applied sciences

MDPI

Article

Addressing Memorization and Aggregation Risks in AI: A Knowledge Graph Approach to Privacy †

Jinhui Zuo [1] and Seok-Won Lee [1,2,*]

[1] Department of Artificial Intelligence, Ajou University, Suwon-si 16499, Republic of Korea; pphui8@ajou.ac.kr
[2] Department of Software and Computer Engineering, Ajou University, Suwon-si 16499, Republic of Korea
* Correspondence: leesw@ajou.ac.kr
† This paper is an extended version published in the Conference paper: PrivGraph: Modeling Personal Privacy Information Using Knowledge Graph. In Proccedings of the 2025 The 25th IEEE International Conference on Software Quality, Reliability and Security (QRS), Hangzhou, China, 16–20 July 2025.

Abstract

Recent studies have shown that AI models can memorize specific data records, resulting in sensitive data exposure through model access. Current privacy-enhancing technologies often overlook the crucial, context-dependent nature of privacy risk as they largely fail to account for the inherent relationships and complex interactions between data records, leading to high risks associated with memorization and potential data aggregation. Our research first investigates two key factors influencing AI privacy risks: implicit connections and data redundancy. These experiments have shown that AI models learn subtle links between private data, even when they are discretely distributed. To address the privacy issue, we introduce PrivGraph, a hierarchically structured knowledge graph for modeling and aggregating private information. Based on PrivGraph, we introduce the Sensitivity Level Factor (SLF) to quantify the degree to which an individual's private information is embedded in the data. In addition, we propose a PrivGraph-based knowledge probing method to facilitate post-training privacy assessments. Our experiments demonstrated that PrivGraph achieves comparable performance to existing models in the Personally Identifiable Information (PII) detection task, while effectively modeling the aggregation of private information even with lengthy texts and data obtained from multiple origins. Finally, we discuss PrivGraph's integration into the AI engineering lifecycle for full-spectrum, full-lifecycle, and traceable privacy protection.

Keywords: data privacy; Artificial Intelligence; personally identifiable information; personal privacy protection; natural language processing

Check for updates

Academic Editors: Tanja Pavleska and Reiner Creutzburg

Received: 13 December 2025
Revised: 5 February 2026
Accepted: 6 February 2026
Published: 11 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The convergence of massive digital data growth and enhanced processing power [1–3] has transitioned Artificial Intelligence (AI) from theoretical research to a practical cornerstone for global automation [4,5]. Modern deep learning systems increasingly rely on immense datasets to extract intricate patterns and hierarchical representations [1,2]. However, this reliance on massive data and the resulting memorization creates severe privacy vulnerabilities [6–8]. Research indicates that deep learning models often inadvertently memorize training data verbatim rather than learning only robust features [6,9]. This phenomenon exposes models to malicious exploitation through data extraction, reconstruction, and membership inference attacks [7], ultimately compromising private information and undermining user trust in AI applications [8].

The OECD AI principles underscore that AI actors must uphold human-centered values, specifically privacy and data protection, throughout the entire system lifecycle [10]. This mandate necessitates the integration of privacy considerations from inception through maintenance. Consequently, establishing traceable privacy protection has become a critical requirement for ensuring accountability and compliance in AI-based systems.

To mitigate privacy vulnerabilities, contemporary research focuses on two paradigms: proactive modifications like knowledge editing and machine unlearning, and traditional PETs such as data anonymization. Proactive techniques surgically alter model parameters to eliminate sensitive information, whereas anonymization neutralizes attributes before they reach the model [11]. However, these methods do not fully secure complex AI systems [12,13]. The opaque nature of deep neural networks hinders verification of internal modifications, leaving it unclear whether sensitive data is truly eradicated or merely dormant. Furthermore, direct labeling often overlooks contextual Personally Identifiable Information (PII) [14], and maintaining data utility remains a significant hurdle [15]. Effectively protecting modern AI requires a multifaceted approach that addresses both algorithmic transparency and complex data dependencies.

AI models, especially deep learning models, exhibit powerful data collection and aggregation capabilities [1,9,16]. These models can collect and merge information from disparate data segments, or even diverse sources through model training. Although isolated instances of PII fragments may seem innocuous, the cumulative aggregation of sensitive information poses significant privacy risks [14]. However, identifying and removing all potential identifiers distorts the semantic content of data, leading to degraded model performance [12,13]. For example, replacing all phone numbers in a given dataset could hinder the ability of the model to recognize phone numbers. Therefore, PETs must be carefully designed and implemented to protect privacy while minimizing data alteration and preserving essential information [17]. However, if a less powerful PET is applied to the data, the model could still associate these remaining PII fragments with a specific individual by analyzing the context. This ultimately enables the model to remember enough personal information to identify the individual [18]. As a result, there is a need for data-centric PETs for AI-based systems to effectively mitigate the risks of sensitive information leakage through model memorization and inference, while also addressing the challenges posed by data aggregation and the contextual nature of private information. These PETs must be designed to not only identify and protect direct PII but also to understand and neutralize potential privacy threats from complex correlations learned by AI models, all while preserving the semantic integrity of the data and minimizing any adverse impact on the performance and utility of the model.

To address these challenges, this work provides a data-centric framework for modeling and mitigating privacy risks. First, we provide empirical evidence of memorization and aggregation risks by quantifying how data redundancy and PII linkability amplify information leakage. We then introduce PrivGraph, a hierarchical knowledge graph that provides a structural privacy representation of PII fragments and their associated identities. To measure these risks, we propose the Sensitivity Level Factor (SLF) as a quantitative privacy metric based on data type and frequency. Furthermore, we develop a PrivGraph-based knowledge probing method for post-training privacy assessment. Finally, we conclude by detailing the integration of this framework into the AI engineering lifecycle to ensure traceable, full-spectrum protection.

2. Related Work

Although privacy issues in AI-based systems have become a key research area, most studies have focused on model-based approaches. Limited attention has been given to adapting data-centric methods for these systems. This section reviews the existing literature

on data-centric privacy protection for textual data and the studies that quantify private information memorization in AI models.

To address the privacy vulnerabilities arising from model memorization, various Privacy-Enhancing Technologies (PETs) have been developed to protect sensitive information within AI-based systems [11,19,20]. These PETs generally fall into two primary categories: algorithm-based and data-centric methods [11]. Algorithm-based methods, such as knowledge editing [19], machine unlearning [20], and differential privacy [21], directly modify the training process of the AI, model architecture, or have an inference mechanism to inherently limit potential data leakage. Despite the promise of these techniques and their widespread adoption in the real world, production-level AI systems have encountered substantial practical challenges. First, stricter privacy controls often lead to less optimized models. Implementing stringent privacy measures can degrade the predictive accuracy or task performance [21]. Moreover, many algorithm-based PETs are computationally expensive, requiring additional resources for training or complex procedures such as model retraining for unlearning tasks [20]. Second, the inherent complexity and unpredictable output of deep neural networks make it difficult to verify and trace the effectiveness of PETs [11]. Third, emerging research indicates that the protection provided by algorithm-based methods might be less robust than initially presumed, potentially being circumvented by increasingly sophisticated adversarial attacks specifically crafted to exploit residual vulnerabilities [15,22,23]. To overcome these limitations, our proposed PrivGraph uses a data-centric framework to address the shortcomings of algorithmic PETs by focusing on granular, traceable data sanitization and the contextual nature of privacy.

On the other hand, data-centric PETs directly detect potential private information within the training data or AI model outputs. Techniques such as pattern matching, keyword filtering, and specialized Named Entity Recognition (NER) models have been applied to identify sensitive elements in the data [22,23]. To enhance the detection of PII within unstructured text, Privacy BERT-LSTM was proposed [24] to combine the contextual embeddings generated by BERT with the sequence labeling capabilities of LSTM. The research specifically defined PII to include sensitive information such as financial data and medical records. Brown et al. [25] introduced a straightforward and effective technique for detecting PII on public websites. To achieve dependable privacy inference with minimal complexity, their approach focused on evaluating how well NLP tools, including sentiment analysis and text classification algorithms can automatically detect PII. Eidan J. Rosado et al. [26] introduced PII-CODEX, a Python library designed to assess the severity of PII. By utilizing a combination of cloud service provider adapters and open-source libraries, PII-CODEX categorized data as unidentifiable, semi-identifiable, and identifiable. The study defined a sensitivity score as a basic aggregation of all sensitive data points, without accounting for their interconnections or whether they could be linked to a single person. PRIVAFRAME [27] utilized a knowledge graph approach, employing a logical-symbolic technique rooted in semantic frames and the Data Privacy Vocabulary. This allowed the categorization of personal data without the need for training datasets. Furthermore, the authors suggested a hybrid model that integrates PRIVAFRAME with deep learning to improve the detection of sensitive information. KG-ERNIE [28] merged ERNIE, an augmented BERT model for the Chinese language, with knowledge graph embeddings. This study also proposed the Text Sensitive Information Intelligent Perception algorithm, which identified sensitive information by conducting text detection across word, statement, and text levels. While traditional data-centric PETs focus on isolated masking and pattern matching, they fail to account for the complex, learnable correlations that enable deep learning models to re-identify individuals through data aggregation. These legacy approaches often necessitate a zero-sum trade-off between privacy and utility, frequently

resulting in degraded model performance or requiring unsustainable levels of manual labeling. Our approach, PrivGraph, overcomes these limitations by shifting from flat data scanning to a hierarchical, graph-based representation of privacy. By utilizing the SLF, we replace indiscriminate data removal with selective sanitization, preserving semantic utility while neutralizing aggregation risks. Furthermore, the PrivGrapher pipeline automates this process through synthetic corpora, eliminating the need for manual labeling and providing a scalable, traceable solution that spans the entire AI engineering lifecycle.

To investigate PII leakage in AI models, Petroni et al. [29] examined whether pretrained Language Models (LMs) retained and reproduced the factual information acquired during their training phase, essentially acting as knowledge bases. They employed fill-in-the-blank cloze statements, such as "Dante was born in [Mask]" and the LAMA probe to evaluate the models' ability to accurately supply missing information to determine their potential benefits compared to conventional knowledge bases. Building on this finding, Lukas et al. [15] broadened the scope of knowledge probing to encompass PII. Their research revealed that pretrained LMs possess a significant capacity to memorize PII, even when countermeasures such as PII scrubbing and differential privacy are implemented. Building on previous research, ProPILE [30] extended the investigation of PII leakage to Large Language Models (LLMs). Their findings highlighted that substantial PII exposure can happen via carefully designed prompts in black-box probing scenarios. However, these methods often fail to capture the context-dependent nature of privacy leakage. PrivGraph advances these techniques by explicitly modeling the interconnectivity and aggregation of discrete PII fragments scattered across large datasets. Unlike basic frequency counts, our framework utilizes the SLF to provide a multidimensional risk assessment that accounts for data type sensitivity and redundancy [31].

