Towards *Contextual* Sensitive Data Detection

Liang Telkamp[1,2] **and Madelon Hulsebos**[2]*
[1]University of Amsterdam, [2]Centrum Wiskunde & Informatica

Abstract

The emergence of open data portals necessitates more attention to protecting sensitive data before datasets get published and used. To do so effectively, we observe the need to refine and broaden our definitions of sensitive data. We argue that *the sensitivity of data depends on its context*, and that this context should be considered in methods for assessing data sensitivity. To this end, we introduce a framework, termed ConSenser, for contextual sensitive data detection based on two aspects: 1) *type contextualization*, which considers the type of data values within the overall context of the dataset or document to assess their true sensitivity, and 2) *domain contextualization*, which assesses the sensitivity of data values informed by domain-specific information external to the dataset, such as geographic origin of a dataset. Experiments instrumented with language models confirm that type-contextualization significantly reduces the number of false positives for type-based sensitive data detection and reaches a recall of 94% compared to 63% with commonly used tools. Moreover, domain-contextualization through retrieval of sensitivity rules effectively grounds sensitive data detection in relevant context in non-standard data domains. A case study with humanitarian data experts also illustrates that context-grounded explanations provide useful guidance in manual data auditing processes. We open-source the code and annotated datasets at `https://github.com/trl-lab/sensitive-data-detection`.

1 Introduction

Data sharing underpins data reuse across sciences, enterprises, and governments through public and private portals (Borgman and Groth, 2025; Worth et al., 2024; Zhang et al., 2024; Wilkinson et al., 2016; Brickley et al., 2019). With the rise of open data sharing portals, comes the need to protect sensitive data from undesired and harmful use. Sensitive data includes direct personal identifiers (e.g., names, emails (European Parliament and Council of the European Union, 2016)) as well as indirect or contextual information that becomes a potential risk under certain conditions (OCHA, 2025). For example, facility locations or military logs may be sensitive depending on geopolitics or timing. This illustrates the complexity of sensitive data detection and the need for nuanced, context-dependent methods. Beyond operational and ethical concerns of publishing sensitive data, Large Language Models (LLMs) increase the risks of leakage: trained on web data, they may memorize and reproduce sensitive information (Subramani et al., 2023; Worth et al., 2024; Carlini et al., 2021; Lukas et al., 2023). Studies show sensitive data is often underspecified in documentation on open data portals like HuggingFace (Yang et al., 2024; Akhtar et al., 2024; Worth et al., 2024), illustrating the importance of accurate methods and tools for (semi-)automated sensitive data protection, particularly on open portals.

Most work on sensitive data detection methods, however, focuses on detecting person-level data, particularly personal identifiable information (PII) (Kužina et al., 2023; Subramani et al., 2023; Akhtar et al., 2024). For unstructured data, PII detection relies on named entity recognition (Subramani et al., 2023); similar approaches exist for tabular data (Raman, 2001; Hulsebos et al., 2019), which is the focus in this paper. Yet *sensitivity* extends far beyond explicit PII. Kober et al. (2023) propose a taxonomy of always-, combination-, context-, and value-sensitive data. For instance, an `address` may be harmless in the context of a public organizational website but expose a private address in another document, causing naïve type-based detection yield false positives. Likewise, hospital `geo-coordinates` may be harmless in one region but sensitive in conflict zones (OCHA, 2025), calling for contextualization beyond detecting personal-sensitive data. These limitations call for a more nuanced perspective and framework for assessing sensitive data.

*Correspondence to madelon@cwi.nl

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 41354–41364
October 24-29, 2026 ©2026 Association for Computational Linguistics

In this paper, we revisit the concept of sensitive data detection, arguing for a more contextual perspective. In our ConSenser framework, we particularly focus on two aspects of contextual sensitive data detection. First, *type contextualization* which takes a more nuanced view beyond the type of values (e.g. `addresses`) by taking the broader context of the data into account. We introduce the technical "detect-then-reflect" mechanism which first detects the type of the values at hand, then reflects on their actual sensitivity based on contextual signals within the document or dataset. Experiments with real-world datasets containing personal identifiable information (PII) show that this mechanism, instrumented by language models, yields a strong improvement in precision while maintaining high recall. Second, we posit *domain contextualization* which addresses sensitivity that depends on external domain-specific factors, beyond merely static types such as PII. We introduce the "retrieve-then-detect" mechanism which first retrieves relevant domain-specific information, such as applicable data policies, to then ground the data sensitivity detection in using a language model. Evaluations illustrate the potential of this mechanism, achieving high recall and improved consistency for detecting non-personal sensitive data. Importantly, the context-grounded domain-specific explanations are deemed highly useful by experts in humanitarian data assessment in a case-study.

2 Revisiting Sensitive Data Detection

The notion of "sensitive data" is multifaceted, spanning legal, theoretical, and domain-specific views. Legal frameworks often equate sensitivity with personal identifiable information. The GDPR defines personal data broadly as "any information relating to an identified or identifiable natural person," specifying sensitive categories like health, ethnicity, or politics requiring extra protection (European Parliament and Council of the European Union, 2016). Similarly, HIPAA defines identifiers that make health information "Protected Health Information (PHI)" when linked to individuals (Centers for Medicare & Medicaid Services, 1996).

