rx-anon—A Novel Approach on the De-Identification of Heterogeneous Data based on a Modified Mondrian Algorithm

F. Singhofer
fabian.singhofer@uni-ulm.de
University of Ulm
Germany

A. Garifullina, M. Kern
{aygul.garifullina,mathias.kern}@bt.com
BT Technology
United Kingdom

A. Scherp
ansgar.scherp@uni-ulm.de
University of Ulm
Germany

ABSTRACT

Traditional approaches for data anonymization consider relational data and textual data independently. We propose rx-anon, an anonymization approach for heterogeneous semi-structured documents composed of relational and textual attributes. We map sensitive terms extracted from the text to the structured data. This allows us to use concepts like k-anonymity to generate a joined, privacy-preserved version of the heterogeneous data input. We introduce the concept of redundant sensitive information to consistently anonymize the heterogeneous data. To control the influence of anonymization over unstructured textual data versus structured data attributes, we introduce a modified, parameterized Mondrian algorithm. The parameter λ allows to give different weight on the relational and textual attributes during the anonymization process. We evaluate our approach with two real-world datasets using a Normalized Certainty Penalty score, adapted to the problem of jointly anonymizing relational and textual data. The results show that our approach is capable of reducing information loss by using the tuning parameter to control the Mondrian partitioning while guaranteeing k-anonymity for relational attributes as well as for sensitive terms. As rx-anon is a framework approach, it can be reused and extended by other anonymization algorithms, privacy models, and textual similarity metrics.

CCS CONCEPTS

• **Security and privacy** → **Data anonymization and sanitization**; • **Computing methodologies** → Information extraction.

KEYWORDS

data anonymization, heterogeneous data, k-anonymity

ACM Reference Format:
F. Singhofer, A. Garifullina, M. Kern, and A. Scherp. 2021. rx-anon—A Novel Approach on the De-Identification of Heterogeneous Data based on a Modified Mondrian Algorithm. In . ACM, New York, NY, USA, 34 pages.

1 INTRODUCTION

Researchers benefit from companies, hospitals, or other research institutions, who share and publish their data. It can be used for predictions, analytics, or visualizations. However, often data to be shared contains Personally Identifiable Information (PII) which does require measures in order to comply with privacy regulations like the Health Insurance Portability and Accountability Act (HIPAA) for medical records in the United States or the General Data Protection Regulation (GDPR) in the European Union. One possible measure to protect PII is to anonymize all personal identifiers. Prior work considered such personal data to be name, age, email address, gender, sex, ZIP, any other identifying numbers, among others [12, 16, 31, 34, 52]. Therefore, the field of Privacy-Preserving Data Publishing (PPDP) has been established which makes the assumption that a data recipient could be an attacker, who might also have additional knowledge (e. g., by accessing public datasets or observing individuals).

Data to be shared can be structured in the form of relational data or unstructured like free texts. Research in data mining and predictive models shows that a combination of structured and unstructured data leads to more valuable insights. One successful example involves data mining on COVID-19 datasets containing full texts of scientific literature and structured information about viral genomes. Zhao and Zhou [62] showed that linking the mining results can provide valuable answers to complex questions related to genetics, tests, and prevention of SARS-CoV-2. Moreover, the combination of structured and unstructured data can also be used to improve predictions of machine learning models. Teinemaa et al. [54] developed a model for predictive process monitoring that benefits from adding unstructured data to structured data. Therefore, links within heterogeneous data should be preserved, even if anonymized.

However, state of the art methods focus either on anonymizing structured relational data [21, 32, 35, 41, 53, 55] or anonymizing unstructured textual data [8, 16, 34, 45, 47, 50], but not jointly anonymizing on both. In example, for structured data the work by Sweeney [53] introduced the privacy concept k-anonymity, which provides a framework for categorizing attributes with respect to their risk of re-identification, attack models on structured data, as well as algorithms to optimize the anonymization process by reducing the information loss within the released version. For unstructured data like texts, high effort has been conducted to develop systems which can automatically recognize PII within free texts using rule based approaches [40, 45, 50], or machine learning methods [8, 16, 24, 34] to allow for replacement in the next step. To the best of our knowledge, the only work that aimed to exploit synergies between anonymizing texts and structured data is by Gardner and Xiong [16]. The authors transferred textual attributes to structural attributes and subsequently applied a standard anonymization approach. However, there is no recoding of the original text, i. e., there is no transfer back of the anonymized sensitive terms. Thus, essentially Gardner and Xiong [16] only anonymize structured data. Furthermore, there is no concept of information redundancy, which is needed for a joined de-anonymization, and there is no weighting parameter to control the influence of relational versus textual

Preprint, Published on arXiv,
2021.

Table 1: Running example of a de-normalized dataset D with relational and textual attributes. A^* is an attribute directly identifying an individual. A_1, ..., A_5 are considered quasi-identifiers and do not directly reveal an individual. X is the textual attribute. See Table 3 for details on notations.

A^*	Relational Attributes A_1, ..., A_5					Textual Attribute X
id	**gender**	**age**	**topic**	**sign**	**date**	**text**
1	male	36	Education	Aries	2004-05-14	My name is Pedro, I'm a 36 years old engineer from Mexico.
1	male	36	Education	Aries	2004-05-15	A quick follow up: I will post updates about my education in more detail.
2	male	24	Student	Leo	2005-08-18	I will start working for a big tech company as an engineer.
3	male	37	Banking	Pisces	2004-05-27	During my last business trip to Canada I met my friend Ben from college.
4	female	24	Science	Aries	2004-01-13	As a scientist from the UK, you can be proud!
4	female	24	Science	Aries	2004-01-17	Four days ago, I started my blog. Stay tuned for more content.
4	female	24	Science	Aries	2004-01-19	2004 will be a great year for science and for my career as a biologist.
5	male	29	indUnk	Pisces	2004-05-15	Did you know that Pisces is the last constellation of the zodiac.
6	female	27	Science	Aries	2004-05-15	Rainy weather again here in the UK. I hope you all have a good day!

attributes as splitting criterion for the data anonymization. Our experiments show, such a weighting is important as otherwise it may lead to a skewed splitting favoring to retain relational attributes over textual attributes.

To illustrate the problem of a joined anonymization of textual and structured data, we consider an example from a blog dataset.[1] As Table 1 indicates, a combined analysis relies on links between the structured and unstructured data. Therefore, it is important to generate a privacy-preserved, but also consistently anonymized release of heterogeneous datasets consisting of structured and unstructured data. Due to the nature of natural language, textual attributes might contain redundant information which is already available in a structured attribute. Anonymizing structured and unstructured parts individually neglects redundant information and leads to inconsistencies in data, since the same information might be anonymized differently. Moreover, for privacy-preserving releases, assumptions on the knowledge of an attacker are made. Privacy might be at risk if the anonymization tasks are conducted individually and without sharing all information about an individual.

We provide a formal problem definition and software framework rx-anon on a joined anonymization of relational data with free texts. We experiment with two real-world datasets to demonstrate the benefits of the rx-anon framework. As baselines, we consider the scenarios where relational and textual attributes are anonymized alone, as it is done by the traditional approaches. We show that we can reduce the information loss in texts under the k-anonymity model. Furthermore, we demonstrate the influence of the λ parameter that influences the weight between relational and textual information and optimize the trade-off between relational and textual information loss.

In summary, our work makes the following contributions:

- We formalize the problem of anonymizing heterogeneous datasets composed of traditional relational and textual attributes under the k-anonymity model and introduce the concept of redundant information.
- We present an anonymization framework based on Mondrian [31] with an adapted partitioning strategy and recoding scheme for sensitive terms in textual data. To this end, we introduce the tuning parameter λ to control the share of information loss in relational and textual attributes in Mondrian.
- We evaluate our approach by measuring statistics on partitions and information loss on two real-world datasets. We adapt the Normalized Certainty Penalty score to the problem of a joined anonymization of relational and textual data.

Below, we discuss related works on data anonymization. We provide a problem formalization in Section 3 and introduce our joined de-anonymization approach rx-anon in Section 4. The experimental apparatus is described in Section 5. We report our results in Section 6. We discuss the results in Section 7, before we conclude.

2 RELATED WORK

Research in the field of anonymization can be differentiated according to the type of data to be anonymized. We present related work on the de-identification of structured data, i. e., traditional relational and transactional data, and unstructured texts. We present works that aim to exploit synergies between these two tasks and contrast them with our rx-anon approach.

Anonymization of Structured Data. Early work of Sweeney [51] showed that individuals, even if obvious identifiers are removed, can be identified by using publicly available data sources and link them to the apparently anonymized datasets. Such attempts to reveal individuals using available linkable data are called record linkage attacks. Her work introduced explicit identifiers and quasi-identifiers. The former category is also called direct identifier and poses information which directly reveals an identity. Attributes of the latter category do not reveal an identity directly, but can reveal an identity if used in combination with other attributes.

This observation led to extensive research on privacy frameworks. An important and very influential approach is k-anonymity, which prevents re-identification attacks relying on record linkage using additional data [53]. k-anonymity describes a privacy model

[1] Note, we used the schema of the Blog Authorship Corpus throughout our running examples in Tables 1, 4, and 5.

where records are grouped and each group is transformed such that their quasi-identifiers are equal. To achieve k-anonymity, Samarati [46] studied suppression and generalization as efficient techniques to enforce privacy. In addition, Meyerson and Williams [38] and LeFevre et al. [31] have shown that optimal k-anonymity in terms of information loss both in the suppression model and for the multidimensional case is NP-hard. Several algorithms have been developed to efficiently compute a k-anonymous version of a dataset while keeping the information loss minimal. Sweeney [52] proposed a greedy approach with tuple suppression to achieve k-anonymity. LeFevre et al. [31] suggested a top-down greedy algorithm Mondrian which implements multidimensional k-anonymity using local recoding models. Ghinita et al. [17] showed how optimal multidimensional k-anonymity can be achieved by reducing the problem to a one-dimensional problem which improves performance while reducing information loss. Based on the k-anonymity model, several extensions have been introduced and studied, where l-diversity and t-closeness are most popular. Machanavajjhala et al. [35] introduced the model of l-diversity to prevent homogeneity and background knowledge attacks on the k-anonymity model. l-diversity uses the concept of sensitive attributes to guarantee diversity of sensitive information within groups of records. Li et al. [32] introduced t-closeness, which extends the idea of diversity by guaranteeing that the distribution within groups does not differ more than a threshold t from the global distribution of sensitive attributes.

While k-anonymity was initially designed to be applied for a single table containing personal data (also called microdata), it has been transferred to different settings. Nergiz et al. [41] investigated the problem of anonymizing multi-relational datasets. They state that k-anonymity in its original form cannot prevent identity disclosure neither on the universal view nor on the local view and therefore modified k-anonymity to be applicable on multiple relations. Gong et al. [18] showed that regular k-anonymity fails on datasets containing multiple entries for one individual (also called 1:M). To anonymize such data, they introduced (k, l)-diversity as a privacy model which is capable of anonymizing 1:M datasets. Terrovitis et al. [55] applied k-anonymity to transactional data. Given a set of items within a transaction, they treated each item to be a quasi-identifier as well as a sensitive attribute simultaneously. The solution introduces k^m-anonymity which adapts the original concept of k-anonymity and extends it by modeling the number of known items of the adversary in the transaction as m. He and Naughton [21] proposed an alternative definition of k-anonymity for transactional data where instead of guaranteeing that subsets are equal in at least k transactions, they require that at least k transactions have to be equal. Finally, Poulis et al. [42] showed how k-anonymity can be applied to data consisting of relational and transactional data and stated that a combined approach is necessary to ensure privacy.

Anonymization of Unstructured Data. In order for textual data to be anonymized, information in texts that may reveal individuals and therefore considered sensitive must be recognized. In recent work, two approaches have been used to extract so called sensitive terms in text. First, Sánchez et al. [47] proposed an anonymization method which makes use of the Information Content (IC) of terms. The IC states the amount of information a term provides and can be calculated as the probability that a term appears in a corpus. The reasoning behind using the IC of terms to detect sensitive information is that terms which provide high information tend to be also sensitive in a sense that an attacker will gain high amounts of information if those terms are disclosed. The advances in the field of Natural Language Processing (NLP) have been used to detect sensitive terms by treating them as named entities. Named Entity Recognition (NER) describes the task of detecting entities within texts and assigning types to them. Named entities reflect instances or objects of the real world, like persons, locations, organizations, or products among others and provide a good foundation for detecting sensitive information in texts. Therefore, recent work formulated and solved the detection of sensitive information as a NER problem [11, 16, 24, 34, 57]. Early work on NER to identify sensitive terms was based on rules and dictionaries [45, 50]. Sweeney [50] suggested a rule-based approach using dictionaries with specialized knowledge of the medical domain to detect Protected Health Information (PHI). Ruch et al. [45] introduced a system for locating and removing PHI within patient records using a semantic lexicon specialized for medical terms. Advances in machine learning led to new approaches on the de-identification of textual data. Gardner and Xiong [16] introduced an integrated system which uses Conditional Random Fields (CRF) to identify PII. Dernoncourt et al. [8] implemented a de-identification system with Recurrent Neural Networks (RNNs) achieving high scores in the 2014 Informatics for Integrating Biology and the Bedside (i2b2) challenge. Liu et al. [34] proposed a hybrid automatic de-identification system which incorporates subsystems using rules as well as CRFs and Bidirectional Long Short-Term Memory (BiLSTM) networks. They argued that a combined approach is preferable since entities such as phone numbers or email addresses can be detected using simple rules, while other entities such as names or organizations require trained models due to their diversity. Fundamental work on transformer neural networks established by Vaswani et al. [58] raises the question, whether transformers can also lead to advances in anonymizing free texts. Yan et al. [61] suggested to use transformers for NER tasks as an improvement to BiLSTM networks. In addition, Khan et al. [27] showed that transformer encoders can be used for multiple NLP tasks and for specific domains such as the biomedical domain. Finally, Johnson et al. [24] were first to propose a de-identification system using transformer models [58]. Their results indicate that transformers are competitive to modern baseline models for anonymization of free texts.

In addition to the detection of sensitive information using NER, important related work is also on replacement strategies for such information in text. Simple strategies involve suppressing sensitive terms with case-sensitive placeholders [45] or with their types [40]. While those strategies are straightforward to implement, a disadvantage is loss of utility and semantics in the anonymized texts. More complex strategies use surrogates as consistent and grammatically acceptable replacements for sensitive terms [11, 57]. In contrast to the generation of surrogates, Sánchez et al. [47] used generalization to transform sensitive terms to a more general version in order to reduce the loss of utility while still hiding sensitive information.

Work Using Synergies Between Both Fields. Anonymization of structured and unstructured data has mostly been considered in isolation. There were few works using synergies between both fields. Chakaravarthy et al. [5] brought a replacement technique for structured data to the field of unstructured texts. They used properties from k-anonymity to determine the sensitive terms to be anonymized within a single document by investigating their contexts. Moreover, to the best of our knowledge, only Gardner and Xiong [16] studied the task of anonymizing heterogeneous datasets consisting of texts and structured data. They provided a conceptual framework including details on data linking, sensitive information extraction, and anonymization. However, their work has no concept of redundant information between structured and textual data, as we introduce in rx-anon. Furthermore, they have no weighting parameter to balance anonymization based on structural versus textual data like we do. Basically, Gardner and Xiong [16] transfer the problem of text anonymization to the structured world and then their approach forgets about where the attributes came from. They do not transfer back the anonymized sensitive terms to recode the original text. So the output of Gardner and Xiong [16]'s anonymization approach is just structured data, which lacks its original heterogeneous form.

3 PROBLEM STATEMENT

We propose a method on automatically anonymizing datasets which are composed of relational attributes and textual attributes. Our approach is unsupervised and applicable across different domains. In order to achieve this task, we need to explain the process of anonymization, formalize the problem of anonymizing heterogeneous data, and describe our anonymization algorithm which is based on k-anonymity.

3.1 Anonymization Process and Data Attributes

For anonymizing a given dataset, multiple steps are necessary to provide a privacy-preserved release. We refer to release as the anonymized version of a given dataset, but a release does not necessarily have to be made publicly available.

In general, the process of anonymization can be divided into three parts, namely preparation, anonymization, and verification [23]. In the preparation phase, the intended audience is assessed, attributes with their types are named, risks of re-identification attacks are analyzed, and the amount of anonymization is calculated based on the results of the prior steps. The next step involves the anonymization itself, where a dataset and determined parameters are taken as an input, and an anonymized dataset depicts the output. Finally, the verification step requires to assess that the required level of anonymization has been achieved (e. g., by removing all PII) while remaining the utility of the anonymized dataset.

Depending on the dataset to be anonymized, there exist several different attributes which need to be anonymized. We have analyzed the literature and categorize the attributes with respect to the scale of the data (i. e., nominal, ordinal, ratio) and their cardinality of relation (i. e., one-to-one, one-to-many, and many-to-many). While the scale is important to know how attributes can be manipulated in order to achieve anonymity, the cardinality of relation provides information how attributes and individuals relate to each other.