3. Background

3.1. Personally Identifiable Information

PII refers to any data that can be used to identify a specific individual. This identification can occur either by using a single piece of information, such as a Social Security number or passport number, or by combining several pieces of information, such as a name, date of birth, and zip code [32]. According to Schwartz, PII can be categorized as direct and indirect identifiers [32]. **Direct identifiers** are PII instances typically associated with one individual and can uniquely identify that individual [32], such as full names, driver's license numbers, and Social Security numbers. In contrast, **indirect identifiers** are PII instances that do not usually identify an individual on their own, but can uniquely specify someone when associated with a certain amount of context or other indirect PII instances. For instance, the job title "lawyer" alone does not identify a particular person as it refers to a group of individuals [32]. In our proposed framework, we utilized both direct and indirect PII to model privacy within the dataset and differentiate individual identities. We focus on the privacy implications of PII in AI models, primarily investigating whether the correlation between PII instances can lead to precise individual localization.

Structurability of PII. Structurability refers to textual PII that follows specific formats or patterns [15]. For example, email addresses typically conform to a pattern such as "username@address.com". This makes identification and extraction relatively easy and requires no labeled data for model training. In our proposed framework, we utilized both Natural Language Processing (NLP)-based techniques and pattern matching to optimize the model performance while minimizing the labeled data requirements.

3.2. Categorization of Personal Data Sensitivity

Different types of PII vary in how they connect to an individual and the level of sensitivity associated with the personal details they expose [33]. For example, a personal identification number is typically considered more sensitive information than a person's age. Sensitivity of an instance of PII is typically related to the following factors:

- *Potential privacy threat.* PII that has a higher probability of being exploited to inflict direct harm (e.g., through identity theft, financial fraud, or physical endangerment) is considered more sensitive.
- *Probability of re-identification.* PII types that readily enable the re-identification of a person, especially when aggregated with other data that appears harmless on its own, are classified as highly sensitive.
- *Context of target application.* The environment in which PII is gathered and utilized plays a crucial role in its sensitivity level. For example, medical information is deemed highly sensitive owing to its private and personal nature. For this study, we envision a scenario of creating publicly available software for general use, which informs our sensitivity scoring.
- *Legal and regulatory requirements.* Numerous laws and regulations, including the General Data Protection Regulation (GDPR) [34], Health Insurance Portability and Accountability Act (HIPAA) [35], and California Consumer Privacy Act (CCPA) [36], define different categories of PII and impose specific requirements for sensitive information protection.

3.3. Training Data Availability Issue

Training a deep learning model to identify the target PII in textual data requires a large amount of labeled training data. This requirement poses a significant challenge because (1) real-world datasets often exhibit biased distributions, where sensitive information comprises a disproportionately small fraction of the total data and is distributed irregularly. Research [37] has demonstrated that sensitive content only constitutes approximately 1.5% of publicly accessible data. Therefore, collecting datasets with high PII content is a challenging task. (2) The manual labeling required to generate sufficiently robust training datasets presents a substantial obstacle, particularly for small-scale projects with limited resources. (3) While Generative AI has been explored as a potential solution for synthetic data generation, its efficacy in mitigating data leakage remains unclear. Moreover, generated data could potentially introduce noise into the dataset, degrading model performance [38]. In our proposed framework, we utilized programmatically generated training data to address the issue of training data availability [6].

3.4. Methodological Motivation: Memorization and Aggregation Risk

The development of the PrivGrapher framework is motivated by the critical need to address two primary drivers of sensitive data exposure in AI: model memorization and data aggregation. Here we introduce two preliminary experiments to demonstrate how these factors impact privacy.

3.4.1. Impact of Data Redundancy on Privacy Risks

This experiment investigates the correlation between data redundancy and the transition from model generalization to memorization. The primary goal is to quantify how varying degrees of instance duplication influence the model's reliance on specific training records rather than the extraction of robust, intrinsic features [1,15,30]. By measuring these shifts, we aim to determine the specific point at which redundancy begins to compromise model utility and data privacy.

Empirical Validation of Redundancy Risks

To evaluate the impact of data redundancy on privacy risks, we collected 10,000 Wikipedia records of famous people and then applied the dataset to fine-tune the DistilBERT [39] and GPT-2 [40] models. Next, we applied knowledge probing to assess the impact of repeated data occurrence on model memorization. We tested the model's ability to recall a specific, concrete piece of information, i.e., a person's name.

As shown in Figure 1, with the increase in frequency of identical name instances, the probability of a data instance being memorized increases significantly for both models. This tendency towards memorization raises a critical privacy challenge: if an instance of PII frequently appears in the training data, the model is likely to memorize and subsequently reveal it in its output. Considering the impact of data duplication on privacy risk, our proposed framework incorporates the influence of data redundancy on model memorization into its SLF assessment. Notably, GPT-2 is not optimized for NER, because its unidirectional architecture struggles with tasks requiring full context, leading to poorer performance than BERT when fine-tuned for token-level tasks.

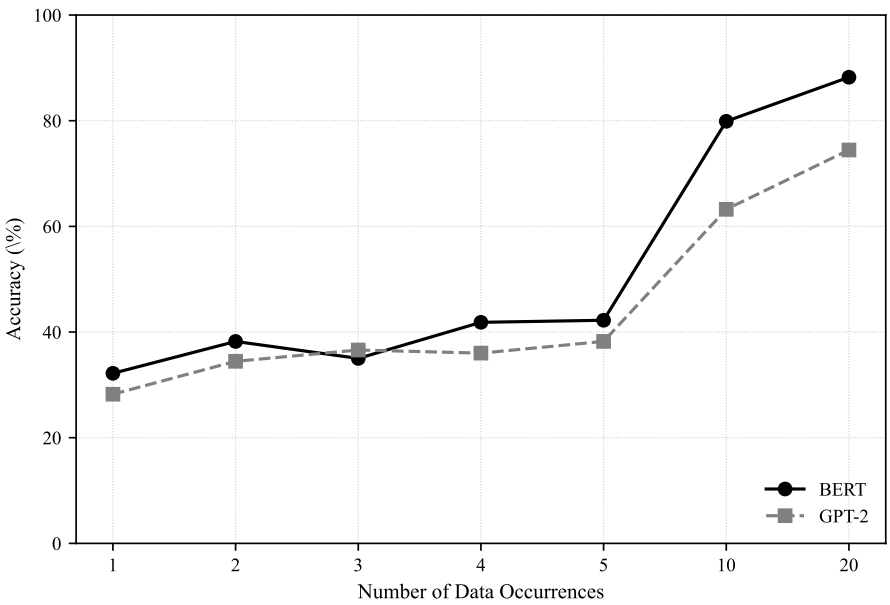

Figure 1. Knowledge probing result on Bert and GPT-2.

3.4.2. Privacy Impact on PII Linkability

In this section, we describe the experiments conducted to illustrate the impact of PII linkability on privacy risks. Specifically, we explore whether the model can discover the potential linkages of PII in the given dataset and memorize these specific PII instances along with their linkages. To this end, we propose a novel three-stage knowledge probing method for sparse PII.

Our Proposed Knowledge Probing Method

As shown in Figure 2, the proposed method begins by optimizing both NER and pattern matching to identify all possible PII instances within the dataset. We then group PII instances together if they fall into the same data segments. To test whether PII fragments dispersed over long distances can still interact, we apply a specific PII association logic: PII instances that appear together in a single data segment are initially placed into the same set and then grouped to form a unified PII set. Subsequently, if a direct PII instance matches an instance already present in a set, both the newly found instance and any of its neighboring PII instances are added to that set.

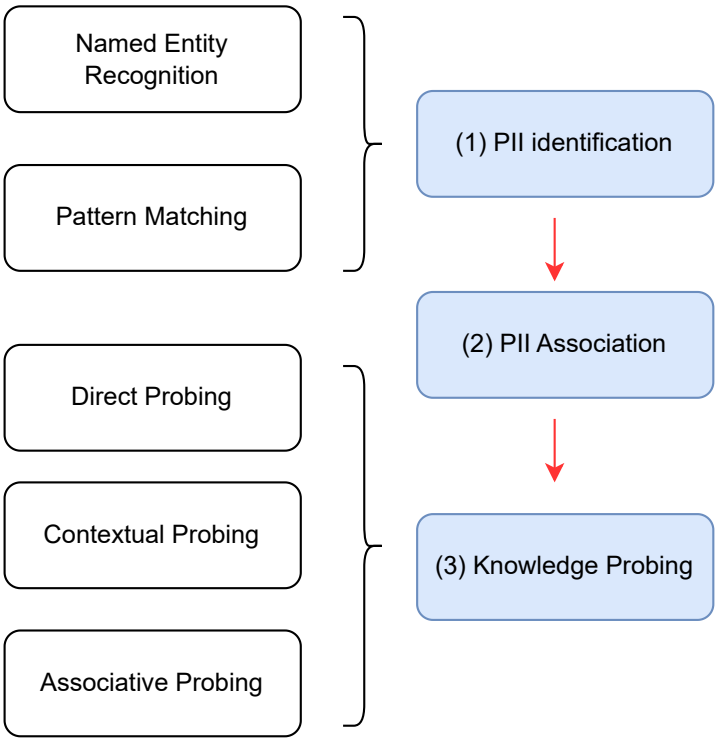

Figure 2. Overview of our proposed knowledge probing method.

We probe the PII memorization of the trained model using three methods: direct probing, contextual probing, and associative probing. Direct Probing examines the memorization ability of a model without any related PII instances. For example, in the case of an identified name instance, we ask the model to complete the masked tag. In this task, context provided is restricted to a single sentence with all other PII instances removed. Contextual Probing evaluates the model's understanding of the contextual information surrounding the PII instance. For this method, prompts are formulated based on identified PII, but parts of the PII or its context are masked (e.g., "In 1996, [MASK] was elected to represent the 13th district in the Illinois Senate"). The model's probability and accuracy in correctly completing the masked portion are measured. Associative Probing examines PII instances within the same PII set. By providing the contextual and relational information contained within a PII set for a target PII instance, we test how other PII instances in the same set—even those originally distributed far apart in the dataset—affect the model's memorization of the target. Because PII sets are collected across the entire dataset, this method clearly demonstrates whether different pieces of private information can influence the memorization of one another, even when they do not originally occur in the same data segment.