Yet, sensitivity is often contextual: seemingly harmless data can become sensitive when combined, repurposed, or shared across contexts (Malkin, 2022).Theoretical perspectives deepen this nuance. Nissenbaum's Contextual Integrity frames sensitivity as the appropriateness of information flows within social norms (Nissenbaum, 2004). Building on this, researchers advocate context-grounded definitions (Malkin, 2022).Kober et al. (2023) distinguish: (1) always-sensitive data (e.g., personal identifiers); (2) sensitivity by linking data; (3) sensitivity by use; and (4) sensitivity from specific values. This echoes that not all personal is equally sensitive, and not all sensitive data is on individuals. To detect sensitive data, of any kind, our framework builds on this conceptual notion of data sensitivity (Def. 1): data protection methods must ensure coverage beyond personal- and static sensitivity definitions.

Definition 1 (Contextual Sensitive Data)**.** *Data whose sensitivity depends on external factors and requires protection due to how, by whom, and in what context it can be misused.*

Informed by this *contextual* definition of sensitive data, we formalize the problem of contextual sensitive data detection as follows:

Problem Statement 1 (Contextual Sensitive Data Detection)**.** *Given a sample of similar-typed values $V = v_1, v_2, \ldots, v_N$ from document $\mathcal{D}$ (e.g., cell values in a table column, extracted similar-typed entities from a text document), a set of internal context factors C_I within $\mathcal{D}$ (e.g., surrounding values, structure), and external context factors C_E outside $\mathcal{D}$ (e.g., specified rules, time/location), the objective of contextual sensitive data detection is to estimate the probability $P(V = sensitive | C_E, C_I) \in [0,1]$, and, based on this probability, classify the sensitivity of V into one of the predefined sensitivity levels $s_1, s_2, \ldots, s_M$, where each s_i corresponds to a distinct sensitivity level (e.g., "sensitive", "medium sensitive", "non-sensitive", etc.).*

3 Contextual Sensitive Data Detection

Protecting sensitive data generally follows a three-step procedure: ① specifying sensitive data, ② detecting sensitive data, ③ remedying data sensitivity. In this work, we focus on the second step, which has received little attention in research. Existing methods for detecting sensitive data suffer from two key limitations to address the contextual nature of sensitive data:

1. *Insufficient specificity*: sensitive data detection methods treat all values of types of interest as equally sensitive, while the subject of the value implies its sensitivity. This indistinctness leads to low precision hence high false positive rates.
2. *Insufficient coverage*: sensitive data detection methods typically focus on personal-level data, particularly PII, and do not generalize to domain-specific sensitive data. The lack of coverage leads to low recall of methods, hence leaving sensitive data unprotected.

To address these challenges in contextual sensitive data detection, we posit two contextual aspects that matter in order to assess the contextual sensitivity of data: *type contextualization* and *domain contextualization*. In our contextual sensitivity detection framework, termed ConSenser, we introduce two mechanisms to incorporate the contextual aspects in sensitive data detection: the *detect-then-reflect* mechanism for type-based contextualization and the *retrieve-then-detect* mechanism for domain contextualization. In what follows, we specifically focus on tabular data but these mechanisms extend to other modalities, such as free-form text documents.

3.1 Type Contextualization

Following existing standards and protocols (European Parliament and Council of the European Union, 2016; Centers for Medicare & Medicaid Services, 1996), sensitive data detection methods often focus on detecting data *types* reflecting personal sensitive data (e.g., names, addresses) (Google, 2018). But the appearance of a name or address does not immediately make a data value sensitive. For example, an address may reflect the address of a person or organization, each having significantly different consequences, hence different sensitivity levels. We term this *type contextualization*: whether a data value corresponding to a sensitivity type of interest (e.g. `name`) truly is sensitive requires considering the context within the document or dataset in which it appears. Type contextualization improves specificity as potential sensitive values are only considered sensitive based on the broader context of the value.

The *detect-then-reflect* mechanism Existing methods classify sensitive data based on signals indicating a type of interest, e.g. `names`, `addresses`, or `phone numbers`. If a value matches this type, it is flagged as sensitive. Type contextualization employs a more nuanced assessment, for which we introduce the two-stage detect-then-reflect mechanism for detecting type-informed sensitive data. Algorithm 1 illustrates this staged procedure for binary sensitivity classification.

In the initial detection step, the potential sensitivity of a set of values is estimated by first classifying the values (V_i) corresponding to sensitive types of interest specified upfront (T_{sens}). This step prioritizes recall by capturing all values that may carry data of the sensitive type. The purpose of the reflection step is to estimate the actual sensitivity of the values (p_{sens}) given their detected sensitive type (t_i) and internal context of the full document (C_I), to finally turn into a discrete sensitivity label (s_i). This step is crucial for mitigating false positives from the type detection stage by incorporating inter-value relationships and document semantics, to filter out values of sensitive types that are not actually sensitive.

Algorithm 1 Type Contextualization

Require: Set of values $V_i = \{v_1, \ldots, v_N\}$ from document D, internal context C_I, sensitive types T_{sens}, sensitivity threshold τ

$t_i \leftarrow \textsc{DetectType}(V_i)$
if $t_i \in T_{\text{sens}}$ **then**
  $p_{\text{sens}} \leftarrow \textsc{Estimate}\ P(V_i = \text{sensitive} \mid t_i, C_I)$
  $s_i \leftarrow (p_{\text{sens}} > \tau)$
else
  $s_i \leftarrow$ false
end if

Tabular Data case. Figure 1 illustrates the detect-then-reflect mechanism for type contextualization on a concrete example with tabular data. Here, table columns are classified to correspond to any sensitive types such as `address` and `phone number` based on the column name and data values. For columns of sensitive types, the *reflection* step assesses whether the column is sensitive in context of the full table context, i.e., the table name as well as column names and values of surrounding columns.