Table 2 contains a non-exhaustive list of attributes, which typically appear in datasets and are critical with respect to re-identification attacks. For the attributes listed in Table 2 we use four scales, namely nominal, ordinal, interval, and ratio. However, interval and ratio can be grouped together as numerical for the anonymization task. Moreover, the cardinality of a relation between an individual and the attribute is to be interpreted as follows: A one-to-many relation means that one individual can have multiple instances of an attribute (e. g., multiple credit card numbers), whereas many-to-one depicts a scenario where many individuals have one property in common (e. g., place of birth). One-to-one and one-to-many attributes directly point to an individual and therefore are considered direct identifiers and must be removed prior to releasing a dataset. However, many-to-one and many-to-many attributes do not reveal an individual directly and therefore are called quasi-identifiers and might remain in an anonymized form in the released version of the dataset.

Even though in Table 2 we present one exclusive cardinality of relation for each attribute, there are always cases where the cardinality of relation depends on context of attributes or whole datasets. An example is home address, where we state that it is a one-to-one attribute. However, this only holds if only one person of a household appears in the dataset. If multiple persons of a household appear in a dataset, we would need to consider it many-to-one. Moreover, if one individual might appear twice with different addresses (e. g., having two delivery addresses in a shop), it would be an one-to-many attribute.

3.2 Problem Formalization

In order to provide a method for anonymizing heterogeneous data composed of relational and textual attributes, we first formalize this problem. In the remainder of this work, we will focus on the task of de-identification of heterogeneous datasets containing traditional relational as well as textual data. We aim to anonymize a dataset by hiding directly identifying attributes. To prevent classical record linkage attacks using quasi-identifying attributes, we use k-anonymity as our privacy model [53]. In general, identification threats based on information within textual documents can be categorized into two categories, where the former poses explicit and the latter poses implicit information leakage [48]. Within texts, we adapt k-anonymity to prevent explicit information leakage, while keeping the structure of the texts as best as possible intact to allow for text mining on implicit information. In other words, using our privacy model, an attacker shall not be able to identify an individual based on attributes, their values, or sensitive terms in texts. However, obfuscating personal writing style as discussed in [13, 37] exceeds this work and is therefore not considered. Table 3 provides an overview of the notations used.

Heterogeneous RX-dataset. Given a dataset D in form of n relations $R_1, ..., R_n$, containing both relational and textual attributes. D contains all data we want to anonymize. We pre-process D for the anonymization process by using the natural join, i. e., $D = R_1 \bowtie ... \bowtie R_n$, i. e., we "flatten" the relational structures. Table 1 shows an example of a dataset composed of two joined relations,

Table 2: Non-exhaustive list of attributes to anonymize with scale and cardinality of relations (sorted by cardinality). Note, for the anonymization task the interval and ratio data can be grouped together as numerical.

Attribute	Scale	Cardinality of Relation
Name [7, 36, 42, 56]	nominal	one-to-one
Social Security Number [36, 56]	nominal	one-to-one
Online identifier [7]	nominal	one-to-one
Passport Numbers [7, 36]	nominal	one-to-one
Home Address [7, 36, 56]	nominal	one-to-one
Credit Card Number [36]	nominal	one-to-many
Phone [36, 56]	nominal	one-to-many
Email Address [7, 36, 56]	nominal	one-to-many
License Plate Number [56]	nominal	one-to-many
IP Address [7, 36, 56]	nominal	one-to-many
Order Reference	nominal	one-to-many
Age [42, 56]	ratio	many-to-one
Sex / Gender [7, 42]	nominal	many-to-one
ZIP / Postcode [56]	nominal	many-to-one
Date of Birth [36, 56]	interval	many-to-one
Zodiac Sign [49]	ordinal	many-to-one
Weight [7, 36]	ratio	many-to-one
Race [7, 36]	nominal	many-to-one
Country	nominal	many-to-one
City [56]	nominal	many-to-one
Salary Figures [14]	ratio	many-to-one
Religion [7, 36]	nominal	many-to-one
Ethnicity [7]	nominal	many-to-one
Employment Information [36]	nominal	many-to-one
Place of Birth [36]	nominal	many-to-one
Skill	nominal	many-to-many
Activities [36]	nominal	many-to-many
Diagnosis / Diseases [14, 36]	nominal	many-to-many
Origin / Nationality [42]	nominal	many-to-many
Purchased Products [42]	nominal	many-to-many
Work Shift Schedules	nominal	many-to-many

where the first relation describes the individuals (*id*, *gender*, *age*, *topic*, *sign*), while the latter relation (*id*, *date*, *text*) contains the posts and links them to an individual with *id* being the foreign key. We call D an RX-dataset, if one attribute A^* directly identifies an individual, one or more traditional relational attributes[2] A_i contain single-valued data, and one textual attribute[3] X is in D. In other words, an RX-dataset is any dataset, which contains at least one directly identifying attribute, one or more quasi-identifying attributes, and one or more textual attributes. For the remainder of this work, we will use relational attributes for attributes we consider traditional relational and textual attributes for attributes with

[2]By traditional relational attributes we refer to numerical, date, or categorical attributes. Categorical attributes might even be composed of multiple terms (e. g., names or full addresses).

[3]For the ease of reading we explore only one textual attribute. However, our approach can be extended for multiple textual attributes $X_1, ..., X_m$.

Table 3: Notation for a given RX-Dataset D.

D	original dataset, $D = R_1 \bowtie ... \bowtie R_n$
R_i	relation of D
A^*	attribute of D identifying an individual directly
A_i	attribute of a relation R_j
X	textual attribute
t	tuple in D
D^*	person centric view on D
r	record (tuple) in D^*
D'	anonymized dataset
X'	set of all non-redundant sensitive terms of X
T	some text in the form of a sequence of tokens
$\mathbf{F}$	set of aggregation functions, $\mathbf{F} = \{\mathbf{F_1}, ..., \mathbf{F_n}, \mathbf{F_{X'}}\}$
E	set of sensitive entity types
er	entity recognition function, $er : T \rightarrow E$
$emap$	mapping function, $emap : \{A_1, ..., A_n\} \rightarrow E$

textual values composed of multiple words or even sentences. In the example in Table 1, the relational attributes are the direct identifier *id* as well as the quasi-identifiers *gender*, *age*, *topic*, and *date*. The textual attribute is *text*. We call a row in D a tuple t. Relational attributes A_i are single-valued and can be categorized into being nominal, ordinal, or numerical (i. e., ratio or interval, which are treated equally in the anonymization process). A textual attribute X is any attribute, where its domain is some form of free text. Therefore, we can state that $t.X$ consists of an arbitrary sequence of tokens $T =< t_1, ..., t_m >$.

Sensitive Entity Types. We define E to be a set of entity types, where each value $e \in E$ represents a distinct entity type (e. g., person or location) and each entity type is critical for the anonymization task. We then define a recognition function er on texts as $er: T \rightarrow E$. The recognition function detects sensitive terms in the text T and assigns a sensitive entity type $e \in E$ to each token $t \in T$. Moreover, we define a mapping function $emap$ on the set of structural attributes as $emap: \{A_1, ..., A_n\} \rightarrow E$. The mapping function $emap$ maps attributes $A_1, ..., A_n$ to a sensitive entity type in E, which is used to match redundant sensitive information with the text.

Redundant Sensitive and Non-redundant Sensitive Terms. Some sensitive information might appear in a textual as well as in a relational attribute. In order to consistently deal with those occurrences, we introduce the *concept of redundant sensitive information*. Redundant sensitive information is any sensitive term $x \in t.X$ with $er(x) = e_j$ for which a relational value $v \in t.A_i$ with $emap(A_i) = e_j$ exists, where $x = v$. In other words, redundant sensitive information is duplicated information, i. e., has the same value which appears under the same sensitive entity type e_j in a relational attribute $t.A_i$ and a sensitive term x in $t.X$.

We introduce the attribute X', which contains all *non-redundant* sensitive information of X. For the remainder of this work, attribute names with apostrophes indicate that these attributes contain the extracted sensitive entities with their types (see *text'* in Table 4). We model X' as a set-valued attribute since in texts of $t.X$, zero or more sensitive terms can appear. Therefore, we explicitly allow empty sets to appear in $t.X'$ if no sensitive information appears in $t.X$. We then replace X in D with X', so that the schema of D becomes $\{A^*, A_1, ..., A_n, X'\}$.

Person Centric view D^* on the Dataset D. If a dataset D is composed of multiple relations, there might be multiple tuples t which correspond to a single individual. In order to apply anonymization approaches on this dataset, we need to group the data in a person centric view similar to Gong et al. [18], where one record r (i. e., one row) corresponds to one individual. Therefore, we define D^* being a grouped and aggregated version of D. This means, that we can retrieve D^* from D as $D^* =_{A^*} G_{\mathbf{F_1}(A_1),...,\mathbf{F_n}(A_n),\mathbf{F_{X'}}(\mathbf{X'})}(D)$, where A^* denotes a directly identifying attribute related to an individual used to group rows of individuals together, G concurrently applies a set of aggregation functions $\mathbf{F_i}$ and $\mathbf{F_{X'}}$ defined on relational attributes A_i as well as sensitive textual terms X'. This aggregation operation should create a person centric view of D by using appropriate aggregation functions $\mathbf{F} = \{\mathbf{F_1}, ..., \mathbf{F_n}, \mathbf{F_{X'}}\}$ on the attributes. For relational attributes A_i, we use **set** as a suitable aggregation function, where two or more distinct values in A_i for one individual result in a set containing all distinct values. For set-based attributes like X', we use the aggregation function **union**, which performs an element-wise union of all sets in X' related to one individual. Table 4 presents a person centric view of our initial example where each record r represents one individual. Dates as well as any non-redundant sensitive terms have been aggregated, as discussed.

k-anonymity in D^*. Based on the notion of equivalence classes [42] and the definition of equality of set-based attributes [21], an equivalence class for D^* can be defined as a partition of records P where for any two records $r, s \in P$ holds $(r.A_1, ..., r.A_n) = (s.A_1, ..., s.A_n)$ and $r.X' = s.X'$. Thus, within an equivalence class each record has the same values for the relational attributes and their sets of sensitive terms have the same values, too. Given our definition of equivalence classes, a person centric dataset D^* is said to be k-anonymous if all equivalence classes of D^* have at least the size k. We refer to the k-anonymous version of D^* as D'. D' protects privacy by hiding direct identifiers. Moreover, since each of the quasi-identifying attributes and sensitive terms in texts appear at least k times, D' also protects against record linkage attacks.

4 RX-ANON APPROACH

Using the definitions from Section 3, we present our anonymization approach rx-anon. We present how we preprocess our data to generate a person centric view. We show how Mondrian [31], a recursive greedy anonymization algorithm, can be used to anonymize *RX*-datasets. Mondrian transforms a dataset into a k-anonymous version by partitioning the dataset into partitions with sizes greater than k and afterwards recodes each partition individually. We introduce an alternative partitioning strategy called Global Document Frequency (GDF) as baseline for partitioning a dataset with sensitive terms. We use the running example (Table 1) to show how an *RX*-dataset is transformed to a privacy-preserved version.

4.1 Pre-processing D to Person-centric View D^*

Prior to anonymizing an *RX*-dataset, it needs to be transformed into a person specific view in order to apply k-anonymity. Using the running example from Table 1, we demonstrate the steps involved to create the person centric view shown in Table 4. First, we identify sensitive terms in the texts and assign sensitive entity types to them. In the remainder of this work, we will use subscripts to indicate the entity type assigned to a sensitive term. Given the first row of the example in Table 1, the text is "My name is Pedro, I'm a 36 years old engineer from Mexico". The sensitive terms are $Pedro_{\text{person}}$, $36\ years\ old_{\text{age}}$, $engineer_{\text{job}}$, and $Mexico_{\text{location}}$. This analysis of texts is executed for all tuples t in D, while there can be multiple sensitive terms from the same entity type within a text, or even no sensitive terms at all. In the next step, we find and mark redundant sensitive information using the results of the prior steps. Therefore, we perform row-wise analyses of relational values with sensitive terms to find links, which actually represent the same information. In our example above, the sensitive term 36 years old_{age} depicts the same information as the value *36* in the relational attribute *age*. Therefore, this sensitive term in the textual attribute is marked as redundant and is not considered as new sensitive information during the anonymization algorithm. Non-redundant sensitive information is stored in the attribute *text'*.

Finally, we build a person-centric view to have a condensed representation of all information available for each individual. Therefore, as described in Section 3, we group the data on a directly identifying attribute to get an aggregated dataset. In the example in Table 1, the directly identifying attribute A^* is *id*. We use **set** as the aggregation function for the relational attributes. Moreover, we collect all sensitive terms mentioned in texts of one individual by performing **union** on the sets of sensitive terms.

Table 4 shows the person-centric view D^* of our dataset D, which has been achieved by aggregating on the attribute *id*. Since the individuals with the *ids 1* and *4* have blogged more than once on different dates, multiple dates have been aggregated as sets. Moreover, since those people also have blogged different texts on different days, all sensitive terms across all blog posts have been collected in the attribute *text'*.

4.2 Compute Anonymized Dataset D' from D^*

Given a person centric dataset D^*, we want to build a k-anonymous version D' by using the definitions of the previous section. In order to achieve anonymization, we adapt the two step anonymization algorithm of Mondrian by LeFevre et al. [31], which first decides on m partitions $P_1, ..., P_m$ (refer to Algorithm 1), and afterwards recodes the values of each partition to achieve k-anonymity. We use Global Document Frequency (GDF) partitioning as baseline partitioning algorithm (see Algorithm 2), which uses sensitive terms and their frequencies to create a greedy partitioning using presence and absence of sensitive terms.

Modified Mondrian Partitioning with Weight Parameter λ. The first step of the algorithm is to find partitions of records with a partition size of at least k. LeFevre et al. [31] introduced multi-dimensional strict top-down partitioning where non-overlapping partitions are found based on all relational attributes. Moreover, they introduced a greedy strict top-down partitioning algorithm Mondrian. Starting with the complete dataset D^* as an input, the partitioning algorithm chooses an attribute to split on and then splits the partition by median-partitioning. The authors suggest to use the attribute which provides the widest normalized range given a sub-partition. For numerical attributes, the normalized range is defined as minimum to maximum. For categorical attributes, the

Table 4: Preprocessed version of the illustrative example. The attribute *date* has been aggregated as set. The attribute *text′* contains sensitive terms of the attribute *text* for all blog posts published by a single individual.

id	gender	age	topic	sign	date	text′
1	male	36	Education	Aries	2004-05-14, 2004-05-15	$\text{Pedro}_{\text{person}}$, $\text{engineer}_{\text{job}}$, $\text{Mexico}_{\text{location}}$
2	male	24	Student	Leo	2005-08-18	$\text{engineer}_{\text{job}}$
3	male	37	Banking	Pisces	2004-05-27	$\text{Ben}_{\text{person}}$, $\text{Canada}_{\text{location}}$
4	female	24	Science	Aries	2004-01-13, 2004-01-17, 2004-01-19	$\text{Four days ago}_{\text{date}}$, $\text{scientist}_{\text{job}}$, $\text{biologist}_{\text{job}}$, $\text{UK}_{\text{location}}$
5	male	29	indUnk	Pisces	2004-05-15	
6	female	27	Science	Aries	2004-05-15	$\text{UK}_{\text{location}}$

range is the number of distinct categories observed in a partition. Sensitive, textual terms are treated as categorical attributes.

In order to properly treat textual terms in this heuristic algorithm, we introduce a weight parameter λ to the modified Mondrian algorithm shown in Algorithm 1. λ can be a value between 0 and 1. It describes the priority to split partitions on relational attributes. $\lambda = 1$ means that the algorithm always favors to split on relational attributes. $\lambda = 0$ leads to splits only based on sensitive terms in textual attributes. $\lambda = 0.5$ does not influence the splitting decisions and therefore is considered as default. The partitioning algorithm stops if no allowable cut can be made such that the criteria of k-anonymity holds for both sub-partitions. Therefore, we can stop splitting partitions if $|P| < 2k$.

Algorithm 1: Modified Mondrian partitioning with weight parameter λ (adapted from LeFevre et al. [31]). It applies a greedy strict top-down partitioning for relational attributes.