Quantification of PII Linkability

To assess the models' memorizability and capability to link PII instances in the given training data, we fine-tuned DistilBERT [39] and GPT-2 [40] models, and then reported the TOP@1 and TOP@5 accuracies for each probing method. As described in the previous section, we collected and labeled the Wikipedia Notable People dataset, which contains approximately 10,000 records.

As listed in Table 1, direct probing for PII resulted in low TOP@1 and TOP@5 accuracies for both models, which suggests that a PII instance alone is difficult to extract from the model without context. On the other hand, the probing accuracy significantly increased when contextual information surrounding the target PII was provided. The addition of

associative knowledge further increased the probing accuracy to more than 95%. These findings demonstrate that PII should not be evaluated in isolation, because a comprehensive assessment requires sufficient contextual information for accurate identification. To this end, our proposed method utilizes a knowledge graph to effectively represent the relationships between PII and its associated individuals. Furthermore, our proposed SLF incorporated these relationships into its assessment. Similar to the results in the previous section, both models exhibited similar trends, although the overall performance of GPT-2 was poorer than BERT. This difference is due to GPT-2's architecture, which can only attend to tokens that appear before the target tag in the sequence.

Table 1. Knowledge Probing Results.

Model	Probing Method	TOP@1	TOP@5
DistilBERT	Direct Probing	15.11	31.12
	Contextual Probing	91.85	95.01
	Associative Probing	**95.64**	**99.83**
GPT-2	Direct Probing	13.29	26.64
	Contextual Probing	55.82	63.01
	Associative Probing	**65.50**	**69.27**

4. PrivGraph: Representation and Formal Definitions

In this section, we introduce PrivGraph, a knowledge graph that identifies and models the aggregation of PII in textual data. First, we define the target private information. Using PII data categories as guidelines, we present the formal definition and structure of PrivGraph.

4.1. Definition of PII Data Category

To ensure the applicability of PrivGraph across diverse domains with varying privacy protection requirements, we employed a top-down methodology to define the target PII, structuring these PII types and their associated abstract concepts into a directed acyclic graph. Moreover, different types of PII vary in how they connect to an individual and the level of sensitivity associated with the personal details they expose [33]. Therefore, a *Data Type Sensitivity* (DTS) score was allocated to every defined data type to quantify its sensitivity.

Definition of PII data category. The PII data category is structured as a Directed Acyclic Graph that hierarchically organizes information from abstract concepts down to measurable data types.

- Concept Nodes (N_c): Represent abstract categories or high-level concepts ($c \in C$). While these nodes do not possess intrinsic DTS scores, they function as aggregation points; the SLF value for a Concept Node is defined as the mathematical sum of the SLF scores of all its subsumed children nodes, which provides the direct observable sensitivity level of different levels of abstraction.
- Leaf Nodes (N_d): Represent specific data types ($d \in D$). Every leaf node is defined as a tuple $N_d = \langle D_d, S_d \rangle$, where S_d is the DTS score.
- Edges: All edges are SUBSUME edges, directed from a higher-level abstract concept to a more specific sub-concept or a terminal data type.

4.2. Specifying Data Category

The identification of PII data categories begins with the analysis of established legal frameworks. In this study, we consider regulations include GDPR [34], the AI Act [36], and CCPA [36]. However, many of these regulations provide lists of abstract definitions without specifying concrete data types. Therefore, we expanded this initial PII catalog by incorporating common public knowledge as a foundational template. Specifically,

our analysis focused on the regulatory definitions of personal data and PII. For instance, GDPR Article 4(1) defines "personal data" as any information relating to an identified or identifiable natural person, i.e., a data subject. Based on these regulatory definitions, we refined these broad concepts and elaborated them into a range of specific data types, while carefully maintaining the hierarchical relationships among them. The developed PII data categorization comprised multiple hierarchical levels and 32 distinct data types.

4.3. Definition of PrivGraph

PrivGraph is a knowledge graph that models each identifiable individuals from the given textual dataset and their related PII. A single PII instance should not be considered sensitive unless it can be linked to multiple identifiers belonging to the same individual, which collectively allow for the identification of that person. We modeled PII with their relationships as follows:

- *Individual node*: Subject that owns the PII data.
- *PII node*: A node representing either an abstract PII concept or a specific PII data type.

Next, we encoded the following relationships between the defined node types as edges:

- *owns edge*: This edge represents the relationship in which an individual *owns* a specified PII data type. For example, an individual owns NAME. We denote the *owns* edge as $owns(I, d)$, or $e_s = I \xrightarrow{\text{OWNS}} d$.
- *subsume edge*: This edge represents the relationship in which a generic PII node (hypernym) *subsumes* a specific term of a PII node (hyponym). For example, an identity number subsumes a personal ID. We denote *subsume* edge as subsume (hyper, hypo), or $e_s = hyper \xrightarrow{\text{SUBSUME}} hypo$.

Definition of PrivGraph. A PrivGraph $P = \langle N_I, N_D; E_O, E_S \rangle$ is a directed acyclic graph constructed for a single identified individual. For each PrivGraph, there is an individual node N_I representing the identified individual, and a set of PII nodes N_d, where $d \in D$, representing PII data types that relate to that individual. Edges represent the relationship between nodes, where an *owns* edge, $e_o \in E_o$, represents the ownership relations between the individual node and specified PII nodes, and a *subsume* edge, $e_s \in E_s$, represents the subsumption relations between PII nodes.

4.4. Sensitivity Level Factor

Based on PrivGraph, we propose the Sensitivity Level Factor (SLF) to quantify the degree to which personal information is included in a given dataset. The SLF score considers several factors contributing to the sensitivity of a PrivGraph, including aggregation of PII, Sensitivity of PII, and the frequently occurring data samples are more likely to be memorized by AI models. This is expressed as Frequency of Appearance (FOA), which represents the number of occurrences of a specific PII element within the dataset.

Definition of Sensitivity Level Factor. For a PrivGraph P that represents an individual's PII profile, and $D_P \in D$ is the set of leaf nodes in P, where each node corresponds to a specific data type d with a sensitivity score S_d and a frequency of appearance f_d. The Sensitivity Level Factor (SLF) of P is the aggregate of the weighted sensitivity and the log-normalized frequency of all data types in D_P. We introduce a weighting coefficient α to prioritize the impact of the data sensitivity, and a scaling constant t to normalize the frequency. To mitigate the impact of high-frequency recurrence, we apply a logarithmic smoothing function:

$$SLF(P) = \sum_{d \in D_p} (\alpha * S_d + ln(1 + \frac{f_d}{t})) \quad (1)$$

5. PrivGrapher: System Pipeline

To extract PII instances and their relationships to data owners from textual data, we implemented an NLP-based pipeline called *PrivGrapher*. Owing to the computational complexity and resource limitations associated with concurrently processing entire large-scale datasets, we adopted a segmentation approach that leverages inherent data attributes and semantic boundaries. For example, Twitter data (https://x.com) can be segmented into individual posts, i.e., tweets, whereas novels or textbooks can be split into paragraphs or chapters. After PrivGraphs are constructed from each data segment, those representing the same individual are subsequently merged.

Figure 3 presents an overview of the *PrivGrapher* pipeline. First, *PrivGrapher* extracts the target PII instances from the dataset and categorizes them according to their PII data type. Next, PII data types are classified as either direct or indirect identifiers. Based on this classification, the Individual Recognition module is applied to identify distinct individuals within the dataset. As shown in Figure 3, the individuals *Harry* and *Ron* are identified from the provided data. Following this, the relationship annotation module assigns the remaining indirect PII to their corresponding owners. As mentioned at the beginning of this section, PrivGraphs representing the same individual are then merged. Finally, these merged PrivGraphs are enriched with parent PII concepts to facilitate the hierarchical demographic abstraction of private information. The remainder of this section describes the implementation of each step in *PrivGrapher* in detail.