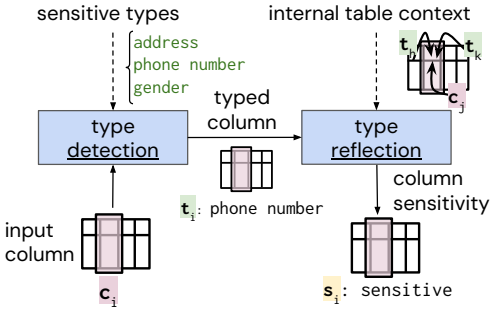


Figure 1: *Type contextualization* for contextually detecting sensitive table columns using the *detect-then-reflect* mechanism to increase precision. This mechanism entails: ① *detecting* potentially sensitive columns based on their (semantic) type, ② *reflecting* on the actual sensitivity of the column within the internal context of the entire table.

3.2 Domain Contextualization

Besides a more fine-grained contextual notion of the sensitivity of certain values, we also argue for a more coarse-grained contextual perspective. Following our notion of contextual sensitive data (Def. 1), we pose that data protection should account for risks beyond identity exposure, for example, harming vulnerable populations or organizations hence consider the data

domain, from humanitarian aid to national security. We refer to this broader perspective on data sensitivity as *domain contextualization*: sensitivity emerges from the dataset's nature and potential misuse, which can arise from external factors. As such, domain contextualization aims at higher detection coverage by grounding the sensitivity assessment in domain-specific context, beyond generic and static predefined types.

The *retrieve-then-detect* mechanism While type-based data sensitivity can often be inferred from internal document context, domain-specific sensitivity typically depends on contextual factors external to the data itself. For domain contextualization, we introduce the *retrieve-then-detect* mechanism (Alg. 2).

Algorithm 2 Domain Contextualization

Require: Set of values $V_i = \{v_1, \ldots, v_N\}$ from document D, external context C_E, sensitivity threshold τ

1: $C_i^{\text{relevant}} \leftarrow \textsc{RetrieveContext}(V_i, C_E)$
2: $p_{\text{sens}} \leftarrow \textsc{Estimate}\ P(V_i = \text{sensitive} \mid C_i^{\text{relevant}})$
3: $s_i \leftarrow (p_{\text{sens}} > \tau)$

Unlike type-informed detection, based on a predefined set of sensitive types, domain-specific sensitive data has no fixed ontology. Instead, the retrieve-then-detect mechanism considers external contextual information, such as rule-based heuristics specified in data governance documents or geopolitical news items, in the detection process. In the retrieval stage, external context is selected from a corpus of domain-specific documents or other external data sources (C_E) based on the relevance to the potential sensitive values at hand (V_i). In the detection stage, the sensitivity of the values at hand (s_i) is assessed given the retrieved relevant external context (C_i^{relevant}).

Tabular Data case. Figure 2 illustrates the retrieve-then-detect mechanism for tabular data. For a given column (or table), the retrieval component fetches information, such as data policy rules or geopolitical information, from external documents or other data sources that are relevant for the given input table. The detection step then assesses whether a table (column), or combination of columns, contains data that should be considered sensitive given the retrieved relevant context.

4 Experimental Setup

Here, we outline the method implementation details, baselines, datasets, and metrics to evaluate our proposed framework for contextually sensitive data detection. Code and datasets can be found at https://github.com/trl-lab/sensitive-data-detection.

4.1 Implementation of the Mechanisms

The core components of the mechanisms in the ConSenser framework are type detection, reflection, and retrieval. Further details of the implementation are provided below.

Type-contextualization In the detection step, given the well-defined nature of PII types, we parse the problem as a multiclass classification task over a taxonomy of personal identifiable types (e.g., `email_address`, `generic_identifier`) derived from GDPR (European Parliament and Council of the European Union, 2016) and UN OCHA guidelines (UN OCHA Centre for Humanitarian Data, 2021) (see the code repository for a complete list). We devise a language model to classify the type of a table column (or return `None`) (Kayali et al., 2024; Shen et al., 2025), by prompting it with the column name and five values. We consider zero-shot and fine-tuned approaches. The zero-shot method leverages the model's inherent ability to follow instructions, while the fine-tuned approach enhances performance by training the models on a synthetic dataset of 1,000 labeled columns. Fine-tuning is done using Low-Rank Adaptation (Hu et al., 2021) for parameter efficiency. The models are trained for a single epoch using an 80/20 train-validation split.