Input : Partition P, weight λ
Output : Set of partitions with size of at least k

```
Function mondrian_partitioning(P, λ):
    if |P| < 2k then // no allowable cut
        return P
    end
    else
        A = next_attribute(λ)
        F = frequency_set(P, A)
        P_l = (r ∈ P | r.A < find_median(F))
        P_r = P \ P_l
        return mondrian_partitioning(P_l) ∪
          mondrian_partitioning(P_r)
    end
```

Global Document Frequency (GDF) Partitioning. Using the idea of a top-down strict partitioning algorithm, we propose with GDF a greedy partitioning algorithm using the presence and absence of sensitive terms. The main goal is to keep the same sensitive terms within the same partition. This is achieved by creating partitions with records which have sensitive terms in common. Algorithm 2 presents the GDF partitioning algorithm, which is based on sensitive terms and their frequencies. Similar to Algorithm 1, we start with the whole dataset as a single partition. Instead of splitting the partition using the median of a relational attribute (Mondrian partitioning), *we split partitions on a chosen sensitive term.* While the first sub-partition contains only records, where the chosen sensitive term appears, the second sub-partition contains the remaining records. For choosing the next term to split on, multiple heuristics are possible. We propose to use the most frequently apparent sensitive term for the remaining texts in the partition as the term to split on. Taking the most frequent term allows us to keep the most frequently appearing term in a majority of texts while suppressing less frequently used terms. The term used to split is then removed and similar to Algorithm 1 the algorithm is recursively called using the first and second partition, respectively. GDF partitioning guarantees that records are partitioned such that sensitive terms in texts are tried to be kept by grouping records with same terms. Moreover, records with no or less frequently used sensitive terms are also included in one partition. Therefore, we build partitions with records which would prevent other partitions from being k-anonymous.

Algorithm 2: Top-down document-frequency-based (GDF) partitioning on sensitive terms in X'.

Input : Partition P, terms with their frequencies F
Output : Set of partitions with size of at least k

```
Function gdf_partitioning(P, F):
    if |P| < 2k then // no allowable cut
        return P
    end
    else
        (x, f) = next_term(P, F)
        P_l = (r ∈ P | x ∈ r.X′)
        P_r = P \ P_l
        F = F \ {(x, f)} // remove considered term
        return gdf_partitioning (P_l, F) ∪ gdf_partitioning
          (P_r, F)
    end
```

Example. Using the running example in Table 4 with $k = 2$ and the GDF partitioning scheme, partitioning is achieved as follows. Starting with the initial complete dataset D^* (person-centric view) as the initial partition P, we determine the most frequent term, which is either *UK* or *engineer*, both appearing twice. Without loss of generality, we assume *engineer* is chosen as the term to split on. Then we split $P = \{1, 2, 3, 4, 5, 6\}$ in $P_l = \{1, 2\}$ containing all records where *engineer* appears and $P_r = \{3, 4, 5, 6\}$ containing the remaining records. For P_l, no allowable cut can be made since $|P_l| = 2$. However, the algorithm continues with P_r since $|P_r| = 4$, and splits P_r on *UK* as the most frequent term appearing twice

in records within P_r. This will lead to two new partitions Using $P_{rl} = \{4, 6\}$ containing records where *UK* appears in the texts and $P_{rr} = \{3, 5\}$ containing the remaining records. Finally, the algorithm results in an optimal partitioning with three partitions, each consisting of two records. In our case, we refer to optimal as a partition layout with the least amount of information loss within the textual attribute.

Recoding. In the next step, each partition is transformed such that values of quasi-identifiers of records are indistinguishable. This process is called recoding. Recoding can either be global [4] or local [31, 59]. Local recoding generalizes values per equivalence class, but equal values from two equivalence classes might be recoded differently. In contrast, global recoding enforces that the same values are recoded equally throughout the entire dataset. Since global recoding requires a global replacement of values with appropriate recoded values, the search space for appropriate replacements may be limited [30]. Therefore, even though global recoding might result in more consistent releases of data, local recoding appears to be more powerful due to its variability in finding good replacements. There are different recoding schemes for the different scales of the attribute. Nominal and ordinal values are usually recoded using Domain Generalization Hierarchies (DGHs) as introduced by Sweeney [52] and used in multiple other works [17, 41, 42, 59]. A DGH describes a hierarchy which is used to generalize distinct values to a more general form such that within a partition all values transform to a single value in the DGH. Generating DGHs is usually considered a manual effort, while there already exist approaches on automatically generating concept hierarchies as introduced by Lee et al. [29], which have also been used in work on anonymization [21]. Alternatively, nominal and ordinal attributes can also be recoded as sets containing all distinct items of one partition. For numerical attributes, LeFevre et al. [31] propose to use either mean or range as a summary statistic. Additionally, numerical attributes can also be recoded using ranges from minimum to maximum. Moreover, for dates El Emam et al. [12] propose an automated hierarchical recoding based on suppressing some information of a date value shown in Figure 1. The leaf nodes represent actual dates appearing in the dataset D (ref. to Table 1). Non-leaf nodes represent automatically generated values by suppressing information on each level.

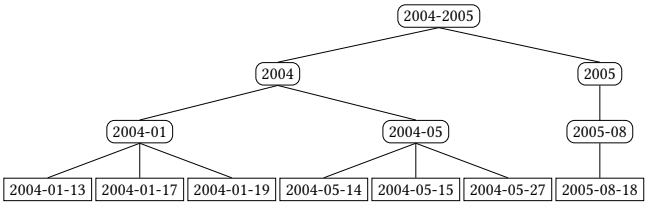


Figure 1: Domain Generalization Hierarchy for date attributes. The leaves depict actual values appearing in the dataset D. The first level of generalization involves suppressing the day. The second level of generalization suppresses the month. Finally, the root can be automatically generated as a range of years.

Since we use a strict-multidimensional partitioning scheme, we apply local recoding as suggested in Mondrian [31]. For numerical attributes, we use range as a summary statistic. For date attributes we use the automatically generated DGH by El Emam et al. [12] as shown in Figure 1. Moreover, since generalization hierarchies for gender, topic, and sign are flat, we recode nominal and ordinal values as sets of distinct values.

Example. After equivalence classes have been determined, relational attributes can be recoded. Table 5 shows how those recoding schemes are applied to the relational attributes of our running example. In addition, a k-anonymous representation of the text attribute X' has to be created. Terms, which are marked as redundant sensitive information, are replaced by the recoded value of its relational representatives. Using the anonymized version of our example in Table 5, the age appearing in the text of the first row is recoded using the value of the attribute *age* of the same row. Moreover, non-redundant sensitive information is recoded using suppression with its entity type. If a sensitive information appears within all records of an equivalence class, retaining this information complies with our definition of k-anonymity for set-valued attributes from Section 3. Therefore, it does not need to be suppressed (see sensitive term *engineer* in Table 5). However, if the same sensitive information is not appearing in every record within an equivalence class, this sensitive information (or the lack of it) violates our definition of k-anonymity and must be suppressed. An example for such a violation in Table 5 is *Mexico*, which appears in the first record, but in no other record of its equivalence class. The result is the k-anonymized dataset D^*.

5 EXPERIMENTAL APPARATUS

We evaluate our framework on two real-world datasets using the modified Mondrian partitioning algorithm with weighting parameter λ as well as the GDF partitioning baseline. We use λ to manipulate the splitting decisions in Mondrian as discussed in Section 4 and measure the resulting partitions as well as information loss.

5.1 Datasets

We require datasets that include a directly identifying attribute A^*, one or more quasi-identifying relational attributes A_i, and one or more textual attributes X containing sensitive information about individuals (refer to the definition of an *RX*-dataset in Section 3). We use the publicly available *Blog Authorship Corpus* and *515K Hotel Reviews Data in Europe* datasets.

Blog Authorship Corpus. The Blog Authorship Corpus[4] was originally used to create profiles from authors [49] but has also been used in privacy research for author re-identification [28]. After cleaning the input data from unreadable characters and others, the corpus contains 681,260 blog posts from 19,319 bloggers, which have been written by a single individual on or before 2006 and published on blogger.com. While the vast majority of blog posts are written in English language, the corpus contains some posts written in other languages. However, non-English blog posts are the minority and therefore do not have a significant impact on the experiment results. A row in the corpus consists of the *id*, *gender*,

[4]https://www.kaggle.com/rtatman/blog-authorship-corpus

Table 5: Anonymized dataset D' of the running example for $k = 2$. Redundant information and remaining sensitive terms are marked bold.

id	gender	age	topic	sign	date	text
1	male	**[24-36]**	(Student,Education)	(Leo,Aries)	[2004-2005]	My name is **person**, I'm a **[24-36]** years old **engineer** from **location**.
1	male	[24-36]	(Student,Education)	(Leo,Aries)	[2004-2005]	A quick follow up: I will post updates about my education in more detail.
2	male	[24-36]	(Student,Education)	(Leo,Aries)	[2004-2005]	I will start working for a big tech company as an **engineer**.
3	male	[29-37]	(indUnk,Banking)	Pisces	2004-05	During my last business trip to **location** I met my friend **person** from college.
4	female	[24-27]	Science	Aries	2004	As a **job** from the **UK**, you can be proud!
4	female	[24-27]	Science	Aries	2004	**Date**, I started my blog. Stay tuned for more content.
4	female	[24-27]	**Science**	Aries	**2004**	**2004** will be a great year for **science** and for my career as a **job**.
5	male	[29-37]	(indUnk,Banking)	**Pisces**	2004-05	Did you know that **Pisces** is the last constellation of the zodiac.
6	female	[24-27]	Science	Aries	2004	Rainy weather again here in the **UK**. I hope you all have a good day!

age, *topic*, and *zodiac sign* of a blogger as well as the *date* and the *text* of the published blog entry. Each row corresponds to one blog post written by one individual, but one individual might have written multiple blog posts. On average, one blogger has published 35 blog posts. We treat *id* as a direct identifier, *gender*, *topic*, and *sign* as categorical attributes, while *age* is treated as a numerical attribute. The attribute *date* is treated as a special case of categorical attribute where we recode dates using the automatically generated DGH shown in Figure 1. The attribute *text* is used as the textual attribute. The attribute *topic* contains 40 different topics, including industry-unknown (indUnk). *Age* ranges from 13 to 48. *Gender* can be male or female. *Sign* can be one of the twelve astrological signs.

In addition to the Blog Authorship Corpus, we run experiments on a second dataset to verify our observations. We chose to use a dataset containing reviews of European hotels. We refer to this dataset as the *Hotel Reviews Dataset*.

Hotel Reviews Dataset. We use the 515K Hotel Reviews Data in Europe dataset[5], called in the following briefly the *Hotel Reviews Dataset*, which contains 17 attributes, of which 15 attributes are relational and two attributes are textual. The textual attributes are *positive* and *negative reviews* of users. Among the relational attributes, we treat *hotel name* and *hotel address* as direct identifiers. The textual attributes are pre-processed and cleaned as described for the Blog Authorship Corpus. *Negative* and *positive word count* as well as *tags* are ignored and therefore considered insensitive attributes. The remaining attributes are treated as quasi-identifiers, with seven numerical, one date, and two nominal attributes. We recode all quasi-identifying attributes similar to the Blog Authorship Corpus. After preparing the Hotel Reviews Dataset, we have 512, 126 reviews for 1, 475 hotels remaining.

5.2 Procedure

As baselines, we consider the scenario where relational and textual attributes are anonymized independently. Usually, sensitive terms within a textual attribute are suppressed completely, which leads to total loss of utility of sensitive terms. With our experiments we want to show that we can improve, i. e., reduce the information loss in texts under the k-anonymity model. Moreover, we want to optimize the trade-off between relational and textual information loss.

Similar to experiments conducted in prior work [17, 18, 42], we run our anonymization tool for different values of $k = 2$, 3, 4, 5, 10, 20, and 50. Regarding our new weighting parameter λ, we used values between 0.0 and 1.0 in steps of 0.1. Sensitive entity types in texts are those detected by spaCy's English models trained on the OntoNotes5 corpus[6]. We added rule-based detectors for the entities MAIL, URL, PHONE, and POSTCODE. We treat all sensitive terms appearing under those entity types as quasi-identifiers. For each value of k, we conduct experiments using different partitioning strategies and parameter settings. In particular, we vary the weight parameter λ to tune Mondrian. To speed up experiment execution times, we ignore redundant sensitive information. Ignoring redundant sensitive information does not influence the experiment results, since both datasets do not provide a relevant amount of overlap between relational attributes and textual attributes. We use local recoding schemes for each experiment to make partitioning results comparable. For the evaluation, we analyze the anonymized dataset with respect to the corresponding partitioning sizes and information loss.

In addition, we repeat the experiments by just considering location entities with entity type GPE (geopolitical entity). We use those experiments to showcase an anonymization task with reduced complexity. We chose location-based entities since they are present in blog posts as well as in hotel reviews. Therefore, they allow for comparison of both datasets.

[5]https://www.kaggle.com/jiashenliu/515k-hotel-reviews-data-in-europe

[6]https://spacy.io/api/data-formats#named-entities

5.3 Measures

In order to evaluate our anonymization approach and compare results of partitioning, we introduce the following measures. In particular, we compare statistics on partitions as well as relational and textual information loss.

Statistics on Partitions. We are interested in the resulting partitions of the anonymized dataset.

Number of splits (based on relational versus textual attributes): We evaluate how partitions are created, based on relational attributes versus textual attributes, and how λ influences splitting decisions. We expect that for $\lambda < 0.5$ we observe more splits on textual attributes and for $\lambda > 0.5$ more splits on relational attributes.

Number of partitions and Partition sizes: In addition to the number of splits, we want to evaluate the size of the resulting partitions since they are closely related to information loss. By the nature of k-anonymity, all partitions need to be at least of size k. Relatively large partitions with respect to k will tend to produce more information loss. Therefore, partition sizes closer to k will be favorable and increase utility. We evaluate resulting partitions by counting the number of partitions, as well as calculating the mean and standard deviation of partition sizes.

Information Loss (Adapted to Heteregenous Datasets). Measuring the information loss of an anonymized dataset is well-known practice for evaluating the amount of utility remaining for a published dataset. We use Normalized Certainty Penalty (NCP) [59] to determine how much information loss has been introduced by the anonymization process. In particular, the NCP assigns a penalty to each data item in a dataset according to the amount of uncertainty introduced. We extend the definitions of NCP to the problem of anonymizing relational and textual data such that for one record r, the information loss is calculated as $NCP(r) = (w_R \cdot NCP_A(r) + w_X \cdot NCP_X(r))/(w_A + w_X)$, where w_A is the importance assigned to the relational attributes, and $NCP_A(r)$ denotes the information loss for relational attributes of record r. Analogously, we define w_X and $NCP_X(r)$ for the textual attribute. For our evaluation, we set w_A and w_X to 1, i. e., weigh the loss stemming from relational data and textual data equally. Note, that this decision is independent of the λ parameter, which decides which attribute or term is actually used for the splitting of the partitions.

For *relational attributes* $A = \{A_1, ..., A_n\}$ we define the information loss $NCP_A(r) = (\sum_{A_i \in A} NCP_{A_i}(r))/|A|$, where $|A|$ denotes the number of relational attributes. NCP_{A_i} is the information loss for a single attribute and depends on the type of attribute. It can be calculated either using NCP_{num} for numerical attributes or NCP_{cat} for categorical attributes. NCP_{num} for numerical values is defined as $NCP_{num}(r) = (z_i - y_i)/|A_i|$, with z_i being the upper and y_i being the lower bound of the recoded numerical interval and $|A_i| = max_{r \in D^*}(r.A_i) - min_{r \in D^*}(r.A_i)$. For categorical values, NCP_{cat} is defined as $NCP_{cat}(r) = \begin{cases} 0 & |u| = 1 \\ \frac{|u|}{|A_i|} & \text{otherwise} \end{cases}$, where $|u|$ denotes the number of distinct values which the recoded value u describes. For categorical values other than dates, $|u|$ will be the number of distinct values appearing in the recoded set. For date attributes, $|u|$ denotes the number of leaves of the subtree below the recoded value (see Figure 1).

For *textual attributes*, we define $NCP_X(r) = (\sum_{x \in r.X'} NCP_x(x))/|r.X'|$, where for each sensitive information x, we calculate the individual information loss $NCP_x(x)$ and normalize it by the number of sensitive terms $|r.X'|$. We define the individual information loss for one sensitive term as $NCP_x(x) = 1$ if x is suppressed, and 0 otherwise.

Finally, we can calculate the total information loss for an entire RX-Dataset D^* as $NCP(D^*) = (\sum_{r \in D^*} NCP(r))/|D^*|$, where for each record r the information loss $NCP(r)$ is calculated and divided by the number of records $|D^*|$.

6 RESULTS

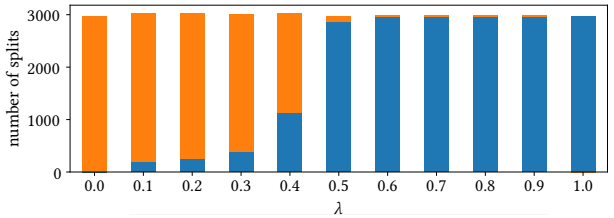
We present the results regarding partition statistics and information loss. For detailed experimental results with plots and tables for all parameter values, we refer to the supplementary material. In particular, details on the influence of λ and k on the splitting decision for both datasets can be found in the supplementary material in Appendix A.3.1. Detailed tables on the influence of λ and k on the partition count and size can be found in Appendix A.3.2. Detailed figures of the information loss, including zoomed plots, for the experiments run on the Blog Authorship Corpus and Hotel Reviews Dataset can be found in Appendix A.4. Figures with a detailed comparison of information loss per entity type in the Blog Authorship Corpus and Hotel Reviews Dataset for different values of λ can also be found in Appendix A.4.