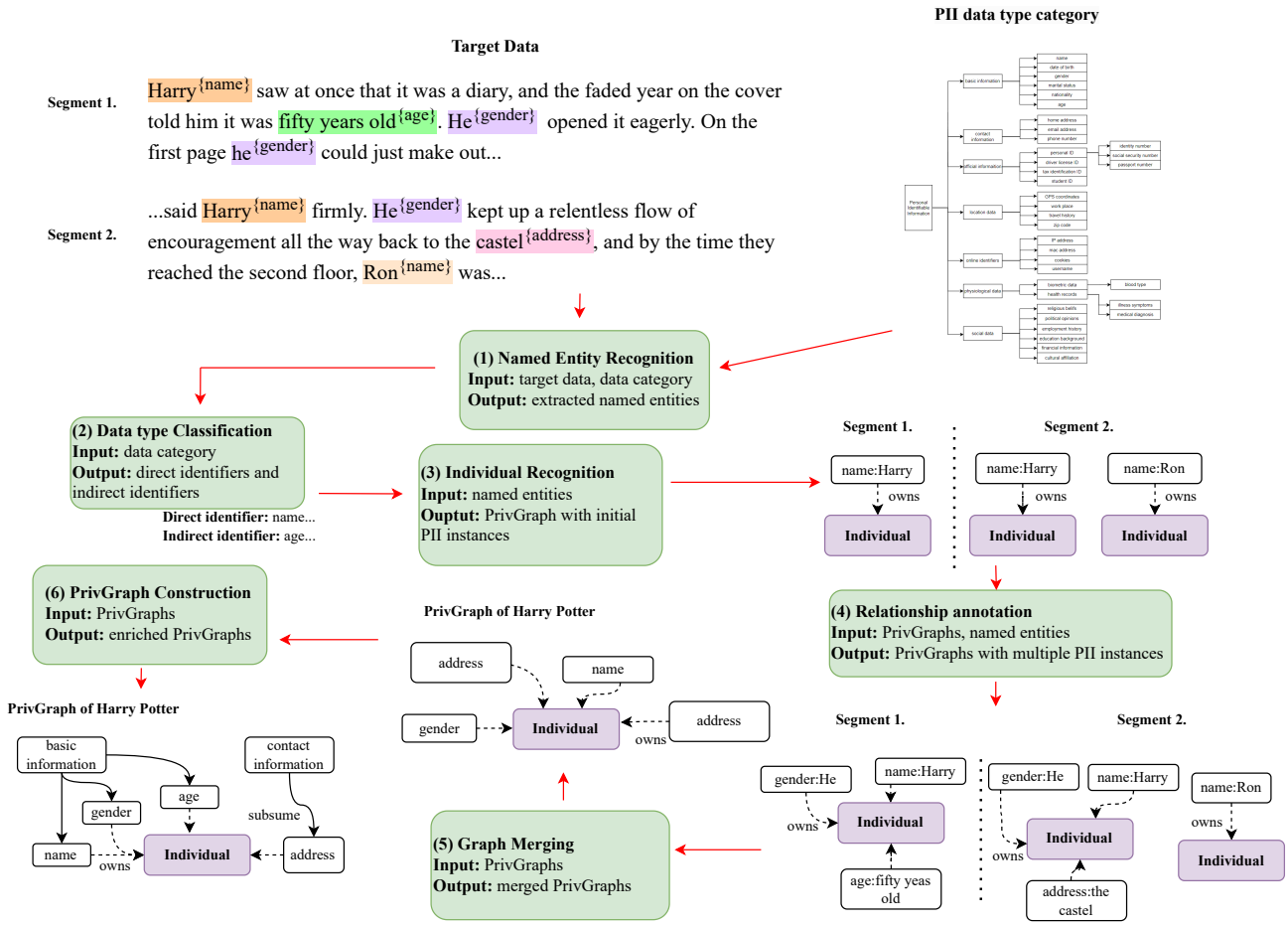

Figure 3. Overview of PrivGraph Construction pipeline. In this graph, the green boxes represent steps, and the red arrows indicate the sequence of those steps.

5.1. NLP-Based Named Entity Recognition

In the first stage, *PrivGrapher* identifies the target PII instances following the established PII data type categorization specified in Section 4.2. To mitigate the data availability problem, we first classified the identified PII into structured, unstructured, and semi-structured types based on their typical data formats and intrinsic properties.

Structured PII. Structured PII exhibits a fixed language pattern. For example, an email would follow the "username@domain.com" format, while a telephone number would typically follow the structure "XXX-XXXX-XXXX".

Unstructured PII. Identifying PII is particularly challenging in unstructured text, where the absence of a fixed schema forces detection models to rely heavily on complex contextual cues. For example, names, addresses, or nationalities can appear in numerous, varied, and unexpected ways when embedded within large blocks of unstructured text.

Semi-structured PII. Semi-structured PII is characterized by its adherence to certain linguistic patterns. However, accurate identification requires an understanding of context. For example, while age is commonly expressed using one or two digits, a standalone numerical figure is not definitively a person's age without further contextual clues.

Next, *PrivGrapher* identifies the target PII using both regular expressions [24] and NER [12]. Specifically, regular expressions are chosen for certain tasks because of their processing speed and lack of training requirements. However, because they only match superficial patterns and overlook contextual or semantic information, their application is directed towards extracting PII that is structured and context-independent, i.e., structured PII. For unstructured PII that do not follow any predefined formats, or PII that require surrounding information for identification (i.e., semi-structured PII), *PrivGrapher* utilizes an NER model to understand and detect contextual information. The system's NER component is specifically built using the spaCy library [41] with its RoBERTa-based NLP pipeline [41].

As discussed in the previous section, training an NER model requires a large quantity of labeled training data, which is a resource-intensive task. To avoid creating a dataset from scratch, we employed a programmatically generated synthetic corpus for training the NER model. To generate training data, we first created templates for each type of unstructured and semi-structured PII. For example, for *NAME*, we first created a list of templates such as "%NAME%, (born | born in | born on) %DATE_OF_BIRTH%, {SUBJECT_PRONOUN} is a ..." After generating permutations of these templates, we obtained 13,644 template instances. Next, we extracted PII instances from existing sources to fill the placeholders in these expanded templates. For unstructured PII, we retrieved the corresponding PII instances from publicly available sources on the Internet to replace placeholders within the expanded templates. For example, for *NAMES*, we collected human names from the NAMES corpus dataset (https://www.kaggle.com/); for *NATIONALITIES*, we sourced country names from Wikipedia (https://www.wikipedia.org/). For the semi-structured PII, placeholders were populated by programmatically generated data. This procedure resulted in a dataset comprising 1,001,954 training records, 134,600 testing records, and 134,600 validation records. The programmatic creation of our synthetic data allowed the automatic assignment of labels to every named entity.

5.2. Data Type Classification

In this step, PII data types are classified into *direct identifiers* and *indirect identifiers* to facilitate individual recognition.

5.3. Individual Recognition

Because it is common for real-world datasets to contain information on several individuals within a single data segment, we analyzed the presence of individuals in each data segment based on direct and indirect identifiers to determine the number of unique individuals.

For *direct identifiers*, *PrivGrapher* first queries existing PrivGraphs to determine if the encountered PII instance is already associated with an individual. If a matching record is located, an additional entry is appended to that PrivGraph to continue tracking the position of the record. If no such record exists in the current PrivGraphs, a new PrivGraph is initiated. In this case, an *owns* edge is created to link the direct identifier instance to the relevant individual node. If ownership cannot be clearly determined through these steps, the Relationship Annotator is utilized to identify the correct owner.

For *indirect identifiers*, we used the Relationship Annotator to determine their corresponding owner, i.e., the corresponding PrivGraph.

5.4. Relationship Annotation

The goal of relationship annotation is to associate PII instances with their corresponding owners, i.e., PrivGraphs. This includes analyzing the presence and interrelationships of identifiers within a data segment to determine the presence of additional individuals. Similar to the NER model in *PrivGrapher*, relationship annotation primarily employs two methodologies: rule-based matching and deep learning-based methods. The relationship annotator in *PrivGrapher* combines the strengths of two distinct methods: the accuracy and efficiency of rule-based systems and the flexibility and adaptability of deep learning-based models. Specifically, relationship annotation leverages Part-Of-Speech (POS) tags and syntactic dependencies [41] to analyze the relationships between identified PII instances and their associated individuals. When rule-based methods fail to extract relationships from the text, a classification model is trained to predict the corresponding owner.

We initiated the relationship annotation process by resolving textual references, i.e., identifying the antecedents of pronouns like “he” and “her”. This task is known as coreference resolution in NLP. For this purpose, we utilized the NLP-based F-coref model [42], which provides an effective balance between accuracy and computational speed. One advantage of the F-coref model is that it is trained on a broad dataset, allowing its immediate application to coreference resolution tasks without requiring additional fine-tuning.

Next, we assign the corresponding owner to each PII instance. This task is different from traditional relation extraction in NLP, because the objective of relationship annotation is to ascertain whether an identified PII can be accurately linked to an individual [43], regardless of the specific nature of their connection. For example, a person can “live at", “maintain", or even “not reside at” a home address. Since all such relationships can potentially disclose personal information, we normalized their representation by constructing an *owns* edge between the PII and individual within the PrivGraph.

We treat PII instances that have already been assigned an owner as annotation targets. Once we determine that a candidate PII instance and target PII instance belong to the same individual, we link the candidate to the matching target. For example, if a PII instance [NAME] has already been associated with an individual, and a subsequent PII instance [EMAIL ADDRESS] is identified as belonging to the same individual, we append the new PII instance to the existing PrivGraph.

PrivGrapher’s rule-based matching strategy utilizes both dependency parsing and Part-of-Speech tagging [41] to systematically analyze sentence structures and pinpoint entities acting as subjects. As outlined in Algorithm 1, the procedure requires four primary inputs: the initial entity e_m, the set of candidate target entities E_t, and syntactic dependency tags D_{sub} (subject relations) and D_{mod} (modifiers or attributes). The algorithm identifies the

target entity e_t through an iterative traversal of the dependency tree, starting from e_m and moving upward toward the root. In each iteration, the algorithm checks if e_m belongs to the target entity set E_t and possesses a dependency tag within D_{sub} or D_{mod}. If both conditions are met, e_m is returned as the target entity e_t. Second, if the current token is identified as a functional head, specifically a $VERB$ or an auxiliary AUX, the algorithm performs a downward search among its children. If no match is found at the current level, the algorithm assigns e_m to its parent (*head*) and continues the search. The process terminates when the target entity is found or when the traversal reaches the $ROOT$ of the dependency tree without a match. By evaluating the starting entity before climbing and unifying the handling of verbal and auxiliary heads, this hierarchical approach effectively captures subject-object relationships across various sentence types, including copula constructions and interrogative forms.

Algorithm 1: Rule-based Relationship Annotation

```
Input: Initial entity e_m, Entity set E_t, Dependency sets D_sub, D_mod.
Output: Target Entity e_t ∈ E_t or None.
while True do
  /* 1. Check if current token is a subject or modifier */
  if dep(e_m) ∈ D_sub ∪ D_mod and e_m ∈ E_t then
    return e_t ← e_m
  end if
  /* 2. If token is a verb or auxiliary, search children for subject */
  if pos(e_m) = VERB or pos(e_m) = AUX then
    for each c ∈ children(e_m) do
      if dep(c) ∈ D_sub and c ∈ E_t then
        return e_t ← c
      end if
    end for
  end if
  /* 3. Exit if ROOT is reached, otherwise move up the tree */
  if dep(e_m) = ROOT or head(e_m) = e_m then
    BREAK LOOP
  end if
  e_m ← head(e_m)
end while

return e_t ← None
```

However, owing to the complexity of natural language and the diversity of expression, rule-based techniques are restricted to analyzing general, predefined sentence patterns. PrivGrapher employs a deep learning methodology to overcome this limitation when examining relationships in complex sentences that do not adhere to these fixed rules. Specifically, the deep learning-based Relationship Annotator operates by attaching a binary classification head to a RoBERTa transformer model [41]. This allows the system to ascertain if a relationship exists between two entities by analyzing their surrounding contextual data. The data produced during NER training primarily details specific individuals. This information is then organized and merged to create extended sentences featuring multiple individuals, which are used to train the relationship annotator. If the model identifies a PII instance with a pre-existing record in a PrivGraph, this new instance is integrated into that same graph. However, if none of the PII within the current data segment are linked to any existing PrivGraph, a new graph is initiated, and an *owns* edge is created from the relevant identifier.