In the reflection phase, the LLM re-evaluates each candidate column corresponding to types of interest within the context of the entire table. We provide the LLM with a markdown-formatted representation of the full table, including headers and five sample rows per column, along with the specific type detected for the target column and, if applicable, the types of other columns which are detected as PII types. The LLM

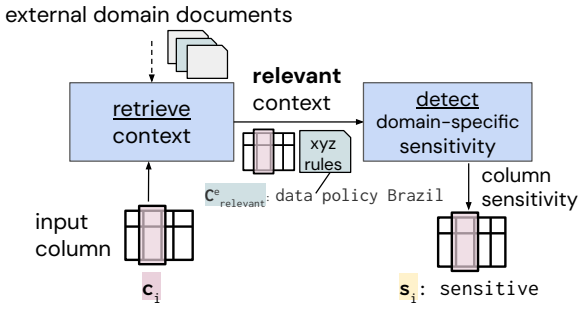


Figure 2: The *retrieve-then-detect* mechanism for detecting sensitive data that requires domain contextualization. This mechanism entails: ① retrieving relevant contextual information such as data policy rules, ② detecting sensitive data based on the table and the retrieved relevant context.

is then instructed to assign one of three contextual sensitivity levels: `non_sensitive`, referring to information that cannot identify a person (e.g., aggregate data, organization names); `moderate_sensitive`, referring to data that could potentially identify a person when combined with other attributes (e.g., demographics, partial information); and `high_sensitive`, referring to data that definitively identifies a person (e.g., full name, email, national ID).

Domain-contextualization Given the domain-specific nature of this mechanism, we conduct a case study with humanitarian data experts from the UN OCHA Humanitarian Data Centre using humanitarian crisis datasets from the Humanitarian Data Exchange (HDX) (United Nations, 2014). For the retrieval step, we aim to identify relevant heuristics indicating sensitivity rules from domain-specific data governance documents. The system retrieves relevant policy fragments from Information Sharing Protocols (ISPs) based on the dataset's geographical context (e.g., country). We use GPT-4o-mini to extract and index the specific sensitivity rules from these documents. If no region-specific ISP is identified, a default ISP template based on cross-country consensus is used.

In the detection phase, the LLM is provided with a curated prompt containing table context (headers and five representative rows), relevant ISP extracts containing explicit instructions to classify each column's sensitivity according to the ISP guidelines while justifying its decision by directly citing the relevant policy passages. Based on the policy definitions, the model assigns one of four sensitivity levels, `non_sensitive`, `moderate_sensitive`, `high_sensitive`, or `severe_sensitive`. This mechanism ensures that sensitivity assessments are grounded in explicit domain policies, thereby enhancing both interpretability and practical usability for domain experts.

4.2 Model Selection

We evaluate closed-source and open-source models to understand trade-offs between these options, if any. While one of the goals of this work is to prevent sensitive data from leaking into proprietary LLMs, these LLMs are typically available under privacy-preserving protocols at some cost which can then provide a viable low-maintenance solution. From closed-source options, we select the GPT-4o-mini (OpenAI et al., 2024) model for its good performance and cost trade-off. We focus on open-source LLMs that enable local deployment, for which we select the Gemma family (2 and 3) (Team, 2025; Team et al., 2024) and Qwen3 in 8B and 14B sizes (Yang et al., 2025). Qwen3 14B was selected given the strong performance of its predecessor Qwen2.5's exceptional performance on benchmarks such as MMLU Pro [1], demonstrating strong reasoning capabilities and robust general understanding. Furthermore, Gemma2 9B and Gemma3 12B were included to strike a balance between computational efficiency and performance, and their support for quantized deployment. We also include versions of Qwen3 8B and Gemma2 9B that are fine-tuned on a synthetic dataset of 1K table columns hand-labeled with 27 PII types, indicated with "FT" suffix.

4.3 Baselines

We compare our methods with two common baselines for sensitive data detection, which are also widely adopted in analyses of sensitive data in public datasets (Larson et al., 2024; Subramani et al., 2023). We include Google Cloud DLP (Google Cloud, 2018), a rule-based system that combines built-in *infoTypes*, predefined detectors for common Personal Identifiable Information (PII) types, which are associated custom-defined dictionaries and regular expressions. For our experiments, we configure the minimum likelihood threshold to "Likely". The second baseline is Microsoft Presidio, an open-source tool that leverages regular expressions, pattern-based rules, context windows, and ML-based named entity recognition to detect PII types. To enable consistent evaluation, each Presidio type has been mapped to a corresponding category in our unified taxonomy (this mapping can be found in the code repository). Both systems output column-level predictions and are widely adopted in enterprise settings. However, neither tool supports contextual reflection or table-wide reasoning; any match to a known PII pattern is automatically treated as sensitive.

4.4 Datasets

To evaluate the contextual sensitive data detection mechanisms we use a combination of synthetic and real-world datasets, to maintain ground-truth validity for sensitive data detection in generic context and the domain-specific context of humanitarian datasets.

- **Type contextualization with sensitive PII**: We extracted 66 anonymized tables, with in total 2,061 columns, from the GitTables corpus (Hulsebos et al., 2023) representing a variety of real-world table schemas. Each column was manually annotated

[1] https://huggingface.co/spaces/TIGER-Lab/MMLU-Pro

with (i) its PII type and (ii) its contextual sensitivity level (non-sensitive, moderate, or highly sensitive).

