ulm university universität
uulm

Ulm University | 89069 Ulm | Germany

Faculty for Engineering, Computer Science and Psychology
Institute of Databases and Information Systems (DBIS)

A Novel Approach on De-Identification of Heterogeneous Data based on a Modified Mondrian Algorithm

Master's Thesis at Ulm University

Presented by:
Fabian Singhofer
fabian.singhofer@uni-ulm.de
1036479

Supervisors:
Prof. Dr.-Ing. Ansgar Scherp
Prof. Dr. rer. nat. Frank Kargl

Industrial Supervisors:
Aygul Garifullina (aygul.garifullina@bt.com)
Mathias Kern, PhD (mathias.kern@bt.com)

2021

Version March 1, 2021

Name: Fabian Singhofer

Student Number: 1036479

Statement of Originality

I hereby declare that I have written the thesis by myself, without contributions from any sources or aids other than those indicated. I confirm that this work has not been submitted or published elsewhere in any other form for the fulfillment of any other degree or qualification.

Ulm, 01.03.2021

Place and Date

Fabian Singhofer

Contents

List of Figures

List of Tables

A Novel Approach on De-Identification of Heterogeneous Data based on a Modified Mondrian Algorithm

Fabian Singhofer
fabian.singhofer@uni-ulm.de
Ulm University

ABSTRACT

Extensive research has been conducted to anonymize traditional relational as well as textual data independently. In contrast, this work proposes a combined anonymization approach for heterogeneous data composed of relational and textual attributes. We map sensitive terms within texts to the structured domain in order to use k-anonymity to generate a privacy preserved version. In addition, we introduce redundant sensitive information as a concept to anonymize heterogeneous data consistently. To address properties of sensitive terms, we introduce a new partitioning strategy GDF based on global document frequencies of terms as well as a tunable Mondrian implementation. We evaluate our approach using two real-world datasets. Experiments show that our approach is capable of reducing information loss under our privacy model by using the tuning parameter λ to control Mondrian partitioning while guaranteeing k-anonymity for relational attributes as well as for sensitive terms.

CCS CONCEPTS

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

KEYWORDS

data anonymization, heterogeneous data, k-anonymity

1 INTRODUCTION

Research benefits from companies, hospitals, or other research institutions, sharing and publishing their data which can be used for predictions, analytics, or visualization. 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 personal data 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 [11, 15, 31, 33, 51]. 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. Examples show that releasing a combination of different kinds of 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. Even though databases and texts can be mined separately, Zhao and Zhou [61] 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. [53] developed a model for predictive process monitoring which benefits from adding unstructured data to structured data. Therefore, links within heterogeneous data should be preserved, even if anonymized.

For structured data, recent work by Sweeney [52] introduced a well-proven privacy concept called 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 [38, 44, 49], or machine learning methods [8, 15, 24, 33] to allow for replacement in the next step.

However, as the example of a blog dataset in Table 1 indicates, a combined analysis of structured and unstructured data relies on links between both. 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.

There is an absence of work on frameworks providing a combined approach for anonymization of heterogeneous datasets. Therefore, we provide a formal problem definition on anonymizing structured data in combination with free texts. Moreover, we present an anonymization framework for heterogeneous datasets, including the required preprocessing, the concept of redundant information, and a solution for k-anonymity on sensitive terms appearing in texts.

Our work includes the following contributions:

- We formalize the problem of anonymizing heterogeneous datasets composed of traditional relational and textual attributes under the k-anonymity model.
- We answer the question how we can use redundant information to generate a consistent anonymization of heterogeneous data.
- We present an anonymization framework based on Mondrian [31] and adapt the partitioning strategy and recoding scheme for sensitive terms in textual data.
- We introduce λ as a tuning parameter to control the share of information loss in relational and textual attributes.
- We evaluate our approach by measuring state-of-the art statistics on partitions as well as information loss on two real-world datasets.