5.5. Graph Merging

In this section, we merge PrivGraphs from different data segments that refer to the same individual. This merging primarily occurs when PrivGraphs share identical direct identifier instances. For identifiers prone to variations, such as "Barack Obama", "Obama" or "Mr. Obama", we utilized a flexible merging technique instead of insisting on strict matches. Notably, to prevent the erroneous association of PII with incorrect persons, PrivGraphs without any direct identifier instances were not considered for merging.

5.6. PrivGraph Construction

PrivGrapher completes the PrivGraph structure by creating parent nodes for all identified PII nodes. These parent nodes correspond to the PII data types or their higher-level concepts from the established categorization. *PrivGrapher* then adds a *SUBSUME* edge extending from each parent node to its respective child node. Figure 4 illustrates a PrivGraph built using information from Barack Obama's Wikipedia page (https://en.wikipedia.org/wiki/Barack_Obama, accessed on 5 February 2026). As shown in the graph, the individual "Barack Obama" was identified, along with many of his PII instances, from the data. For each specified PII data type, multiple occurrences of the same PII value (e.g., the same specific email address appearing several times) may be identified from the data. We recorded all such data instances in the "Records" field to keep track of all the private information. For an abstract data type, the SLF score is calculated based on the sum of all its subsumed PII types. Finally, for each PrivGraph, an overall SLF score is calculated for the individual.

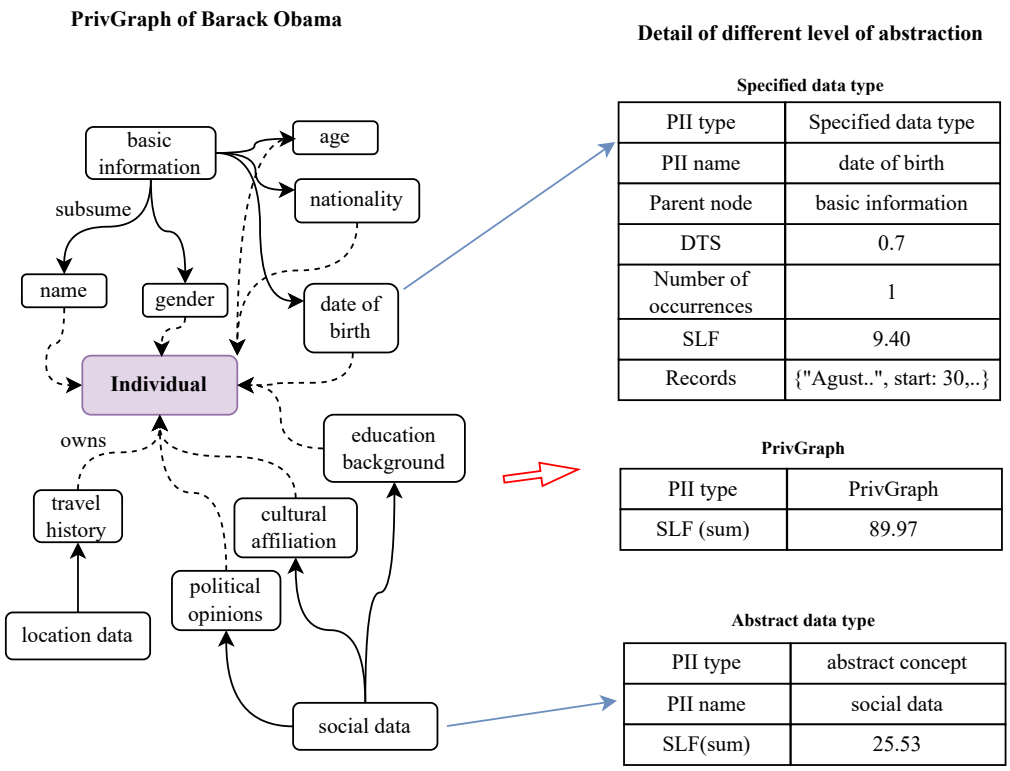

Figure 4. PrivGraph constructed from Wikipedia of Barack Obama.

6. Evaluation

This section assesses PrivGraph's proficiency in modeling and consolidating sensitive information. As prior studies on aggregate PII are unavailable, our evaluation first reports and compares the performance of PrivGraph's NER module with baseline systems. Subsequently, we demonstrate PrivGraph's performance in information aggregation tasks.

6.1. Dataset

As discussed in Section 3.3, typical textual datasets for general use are often unsuitable for training or evaluating PII-detection models due to the extremely low proportion and uneven distribution of sensitive information. To effectively assess PrivGraph's PII extraction capabilities, we compiled and manually annotated a new dataset comprising Wikipedia entries of 100 prominent individuals. As an additional demonstration of PrivGraph's ability to uncover private information from diverse data platforms, we manually gathered data from Twitter (https://x.com) and Reddit (https://www.reddit.com/). This data collection was confined to the period between January 2022 and January 2023, with the aim of identifying instances in which PII from these distinct sources could be aggregated. Our evaluation utilized distinct data samples, including 10,000 randomly chosen conversations and threads from the Reddit dataset and a similar random sample of 10,000 tweets from the Twitter dataset. As shown in Table 2, the textual data from Reddit discussions is generally longer than that from Twitter, because of the different community dynamics. To further assess PrivGraph's capability with large-scale, continuous narratives, we also processed the complete Harry Potter novel series. We segmented this dataset by chapter, with each chapter comprising an average of approximately 5661 words.

Table 2. Data statistics of datasets.

Data Sources	Number of Records	Statistics
Wikipedia of famous people	100	Each record contains 299.54 words on average
Reddit conversations	10,000	Each record contains 1161.07 words on average
Twitter tweets	10,000	Each record contains 180.86 words on average
Harry Potter	199 chapters	Each chapter contains 5661 words on average

6.2. Baseline Models and Experiment Settings

To establish a robust comparative framework, we selected baseline models based on two primary criteria: model performance and transparency. We chose the models because they represent the current state-of-the-art in sensitive information identification. Furthermore, these models are publicly accessible and open-source.

- PRIVAFRAME [27] presents a hybrid strategy for detecting sensitive information, by merging deep learning with logical-symbolic techniques. In this model, a deep learning component is responsible for pinpointing sensitive sentences, while a frame-based knowledge graph is employed to determine specific types of personal data with fine granularity.
- KG-ERNIE [28] proposed a model for sensitive information perception rooted in ERNIE and knowledge graph embedding. This approach employs a knowledge graph-based technique for entity embedding along with the ERNIE pretrained model to extract semantic details and features. A Convolutional Neural Network processed the encoded information for recognition and categorization.
- Privacy BERT-LSTM [24] leverages the combined strengths of BERT for generating contextual embeddings and LSTM for its effectiveness in sequential information processing. The model also incorporates an attention mechanism to identify and emphasize important regions within textual documents.

The experiments were conducted using the spaCy library [41] and its RoBERTa-based NLP pipeline [41] for the core NER components. For the deep learning-based Relationship Annotator, we attached a binary classification head to a RoBERTa transformer model. PrivGrapher and the baseline models (KG-ERNIE and Privacy BERT-LSTM) was trained on our programmatically generated synthetic corpus comprising 1,001,954 training records, 134,600 testing records, and 134,600 validation records.

To synthesize the combined influence of DTS and Frequency of Appearance on the SLF, we set the parameters at $\alpha = 10$ and $t = 0.1$. This configuration was determined empirically to serve as a calibration mechanism; since DTS scores and raw frequency counts operate on different numerical scales, these values align their contributions so that neither factor disproportionately dominates the final SLF. Specifically, the weight α scales the discrete sensitivity levels to a range comparable with the log-transformed frequency, while t ensures the logarithmic term remains sensitive to low-frequency PII occurrences that still carry significant privacy risks.

6.3. Results

According to Figure 5, KG-ERNIE delivered the best performance with a precision of 0.9001, recall of 0.9220, and an F1 score of 0.9101. Privacy BERT-LSTM was the second best, with a precision, recall, and F1 score of 0.8741, 0.8721, and 0.8731, respectively. However, both models required considerable labeled training data for optimal performance. Conversely, PrivGrapher and PRIVAFRAME, which do not require manually labeled data, performed comparably well. PrivGrapher recorded a precision, recall, and F1 score of 0.7835, 0.9121, and 0.8429, respectively. PRIVAFRAME recorded a precision, recall, and F1 score of 0.7823, 0.8746, and 0.8260, respectively. These results demonstrate that PrivGrapher and PRIVAFRAME are competitive options, especially when manual labeling is unfeasible.

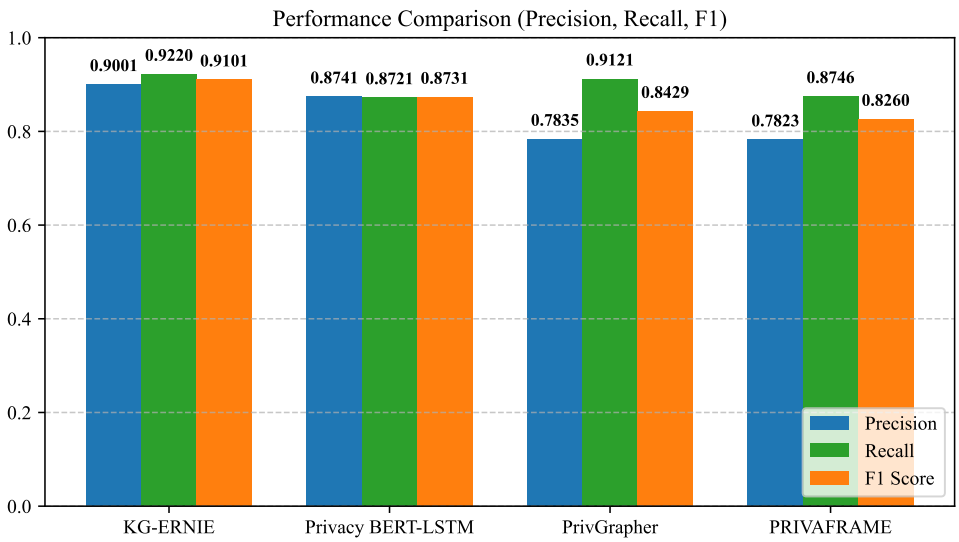

Figure 5. Experimental results of the NER task comparing baselines and our proposed method on the Wikipedia of famous people dataset.

