergers/acquisitions/partnerships, marketing and sales strategies (like detail sales projections, campaign information)
6. Research and Development: Latest research initiatives, ideas, unpublished intellectual property

There may be multiple data leakage types present in a Query sent to an LLM. Identify whether the following queries have data leakage levels of HIGH or LOW. In case of HIGH data leakage level, also identify the types of data leakage types present in the Query.

Query: $\langle training\_query_1 \rangle$
Data Leakage Level: HIGH / LOW
Data Leakage Types: $\langle data\_leakage\_types\_in\_query_1 \rangle$

. . .
Query: $\langle training\_query_8 \rangle$
Data Leakage Level: HIGH / LOW
Data Leakage Types: $\langle data\_leakage\_types\_in\_query_8 \rangle$

Query: $\langle test\_query \rangle$
Data Leakage Level: language model to generate its response here...

Table 8: Few-shot in-context learning prompt used for Tasks T_1 and T_3 by the decoder-only models (Mistral-7B-Instruct and GPT-4o-mini)

Original Query: *SecureLogin, a company specializing in cybersecurity solutions, is building a user authentication system for a new e-commerce platform like TrendyThreads. The platform needs to offer flexible login options like username/password and social media logins, while maintaining robust security measures. Can you suggest best practices for multi-factor authentication (MFA) and secure password hashing techniques to protect user data?* **Gold-standard Rephrased Query**: *A company is building a user authentication system for an e-commerce platform. Can you suggest best practices for multi-factor authentication (MFA) and secure password hashing techniques to protect user data?* **T5-base_CL Rephrased Query**: *Can you suggest best practices for multi-factor authentication (MFA) and secure password hashing techniques to protect user data for a user authentication system for an e-commerce platform like TrendyThreads?* **Mistral-7B-Instruct (few-shot) Rephrased Query**: *Suggest best practices for multi-factor authentication (MFA) and secure password hashing techniques to protect user data for a user authentication system for an e-commerce platform, which is being built by SecureLogin, a cybersecurity solutions company. The platform should offer flexible login options like username/password and social media logins while maintaining robust security measures.* **Mistral-7B-Instruct_CL Rephrased Query**:*Can you suggest best practices for multi-factor authentication (MFA) and secure password hashing techniques to protect user data for a new e-commerce platform like TrendyThreads?*
Original Query: *What are the latest trends in employee benefits that we can incorporate into our benefits package, considering our current offerings such as health insurance plans, retirement savings programs, tuition reimbursement, and wellness initiatives?* **Gold-standard Rephrased Query**: *What are the latest trends in employee benefits to incorporate into benefits packages?* **T5-base_CL Rephrased Query**: *What are the latest employee benefits that an organization can incorporate into their benefits package, considering their current offerings such as health insurance plans, retirement savings programs, tuition reimbursement, and wellness initiatives?* **Mistral-7B-Instruct (few-shot) Rephrased Query**: *What are the latest trends in employee benefits that can be incorporated into a benefits package?* **Mistral-7B-Instruct_CL Rephrased Query**:*What are the latest trends in employee benefits that can be incorporated into a benefits package, considering the current offerings such as health insurance plans, retirement savings programs, tuition reimbursement, and wellness initiatives?*
Original Query: *Our client, XYZ Pharmaceuticals, requires a mobile app to track patient medication adherence for a new experimental drug undergoing FDA approval. Develop a project plan outlining key milestones and deliverables.* **Gold-standard Rephrased Query**: *Develop a project plan for a mobile app that tracks patient medication adherence for a new experimental drug undergoing FDA approval, outlining key milestones and deliverables.* **T5-base_CL Rephrased Query**: *A pharmaceutical company requires a mobile app to track patient medication adherence for a new experimental drug undergoing FDA approval. Develop a project plan outlining key milestones and deliverables.* **Mistral-7B-Instruct (few-shot) Rephrased Query**: *Our client, a pharmaceutical company, requires a mobile app to track patient medication adherence. Develop a project plan outlining key milestones and deliverables.* **Mistral-7B-Instruct_CL Rephrased Query**: *A Pharmaceutical company requires a mobile app to track patient medication adherence for a new experimental drug undergoing FDA approval. Develop a project plan outlining key milestones and deliverables.*
Original Query: *How can we improve our patent application process for our latest innovation in renewable energy technology, which involves a proprietary method for enhancing solar panel efficiency? We want to ensure that our patent filings are comprehensive and protect all aspects of our technology, including the specific materials and fabrication techniques we've developed.* **Gold-standard Rephrased Query**: *How can a company improve its patent application process for its latest innovation, which involves a novel method for enhancing a system's efficiency? It is required to ensure that the patent filings are comprehensive and protect all aspects of the innovation.* **T5-base_CL Rephrased Query**: *How can a company improve their patent application process for their latest innovation in renewable energy technology, which involves a proprietary method for enhancing solar panel efficiency? They want to ensure that their patent filings are comprehensive and protect all aspects of their technology, including the specific materials and fabrication techniques they've developed.* **Mistral-7B-Instruct (few-shot) Rephrased Query**: *How can an organization improve their patent application process? Ensure comprehensive patent filings protect all aspects of the technology, including specific materials and fabrication techniques.* **Mistral-7B-Instruct_CL Rephrased Query**: *How to improve patent application process for a renewable energy technology, which involves a proprietary method for enhancing solar panel efficiency? This should ensure that patent filings are comprehensive and protect all aspects of the technology, including the specific materials and fabrication techniques.*

Table 9: Some examples of rephrasing by T5-base_CL and Mistral-7B-Instruct. Text highlighted in red indicates sensitive content that has still remained in model-generated rephrased queries when compared to the gold-standard rephrased queries.