- **Domain contextualization with humanitarian data**: To evaluate the retrieve-then-detect mechanism for domain-specific sensitive data, we conduct a case study with humanitarian experts from the UN OCHA Humanitarian Data Centre. We collect 23 tabular datasets, with in total 326 columns, resembling typical humanitarian datasets (e.g., for displacement tracking, public health monitoring) for different regions. With rich input from humanitarian experts to ensure realistic properties, we synthesize 9 tabular datasets with one or more actual contextually sensitive data in line with region-specific Information Sharing Protocols (ISPs). Two humanitarian data experts annotated columns (100 in total, of which 70 contain sensitive data) with a sensitivity level (low, moderate, high, severe). The other 14 tabular datasets are similar but real tabular datasets from the UN's Humanitarian Data Exchange[2] and have been manually curated as challenging datasets without actual sensitive data, to evaluate the false positive rate.

Evaluation granularity For evaluation, sensitivity levels are binarized: *low* is treated as non-sensitive, while *moderate*, *high*, and *severe* are grouped as sensitive. This aligns with operational practice in humanitarian contexts where any moderate or higher sensitivity triggers safeguards, consistent with UN OCHA guidance that such data warrants protective measures (OCHA, 2025). For domain contextualization, we assess performance on table-level to align with the ISP's coarse-grained data policy instructions, despite having column-level annotations. Tables are labeled sensitive if one of its columns is annotated as sensitive (moderate, high or severe).

4.5 Metrics

We evaluate accuracy along three axes: PII detection, contextual reflection for PII, and non-PII sensitivity detection. To account for class imbalance, we report (support-)weighted averages. Performance is measured at column and table level. For type contextualization we evaluate column-level accuracy, PII detection is treated as a multiclass task over 27 PII types (plus a "None" category), while sensitivity reflection reduces to a binary decision between sensitive (moderate/high) and non-sensitive (low). For domain contextualization we evaluate table-level accuracy: a table is classified as sensitive if it contains any moderate/high/severe sensitive information, resulting in a binary task (sensitive vs. not). Precision, recall, and F1 are used as evaluation metrics.

Besides accuracy metrics, we report the average runtime per column to assess deployment considerations.

[2] https://data.humdata.org/

5 Results

We present a comprehensive evaluation of the mechanisms in the ConSenser framework for contextually sensitive data detection in tabular datasets, through type- and domain-contextualization. The Type Contextualization experiments, instrumented by the detect-then-reflect mechanism, are focused on detecting personal-identifiable information (PII) and Domain Contextualization, instrumented by the retrieve-then-detect mechanism, focuses on domain-specific sensitivity through a case-study with humanitarian datasets.

5.1 Type contextualization improves specificity.

First, we assess the *detect-then-reflect* mechanism for detecting sensitive data through type contextualization. We report on baselines representing existing state-of-the-art tools for detecting sensitive data which automatically treat all columns of interest types as sensitive above the line. Below the line, we report the contextual LLM-assisted method that detects columns of the types of interest which are directly labeled as sensitive *without reflection* (left column), and are labeled as sensitive only if this is the output of the *reflection* step (right column).

Table 1: Type contextualization for PII sensitive data detection, with baselines and the *detect-then-reflect* mechanism. The LLMs are shown with and without the reflection step. We observe that reflection consistently improves precision and F1.

system/model	w/o reflection			w/ reflection		
	prec.	rec.	F1	prec.	rec.	F1
Google DLP	0.53	0.63	0.58	–	–	–
Presidio	0.52	0.62	0.57	–	–	–
GPT-4o-mini	**0.86**	0.64	0.73	**0.94**	0.63	0.76
Gemma 2 9B	0.74	0.82	0.78	0.80	0.79	0.80
Gemma 3 12B	0.49	0.94	0.64	0.75	0.81	0.78
Qwen3 8B	0.74	0.87	**0.80**	0.75	0.87	0.80
Qwen3 14B	0.57	**0.97**	0.71	0.73	**0.94**	0.82
Qwen3 8B FT -> GPT-4o-mini	–	–	–	0.90	0.86	**0.88**

Without reflection, LLMs mark any detected PII-type column as sensitive, leading to high recall but lower precision; Gemma 3 12B and Qwen3 14B

reach a recall of 0.94 and 0.97 but precision of only 0.49 and 0.57 respectively (Table 1). GPT-4o-mini achieves higher precision, 0.86, at lower recall of 0.64. With reflection using full-table context, substantially improves precision across models, e.g., Qwen3 14B increases to 0.73 precision with 0.94 recall, trading a small decrease in recall for a significant gain on precision. A modular pipeline combining Qwen3 8B for type detection with GPT-4o-mini for reflection achieves F1 0.88, but is closely matched by Qwen3 14B which yields higher recall (0.94 compared to 0.86). These results illustrate the stark difference compared to existing baselines reported in the upper left quadrant, which suffer from insufficient recall as well as precision. Comparing the detect-then-reflect mechanism powered by open-source LLMs with commonly used pattern-matching tools DLP and Presidio, we observe a significant improvement in both precision and recall, addressing the key limitations of existing tools for deployment in practice. Overall, the *detect-then-reflect* mechanism corrects the naive assumption that all columns of sensitive *types* are indeed always sensitive, boosting precision while maintaining recall.