The evaluation of the full PrivGrapher pipeline demonstrates its robust end-to-end capability, successfully identifying 144 PrivGraphs from 100 heterogeneous input records. As shown in Table 3, while the rule-based relation annotation achieves high precision (98.75%) for standard patterns, it only identifies approximately 60% of the relations between PII and individuals in practical operation. On the other hand, the model-based component provides the critical adaptability required for complex, non-standard PII. Although the integrated pipeline faces the inherent challenge of error propagation common in multi-stage architectures, an overall F1 score of 79.93% underscores PrivGrapher's competitive performance in modeling and aggregating PII without the need for manual labeling. These results confirm that the pipeline remains a highly effective and scalable solution for automated privacy graph generation.

Table 3. Privgraph Performance on Privgraph Generation.

Tasks	Precision	Recall	F1
NER	78.35	91.21	84.29
Relation annotation (Rule-based)	98.75	1.00	99.37
Relation annotation (Model-based)	91.32	93.84	92.56
Overall	72.01	89.82	79.93

6.4. PrivGraph Application

In this section, we demonstrate PrivGraph's capability to effectively structure and consolidate PII from various origins and substantial, continuous textual data. In addition, building upon PrivGraph's framework, we introduced a novel knowledge probing technique designed to quantify specific private data retained by a model.

6.4.1. Modeling PII from Different Sources

To illustrate PrivGraph's ability to handle PII instances from various sources, we evaluated its performance when modeling data consolidated from distinct sources. Our methodology involves deploying *PrivGrapher* on datasets comprising Twitter tweets and Reddit conversations separately. As a result, the PrivGraphs generated from each individual source were combined.

Experimental result. Table 4 lists the results of this study. PrivGrapher detected 2693 PrivGraphs from the Twitter tweets dataset and 5347 from the Reddit conversations dataset. After combining PrivGraphs built from these two sources, a total of 8002 unique PrivGraphs were formed, which reflect the consolidation of 38 instances in which PrivGraphs from the individual datasets overlapped.

Analysis of the merged dataset reveals an average of 2.29 specified PII instances for each PrivGraph, which indicates a generally sparse concentration of sensitive information within the overall dataset. Despite this general sparseness, the presence of PrivGraphs exhibiting high SLF values signifies a non-uniform distribution of private data. Such randomness and unevenness could foster high-risk privacy situations and underscore the need to manage specific cases involving excessive personal information exposure. The fact that 38 PrivGraphs were consolidated from both Twitter and Reddit sources illustrates the tangible risks of private information being aggregated for individuals who use several social media platforms.

Table 4. Statistics of PrivGraphs built on Twitter and Reddit datasets.

Metric	Twitter	Reddit	Merged
Number of PrivGraphs	2693	5347	8002
Average specified PII types per PrivGraph	1.73	4.32	2.29
Min SLF	6.10	6.40	6.10
Max SLF	42.24	135.73	135.73
Average SLF	10.21	21.84	13.44
Number of Merged Graphs	–	–	38

6.4.2. Long Textual Data Analysis

In this section, we examine PrivGraph's ability to model PII when dealing with extensive and continuous datasets. Specifically, we applied *PrivGrapher* to the complete collection of Harry Potter novels. We segmented this novel series at its chapter boundaries, which produced 199 data segments, with an average chapter length of 5661 words.

Experimental result. The data presented in Table 5 indicated that PrivGrapher successfully identified 838 distinct individuals within the novel series, with each PrivGraph containing an average of 11.32 specified types of PII. In contrast to the outcomes from the

preceding section, the analysis of the Harry Potter novel series showed a markedly higher count of specified PII types per PrivGraph and an increased average SLF. These results imply that the volume of sensitive information about individuals in a fictional narrative is typically greater than observed in informal discussions or arbitrary posts on social media.

Table 5. Statistics of PrivGraphs built on harry potter.

Metric	Result
PrivGraphs	838
Average specified PII types per PrivGraph	11.32
Min SLF	8.30
Max SLF	507.73
Average SLF	132.92

The experimental output reveals substantial PII disclosure for the novel's main characters. As shown in Figure 6, the main character Harry Potter was identified in 58,185 instances of PII, with a SLF score of 507.73; Hermione Granger was identified in 18,795 instances of PII, with a SLF score of 482.23; and Ron Weasley was identified in 26,904 instances of PII, with a SLF score of 431.29. When these findings were contrasted with results from the Twitter and Reddit datasets, it becomes evident that text sources emphasizing character development, such as novels, tend to encompass a greater volume of PII. For key protagonists such as Harry Potter and Ron Weasley, this PII is often dispersed across the entirety of the narrative.

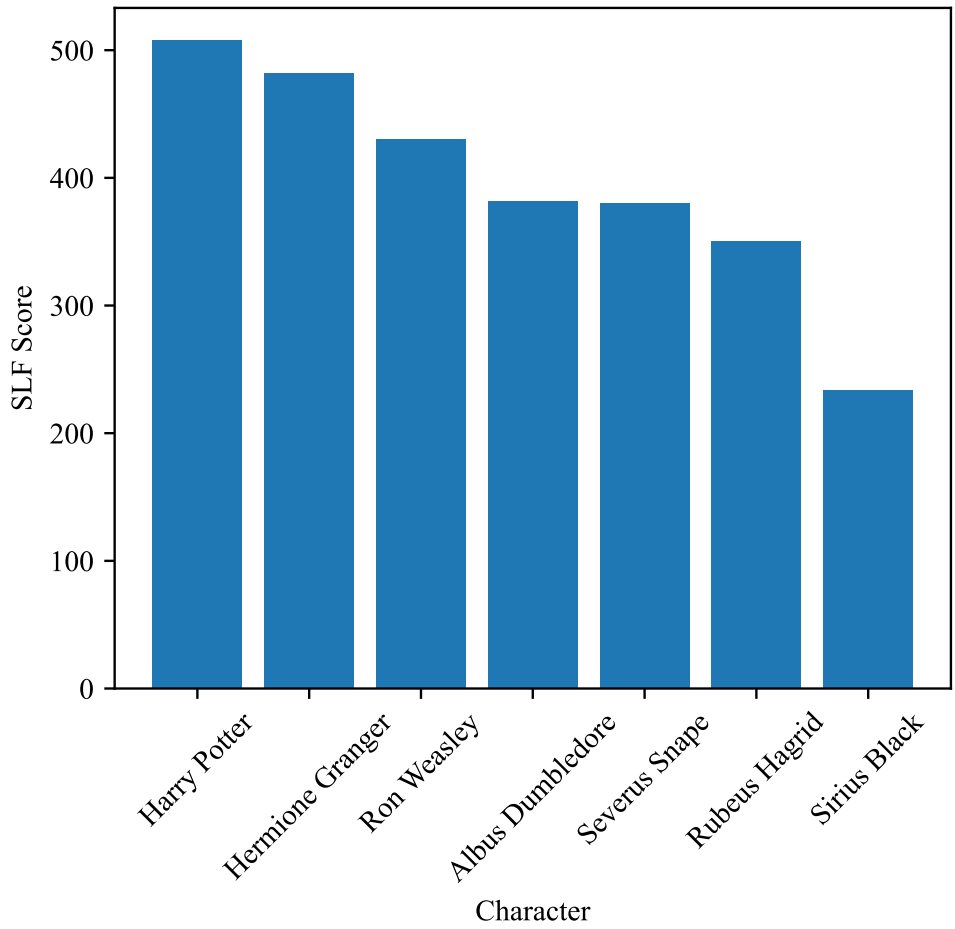

Figure 6. The SLF score of the main characters in Harry Potter.

6.4.3. PrivGraph-Based Privacy Knowledge Probing

Another application of PrivGraph is as a ground truth baseline when investigating how aggregated sensitive information is memorized within a model's post-training parameters. This facilitates the assessment of PETs implemented in the model. To effectively safeguard an individual associated with a PrivGraph P, it is necessary to reduce the frequency and repetition of sensitive data types. Accordingly, for a PrivGraph P with a set of leaf nodes D_P, the Model SLF is the aggregate of the probing accuracy for each data type, weighted by its corresponding sensitivity score S_d. Formally:

$$ModelSLF(P) = \sum_{d \in D_p} ProbingAcc(d) * S_d \tag{2}$$

where $\mathrm{Acc}(d)$ denotes the probing accuracy achieved by the model for the specific data type d.

The procedure for assessing the aggregation of privacy knowledge within the trained model starts by creating structured templates tailored to each category of PII data, such as "[name], born in [date of birth], ..." These templates are then populated using specific PII instances extracted from the target PrivGraph. Subsequently, we introduce random masking, which obscures certain PII entries within these populated templates. Finally, we evaluate the model's performance in predicting the concealed PII details, relying solely on the unmasked contextual information provided.

In this experiment, 10,000 entries were randomly sampled from a Wikipedia dataset of famous people. We used these samples to fine-tune the DistilBERT [39] and GPT-2 [40] models. For DistilBERT, we employed the Masked Language Modeling (MLM) task [30]. Since the GPT-2 model can only attend to tokens that appear before it in the sequence, we replaced MLM with the next word prediction task for this model. Following the fine-tuning process, we examined the potential privacy vulnerabilities of the models and the reduction in their performance when subjected to different PETs. For comparison, the set of PETs evaluated in this study includes:

- **Differential privacy** [21]. In differential privacy, the parameter ϵ is applied to adjust the privacy budget.
- **NER-based PII identification and random anonymization**. We applied Privacy BERT-LSTM [24] to identify potential PII from the dataset, and then performed random anonymization by replacing each identified PII instance with a corresponding PII tag, irrespective of its specific attribute or type.
- **NER-based single PII anonymization**. After identifying all PII instances using the same NER model as in random anonymization, we removed all instances of several PII types that were generally recognized as sensitive, such as name, gender, and marital status.
- **PrivGraph-based privacy protection**. This method removes the PII that contributes the most to SLF scores in PrivGraphs above a certain threshold.

PrivGraph-based privacy protection offers a focused approach by identifying the total privacy footprint of each individual. This empowers developers and privacy engineers to concentrate on individuals with the highest exposure risks. As a result, developers can precisely remove data elements that markedly elevate an SLF score, thereby avoiding less targeted methods, such as random PII deletion or the wholesale removal of all data categories. Moreover, PrivGraph supports calculated removal of alternative PII if conflicts with system operations arise, thereby enhancing privacy safeguards while ensuring minimal disruption of functionality.

To quantify the degradation in model performance, we reported *Perplexity* scores [30]. We employed random probing (RandP) to measure the general memorization loss. Fur-

thermore, the model's capacity to memorize sensitive data was evaluated within the MLM task, for which we calculated the TOP@1 and TOP@5 accuracies based on our curated labeled records. To enable fair comparison, we fine-tuned each PET to attain similar ModelSLF scores. For differential privacy, we adjusted the privacy budget ϵ. For NER-based PII identification, random anonymization, NER-based single PII anonymization, and PrivGraph-based privacy protection, we controlled the strength of privacy protection by increasing or decreasing the number of anonymized entities.