Table 1: Illustrative example for a de-normalized version of a heterogeneous RX-dataset D consisting of traditional relational data and textual data. 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!

2 RELATED WORK

Automatic de-identification has been the subject of research in the past years. 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) as well as on unstructured texts. Finally, we conclude this section by summarizing the most beneficial insights of recent research for our work.

2.1 Anonymization of Structured Data

Early work of Sweeney [50] showed that individuals, even if obvious identifiers are removed, can be identified by using publicly available data sources and link them to apparently anonymized datasets. Such attempts to reveal individuals using available linkable data are called *record linkage attacks*. Moreover, her work introduced and distinguished 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. Among them, k-anonymity represents an early and straightforward approach which prevents re-identification attacks relying on record linkage using additional data [52]. k-anonymity describes a privacy model where records are grouped and each group is transformed such that their quasi-identifiers are equal. To achieve k-anonymity, Samarati [45] studied suppression and generalization as efficient techniques to enforce privacy.

In addition, recent work by Meyerson and Williams [37] and LeFevre et al. [31] has shown that optimal k-anonymity in terms of information loss both in the suppression model and for the multidimensional case is *NP*-hard. Therefore, several algorithms have been developed to efficiently come up with a k-anonymous version of a dataset while keeping the information loss minimal. Sweeney [51] proposed a greedy approach with tuple suppression to achieve k-anonymity. Moreover, LeFevre et al. [31] suggested a top-down greedy algorithm called Mondrian which implements multidimensional k-anonymity using local recoding models. Additionally, Ghinita et al. [17] showed how optimal multidimensional k-anonymity can be achieved by reducing the problem to an 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. [34] analyzed weaknesses of k-anonymity and therefore introduced the model of ℓ-diversity to prevent *homogeneity* and *background knowledge attacks* possible in the k-anonymity model. ℓ-diversity uses the concept of sensitive attributes to guarantee diversity of sensitive information within groups of records. Moreover, 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 compared to 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. [39] 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. In addition, 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 came up with (k, l)-diversity as a privacy model which is capable of anonymizing 1:M datasets.

k-anonymity has not only been studied for relational data, but also for transactional data. Terrovitis et al. [54] defined the problem where transactional data needs to be de-identified. Given a set of items within a transaction, they treated each item to be a quasi-identifier as well as a sensitive attribute simultaneously. Their 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. [41] showed how k-anonymity can be applied to heterogeneous data consisting of relational and transactional data and stated that a combined approach is necessary to ensure privacy.

2.2 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.

First, Sánchez et al. [46] 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.

Alternatively, 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, the majority of recent work formulated and solved the detection of sensitive information as a NER problem [10, 15, 24, 33, 56].

Early work on NER investigated the possibility to recognize entities using rules and dictionaries. Sweeney [49] suggested a rule-based approach using dictionaries with specialized knowledge of the medical domain to detect Protected Health Information (PHI). Moreover, Ruch et al. [44] 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 de-identification of textual data. Gardner and Xiong [15] introduced an integrated system which uses Conditional Random Fields (CRF) to identify PII. Furthermore, 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. [33] 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. [57] arises the question, whether transformers can also lead to advances in anonymizing free texts. Yan et al. [60] 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 transformers. Their results indicate that transformers are competitive to modern baseline models for anonymization of free texts.

In addition to research on the detection of sensitive information, recent work came up with replacement strategies. Simple strategies involve suppressing sensitive terms with case-sensitive placeholders [44] or with their types [38]. While those strategies are straightforward to implement, a disadvantage is loss of utility and semantics in the anonymized texts. Therefore, more complex strategies involve the generation and usage of surrogates as consistent and grammatically acceptable replacements for sensitive terms [10, 56]. In contrast to the generation of surrogates, Sánchez et al. [46] 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 has been conducted to bring anonymization and replacement techniques studied for structured data to the field of unstructured texts. Chakaravarthy et al. [5] introduced the concept of K-safety and used it to determine the sensitive terms to be anonymized within a single document by investigating their contexts.