Ablation: reflection without detection. To investigate the value of the type-detection step, and represent the performance of a common baseline method (Dubniczky et al., 2026), we also task language models with identifying sensitive columns using directly the full table context without first detecting the presence of columns of specific types (Table 1). The results in Table 2 illustrate that LLMs yield good performance for sensitive columns using only table context without prior type detection, but generally yield significant improvement across models when explicit types are detected first. Qwen3 14B achieves the highest F1 (0.82) followed by Gemma 2 9B (F1 0.81). GPT-4o-mini excels in precision (0.96) but has low coverage with a recall of 0.56. Comparing the performance with the detect-then-reflect pipeline in Table 1, we observe that the reflection-only approach, without type detection, yields particularly a lower coverage, therefore leaving more sensitive data undetected. For example, looking closer into the performance of the best performing models, the recall of Qwen3 14B drops from 0.94 to 0.89, and Gemma2 9B from 0.79 to 0.75. These results indicate that LLMs can reason about sensitivity from the table alone, but the full *detect-then-reflect* mechanism yields preferable trade-off in precision and recall.

Table 2: Reflection-only performance without prior type detection, for identifying contextually sensitive columns. Evaluated across models using precision, recall, and F1 scores.

model	precision	recall	F1
GPT-4o-mini	**0.96**	0.56	0.71
Gemma 2 9B	0.88	0.75	0.81
Qwen3 8B	0.74	0.78	0.76
Qwen3 14B	0.76	**0.89**	**0.82**

5.2 Domain contextualization provides coverage and context-grounded explainability.

Shifting towards contextual sensitive data which cannot be assessed through a fixed set of types, we evaluate the *retrieve-then-detect* mechanism on a combined set of synthetic and real humanitarian tables from the HDX platform. Without external domain knowledge, all models exhibit a highly conservative bias, achieving perfect recall but all with precision lower than 0.56 (Table 3; baseline). Integrating retrieved domain-specific context, i.e., sensitivity guidelines obtained from information sharing protocols (ISP's) for humanitarian data sharing, into the input for reflection by the language model, consistently improves precision. GPT-4o-mini's precision increases from 0.47 to 0.69 (F1=0.82), and Qwen3 14B's from 0.56 to 0.64 (F1=0.78), both maintaining perfect recall (1). Table 3 summarizes the results on the 23 datasets humanitarian datasets.

Table 3: Table-level classification performance with and without context-grounding in humanitarian data sensitivity rules. The retrieved rules improve precision while maintaining high recall.

system/model	w/o domain knowledge			w/ domain knowledge		
	prec.	rec.	F1	prec.	rec.	F1
Baseline: all sensitive	0.375	**1.000**	0.545	–	–	–
GPT-4o-mini	0.474	**1.000**	0.643	0.692	**1.000**	**0.818**
Gemma 2 9B	0.375	**1.000**	0.545	0.429	**1.000**	0.600
Gemma 3 12B	0.529	**1.000**	0.692	0.500	**1.000**	0.667
Qwen3 8B	**0.562**	**1.000**	**0.720**	**0.778**	0.778	0.778
Qwen3 14B	**0.562**	**1.000**	**0.720**	0.643	**1.000**	0.783

Qualitative analysis surfaces that the context retrieval mechanism enables models to produce interpretable and consistent justifications, thereby enhancing the transparency and auditability of the sensitivity assessments. The model outputs containing sensitivity classifications and explanations were reviewed by two humanitarian data quality experts, who considered the context-grounded explanations accurate, and highly informative for guiding and

standardizing human-assessment. The below excerpt shows such context-grounded LLM assessment on a table titled "Reported location on NFIs suppliers" with (1) a description of the data at hand, along with (2) the LLM's sensitivity assessment, and (3) explanation grounded in retrieved ISP rules.

```
1. The column "nfi_supplier_loc-value-
from_Iran" indicates the origin of NFIs
suppliers.

2. This information, even with country-level
disaggregation, doesn't reveal personal or
sensitive data. It relates to the location
of suppliers, which is generally not consid-
ered sensitive.

3. It aligns with the ISP's guideline on
"Facility data (e.g., health, education, water
points) at national or regional level, unless
explicitly restricted by clusters." Since
this column is about locations of suppliers
and not about individuals or specific opera-
tions, it is categorized as non-sensitive.
```

5.3 Error analysis

Type contextualization On data with PII, we particularly observed misclassification of columns of generic types, as well as explicit personal types. For example, the "GENERIC_ID" type is well detected, but classified as non-sensitive by most LLMs causing very low recall. For types more explicitly relating to personal information (like names), columns are often classified as sensitive in reflection, while not always annotated as such. This highlights the potential for more specific context- and type-specific instructions during reflection.

Domain contextualization In our experiments with humanitarian data, we observed mainly false positives for LLMs, which generally appear cautious with humanitarian data. On these false positives, analysis surfaces that the Qwen model has most confusion around borderline cases that are annotated as non-sensitive but classified as medium-sensitive, while gpt-4o-mini deems most of these tables highly sensitive (75%). This illustrates the need for more specific instructions for borderline cases.

5.4 Latency of contextual sensitive data detection.

Table 4 gives an overview of wall-clock latency measurements of the end-to-end pipeline averaged over all columns, as well as cost indications and memory measurements for local models. Despite the more elaborate mechanisms for contextualized sensitive data detection, the latency delta between the contextual sensitive LM-powered pipelines versus Google DLP (which only conducts type-based sensitivity detection) might be acceptable given the significant accuracy gain that the introduced mechanisms provide. If efficiency is key, smaller models like Qwen3 8B can yield an favorable trade-off, whereas larger or more costly models for more sensitive scenarios can be feasible for smaller-scale evaluations.