Experimental results. As shown in Table 6, all PETs were fine-tuned to exhibit comparable ModelSLF scores. In terms of privacy protection efficacy, NER-based random anonymization demonstrated the lowest privacy knowledge probing accuracy (0.01 TOP@1 and 0.03 TOP@5 for the BERT model), suggesting its potential for effective PII safeguarding. However, this approach substantially impaired both the model's general knowledge memorability and its performance as measured by perplexity. Similarly, NER-based single PII anonymization yielded negative perplexity and RandP scores. Its low TOP@1 and high TOP@5 accuracies suggest that it only protects PII that is removed from the dataset, failing to safeguard remaining privacy attributes. Moreover, while the differential privacy-based method offered similar levels of privacy protection in knowledge probing, it did so at the expense of increased model perplexity.

In contrast, the PrivGraph-based strategy, which selectively removes PII with high-sensitivity contributions while preserving PrivGraphs with low SLF scores, successfully protects privacy and minimizes data noise. PrivGraph successfully balanced the trade-off between privacy and data utility by maintaining high model performance with minimal degradation in perplexity.

Table 6. Privacy protection results by model.

Model	PETs	Perplexity	RandP	TOP@1	TOP@5	Avg Model SLF
GPT-2	No PETs	5.32	0.69	0.60	0.82	10.21
	Differential Privacy	7.44	0.54	0.02	0.06	4.12
	NER-based random anonymization	10.56	0.51	0.01	0.02	3.75
	NER-based single PII anonymization	9.56	0.56	0.32	0.53	3.54
	PrivGraph	6.32	0.60	0.43	0.57	4.01
DistilBERT	No PETs	2.70	0.81	0.92	0.95	16.04
	Differential Privacy	4.69	0.69	0.03	0.17	2.08
	NER-based random anonymization	3.78	0.73	0.01	0.03	1.89
	NER-based single PII anonymization	4.03	0.71	0.49	0.72	2.02
	PrivGraph	2.73	0.79	0.68	0.69	1.98

7. Discussion and Practical Implications

7.1. Discussion of Experiments

The experimental results offer a deeper understanding of how PrivGraph bridges the gap between empirical data patterns and the practical privacy risks inherent in modern AI engineering. By evaluating our framework across diverse data distributions, we have established that the linkability of information is a more significant threat than isolated data leakage. In the context of aggregation across sources, our findings from Twitter and Reddit data reveal that while individual fragments of information might appear innocuous when viewed in isolation, their synthesis into consolidated profiles across platforms creates a severe privacy vulnerability. For real-world AI systems, this necessitates a shift in perspective where privacy-enhancing technologies must move beyond treating records as independent entities and instead adopt a graph-based approach to account for the cumulative reconstruction of individual identities through disparate data sources.

Building upon this, the analysis of long-text aggregation through narratives like the Harry Potter series highlights a specific vulnerability regarding the depth of information. Text sources that involve significant character development or continuous factual narratives allow deep learning models to act as unintended knowledge bases. Because PII is often dispersed throughout a long-form text, models can memorize and reproduce high-fidelity profiles based on repeated, contextually rich descriptions. To mitigate this in production-level AI, it is essential to neutralize long-distance interactions between PII fragments. This ensures that the model does not build a lookup mechanism for specific individuals, thereby preserving the utility of the training data while preventing the memorization of high-sensitivity narratives.

Furthermore, our results regarding model privacy auditing and knowledge probing demonstrate that PrivGraph serves as an effective ground-truth baseline for quantifying the private data actually retained within model parameters. The high accuracy of associative probing confirms that models naturally learn to link scattered PII, yet the PrivGraph approach offers a more surgical solution than traditional methods like Differential Privacy. By selectively removing the PII that contributes most to the SLF, developers can maintain model utility and avoid the performance degradation typically associated with black-box anonymization.

While our current evaluation focuses on GPT-2 and BERT to establish a foundational baseline for PrivGraph, we recognize that the NLP landscape has shifted toward much larger models. Future work will investigate whether the observed sensitivity patterns and performance gains scale linearly as parameter counts increase. Furthermore, LLMs have demonstrated superior performance in processing unstructured data through prompt engineering, which offers a more efficient alternative to data-intensive and computationally heavy fine-tuning. We plan to replace current rule-based components with LLM-driven automation in future iterations. By leveraging advanced prompt engineering to handle complex relationship annotations, we aim to minimize manual overhead and enhance the framework's adaptability across diverse and unstructured datasets. Additionally, we are considering expanding our evaluation by incorporating datasets from various other platforms. This expansion will allow us to further demonstrate the generalizability and robustness of our approach across different data distributions and domains.

7.2. Discussion of Integrating PrivGraph with the AI Engineering Lifecycle

The integration of PrivGraph into the AI engineering lifecycle establishes a data-centric framework for traceable privacy management that spans from initial requirements to post-deployment monitoring. As illustrated in Figure 7, this process begins during requirements elicitation and analysis, where standards, regulations, and user needs are translated into specific PII data categories that define the modeling scope. As the lifecycle moves into data collection and private data modeling, PrivGraph acts as a private data modeler to extract sensitive information from raw datasets, particularly in large-scale contexts like LLM training where it allows for fine-grained detection of unexpected exposure and offers users flexibility in selecting which personal data is processed. During the cleaning, training, and evaluation phases, the tool provides a visual model that enables engineers to aggregate private information effectively, minimizing data noise while ensuring that even in cases of functional conflict, such as bank loan analysis, irrelevant identifiers like names or Social Security numbers are excluded to prevent individual identification. Finally, in the deployment and monitoring stage, PrivGraph serves as a supervisor and final defense mechanism by analyzing model outputs and web-retrieved inputs in Retrieval-Augmented Generation contexts [44] to detect potential leakage or inference attacks, thereby ensuring that personal information remains protected through every evolution of the system.

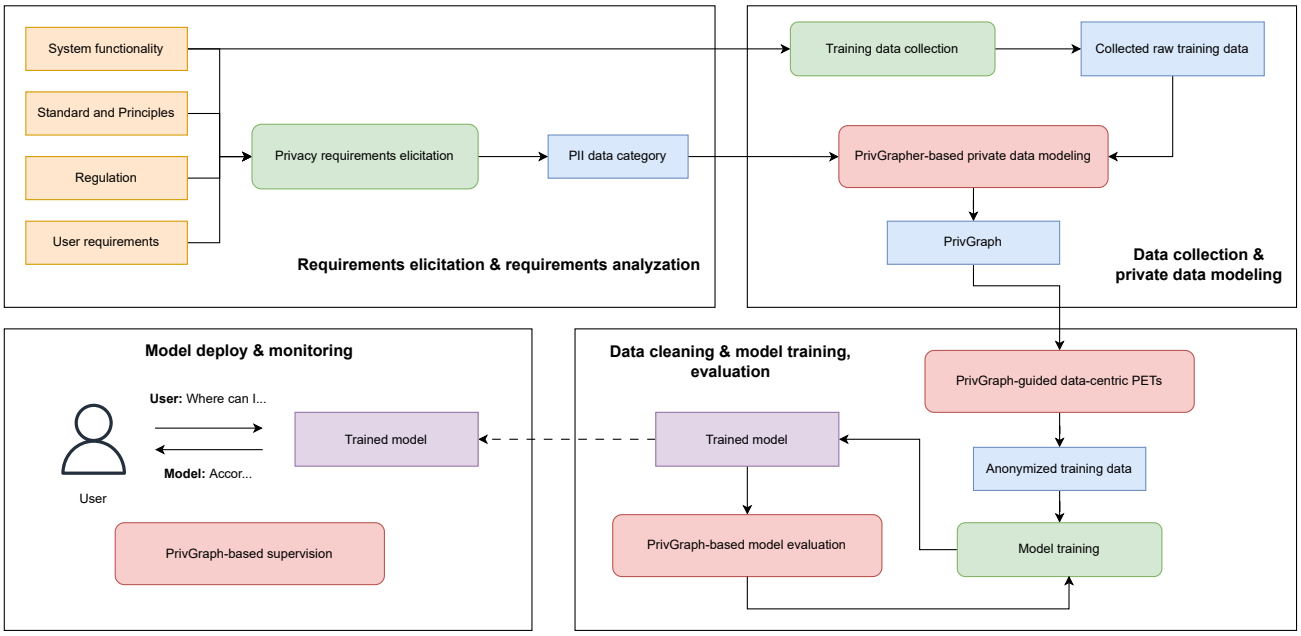

Figure 7. Our proposed privacy engineering process for AI-based systems.

8. Conclusions

In summary, this study addressed the growing concerns of data aggregation for AI-based systems by investigating two critical, yet often nuanced, aspects of privacy vulnerabilities: the impact of data redundancy on amplifying privacy risks and the mechanisms through which PII linkability can lead to re-identification and sensitive disclosures. Our analysis revealed how these factors contributed to the potential for privacy breaches, even when basic data protection measures are in place.

To address this issue, we introduce a knowledge graph to effectively collect, represent, and quantify the amount of sensitive information present in the training data of AI models. Specifically, we define and specify PII categories in a hierarchical manner to clearly manage target PII patterns. We propose PrivGraph, a knowledge graph representation of PII that enables granular modeling and control of private information within datasets. Based on PrivGraph, we develop the SLF (Sensitivity Level Factor) to quantify the sensitivity of a PrivGraph.

Our experiments across Wikipedia, social media, and long-form narratives demonstrate that PrivGrapher achieves highly accurate privacy detection without requiring manually labeled training data. The framework successfully identified high-risk PII concentrations in long-form content and mapped sensitive cross-platform identities that traditional methods often overlook. Furthermore, PrivGrapher outperformed standard privacy protection techniques by maintaining superior data utility, proving that selective, graph-based anonymization can effectively safeguard privacy without compromising model performance. Finally, we discuss the integration of PrivGraph throughout the AI engineering lifecycle to comprehensively enhance privacy.

Future research will focus on investigating whether PrivGraph's sensitivity patterns and performance benefits scale effectively with LLMs while replacing existing rule-based components with LLM-driven automation and prompt engineering to enhance adaptability. Additionally, we aim to expand our evaluation to a broader range of platforms to further demonstrate the framework's robustness across diverse data distributions.