6.1 Partitions Splits, Counts, and Size

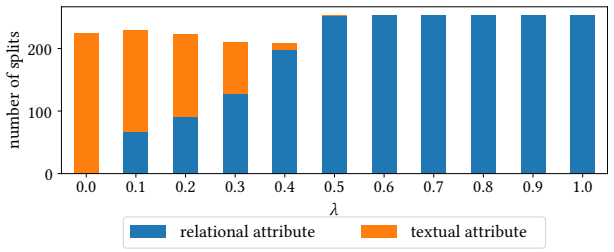
To modify splitting decisions and therefore the distribution of information loss between relational and textual attributes, we introduced the tuning parameter λ to Mondrian partitioning. Thus, first, we verify how λ impacts splitting decisions. We count for a particular λ how often partitions are effectively split on a relational attribute and compare this metric to the number of splits on sensitive terms of textual attributes. We also evaluate the count of the resulting partitions and the partitions' sizes.

Partition Splits. Figure 2a shows the distribution of splitting decisions for experiments run on the Blog Authorship Corpus for $k = 5$ and $\lambda = 0.0$ to 1.0. As λ was designed, $\lambda = 1$ results in only splits on relational attributes whereas $\lambda = 0$ results in splits only on sensitive terms. As our results show, an unbiased run of Mondrian with $\lambda = 0.5$ causes partitions to be split mostly on relational attributes. Since the span of relational attributes is lower compared to sensitive terms, relational attributes provide the widest normalized span and are therefore favored to split on. For $\lambda > 0.5$, the majority of the weight for splitting is given to the relational attributes. Thus, there is no relevant change since relational attributes are considered almost every time throughout the partitioning phase. However, for $\lambda < 0.5$, we observe that more and more splits are made based on textual attributes. For $\lambda = 0.4$, already more than half of the splits are based on textual terms. If only locations, i. e., entities of type GPE, are considered, λ is not in all cases able to control the share of splits between relational and textual attributes, since low values for λ do not result in more splits on textual attributes. Plots are omitted for brevity here and can be found in the supplementary materials.

For the Hotel Reviews Dataset, shown in Figure 2b, the number of splits is generally lower (see also partition sizes, below), since it contains less records. Also the splitting on textual attributes is less likely for hotel reviews compared to blog posts. In the case of experiments considering only location entities, the impact of λ is even smaller and splits are mostly performed on relational attributes. Details are provided in the supplementary materials.



(a) Textual vs. relational splits on the Blog Authorship Corpus



(b) Textual vs. relational splits on the Hotel Reviews Dataset

Figure 2: Number of splits based on textual attributes (orange) versus relational attributes (blue) using Mondrian partitioning ($k = 5$) with varying weights λ on the (a) Blog Authorship Corpus and (b) Hotel Reviews Dataset.

Partition Count and Size. Regarding the number of partitions and their size, Table 6 provides statistics on partitions using Mondrian partitioning with varying λ as well as GDF partitioning for the Blog Authorship Corpus. In the table, *count* refers to the number of partitions produced under the specific values of k and λ, while *size* refers to the average number of records per partition. The Mondrian partitioning algorithm produces the same partitioning layout for λ between 0.6 and 0.9. This observation matches statistics on partition splits, since for these values of λ the Mondrian algorithm decides to use the same attributes to split on. Furthermore, GDF partitioning is not able to generate partition sizes close to k, compared to Mondrian partitioning. Table 7 shows the results when only location entities are considered. Here, $\lambda = 0$ leads to bigger and fewer partitions compared to other settings for λ. Comparing GDF to Mondrian with $\lambda = 0$, we observe that for low numbers of k, GDF partitioning achieves in general smaller, but more variable partitions with regard to size. However, for larger values of k, Mondrian partitioning achieves better distribution of partitions and therefore better distribution of sensitive terms.

The results for the Hotel Reviews Dataset are shown in Table 8. Table 9 shows the results for the location type only. We make the same observations as for the Blog Authorship Dataset. However, due to the lower number of records in the Hotel Reviews Dataset, the total count of partitions is comparatively smaller.

Table 6: Statistics on partitions for Blog Authorship Corpus considering all entity types. Count refers to the number of partitions found, while size refers to the average number of records per partition.

	λ	k	3	4	5	10	20
GDF	-	count	2479	1512	1078	352	92
		size	7.79	12.78	17.92	54.88	209.99
Mondrian	0	count	5162	3795	2971	1412	692
		size	3.74	5.09	6.50	13.68	27.92
	0.3	count	5236	3841	3023	1462	707
		size	3.69	5.03	6.39	13.21	27.33
	0.5	count	5198	3800	2979	1441	711
		size	3.72	5.08	6.49	13.41	27.17
	0.6 - 0.9	count	5180	3781	2987	1441	711
		size	3.73	5.11	6.47	13.41	27.17
	1	count	5128	3749	2964	1431	703
		size	3.77	5.15	6.52	13.50	27.48

Table 7: Statistics on resulting partitions for Blog Authorship Corpus considering only GPE entities

	λ	k	3	4	5	10	20
GDF	-	count	737	301	129	2	1
		size	26.21	64.18	149.76	9659.50	19319
Mondrian	0	count	1164	828	703	392	242
		size	16.60	23.33	27.48	49.28	79.83
	0.3	count	5248	3780	2994	1443	700
		size	3.68	5.11	6.45	13.39	27.60
	0.5	count	5154	3762	2970	1433	703
		size	3.75	5.14	6.50	13.48	27.48
	0.6 - 0.9	count	5153	3761	2970	1433	703
		size	3.75	5.14	6.50	13.48	27.48
	1	count	5128	3749	2964	1431	703
		size	3.77	5.15	6.52	13.50	27.48

6.2 Information Loss

In addition to the statistics on partition splits, counts, and sizes, we are interested in how the partitioning performs with respect to the introduced information loss measure. Figure 3a provides an overview on relational information loss NCP_A (y-axis) for different values of k between 2 and 50 (x-axis) for the Blog Authorship Corpus. Figure 3b shows the textual information loss NCP_X. Results for λ between 0.6 and 0.9 are not plotted, since they are almost identical to the run using $\lambda = 0.5$. Figures 4a and 4b provide the information loss for experiments run on the Hotel Reviews dataset.

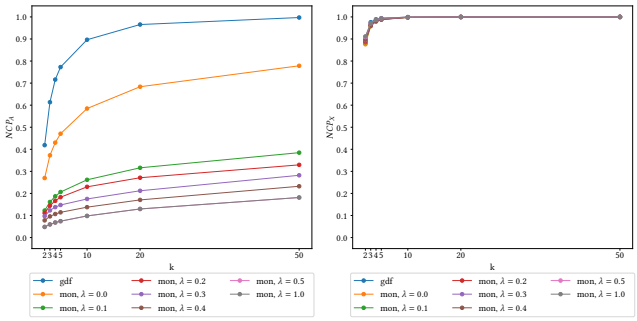
Relational Information Loss. The information loss increases with larger k throughout all experiments. Higher information loss is caused by having larger partitions and therefore higher efforts in recoding. Furthermore, we can state that information loss in the relational attributes increases if the tuning parameter λ decreases (see

Table 8: Statistics on partitions for Hotel Reviews Dataset considering all entity types. Count refers to the number of partitions found, while size refers to the average number of records per partition.

	λ	k	3	4	5	10	20
GDF	-	count	272	163	127	43	16
		size	5.42	9.05	11.61	34.30	92.19
Mondrian	0	count	398	293	226	117	54
		size	3.71	5.03	6.53	12.61	27.31
	0.3	count	404	285	212	106	50
		size	3.65	5.18	6.96	13.92	29.50
	0.5	count	415	256	255	128	64
		size	3.55	5.76	5.78	11.52	23.05
	0.6 - 1	count	417	256	255	128	64
		size	3.54	5.76	5.78	11.52	23.05

Table 9: Statistics on resulting partitions for Hotel Reviews Dataset considering only GPE entities.

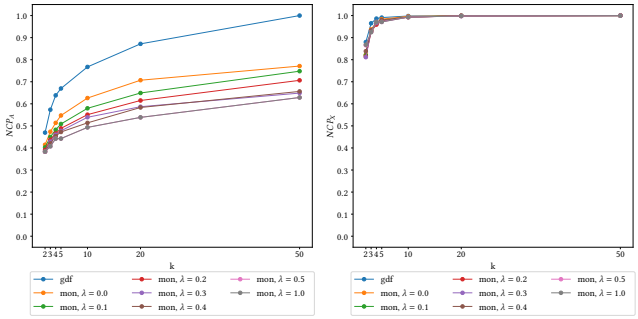
	λ	k	3	4	5	10	20
GDF	-	count	107	47	15	1	1
		size	13.79	31.38	98.33	1475	1475
Mondrian	0	count	49	35	32	17	13
		size	30.10	42.14	46.09	86.76	113.46
	0.3	count	376	312	211	101	50
		size	3.92	4.73	6.99	14.60	29.50
	0.5 - 1	count	417	256	255	128	64
		size	3.54	5.76	5.78	11.52	23.05



(a) Relational information loss NCP_A **(b) Textual information loss NCP_X**

Figure 3: Information loss for relational attributes (a), and textual attributes (b) on the Blog Authorship Corpus.

Figure 3a). This observation coincides with statistics on splitting decisions, since for lower values of λ, Mondrian more frequently decides to split on sensitive terms in textual attributes. This leads to more variations in relational values of partitions, which ultimately increases the relational information loss. In experiments where only



(a) Relational information loss NCP_A **(b) Textual information loss NCP_X**

Figure 4: Information loss for relational attributes (a), and textual attributes (b) on the Hotel Reviews Dataset.

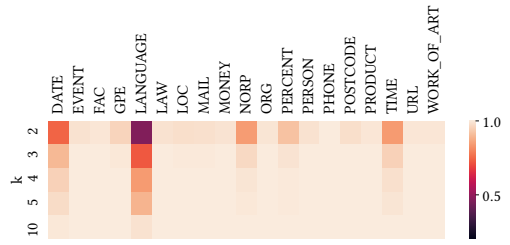
locations are considered, GDF partitioning as well as Mondrian partitioning with $\lambda = 0$ result in relatively high relational information loss compared to other experiment runs (see figures in the supplementary material). In both cases, the high relational information loss is caused by having partitions split only based on one option, namely the recognized sensitive locations appearing in the textual attribute (cf. previous section). Comparing with Figure 4a, we can state that relational information loss appears to be higher for the Hotel Reviews Dataset compared to the Blog Authorship Corpus. However, we still observe the same behavior where higher values of λ result in relatively lower relational information loss.

Textual Information Loss. Analyzing the information loss in the textual attribute, see Figure 3b, one observation is that for values of $k \geq 10$ the information loss in texts tends to become 1. This equals suppressing all sensitive terms in texts. Moreover, our modified Mondrian partitioning performs better compared to the naive partitioning strategy GDF. GDF partitioning results in partitions with unequal and larger sizes and therefore ends up with large partitions, which significantly increase information loss. Moreover, GDF partitioning decides on splitting partitions taking a single global maximum (most frequent term) ignoring the multi-dimensionality and diversity of sensitive terms in texts. We make the same observations on the Hotel Reviews Dataset plotted in Figure 4b. However, information loss for $k \leq 5$ tends to be slightly lower. If only locations are considered, textual information loss in hotel reviews can significantly be reduced (see figures in the supplementary material). Since the Hotel Reviews Dataset only contains reviews for hotels in Europe, there is a limited number of locations that are included. This leads to significant preservation of sensitive terms even for values of $k \leq 10$.

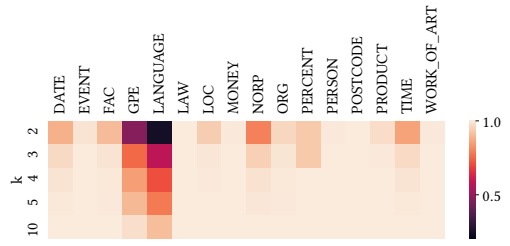
6.3 Attribute-level Textual Information Loss

To get a deeper understanding of textual attributes on the anonymization process, we analyzed textual information loss on entity type level. Figure 5a provides an overview of information loss per different entity type extracted from text in the Blog Authorship Corpus for k is 2 to 50 and a fixed $\lambda = 0.2$. It shows that there is a high information loss for most attributes, even for small k. However, sensitive terms of type LANGUAGE may be reduced for values of

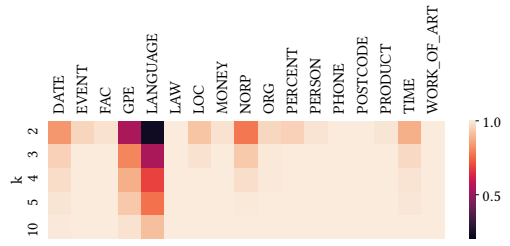
$k \leq 5$. Since the number of distinct entities of type LANGUAGE is much lower compared to other entity types in the Blog Authorship Corpus like EVENT and PERSON, the entities (i. e., number of sensitive terms) of type LANGUAGE can be better preserved. We obtain similar results for Mondrian partitioning with other values of $\lambda \leq 0.4$. We make the same observations on the Hotel Reviews dataset for both textual attributes, the positive reviews and negative reviews (see Figures 5b and 5c). In addition to LANGUAGE entities, sensitive locations (GPE) can also be preserved for both textual attributes.



(a) Entity-level textual loss on the Blog Authorship Corpus



(b) Entity-level textual loss on positive Hotel Review Dataset



(c) Entity-level textual loss on negative Hotel Review Dataset

Figure 5: Textual information loss per entity type with Mondrian ($\lambda = 0.2$). Not all entity types appear in all datasets.

7 DISCUSSION

7.1 Key Results

Due to heterogeneity of sensitive terms in texts, by default, they are less likely considered to split on. By introducing the tuning parameter λ in our framework, we were able to control the Mondrian algorithm to preserve more information in either relational or textual attributes. Our experiments show that the partitioning parameter may be tuned in order to favor information preservation in textual attributes over relational attributes. We observe that a value of λ between 0.4 and 0.5 results in balanced splits, i. e., about the same number of splits are based on relational attributes versus textual terms. Our anonymization approach allows us to reduce the information loss in texts under the k-anonymity privacy model. In contrast, in the related works [8, 34, 50] sensitive terms have been completely suppressed. Furthermore, our experiments show that for $k \leq 5$, not all sensitive terms need to be suppressed. In case of entities of type LANGUAGE, our approach could preserve about 60% for $k = 2$ in the Blog Authorship Corpus (see Figure 5a) and up to 80% of terms for $k = 2$ in the Hotel Reviews Dataset (see Figures 5b and 5c). Generally, when applying k-anonymity on sensitive terms, it works better for texts from a specific domain (e. g., hotels) than cross-domain datasets (e. g., blogs), as the latter have a higher diversity.

7.2 Threads to Validity

While our approach presents a general framework to anonymize heterogeneous data, our choices on detecting and comparing sensitive terms may have an impact on the experiments' outcomes. We consider all sensitive terms in the texts to be quasi-identifiers. However, in certain situations, sensitive entity types should—similar to relational attributes—also be distinguished in direct and quasi-identifying attributes. Having a distinction between direct and quasi-identifiers is necessary in cases where texts include many names, or other identifiers appearing for multiple records. There may be a possible over-anonymization or under-anonymization in our rx-anon approach, influenced by the accuracy of the detected sensitive terms. Over-anonymization resembles the case where sensitive terms are falsely suppressed. It is caused by low precision and reduces utility of the anonymized data. This happens, when terms, which do not pose any risk of identity disclosure, are anonymized and the text loses important structures due to the missing terms. If sensitive terms are labeled with false entity types, they might also falsely be anonymized, since our strict definition of k-anonymity requires also entity types to be equal. Under-anonymization describes a case where sensitive terms are falsely kept. This case is generally considered more critical than falsely suppressing terms and is related to low recall. If entities which should have been anonymized are not detected at all, the information they provide will appear in the released dataset and might reveal information which should not have been disclosed. We address this thread of validity and use a state-of-the-art NLP library spaCy to extract named entities from text. We use spaCy's recent transformer-based language model [9] for English (Version 3.0.0a0)[7], which has an an F1-score for NER tasks of 89.41. However, there are cases that we are yet missing. There can be different writings of the same sensitive information which leads to over-anonymization. For example, the capital city of Germany may be referred to simply by its actual name "Berlin" or indirectly referred to as "Germany's capital". It is not possible for our current system to resolve such linkage. We refer to such cases as false negative matches. There may also be identical terms which actually have different semantics, which leads to under-anonymization. In example, consider the phrases "I live in Berlin" and "I love Berlin", which appear in two different records and would

[7] Available at https://github.com/explosion/spacy-models/releases//tag/en_core_web_trf-3.0.0a0

happen to be grouped into the same partition. Our approach would treat both appearances of "Berlin" the same way even though in the first case it is referring to a place of residence while in the second case it is an expression of preference. We refer to such a scenario as false positive matches. In order to mitigate such false positive and negative cases, one can integrate more advanced text matching functions to our rx-anon framework, potentially depending on the requirements of a specific use case. For false negative matches, one may introduce synonym tables, semantic rules, and metrics such as Levenshtein distance to cope with spelling mistakes. To cope with false positive matches, one suggestion is to consider the surrounding context by comparing Part-of-Speech-Tags and dependencies of terms within and across sentences. For example, one could use contextualized word vectors [9, 39]. Note, in this work, we focus on showing that heterogeneous data can be anonymized using our rx-anon approach and demonstrate the influence of the splitting parameter λ on the creation of the partitions. Using different extensions to rx-anon such as word matching functions is prepared by proposing a framework approach and can be integrated and evaluated as required by a different use case or dataset.