2.3 Summary

With the fundamental work on k-anonymity by Sweeney [52], enforcing privacy for publicly available structured data became an interesting research topic. Work in the field of k-anonymity focused on efficient implementations [17, 31, 51], extensions to tackle the model's weaknesses [32, 34], anonymization of relational data in different settings [18, 39], and its applicability on other forms of data such as transactional data [21, 41, 54].

Moreover, literature indicates that a majority of anonymization systems for textual data typically apply NER to detect sensitive terms in texts [8, 24, 33, 56]. In addition, recent work in the field of NLP shows that transformers lead to comparable results for NER tasks while providing good performance [27, 60]. However, Liu et al. [33] emphasizes the importance of a combined approach using rules in addition to machine learning models for sensitive terms which have specific structures such as phone numbers or URLs.

In recent work, anonymization of structured and unstructured data has mostly been considered as two distinct areas of research. However, there has been some work using synergies between both fields. Chakaravarthy et al. [5] used properties from k-anonymity to propose K-safety as a privacy model which can be used to guarantee privacy in text using contexts of sensitive terms. Moreover, to the best of our knowledge, only Gardner and Xiong [15] 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, but only conducted experiments on a small dataset from the medical domain.

3 TOWARDS ANONYMIZATION OF RELATIONAL AND TEXTUAL DATA

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. For

an extensive description of the anonymization process, refer to Section B.

Depending on the dataset to be anonymized, there exist several attributes which need to be anonymized. Table 2 contains a non-exhaustive list of attributes, which typically appear in datasets and are critical with respect to re-identification attacks. We categorize these attributes with respect to their scale and their cardinality of relation. 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. 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 relation depicts the cardinality of a relation between an individual and the attribute. For example, one-to-many 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 an 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.

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. For the purpose of our work, we want to anonymize a given dataset by hiding directly identifying attributes. Moreover, to prevent classical record linkage attacks using quasi-identifying attributes, we use k-anonymity as our privacy model [52]. 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 [47]. Within texts of textual attributes, we adapt k-anonymity to prevent explicit information leakage, while keeping the structure of texts 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 [12, 36] exceeds this work and is therefore not considered.

3.1 Problem Statement

In order to provide a method for anonymizing heterogeneous data composed of relational and textual attributes, we first formalize this problem with respect to its anonymization task. Therefore, we define the properties of the datasets, and its anonymity. Table 3 provides an overview of the notations used.

Table 2: Non-exhaustive list of attributes to anonymize including their scales and cardinality of relations (sorted by cardinality of relation).

Attribute	Scale	Cardinality of Relation
Name [7, 35, 41, 55]	nominal	one-to-one
Social Security Number [35, 55]	nominal	one-to-one
Online identifier [7]	nominal	one-to-one
Passport Numbers [7, 35]	nominal	one-to-one
Home Address [7, 35, 55]	nominal	one-to-one
Credit Card Number [35]	nominal	one-to-many
Phone [35, 55]	nominal	one-to-many
Email Address [7, 35, 55]	nominal	one-to-many
License Plate Number [55]	nominal	one-to-many
IP Address [7, 35, 55]	nominal	one-to-many
Order Reference	nominal	one-to-many
Age [41, 55]	ratio	many-to-one
Sex / Gender [7, 41]	nominal	many-to-one
ZIP / Postcode [55]	nominal	many-to-one
Date of Birth [35, 55]	interval	many-to-one
Zodiac Sign [48]	ordinal	many-to-one
Weight [7, 35]	ratio	many-to-one
Race [7, 35]	nominal	many-to-one
Country	nominal	many-to-one
City [55]	nominal	many-to-one
Salary Figures [13]	ratio	many-to-one
Religion [7, 35]	nominal	many-to-one
Ethnicity [7]	nominal	many-to-one
Employment Information [35]	nominal	many-to-one
Place of Birth [35]	nominal	many-to-one
Skill	nominal	many-to-many
Activities [35]	nominal	many-to-many
Diagnosis / Diseases [13, 35]	nominal	many-to-many
Origin / Nationality [41]	nominal	many-to-many
Purchased Products [41]	nominal	many-to-many
Work Shift Schedules	nominal	many-to-many