Author Contributions: Conceptualization, J.Z. and S.-W.L.; Methodology, J.Z. and S.-W.L.; Software, J.Z.; Validation, J.Z.; Formal Analysis, J.Z.; Investigation, J.Z.; Data Curation, J.Z.; Writing—Original Draft Preparation, J.Z.; Writing—Review and Editing, J.Z. and S.-W.L.; Visualization, J.Z.; Supervision,

S.-W.L.; Project Administration, S.-W.L.; Funding Acquisition, S.-W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the BK21 FOUR program of the National Research Foundation of Korea funded by the Ministry of Education (NRF5199991014091). This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2026-RS-2023-00255968) grant funded by the Korean government (MSIT).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Our code is available at: https://github.com/pphui8/PrivGraph (accessed on 5 February 2026), our dataset is available at https://drive.google.com/drive/folders/1MPmpHUsKpYiullqZNOnTNqrVzE7GJ4Ws?usp=drive_link (accessed on 5 February 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PETs	Privacy-Enhancing Technologies
PII	Personally Identifiable Information
SLF	Sensitivity Level Factor
NER	Named Entity Recognition
NLP	Natural Language Processing
LMs	Language Models
LLMs	Large Language Models
DP	Differential Privacy
DTS	Data Type Sensitivity
GDPR	General Data Protection Regulation
CCPA	California Consumer Privacy Act
POS	Part-of-Speech
MLM	Masked Language Modeling

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
2. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. 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)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186.
3. Wu, S.; Fei, H.; Qu, L.; Ji, W.; Chua, T.-S. Next-GPT: Any-to-Any Multimodal LLM. In Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024.
4. Islam, S.B.; Rahman, M.A.; Hossain, K.T.; Hoque, E.; Joty, S.; Parvez, M.R. Open-RAG: Enhanced Retrieval Augmented Reasoning with Open-Source Large Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024), Miami, FL, USA, 12–16 November 2024*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024.
5. Targ, S.; Almeida, D.; Lyman, K. Resnet in Resnet: Generalizing Residual Architectures. *arXiv* **2016**, arXiv:1603.08029. [CrossRef]
6. Carlini, N.; Tramèr, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, Ú. Extracting Training Data from Large Language Models. In *Proceedings of the 30th USENIX Security Symposium, Vancouver, BC, Canada, 11–13 August 2021*; USENIX Association: Berkeley, CA, USA, 2021; pp. 2633–2650.
7. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–24 May 2017; pp. 3–18.
8. Rahman, M.M.; Arshi, A.S.; Hasan, M.M.; Mishu, S.F.; Shahriar, H.; Wu, F. Security Risk and Attacks in AI: A Survey of Security and Privacy. In Proceedings of the 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), Torino, Italy, 26–30 June 2023; pp. 1834–1839.

9. Petroni, F.; Rocktäschel, T.; Riedel, S.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A. Language Models as Knowledge Bases? In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 2463–2473.
10. OECD.AI. AI Principles. Available online: https://oecd.ai/en/ai-principles (accessed on 13 March 2025).
11. Nguyen, T.T.; Huynh, T.T.; Ren, Z.; Nguyen, T.T.; Nguyen, P.L.; Yin, H.; Nguyen, Q.V.H. Privacy-preserving explainable AI: A survey. *Sci. China Inf. Sci.* **2025**, *68*, 111101. [CrossRef]
12. Tesfay, W.B.; Serna, J.; Rannenberg, K. Privacybot: Detecting Privacy Sensitive Information in Unstructured Texts. In Proceedings of the 6th International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, 22–25 October 2019; pp. 53–60.
13. Yermilov, O.; Raheja, V.; Chernodub, A. Privacy- and Utility-Preserving NLP with Anonymized Data: A Case Study of Pseudonymization. *arXiv* **2023**, arXiv:2306.05561. [CrossRef]
14. Hassanpour, A.; Utsash, M.M.; Yang, B. The Impact of Linkability on Privacy Leakage. In *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Istanbul, Turkey, 10–12 November 2022*; ACM: New York, NY, USA, 2020; pp. 364–370.
15. Lukas, N.; Ahmed, S.C.; Li, B.; Salem, K.; Kerschbaum, F. Analyzing Leakage of Personally Identifiable Information in Language Models. In Proceedings of the 2023 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 21–25 May 2023; pp. 346–363.
16. Goethals, S.; Sörensen, K.; Martens, D. The Privacy Issue of Counterfactual Explanations: Explanation Linkage Attacks. *ACM Trans. Intell. Syst. Technol.* **2023**, *14*, 83. [CrossRef]
17. Rigaki, M.; Garcia, S. A Survey of Privacy Attacks in Machine Learning. *ACM Comput. Surv.* **2024**, *56*, 101. [CrossRef]
18. Carlini, N.; Liu, C.; Erlingsson, Ú.; Kos, J.; Song, D. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *Proceedings of the 28th USENIX Security Symposium, Santa Clara, CA, USA, 14–16 August 2019*; USENIX Association: Berkeley, CA, USA, 2019; pp. 267–284.
19. Wang, S.; Zhu, Y.; Liu, H.; Zheng, Z.; Chen, C.; Li, J. Knowledge Editing for Large Language Models: A Survey. *ACM Comput. Surv.* **2024**, *57*, 59. [CrossRef]
20. Liu, Z.; Huang, Y.; Wen, Y.; Zhao, J.; Lin, J. Breaking the Trilemma of Privacy, Utility, and Efficiency via Controllable Machine Unlearning. In *Proceedings of the ACM Web Conference 2024 (WWW '24), Singapore, 13–17 May 2024*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 1260–1271.
21. Zhu, T.; Ye, D.; Wang, W.; Zhou, W.-C.; Yu, P.S. More Than Privacy: Applying Differential Privacy in Key Areas of Artificial Intelligence. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 2824–2843. [CrossRef]
22. Shumailov, I.; Shumaylov, Z.; Zhao, Y.; Gal, Y.; Papernot, N.; Anderson, R. Ununlearning: Unlearning Is Not Sufficient for Content Regulation in Advanced Generative AI. *arXiv* **2024**, arXiv:2407.00106.
23. Zhang, Z.; Wang, F.; Li, X.; Wu, Z.; Tang, X.; Liu, H.; He, Q.; Yin, W.; Wang, S. Catastrophic failure of LLM unlearning via quantization. In Proceedings of the 13th International Conference on Learning Representations (ICLR), Singapore, 24–28 April 2025.
24. Muralitharan, J.; Arumugam, C. Privacy BERT-LSTM: A Novel NLP Algorithm for Sensitive Information Detection in Textual Documents. *Neural Comput. Appl.* **2024**, *36*, 15439–15454. [CrossRef]
25. Brown, C.; Morisset, C. Simple and Efficient Identification of Personally Identifiable Information on a Public Website. In Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2022; pp. 4246–4255.
26. Rosado, E.J. PII-Codex: A Python Library for PII Detection, Categorization, and Severity Assessment. *J. Open Source Softw.* **2023**, *8*, 5402. [CrossRef]
27. Gambarelli, G.; Gangemi, A. PRIVAFRAME: A Frame-Based Knowledge Graph for Sensitive Personal Data. *Big Data Cogn. Comput.* **2022**, *6*, 90. [CrossRef]
28. Huo, L.; Jiang, J. Research on Intelligent Perception Algorithm for Sensitive Information. *Appl. Sci.* **2023**, *13*, 3383. [CrossRef]
29. Petroni, F.; Rocktäschel, T.; Lewis, P.; Bakhtin, A.; Wu, Y.; Miller, A.H.; Riedel, S. Language Models as Knowledge Bases? In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), Hong Kong, China, 3–7 November 2019; pp. 2463–2473.
30. Kim, S.; Lee, S.; Kwak, H.; Oh, A. Propile: Probing Privacy Leakage in Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS 2023)*; Curran Associates, Inc.: New Orleans, LA, USA, 2023; Volume 36, pp. 20750–20762.
31. Zuo, J.; Lee, S.-W. PrivGraph: Modeling personal privacy information using knowledge graph. In Proceedings of the 2025 25th International Conference on Software Quality, Reliability, and Security (QRS), Hangzhou, China, 16–20 July 2025; pp. 432–441.
32. Schwartz, P.M.; Solove, D.J. The PII Problem: Privacy and a New Concept of Personally Identifiable Information. *NYU Law Rev.* **2011**, *86*, 1814.
33. Rumbold, J.M.M.; Pierscionek, B.K. What Are Data? A Categorization of the Data Sensitivity Spectrum. *Big Data Res.* **2018**, *12*, 49–59. [CrossRef]

34. European Commission. 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 (General Data Protection Regulation). *Off. J. Eur. Union* **2016**, *L119*, 1–88.
35. U.S. Government. *Health Insurance Portability and Accountability Act of 1996 (HIPAA)*; Pub. L. No. 104–191, 110 Stat. 1936; U.S. Government: Washington, DC, USA, 1996.
36. European Commission. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act). *Off. J. Eur. Union* **2024**, *L*, 1–81.
37. Zhao, Y.; Chen, J. A Survey on Differential Privacy for Unstructured Data Content. *ACM Comput. Surv.* **2022**, *54*, 207. [CrossRef]
38. Stadler, T.; Oprisanu, B.; Troncoso, C. Synthetic Data–Anonymisation Groundhog Day. In *Proceedings of the 31st USENIX Security Symposium, Boston, MA, USA, 10–12 August 2022*; USENIX Association: Berkeley, CA, USA, 2022; pp. 1451–1468.
39. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv* **2019**, arXiv:1910.01108.
40. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. *Language Models Are Unsupervised Multitask Learners*; OpenAI Technical Report; OpenAI: San Francisco, CA, USA, 2019.
41. Explosion. spaCy: Industrial-Strength Natural Language Processing. Available online: https://spacy.io/ (accessed on 13 March 2025).
42. Otmazgin, S.; Cattan, A.; Goldberg, Y. F-coref: Fast, Accurate and Easy to Use Coreference Resolution. In Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: System Demonstrations, Taipei, Taiwan, 20–23 November 2022; pp. 48–56.
43. Zhong, Z.; Chen, D. A Frustratingly Easy Approach for Entity and Relation Extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2021), Online, 6–11 June 2021*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 50–61.
44. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.T.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*; Curran Associates, Inc.: San Diego, CA, USA, 2020; Volume 33, pp. 9459–9474.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.