Table 4: Comparison of pipeline latency per column and resource cost across different methods and models.

tool/model	total pipeline (s/col)	cost or GPU memory
Google DLP[3]	0.16	subscription
GPT-4o-mini	1.20	$0.05 per table
Gemma 2 9B	2.89	7.8 GB
Gemma 3 12B	2.94	8.0 GB
Qwen3 8B	0.46	8.0 GB
Qwen3 14B	1.62	11.4 GB

6 Related Work

Existing methods for detecting sensitive data concentrate on detecting PII types, or otherwise static sensitive types. To this end, industry tools like Microsoft Presidio (Microsoft, 2018) and Google Cloud DLP (Google Cloud, 2018) detect correspondences between values and structured types, such as email addresses and phone numbers, based on rule-based heuristics and syntactic pattern matching. These tools are also widely adopted in recent research on data sensitivity detection and analysis (Larson et al., 2024; Subramani et al., 2023). Recent research has expanded the type detection scope by utilizing large neural models to learn mappings between high-dimensional semantic representations of values, e.g. table columns, and their semantic types (Hulsebos et al., 2019; Kužina et al., 2023).

Language models have also emerged as powerful tools for sensitive data detection, as particularly shown for PII detection, outperforming both rule-based and traditional machine learning methods in accuracy and adaptability (Shen et al., 2025; Yang et al., 2023; Dubniczky et al., 2026). However, while language models excel at identifying known PII types, they struggle with assessing *contextual* sensitivity, such as domain-specific sensitive types, unless explicitly provided with the relevant operational context. In this work, we address this gap: our contextual sensitive data detection framework addresses both personal and non-personal sensitive data by considering internal and external context factors.

[3]For Google DLP, we only measure type-based sensitivity detection as it is not suitable for domain-specific sensitivity.

7 Conclusion

With the rise of open data portals to democratize data access, comes the risk of exposing sensitive data to malicious actors or language models fueled by public data. Protecting sensitive data is therefore key, but methods for this are limited: *they are too specific to capture all sensitive data, and too generic to only suppress data that is actually sensitive*. We argue for *contextualizing* sensitive data detection along two dimensions. First, *type contextualization*, which considers fine-grained aspects beyond static sensitivity types. Our detect-then-reflect mechanism shows that first detecting if data values belong to a sensitive type, then reflecting on their true sensitivity, increases recall over existing tools with up to 49%. Second, we introduce *domain contextualization* which considers domain-specific factors external to the data in order to assess its sensitivity. A case study with humanitarian datasets highlights that contextualizing the sensitivity assessment in relevant domain-specific rules through a retrieval stage, extends the type coverage and enables domain-grounded explanations. We believe that this work provides a stepping stone in the important direction of contextual sensitive data detection.

Limitations

Evaluation coverage In our humanitarian case study, we show that domain contextualization leads to higher precision while preserving coverage, and also provides useful grounded explanations to data quality reviewers. The limited availability of labeled data in this case study, might affect generalization of the insights for domain-contextualized sensitive data detection. Moreover, while the conceptual framework and mechanisms are modality agnostic (Section 3), the lack of datasets with labeled sensitivity and the data setting of the humanitarian case study have resulted in evaluations for tabular data only, for which we constructed and publish hand-labeled datasets. We call for more diverse labeled datasets, to advance this important problem, and evaluate the framework and mechanisms for different modalities, languages, and domains.

Modality coverage Despite the focus of our implementation and evaluation on tabular data. The proposed mechanisms easily extend to other modalities, such as free-form text data: 1) type-informed sensitivity in free-form text can be implemented efficiently by concatenating a decomposition component with a more lightweight learned type predictor (Hulsebos et al., 2019) and performing reflection in batches, 2) for domain-contextualization of free-form text, a hierarchical reflection procedure could be efficient. For example, starting with assessing high-level segments of the text (document-level, paragraph-level, sentence-level, etc (Lin et al., 2025)), and incrementally narrowing down.

Sensitivity specification dependency The presented mechanisms for detecting sensitive data can enable automation but also assist humans in data reviewing by providing context-grounded explanations with explicit citations to relevant sensitivity rules. For successful adoption in practice, however, the mechanisms depend on having access to reasonable sensitivity instructions in the form of 1) fine-grained semantic types of data that are deemed sensitive for type contextualization, and 2) reasonably specified assessment rules in domain-specific documents or (web) data sources.

Data access delay and further automation A potential negative consequence of automated data sensitivity detection is that critical datasets might be blocked that are time-sensitive and could be instrumental for solving societal problems, for example, in humanitarian domains. The mechanisms could be tailored around the desired precision/recall trade-off to reduce false alarms. In this work, we only present mechanisms for detecting contextual data sensitivity, and not for automated remedies. In practice, a report could be generated based on existing cases to instruct the user on how to remedy the sensitive data, or potentially even automatically remedy this, which is also an important direction for future research.

References

Mubashara Akhtar, Omar Benjelloun, Costanza Conforti, Luca Foschini, Pieter Gijsbers, Joan Giner-Miguelez, Sujata Goswami, Nitisha Jain, Michalis Karamousadakis, Satyapriya Krishna, Michael Kuchnik, Sylvain Lesage, Quentin Lhoest, Pierre Marcenac, Manil Maskey, Peter Mattson, Luis Oala, Hamidah Oderinwale, Pierre Ruyssen, and 12 others. 2024. Croissant: A metadata format for ml-ready datasets. In *Advances in Neural Information Processing Systems*, volume 37, pages 82133–82148. Curran Associates, Inc.