Finally, false negative matches and false positive matches can also occur on redundant sensitive information. While false negative matches result in inconsistencies in the released data, false positive matches obfuscate semantic meaning of sensitive terms in texts.

7.3 Generalizability

Our work has multiple implications which can be beneficial for other work. We showed that anonymizing unstructured text data can be achieved by extracting sensitive terms and casting the task into a structured anonymization problem. One may generalize the concept also for semi-structured data such as JSON documents. The idea of linking relational fields to attributes of other data types could be extended in order to retrieve a consistent, and privacy-preserved version of heterogeneous JSON documents. In addition, tuning the partitioning using a parameter like λ is not only relevant in the context of anonymizing heterogeneous data, but could also be adapted to an attribute level to favor distinct attributes over others. An adjustable attribute-level bias within the partitioning phase of Mondrian would allow users to prioritize preservation of information in specific attributes. Suppose that one department within an organization shares data with a second department, which should do an age-based market analysis of sold products, but should not get access to raw data and therefore receive an anonymized version. As a consequence, the department providing data could adjust the anonymization using a bias to preserve more information in relevant attributes (i. e., age), and less information in others.

8 CONCLUSION

We introduced rx-anon as a step towards a framework for anonymizing hybrid documents consisting of relational as well as textual attributes. We have formally defined the problem of jointly anonymizing heterogeneous datasets by transferring sensitive terms in texts to an anonymization task of structured data, introduced the concept of redundant sensitive information, and the tuning parameter λ to control and prioritize information loss in relational and textual attributes. We have demonstrated the usefulness of rx-anon at the example of two real-world datasets using the privacy model k-anonymity [53].

Data Availability and Reproducibility: Although extensive success has been achieved in anonymizing different types of data, there is limited work in the field of anonymizing heterogeneous data. Therefore, we would like to emphasize the importance and encourage researchers to investigate combined anonymization approaches for heterogeneous data to receive a consistent and privacy-preserved release of data. The source code of rx-anon will be made publicly available to encourage reproduction and extension of our work.

As a framework approach, rx-anon can be extended in all aspects of the anonymization pipeline, namely the *partitioning, string matching, recoding, privacy model, and supported entity types*. Particularly, we are interested to see how anonymizing heterogeneous data can be achieved using other anonymization techniques than k-anonymity and use contextualized text similarity functions [9, 39]. A detailed discussion of the extensibility of our framework is provided in Appendix A.5.

Supplementary Materials

A EXTENDED EXPERIMENTAL RESULTS

The following sections contain extended experiment results. In particular, we provide numbers of distinct entities, give information about the performance of our framework, share statistics relevant for partitioning, and present details on information loss.

A.1 Distinct Terms

Table 10 provides an overview of the number of distinct terms appearing in textual attributes. In general, the texts of the Blog Authorship Corpus contain significantly more distinct entities.

Table 10: Numbers of distinct terms per entity type. The Blog Authorship Corpus contains one textual attribute *text*. The Hotel Reviews Dataset contains two textual attributes, namely *negative review* and *positive review*.

	Blog Authorship Corpus	Hotel Reviews Dataset	
entity type	text	negative review	positive review
DATE	83,972	2,672	1,993
EVENT	13,883	161	244
FAC	32,864	3,452	14,070
GPE	34,639	1,058	2,512
LANGUAGE	761	30	30
LAW	5,153	12	3
LOC	13,635	480	1,370
MAIL	3,225	0	0
MONEY	16,050	2,089	625
NORP	9,676	293	344
ORG	162,555	3,887	9,444
PERCENT	4,104	30	25
PERSON	245,667	2,273	5,728
PHONE	442	1	0
POSTCODE	739	7	8
PRODUCT	48,207	842	892
TIME	61,669	4,311	2,366
URL	29,297	0	0
WORK_OF_ART	145,421	290	349

A.2 Performance

Table 11 provides valuable insights in execution times of the experiments. Each experiment was executed on a single CPU core and did not require to analyze the texts, since the processed NLP state is read from cached results. In the case of experiments run on the Blog Authorship Corpus, execution times were significantly higher compared to the Hotel Reviews dataset. One observation is that if only relational attributes are considered (Mondrian, $\lambda = 1$), execution times come down to a fraction of experiments where sensitive terms are considered during the partitioning phase.

Considering memory consumption, running a single experiment on the Blog Authorship Corpus required 25.2 GB for all entities and 13.4 GB in the case of just considering GPE entities (locations). In the case of the Hotel Review dataset, 5.4 GB and 4.2 GB were required respectively.

Table 11: Execution times of experiments in hh:mm:ss.

		Blog Authorship Corpus		Hotel Reviews Dataset	
	λ	all	GPE	all	GPE
GDF	-	10:25:12	03:50:13	00:12:55	00:10:26
Mondrian	0	15:19:31	02:46:14	00:29:26	00:11:44
	0.1	15:01:34	01:32:25	00:29:56	00:13:16
	0.2	14:43:37	01:30:11	00:29:44	00:12:58
	0.3	14:32:23	01:24:31	00:29:17	00:12:49
	0.4	13:59:02	01:15:58	00:27:59	00:12:44
	0.5	11:03:43	01:04:59	00:25:13	00:12:18
	0.6	11:05:29	01:05:04	00:25:12	00:12:20
	0.7	11:06:26	01:04:47	00:25:14	00:12:20
	0.8	11:03:33	01:04:50	00:25:01	00:12:23
	0.9	11:02:57	01:04:32	00:25:12	00:12:19
	1	01:48:41	00:47:22	00:12:48	00:11:17

A.3 Partitions

In our experiments, we evaluate statistics on partition splits to gain insights how λ influences splitting decisions of Mondrian partitioning. Moreover, we also share statistics on resulting partitions.

A.3.1 Partition Splits. Figure 6 provides an overview of the distribution of splitting decisions between relational and textual attributes for experiments run on the Blog Authorship Corpus for all values of k. The left column includes experiments considering all entities, while the right column presents results for experiments run only considering location (GPE) entities.

A noteworthy observation is that for a fixed λ, the number of splits on textual attributes decreases if k increases. Since we are only considering valid splits, sensitive terms have to appear at least $2k$ times within a partition to be split on. Therefore, in case of $k = 50$, sensitive terms are required to appear 100 times, which is less likely due to heterogeneity of blog post texts.

If only locations, i. e., entities of type GPE, are considered, λ is not in all cases able to control the share of splits between relational and textual attributes, since low values for λ do not result in more splits on textual attributes. This effect is caused by the lack of multi-dimensionality. Since only one category of sensitive entity types is considered, Mondrian has only one option (namely split on sensitive terms with type GPE) to split on textual attributes. If splits on GPE terms fail (e. g., if there are none), Mondrian will ultimately continue to split on a relational attribute.

Similarly, Figure 7 highlights the impact of λ on partition splits for experiments run on the Hotel Reviews Dataset for all values of k.

A.3.2 Partition Count and Size. We present the results regarding partition statistics. Table 12 provides valuable insights on the number of partitions as well as the size and standard deviation regarding partition sizes for the experiments on the Blog Authorship Corpus considering all entities. Similarly, Table 13 provides an overview of the same metrics for the Blog Authorship Corpus only considering GPE entities (locations). Tables 14 and 15 share insights on partition statistics for the Hotel Reviews dataset.

A.4 Information Loss

In addition to evaluating resulting partitions, we are also interested in the actual information loss which is introduced by anonymizing a given dataset. Figure 8 provides an overview of information loss for experiments run on the Blog Authorship Corpus. In particular, Figure 8a visualizes relational and Figure 8b textual information loss for experiments considering all entity types. Similarly, Figure 8c and Figure 8d provide an overview of information loss if only GPE entities are considered. Figure 11 provides the same statistics for the Hotel Reviews Dataset.

Moreover, Figures 9 and 10 provide a zoomed version of Figure 8b and Figure 8d showing textual information loss for experiments on the Blog Authorship Corpus. Similarly, we visualize zoomed textual information loss for the Hotel Reviews Dataset in Figure 12 and Figure 13.

In addition to high-level charts on information loss, Figure 14 provides a detailed analysis of information loss per entity type for the attribute *text* in the Blog Authorship Corpus. Additionally, Figure 15 and Figure 16 visualize the textual information loss per entity type for the attributes *negative review* and *positive review* respectively.

A.5 Directions of Extending the rx-anon Framework

As a framework approach, rx-anon enables several paths for future work. These include all aspects of the anonymization pipeline, namely the partitioning, string matching, recording, privacy model, and supported entity types. We provide examples below.

Partitioning. We showed how decisions on partitions significantly influence information loss. While the naive partitioning strategy GDF can deal with sparse, but diverse sets of sensitive terms, there might be partitioning strategies better suited to attributes with such properties. It would be interesting to see if clustering algorithms applied on sensitive terms lead to improved partitioning. Such clustering algorithms require a minimum lower bound on the partition sizes of at least k. Abu-Khzam et al. [1] presents a general framework for clustering algorithms with a lower bound on the cluster size.

String matching. The matching of relational and textual attributes is currently using an exact string match. Another interesting research topic to build on our work is to investigate sophisticated methods to find non-trivial links within the dataset. Non-trivial links are links which cannot be detected using simple string matching. Mechanisms to reveal non-trivial links are discussed by Hassanzadeh et al. [20]. They studied approximations on string matching as well as semantic mechanisms based on ontology and created a declarative framework and specification language to resolve links in relational data. Those mechanisms would also be applicable to find links between relational data and sensitive entities. It would also be interesting to use string matching based on models using word embeddings [39] or transformer-based similarity functions [9].

Recoding. Moreover, our current recoding strategy for sensitive terms in texts uses suppression to generate a k-anonymous version of texts. However, suppression tends to introduce more information loss compared to generalization. Therefore, it would be interesting to introduce an automatic generalization mechanism for sensitive terms and evaluate it. One way to automatically generate DGHs for sensitive terms is to use hypernym-trees as discussed by Lee et al. [29] and used by Anandan et al. [3] to anonymize texts.

Privacy model. We used k-anonymity as the privacy model to prevent identity disclosure. Even though k-anonymity establishes guarantees on privacy, it does not guard against attacks where adversaries have access to background knowledge. Differential privacy introduced by Dwork [10] resists such attacks by adding noise to data. Our rx-anon can be extended by using such an alternative privacy model. An interesting question to answer would be how differential private methods defined on relational data could be combined with work on creating a differential private representation of texts [13, 60].

Entity types. For the recognition of sensitive entities, we chose to use spaCy and its entity types trained on the OntoNotes5 corpus. We chose to use the OntoNotes5 entity types scheme since it provides more distinguishment and therefore more semantics to entities compared to WikiNer annotations. However, there are still cases, where more fine-grained entity recognition will reduce false positive matches. One example is the term "Georgia" which can refer to the country in the Caucasus, the U.S. state, or to a city in Indiana. Ling and Weld [33] presents a fine-grained set of 112 entity-types which would cover the explained example and state that a fine-grained entity recognition system benefits from its accuracy.

B EXTENDED RELATED WORK

We present related work for anonymization of other types of data. Moreover, we present an overview of the existing regulations on anonymization and their view on PII. Finally, we present a non-exhaustive overview of anonymization tools and frameworks available.

B.1 Research in Data Anonymization for Audio, Images, and Video

Even though this work only focuses on structured data and free text, recent work on anonymization of other forms of data is worth mentioning. For de-identification of images showing faces, Gross et al. [19] highlighted that pixelation and blurring offers poor privacy and suggested a model-based approach to protect privacy while preserving data utility. In contrast, recent work by Hukkelås et al. [22] applied methods from machine learning by implementing a simple Generative Adversarial Network (GAN) to generate new faces to preserve privacy while retaining original data distribution.

For audio data, recent work focused either on anonymization of the speaker's identity or the speech content. Justin et al. [25] suggested a framework which automatically transfers speech into a de-identified version using different acoustical models for recognition and synthesis. Moreover, Cohn et al. [6] investigated the task of de-identifying spoken text by first using Automatic Speech Recognition (ASR) to transcribe texts, then extracting entities using NER, and finally aligning text elements to the audio and suppressing audio segments which should be de-identified.

Additionally, recent work by Agrawal and Narayanan [2] showed that de-identification of people can also be applied to whole bodies within videos whereas Gafni et al. [15] focused on live de-identification of faces in video streams.

Finally, McDonald et al. [37] developed a framework for obfuscating writing styles which can be used by authors to prevent stylometry attacks to retrieve their identities. When it comes to unstructured text, their approach anonymizes writing styles in text documents by analyzing stylographic properties, determining features to be changed, ranking those features with respect to their clusters, and suggesting those changes to the user.

B.2 What is considered Personally Identifiable Information?

In order to understand what fields should be anonymized, a common understanding on what Personally Identifiable Information (PII) is needs to be established. Therefore, we provide a broad overview on regulations such as the Health Insurance Portability and Accountability Act (HIPAA), the General Data Protection Regulation (GDPR), and definitions by National Institute of Standards and Technology (NIST) to get an understanding for PII.

B.2.1 Health Insurance Portability and Accountability Act. First, we want to consider the Health Insurance Portability and Accountability Act (HIPAA) providing regulations to ensure privacy within medical data in the USA [56]. Even though the HIPAA privacy rule uses the terminology Protected Health Information (PHI), in general we can transfer their identifiers to the domain of PII. The HIPAA states that any information from the past, present, or future which is linked to an individual is considered PHI. In addition to domain experts defining PHI, the Safe Harbor Method defined in the HIPAA provides an overview of attributes which should be anonymized by removing [56]. Those attributes are in particular:

(1) Names
(2) Geographic entities smaller than states (street address, city, county, ZIP, etc.)
(3) Dates (except year)
(4) Phone numbers
(5) Vehicle identifiers and serial numbers
(6) Fax numbers
(7) Device identifiers and serial numbers
(8) Email addresses
(9) URLs
(10) Social security numbers
(11) IP addresses
(12) Medical record numbers
(13) Biometric identifiers, including finger and voice prints
(14) Health plan beneficiary numbers
(15) Full-face photographs
(16) Account numbers
(17) Any other unique identifying number, characteristic, code, etc.
(18) Certificate and license numbers

B.2.2 General Data Protection Regulation. In Europe, one important privacy regulation is the General Data Protection Regulation (GDPR) [7]. Instead of using the term PII, the GDPR refers to the term *personal data*. The regulation states that *"'Personal data' means any information relating to an identified or identifiable natural person ..."* [7]. Even though the GDPR does not explicitly state a list of attributes considered personal data, they provide some guidance on which properties are considered personal data. In particular the GDPR states that personal data is any data which can identify an individual directly or indirectly *"by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person"* [7].

B.2.3 Guidelines by NIST. In contrast to the GDPR, the National Institute of Standards and Technology (NIST) provides guidance on protecting PII [36]. The NIST distinguishes PII in two categories. The first category includes "... any information that can be used to distinguish or trace an individual's identity ..." [36]. In particular, they list the following attributes:

- Name
- Social Security Number
- Date and place of birth
- Mother's maiden name
- Biometric records

Moreover, the NIST labels "... any other information that is linked or linkable to an individual ..." also as PII [36]. Examples for linked or linkable attributes are:

- Medical information
- Educational information
- Financial information
- Employment information

B.3 Existing Anonymization Tools and Frameworks

Multiple publicly available tools and frameworks for anonymization of data have been released. *ARX*[8] is an open source comprehensive software providing a graphical interface for anonymizing structured datasets [43, 44]. ARX supports multiple privacy and risk models, methods for transforming data, and concepts for analyzing the output data. Among the privacy models, it supports syntactic privacy models like k-anonymity, l-diversity, and t-closeness, but also supports semantic privacy models like ε-differential privacy. Moreover, *Amnesia*[9] is a flexible data anonymization tool which allows to ensure privacy on structured data. Amnesia supports k-anonymity for relational data as well as k^m-anonymity for datasets

[8] https://arx.deidentifier.org/
[9] https://amnesia.openaire.eu/

containing set-valued data fields. Finally, *Privacy Analytics*[10] offers a commercial Eclipse plugin which can be used to anonymize structured data.