Christine L Borgman and Paul Groth. 2025. From data creator to data reuser: Distance matters. *Harvard Data Science Review*.

Dan Brickley, Matthew Burgess, and Natasha Noy. 2019. Google dataset search: Building a search engine for datasets in an open web ecosystem. In *The world wide web conference*, pages 1365–1375.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam

Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, and 1 others. 2021. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650.

Centers for Medicare & Medicaid Services. 1996. The Health Insurance Portability and Accountability Act of 1996 (HIPAA). Online at http://www.cms.hhs.gov/hipaa/.

Richard A. Dubniczky, Bertalan Borsos, Tamas Bisztray, and Norbert Tihanyi. 2026. You have been LaTeXpOsEd: A Large-Scale systematic analysis of information leakage in preprint archives using large language models. In *20th USENIX WOOT Conference on Offensive Technologies (WOOT 26)*, pages 247–261, Baltimore, MD. USENIX Association.

European Parliament and Council of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

Google. 2018. Data loss prevention.

Google Cloud. 2018. Cloud Data Loss Prevention (DLP) API – General Availability. Web documentation. Accessed: 2025-07-22.

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. *Preprint*, arXiv:2106.09685.

Madelon Hulsebos, Çagatay Demiralp, and Paul Groth. 2023. Gittables: A large-scale corpus of relational tables. *Proceedings of the ACM on Management of Data*, 1(1):1–17.

Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. 2019. Sherlock: A deep learning approach to semantic data type detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM.

Moe Kayali, Anton Lykov, Ilias Fountalis, Nikolaos Vasiloglou, Dan Olteanu, and Dan Suciu. 2024. Chorus: Foundation models for unified data discovery and exploration. *Proceedings of the VLDB Endowment*, 17(8):2104–2114.

Maria Kober, Jordan Samhi, Steven Arzt, Tegawendé F Bissyandé, and Jacques Klein. 2023. Sensitive and personal data: What exactly are you talking about? In *2023 IEEE/ACM 10th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, pages 70–74. IEEE.

Vjeko Kužina, Ana-Marija Petric, Marko Barišić, and Alan Jović. 2023. Cassed: Context-based approach for structured sensitive data detection. *Expert Systems with Applications*, 223:119924.

Stefan Larson, Nicole Cornehl Lima, Santiago Pedroza Diaz, Amogh Manoj Joshi, Siddharth Betala, Jamiu Tunde Suleiman, Yash Mathur, Kaushal Kumar Prajapati, Ramla Alakraa, Junjie Shen, Temi Okotore, and Kevin Leach. 2024. De-identification of sensitive personal data in datasets derived from IIT-CDIP. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21494–21505, Miami, Florida, USA. Association for Computational Linguistics.

Yiming Lin, Madelon Hulsebos, Ruiying Ma, Shreya Shankar, Sepanta Zeighami, Aditya G Parameswaran, and Eugene Wu. 2025. Querying templatized document collections with large language models. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pages 2422–2435. IEEE.

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. 2023. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363. IEEE.

Nathan Malkin. 2022. Contextual integrity, explained: A more usable privacy definition. *IEEE Security & Privacy*, 21(1):58–65.

Microsoft. 2018. Microsoft presidio.

Helen Nissenbaum. 2004. Privacy as contextual integrity. *Wash. L. Rev.*, 79:119.

UN OCHA. 2025. Data responsibility guidelines 2025.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Vijayshankar Raman. 2001. Potter's wheel: An interactive data cleaning system. In *VLDB*, volume 1, pages 381–390.

Y. Shen, Z. Ji, J. Lin, and K. R. Koedginer. 2025. Enhancing the de-identification of personally identifiable information in educational data. *Preprint*, arXiv:2501.09765.

Nishant Subramani, Sasha Luccioni, Jesse Dodge, and Margaret Mitchell. 2023. Detecting personal information in training corpora: an analysis. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 208–220, Toronto, Canada. Association for Computational Linguistics.

Gemma Team. 2025. Gemma 3.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

UN OCHA Centre for Humanitarian Data. 2021. The ocha data responsibility guidelines. Blog post.

United Nations. 2014. The humanitarian data exchange. https://data.humdata.org/. Accessed: 2015-05-31.

Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, and 1 others. 2016. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9.

Sophia Worth, Ben Snaith, Arunav Das, Gefion Thuermer, and Elena Simperl. 2024. Ai data transparency: an exploration through the lens of ai incidents. *arXiv preprint arXiv:2409.03307*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Jianliang Yang, Xiya Zhang, Kai Liang, and Yuenan Liu. 2023. Exploring the application of large language models in detecting and protecting personally identifiable information in archival data: A comprehensive study*. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2116–2123.

Xinyu Yang, Weixin Liang, and James Zou. 2024. Navigating dataset documentations in ai: A large-scale analysis of dataset cards on hugging face. *Preprint*, arXiv:2401.13822.

Jiayao Zhang, Yuran Bi, Mengye Cheng, Jinfei Liu, Kui Ren, Qiheng Sun, Yihang Wu, Yang Cao, Raul Castro Fernandez, Haifeng Xu, and 1 others. 2024. A survey on data markets. *arXiv preprint arXiv:2411.07267*.