Besides toolings for de-identification of structured data, there also exist frameworks or modules to achieve anonymization. *python-datafly*[11] is a Python implementation of the Datafly algorithm introduced by Sweeney [52] as one of the first algorithms to transfer structured data to match k-anonymity. Additionally, *Crowds*[12] is an open-source python module developed to de-identify a dataframe using the Optimal Lattice Anonymization (OLA) algorithm as proposed by El Emam et al. [12] to achieve k-anonymity. Finally, an example for an implementation of the Mondrian algorithm [31] is available for Python[13] to show how k-anonymity, l-diversity, and t-closeness can be used as privacy models.

There are multiple tools and frameworks for de-identification of free text. *NLM-Scrubber*[14] is a freely available tool for de-identification of clinical texts according to the Safe Harbor Method introduced in the HIPAA Privacy Rule. Moreover, *MITRE Identification Scrubber Toolkit (MIST)*[15] is a suite of tools for identifying and redacting PII in free-text medical records [26]. *deid*[16] is a tool which allows anonymization of free texts within the medical domain. Finally, *deidentify*[17] is a Python library developed especially for de-identification of medical records and comparison of rule-, feature-, and deep-learning-based approaches for de-identification of free texts [57].

REFERENCES

[1] Faisal N. Abu-Khzam, Cristina Bazgan, Katrin Casel, and Henning Fernau. 2018. Clustering with Lower-Bounded Sizes: A General Graph-Theoretic Framework. *Algorithmica* 80, 9 (2018), 2517–2550.

[2] Prachi Agrawal and P. J. Narayanan. 2011. Person De-Identification in Videos. *IEEE Trans. on Circuits and Systems for Video Technology* 21, 3 (2011), 299–310.

[3] Balamurugan Anandan, Chris Clifton, Wei Jiang, Mummoorthy Murugesan, Pedro Pastrana-Camacho, and Luo Si. 2012. t-Plausibility: Generalizing words to desensitize text. *Trans. on Data Privacy* 5, 3 (2012), 505–534.

[4] Roberto J. Bayardo and Rakesh Agrawal. 2005. Data Privacy through Optimal k-Anonymization. In *Int. Conf. on Data Engineering (ICDE)*. IEEE.

[5] Venkatesan T. Chakaravarthy, Himanshu Gupta, Prasan Roy, and Mukesh K. Mohania. 2008. Efficient techniques for document sanitization. In *Int. Conf. on Information and Knowledge Mining (CIKM)*. ACM.

[6] Ido Cohn, Itay Laish, Genady Beryozkin, Gang Li, Izhak Shafran, Idan Szpektor, Tzvika Hartman, Avinatan Hassidim, and Yossi Matias. 2019. Audio De-identification - a New Entity Recognition Task. In *Conf. of the North American Chapter of the Association for Computational Linguistics*. ACL. arXiv:1903.07037

[7] Council of European Union. 2016. EU General Data Protection Regulation (GDPR). https://eur-lex.europa.eu/eli/reg/2016/679/oj

[8] Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. 2017. De-identification of patient notes with recurrent neural networks. *J. of the American Medical Informatics Association* 24, 3 (2017), 596–606. arXiv:1606.03475

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. ACL.

[10] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming*, Vol. 4052. Springer, 1–12.

[11] Elisabeth Eder, Ulrike Krieg-Holz, and Udo Hahn. 2019. De-Identification of Emails: Pseudonymizing Privacy-Sensitive Data in a German Email Corpus. In *Int. Conf. on Recent Advances in Natural Language Processing*. Incoma Ltd.

[10] https://privacy-analytics.com/health-data-privacy/
[11] https://github.com/alessiovierti/python-datafly
[12] https://github.com/leo-mazz/crowds
[13] https://github.com/Nuclearstar/K-Anonymity
[14] https://scrubber.nlm.nih.gov/
[15] http://mist-deid.sourceforge.net/
[16] https://www.physionet.org/content/deid/1.1/
[17] https://github.com/nedap/deidentify

[12] Khaled El Emam, Fida Kamal Dankar, Romeo Issa, and others. 2009. A Globally Optimal k-Anonymity Method for the De-Identification of Health Data. *J. of the American Medical Informatics Association* 16, 5 (2009), 670–682.

[13] Natasha Fernandes, Mark Dras, and Annabelle McIver. 2019. Generalised Differential Privacy for Text Document Processing. In *Principles of Security and Trust*. Springer.

[14] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. 2010. Privacy-preserving data publishing: A survey of recent developments. *Comput. Surveys* 42, 4 (2010), 1–53.

[15] Oran Gafni, Lior Wolf, and Yaniv Taigman. 2019. Live Face De-Identification in Video. In *Int. Conf. on Computer Vision (ICCV)*. IEEE.

[16] James Gardner and Li Xiong. 2008. HIDE: An Integrated System for Health Information DE-identification. In *Int. Symposium on Computer-Based Medical Systems*. IEEE.

[17] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. 2007. Fast data anonymization with low information loss. In *VLDB*. ACM.

[18] Qiyuan Gong, Junzhou Luo, Ming Yang, Weiwei Ni, and Xiao Bai Li. 2017. Anonymizing 1:M microdata with high utility. *Knowledge-Based Systems* 115 (2017), 15–26.

[19] Ralph Gross, Latanya Sweeney, F. de la Torre, and Simon Baker. 2006. Model-Based Face De-Identification. In *Computer Vision and Pattern Recognition*. IEEE.

[20] Oktie Hassanzadeh, Lipyeow Lim, Anastasios Kementsietsidis, and Min Wang. 2009. A declarative framework for semantic link discovery over relational data. In *Int. World Wide Web Conference*. ACM.

[21] Yeye He and Jeffrey F. Naughton. 2009. Anonymization of Set-Valued Data via Top-Down, Local Generalization. *VLDB* 2, 1 (2009), 934–945.

[22] Håkon Hukkelås, Rudolf Mester, and Frank Lindseth. 2019. DeepPrivacy: A Generative Adversarial Network for Face Anonymization. In *Advances in Visual Computing*, Vol. 11844. Springer.

[23] Information and Privacy Commissioner of Ontario. 2016. *De-identification Guidelines for Structured Data*. Technical Report June. Information and Privacy Commissioner of Ontario. https://www.ipc.on.ca/wp-content/uploads/2016/08/Deidentification-Guidelines-for-Structured-Data.pdf

[24] Alistair E. W. Johnson, Lucas Bulgarelli, and Tom J. Pollard. 2020. Deidentification of free-text medical records using pre-trained bidirectional transformers. In *ACM Conf. on Health, Inference, and Learning*. ACM.

[25] Tadej Justin, Vitomir Struc, Simon Dobrisek, Bostjan Vesnicer, Ivo Ipsic, and France Mihelic. 2015. Speaker de-identification using diphone recognition and speech synthesis. In *11th IEEE Int. Conf. and Workshops on Automatic Face and Gesture Recognition*. IEEE.

[26] Mehmet Kayaalp, Allen C. Browne, Zeyno A. Dodd, Pamela Sagan, and Clement J. McDonald. 2014. De-identification of Address, Date, and Alphanumeric Identifiers in Narrative Clinical Reports. In *American Medical Informatics Association Annual Symposium*. AMIA.

[27] Muhammad Raza Khan, Morteza Ziyadi, and Mohamed AbdelHady. 2020. MT-BioNER: Multi-task Learning for Biomedical Named Entity Recognition using Deep Bidirectional Transformers. (2020). arXiv:2001.08904

[28] Moshe Koppel, Jonathan Schler, Shlomo Argamon, and Eran Messeri. 2006. Authorship attribution with thousands of candidate authors. In *Int. Conf. on Research and Development in Information Retrieval (SIGIR)*, Vol. 2006. ACM.

[29] Sangno Lee, Soon Young Huh, and Ronald D. McNiel. 2008. Automatic generation of concept hierarchies using WordNet. *Expert Systems with Applications* 35, 3 (2008), 1132–1144.

[30] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. 2005. Incognito: efficient full-domain K-anonymity. In *2005 ACM SIGMOD Int. Conf. on Management of data*. ACM.

[31] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. 2006. Mondrian Multidimensional K-Anonymity. In *Int. Conf. on Data Engineering (ICDE)*. IEEE.

[32] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *Int. Conf. on Data Engineering (ICDE)*. IEEE.

[33] Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. *National Conf. on Artificial Intelligence* 1 (2012), 94–100.

[34] Zengjian Liu, Buzhou Tang, Xiaolong Wang, and Qingcai Chen. 2017. De-identification of clinical notes via recurrent neural network and conditional random field. *J. of Biomedical Informatics* 75 (2017), S34–S42.

[35] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. 2006. L-diversity: privacy beyond k-anonymity. In *Int. Conf. on Data Engineering (ICDE)*. IEEE.

[36] Erika McCallister, Timothy Grance, and Karen A. Scarfone. 2010. *Guide to protecting the confidentiality of Personally Identifiable Information (PII)*. Technical Report. National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-122.pdf

[37] Andrew W. E. McDonald, Sadia Afroz, Aylin Caliskan, Ariel Stolerman, and Rachel Greenstadt. 2012. Use Fewer Instances of the Letter "i": Toward Writing Style Anonymization. In *Privacy Enhancing Technologies*, Vol. 7384. Springer.

[38] Adam Meyerson and Ryan Williams. 2004. On the complexity of optimal k-anonymity. In *Symposium on Principles of Database Systems*. ACM.

[39] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger (Eds.). 3111–3119. https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html

[40] Ishna Neamatullah, Margaret M. Douglass, Li-wei H Lehman, and others. 2008. Automated de-identification of free-text medical records. *BMC Medical Informatics and Decision Making* 8 (2008), 32.

[41] Mehmet Ercan Nergiz, Christopher Clifton, and Ahmet Erhan Nergiz. 2007. MultiRelational k-Anonymity. In *Int. Conf. on Data Engineering (ICDE)*, Vol. 21. IEEE.

[42] Giorgos Poulis, Grigorios Loukides, Aris Gkoulalas-Divanis, and Spiros Skiadopoulos. 2013. Anonymizing Data with Relational and Transaction Attributes. In *Machine Learning and Knowledge Discovery in Databases*. Springer.

[43] Fabian Prasser, Johanna Eicher, Helmut Spengler, Raffael Bild, and Klaus A. Kuhn. 2020. Flexible data anonymization using ARX—Current status and challenges ahead. *Software: Practice and Experience* 50, 7 (2020), 1277–1304.

[44] Fabian Prasser, Florian Kohlmayer, Ronald Lautenschläger, and Klaus A. Kuhn. 2014. ARX–A Comprehensive Tool for Anonymizing Biomedical Data. In *American Medical Informatics Association Annual Symposium*. AMIA.

[45] Patrick Ruch, Robert H. Baud, Anne-Marie Rassinoux, Pierrette Bouillon, and Gilbert Robert. 2000. Medical document anonymization with a semantic lexicon. In *American Medical Informatics Association Annual Symposium*. AMIA.

[46] Pierangela Samarati. 2001. Protecting respondents identities in microdata release. *IEEE Trans. on Knowledge and Data Engineering* 13, 6 (2001), 1010–1027.

[47] David Sánchez, Montserrat Batet, and Alexandre Viejo. 2013. Automatic general-purpose sanitization of textual documents. *IEEE Trans. on Information Forensics and Security* 8, 6 (2013), 853–862.

[48] Yücel Saygin, Dilek Hakkani-Tür, and Gökhan Tür. 2009. Sanitization and Anonymization of Document Repositories. In *Database Technologies: Concepts, Methodologies, Tools, and Applications*. IGI Global.

[49] Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James Pennebaker. 2006. Effects of age and gender on blogging. In *Computational Approaches to Analyzing Weblogs*. AAAI.

[50] Latanya Sweeney. 1996. Replacing personally-identifying information in medical records, the Scrub system. In *Proceedings : a Conf. of the American Medical Informatics Association. AMIA Fall Symposium*. AMIA.

[51] Latanya Sweeney. 2000. *Simple Demographics Often Identify People Uniquely*. Carnegie Mellon U.

[52] Latanya Sweeney. 2002. Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 5 (2002), 571–588.

[53] Latanya Sweeney. 2002. k-Anonymity: A Model for Protecting Privacy. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 5 (2002), 557–570.

[54] Irene Teinemaa, Marlon Dumas, Fabrizio Maria Maggi, and Chiara Di Francescomarino. 2016. Predictive business process monitoring with structured and unstructured data. In *Business Process Management*, Vol. 9850. Springer.

[55] Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. 2008. Privacy-preserving anonymization of set-valued data. *VLDB* 1, 1 (2008), 115–125.

[56] The Office for Civil Rights (OCR) and Bradley Malin. 2012. *Guidance Regarding Methods for de-identification of protected health information in accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule*. Technical Report. U.S. Department of Health & Human Services. https://www.hhs.gov/sites/default/files/ocr/privacy/hipaa/understanding/coveredentities/De-identification/hhs_deid_guidance.pdf

[57] Jan Trienes, Dolf Trieschnigg, Christin Seifert, and Djoerd Hiemstra. 2020. Comparing Rule-based, Feature-based and Deep Neural Methods for De-identification of Dutch Medical Records. In *Health Search and Data Mining*, Vol. 2551. CEUR.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Conf. on Neural Information Processing Systems*. Curran.

[59] Jian Xu, Wei Wang, Jian Pei, Xiaoyuan Wang, Baile Shi, and Ada Wai-Chee Fu. 2006. Utility-based anonymization using local recoding. In *Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD)*. ACM.

[60] Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. 2020. A Differentially Private Text Perturbation Method Using Regularized Mahalanobis Metric. (2020). arXiv:2010.11947

[61] Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: Adapting Transformer Encoder for Named Entity Recognition. (2019). arXiv:1911.04474

[62] Ying Zhao and Charles C. Zhou. 2020. Link Analysis to Discover Insights from Structured and Unstructured Data on COVID-19. In *Bioinformatics, Computational Biology and Health Informatics*. ACM.

Figure 6: Splitting statistics for the Blog Authorship Corpus. Left plots are results for experiments run considering all entities. Right plots represent statistics for experiments run only considering GPE entities.

Figure 7: Splitting statistics for Hotel Reviews Dataset. Left plots are results for experiments run considering all entities. Right plots represent statistics for experiments run only considering GPE entities.

Table 12: Statistics on resulting partitions for Blog Authorship Corpus considering all entity types.

	λ	k	2	3	4	5	10	20	50
GDF	-	count	5810	2479	1512	1078	352	92	7
		size	3.33	7.79	12.78	17.92	54.88	209.99	2759.86
		std	26.78	103.95	199.22	292.95	726.15	1742.66	7146.68
Mondrian	0	count	8219	5162	3795	2971	1412	692	278
		size	2.35	3.74	5.09	6.50	13.68	27.92	69.49
		std	0.76	1.15	1.48	1.91	3.56	6.76	18.28
	0.1	count	8277	5226	3841	3028	1471	710	286
		size	2.33	3.70	5.03	6.38	13.13	27.21	67.55
		std	0.47	0.78	1.06	1.35	2.74	5.77	14.40
	0.2	count	8243	5234	3829	3031	1460	715	283
		size	2.34	3.69	5.05	6.37	13.23	27.02	68.27
		std	0.47	0.79	1.06	1.36	2.79	5.67	14.64
	0.3	count	8302	5236	3841	3023	1462	707	280
		size	2.33	3.69	5.03	6.39	13.21	27.33	69.00
		std	0.47	0.78	1.06	1.34	2.82	5.81	14.47
	0.4	count	8307	5238	3855	3024	1466	720	283
		size	2.33	3.69	5.01	6.39	13.18	26.83	68.27
		std	0.47	0.78	1.06	1.36	2.77	5.57	14.50
	0.5	count	8186	5198	3800	2979	1441	711	278
		size	2.36	3.72	5.08	6.49	13.41	27.17	69.49
		std	0.48	0.77	1.07	1.34	2.76	5.68	14.59
	0.6 - 0.9	count	8168	5180	3781	2987	1441	711	276
		size	2.37	3.73	5.11	6.47	13.41	27.17	70.00
		std	0.48	0.77	1.07	1.35	2.76	5.69	14.43
	1	count	8080	5128	3749	2964	1431	703	273
		size	2.39	3.77	5.15	6.52	13.50	27.48	70.77
		std	0.51	0.80	1.10	1.38	2.80	5.78	14.63

Table 13: Statistics on resulting partitions for Blog Authorship Corpus considering only GPE entities

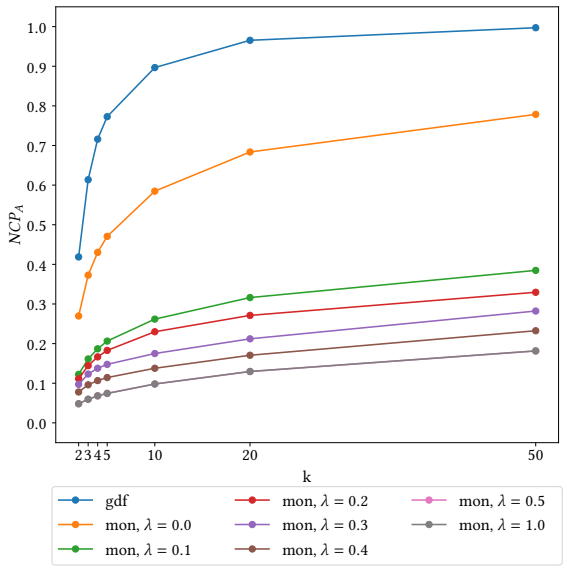
	λ	k	2	3	4	5	10	20	50
GDF	-	count	2140	737	301	129	2	1	1
		size	9.03	26.21	64.18	149.76	9659.50	19319	19319
		std	575.34	1016.03	1629.78	13645.04	0	0	
Mondrian	0	count	1447	1164	828	703	392	242	121
		size	13.35	16.60	23.33	27.48	49.28	79.83	159.66
		std	113.84	126.77	150.14	162.72	217.34	275.39	383.25
	0.1	count	8137	5265	3790	3019	1460	704	281
		size	2.37	3.67	5.10	6.40	13.23	27.44	68.75
		std	0.49	0.77	1.09	1.36	2.83	5.82	15.32
	0.2	count	8186	5227	3801	3016	1451	709	276
		size	2.36	3.70	5.08	6.41	13.31	27.25	70.00
		std	0.48	0.79	1.08	1.37	2.85	5.87	14.48
	0.3	count	8167	5248	3780	2994	1443	700	279
		size	2.37	3.68	5.11	6.45	13.39	27.60	69.24
		std	0.48	0.78	1.09	1.34	2.86	6.08	14.55
	0.4	count	8153	5200	3780	3022	1473	707	280
		size	2.37	3.72	5.11	6.39	13.12	27.33	69.00
		std	0.48	0.79	1.08	1.35	2.74	5.88	14.07
	0.5	count	8153	5154	3762	2970	1433	703	273
		size	2.37	3.75	5.14	6.50	13.48	27.48	70.77
		std	0.49	0.79	1.09	1.37	2.79	5.82	14.88
	0.6 - 0.9	count	8146	5153	3761	2970	1433	703	273
		size	2.37	3.75	5.14	6.50	13.48	27.48	70.77
		std	0.49	0.79	1.09	1.37	2.79	5.78	14.63
	1	count	8080	5128	3749	2964	1431	703	273
		size	2.39	3.77	5.15	6.52	13.50	27.48	70.77
		std	0.51	0.80	1.10	1.38	2.80	5.78	14.63

Table 14: Statistics on resulting partitions for Hotel Reviews Dataset considering all entity types.

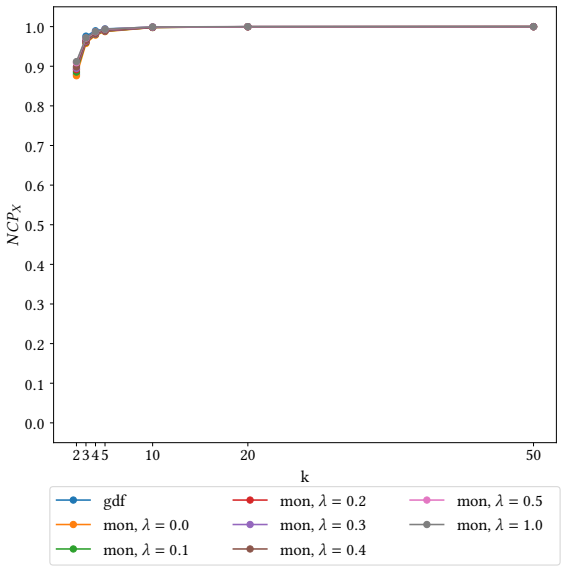
	λ	k	2	3	4	5	10	20	50
GDF	-	count	523	272	163	127	43	16	1
		size	2.82	5.42	9.05	11.61	34.30	92.19	1475
		std	3.45	11.51	24.56	35.47	112.32	264.30	0
Mondrian	0	count	645	398	293	226	117	54	21
		size	2.29	3.71	5.03	6.53	12.61	27.31	70.24
		std	0.47	0.77	1.11	1.45	2.60	5.59	15.67
	0.1	count	635	401	294	231	115	56	22
		size	2.32	3.68	5.02	6.39	12.83	26.34	67.05
		std	0.47	0.76	1.04	1.39	2.78	5.22	14.78
	0.2	count	641	404	295	224	112	51	20
		size	2.30	3.65	5	6.58	13.17	28.92	73.75
		std	0.46	0.75	1.08	1.41	2.71	5.03	15.00
	0.3	count	645	404	285	212	106	50	24
		size	2.29	3.65	5.18	6.96	13.92	29.50	61.46
		std	0.45	0.79	1.03	1.41	2.43	3.65	6.47
	0.4	count	610	406	268	209	113	48	20
		size	2.42	3.63	5.50	7.06	13.05	30.73	73.75
		std	0.49	0.84	0.85	1.76	2.32	6.71	6.45
	0.5	count	558	415	256	255	128	64	20
		size	2.64	3.55	5.76	5.78	11.52	23.05	73.75
		std	0.48	0.84	0.70	0.72	1.19	2.22	19.13
	0.6 - 1	count	554	417	256	255	128	64	20
		size	2.66	3.54	5.76	5.78	11.52	23.05	73.75
		std	0.47	0.83	0.70	0.71	1.19	2.22	19.13

Table 15: Statistics on resulting partitions for Hotel Reviews Dataset considering only GPE entities.

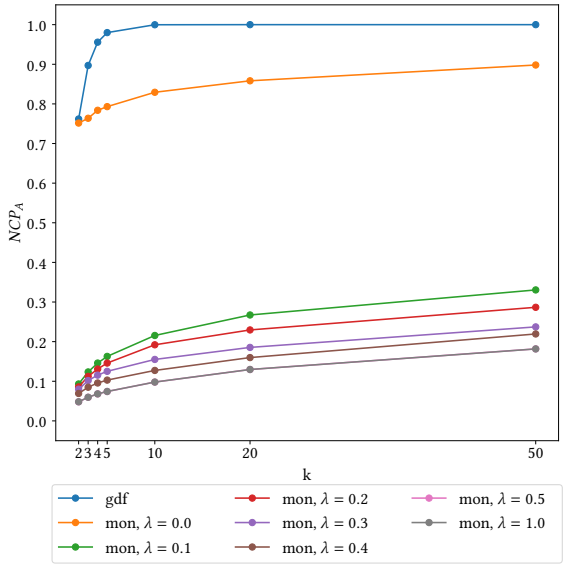
	λ	k	2	3	4	5	10	20	50
GDF	-	count	303	107	47	15	1	1	1
		size	4.87	13.79	31.38	98.33	1475	1475	1475
		std	18.59	90.92	179.98	358.71	0	0	0
Mondrian	0	count	74	49	35	32	17	13	7
		size	19.93	30.10	42.14	46.09	86.76	113.46	210.71
		std	61.71	74.77	86.30	89.59	112.62	121.77	135.59
	0.1	count	603	408	276	236	117	59	18
		size	2.45	3.62	5.34	6.25	12.61	25	81.94
		std	0.50	0.72	1.10	1.06	1.82	3.03	13.70
	0.2	count	647	380	318	201	95	45	21
		size	2.28	3.88	4.64	7.34	15.53	32.78	70.24
		std	0.45	0.70	0.89	1.45	2.70	4.46	8.09
	0.3	count	634	376	312	211	101	50	21
		size	2.33	3.92	4.73	6.99	14.60	29.50	70.24
		std	0.47	0.75	0.89	1.55	2.91	5.75	7.42
	0.4	count	657	362	327	202	99	50	21
		size	2.25	4.07	4.51	7.30	14.90	29.50	70.24
		std	0.43	0.66	0.75	1.51	2.82	5.75	7.42
	0.5 - 1	count	554	417	256	255	128	64	20
		size	2.66	3.54	5.76	5.78	11.52	23.05	73.75
		std	0.47	0.83	0.70	0.71	1.19	2.22	19.13



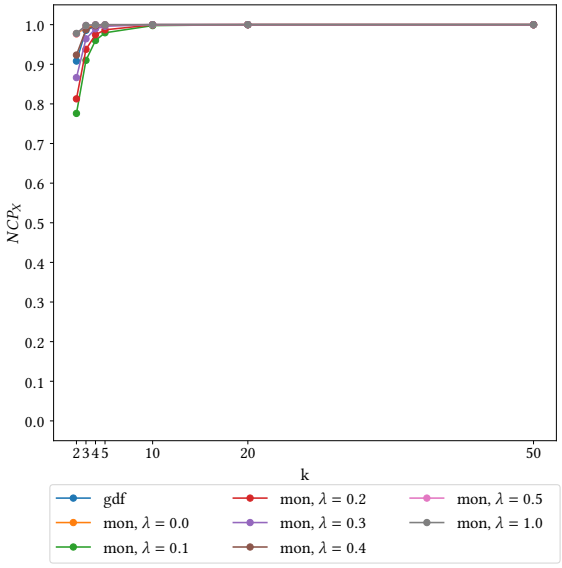
(a) Relational information loss NCP_A, all entity types



(b) Textual information loss NCP_X, all entity types



(c) Relational information loss NCP_A, only GPE entities



(d) Textual information loss NCP_X, only GPE entities

Figure 8: Relational and textual information loss for experiments run on the Blog Authorship Corpus. Results for relational (a) and textual information loss (b) for experiments considering all entities. Results for relational (c) and textual information loss (d) for experiments considering only GPE entities.

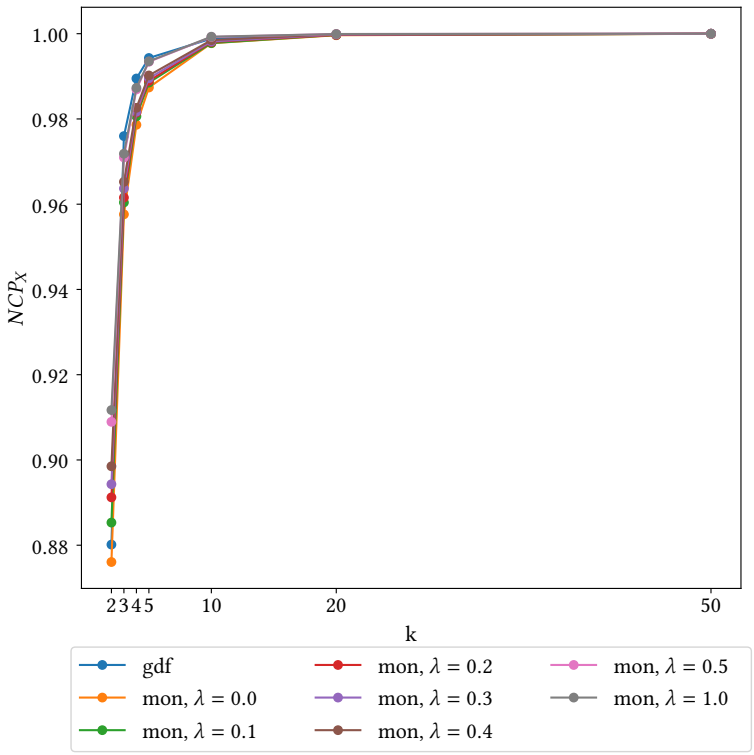


Figure 9: Zoomed textual information loss for experiments run on the Blog Authorship Corpus considering all entities.

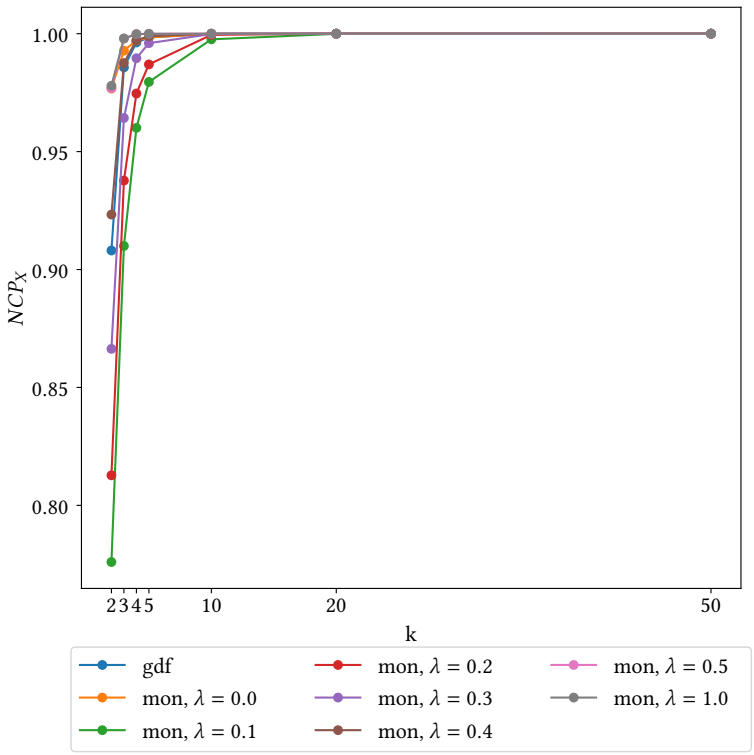
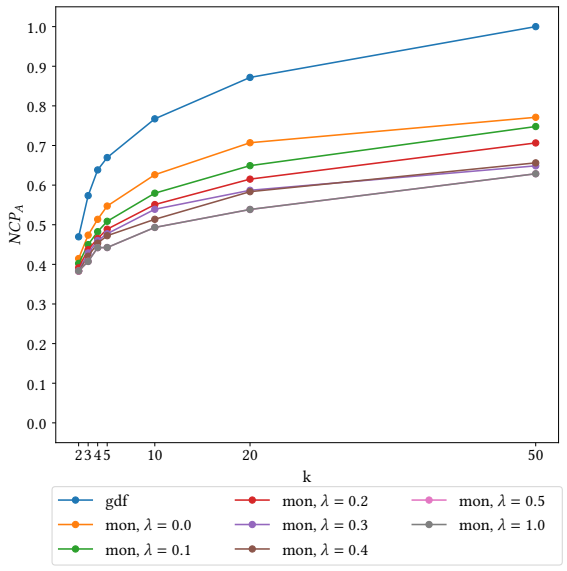
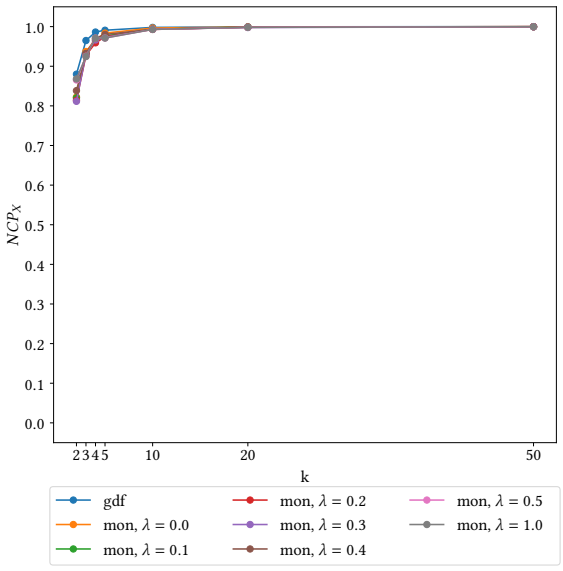


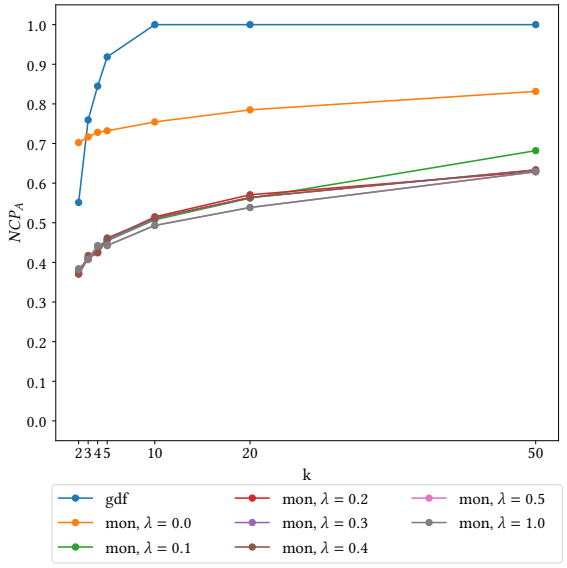
Figure 10: Zoomed textual information loss for experiments run on the Blog Authorship Corpus considering only GPE entities.



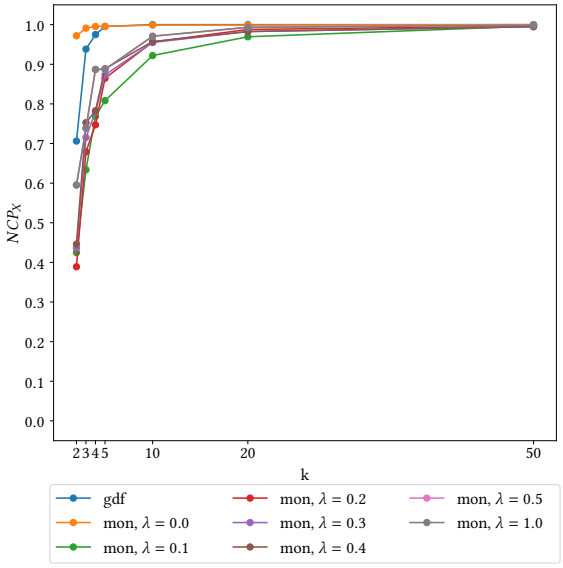
(a) Relational information loss NCP_A, all entity types



(b) Textual information loss NCP_X, all entity types



(c) Relational information loss NCP_A, only GPE entities



(d) Textual information loss NCP_X, only GPE entities

Figure 11: Relational and textual information loss for experiments run on the Hotel Reviews Dataset. Results for relational (a) and textual information loss (b) for experiments considering all entities. Results for relational (c) and textual information loss (d) for experiments considering only GPE entities.

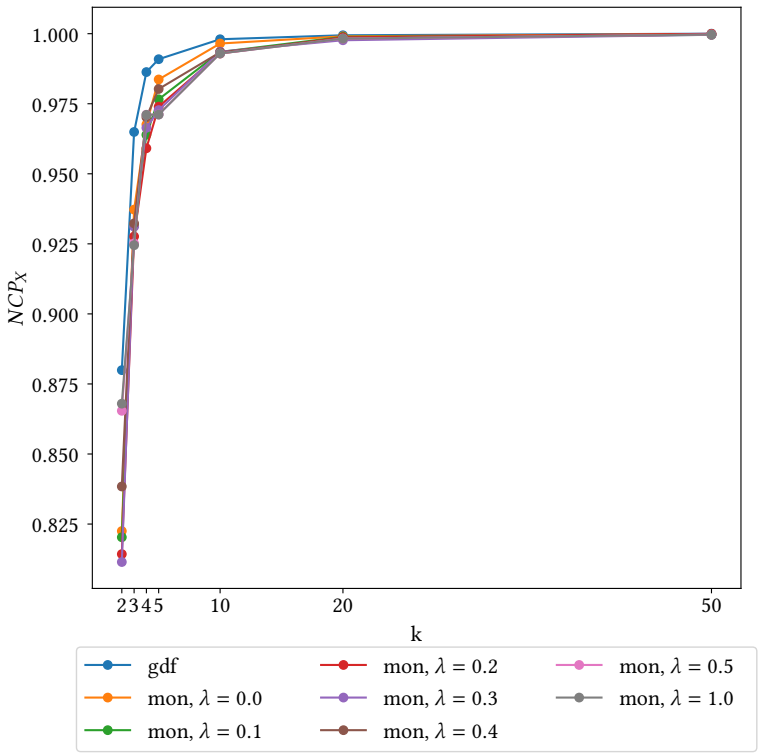


Figure 12: Zoomed textual information loss for experiments run on the Hotel Reviews Dataset considering all entities.

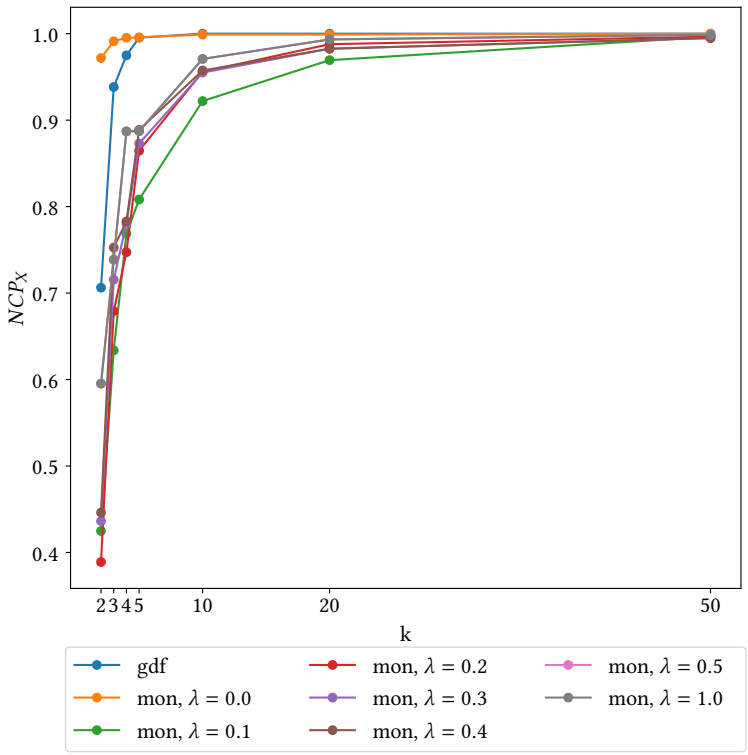


Figure 13: Zoomed textual information loss for experiments run on the Hotel Reviews Dataset considering only GPE entities.

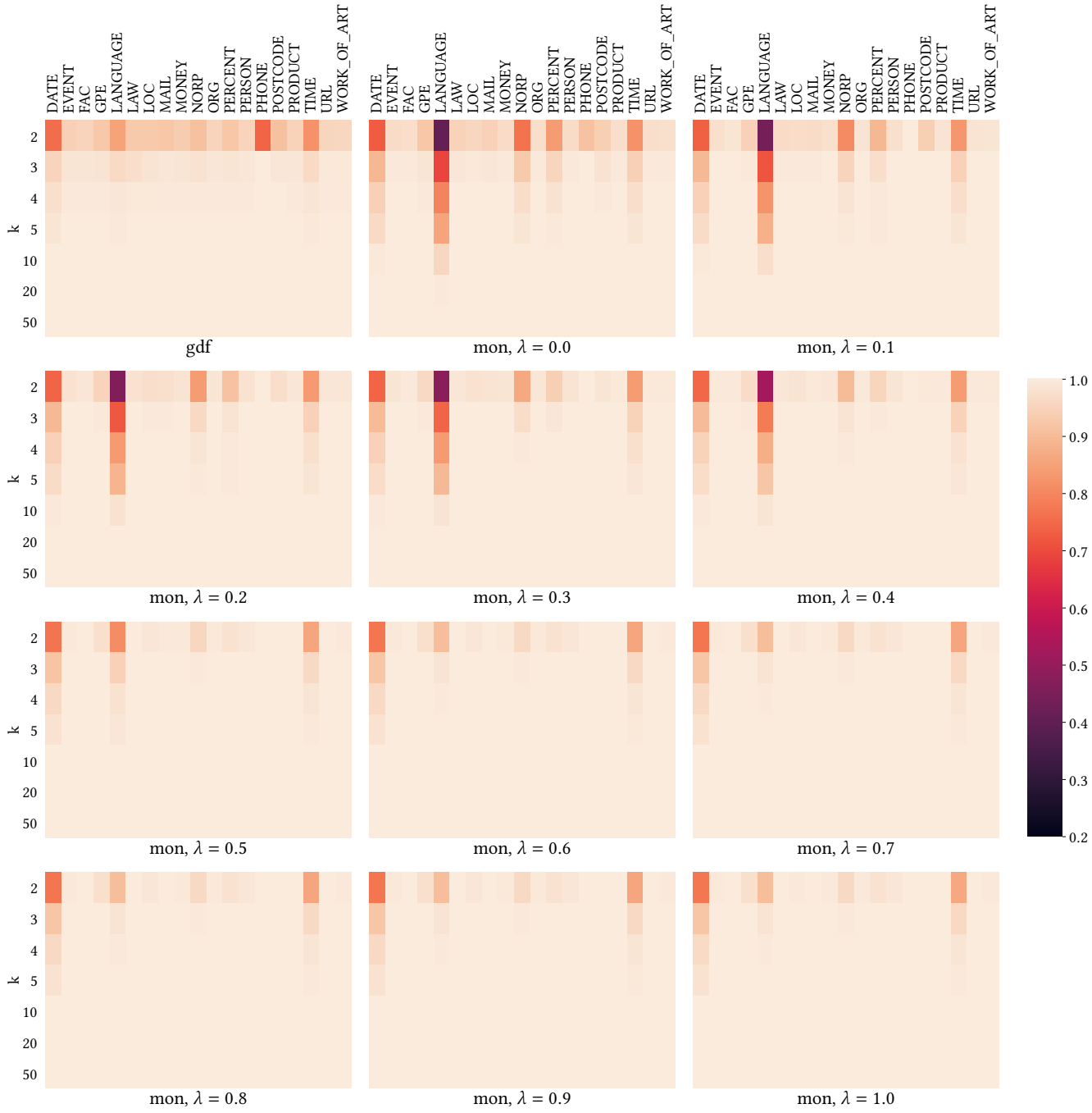


Figure 14: Textual information loss of the attribute *text* per entity types for experiments run on the Blog Authorship Corpus. Information loss is visualized for GDF and Mondrian partitioning with varying λ.

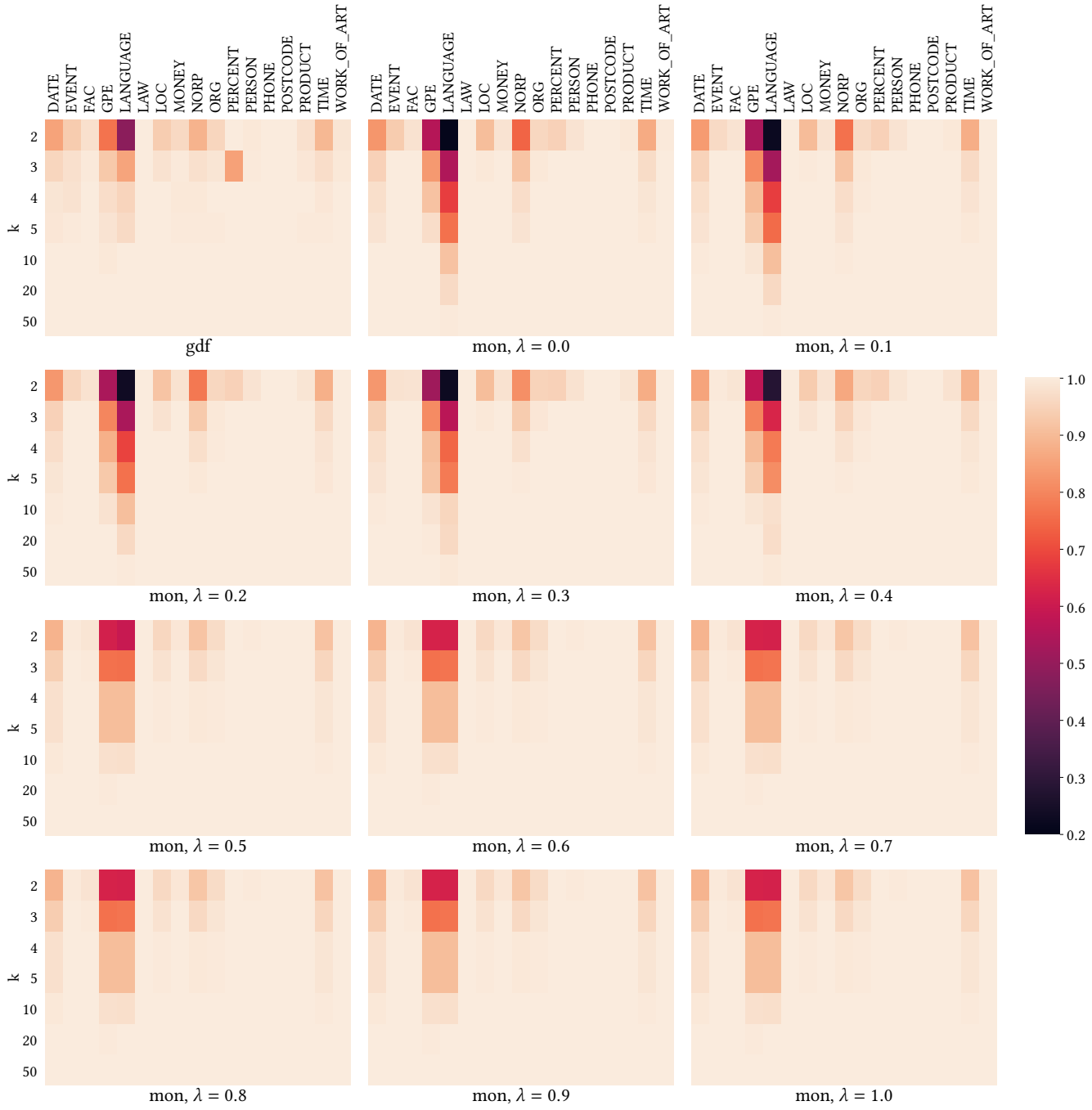


Figure 15: Textual information loss of the attribute *negative review* per entity type for experiments run on the Hotel Reviews Dataset. Information loss is visualized for GDF and Mondrian partitioning with varying λ.

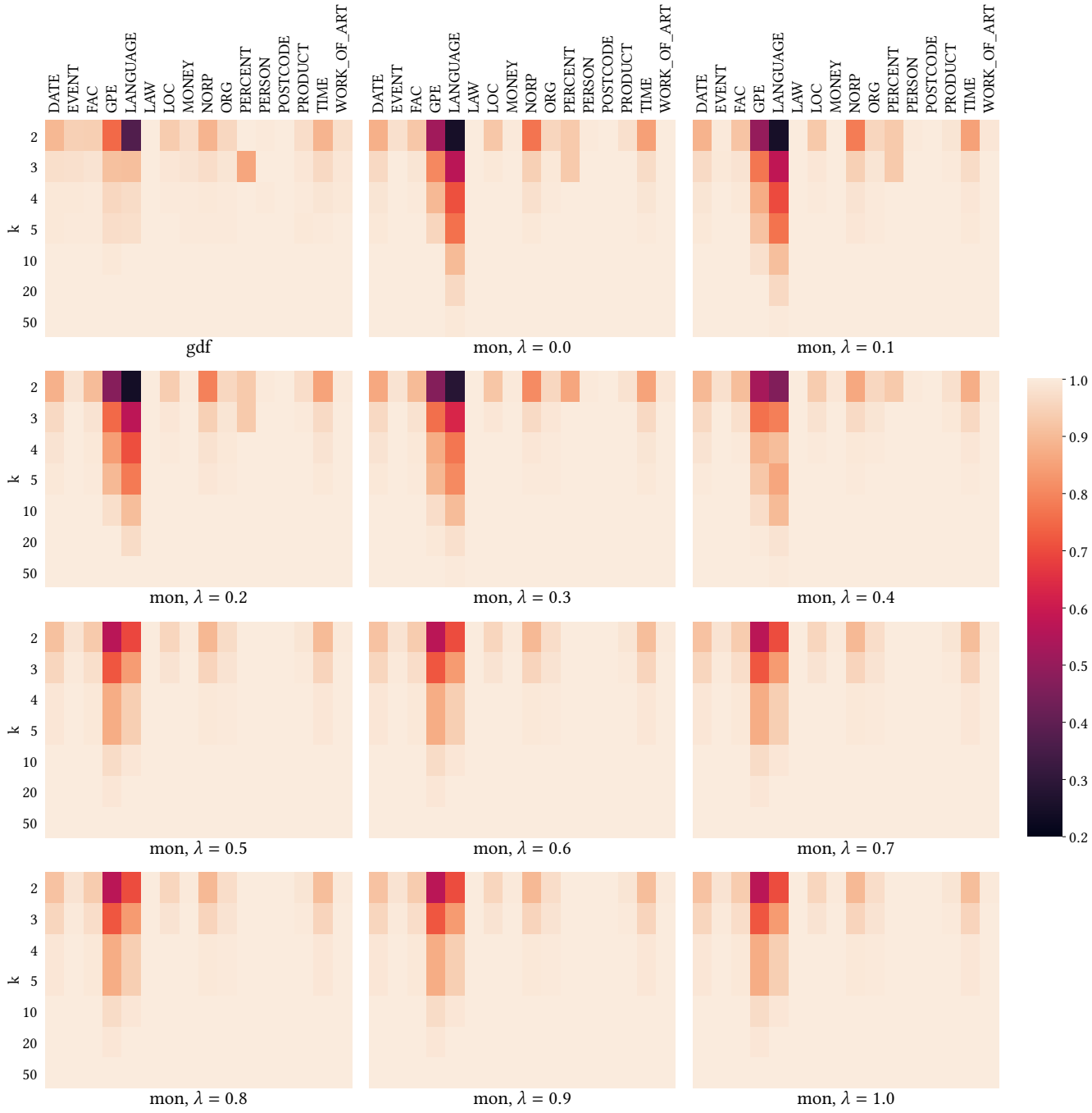


Figure 16: Textual information loss of the attribute *positive review* per entity type for experiments run on the Hotel Reviews Dataset. Information loss is visualized for GDF and Mondrian partitioning with varying λ.