3.1.1 RX-dataset. Given data of a dataset D in form of n relations $R_1, ..., R_n$, we can construct our input dataset by using the natural join, i.e. $D = R_1 \bowtie ... \bowtie R_n$. Table 1 shows an example of a dataset composed of two joined relations, 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 define that D is an *RX*-dataset, if one attribute A^* directly identifies an individual, one or more traditional relational attributes A_i[1] contain single-valued data, and one textual attribute X[2] 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 textual values composed of multiple words or even sentences. In the example in Table 1, the relational attributes are the direct

[1] 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).

[2] 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-based attribute which contains 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, $emap : \{A_1, ..., A_n\} \rightarrow E$

identifier *id* as well as the quasi-identifiers *gender*, *age*, *topic*, and *date*. The textual attribute is *text*. We call one row in D a tuple t. Relational attributes A_i are single-valued and can be categorized into being nominal, ordinal, or numerical. 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 $T =< t_1, ..., t_m >$. D contains all data we want to anonymize.

3.1.2 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 $t \in T$ which happens to be sensitive. Moreover, we define a mapping function $emap$ on the set of 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 and is used to match redundant sensitive information, which we discuss below.

3.1.3 Redundant sensitive information. Due to the nature of free texts, some sensitive information might be appearing in a textual attribute as well as in a relational attribute. In order to consistently deal with those attributes, we introduce the concept of redundant sensitive information. Redundant sensitive information is any sensitive term $x \in t.X$ with $er(x) = e_1$ for which a relational value $v \in t.A_i$ with $emap(A_i) = e_2$ exists, where $e_1 = e_2$ and $x = v$. In other words, redundant sensitive information is duplicated information which appears under the same sensitive entity type in a relational attribute $t.A_i$ and as a sensitive term x in $t.X$.

3.1.4 Collection of non-redundant sensitive terms X'. We then 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 schema of D becomes $\{A^*, A_1, ..., A_n, X'\}$.

3.1.5 Person centric view 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

$$D^* =_{A^*} G_{\mathbf{F_1}(A_1),...,\mathbf{F_n}(A_n),\mathbf{F_{X'}}(X')}(D), \quad (1)$$

where A^* denotes a directly identifying attribute related to an individual used to group rows of individuals together, $\mathbf{F_i}$ denotes an aggregation function defined on A_i, and $\mathbf{F_{X'}}$ being the aggregation function for the sensitive terms. 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 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 and dates as well as non-redundant sensitive terms have been aggregated as previously discussed.

3.1.6 Equivalence class in D^*. Based on the notion of equivalence classes of Poulis et al. [41] and the definition of equality of set-based attributes by He and Naughton [21], an equivalence class for D^* can be defined as a partition of records P where for any two records $r, s \in P$ $(r.A_1, ..., r.A_n) = (s.A_1, ..., s.A_n)$ and $r.X' = s.X'$. In particular this means that within an equivalence class, each record has the same values within the relational attributes and their sets of sensitive terms contain the same values.

3.1.7 k-anonymity in D^*. 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.

3.2 Anonymization Approach

Using the definitions from Section 3.1, we present our anonymization approach. In detail, we present how we preprocess our data to generate a person centric view. Moreover, 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. Additionally, we present an alternative partitioning strategy called Global Document Frequency (GDF) partitioning which presents a baseline for partitioning a dataset given sensitive terms. Finally, we use the running example from Table 1 to show how an *RX*-dataset is transformed into a privacy preserved version.

3.2.1 Preparation and Preprocessing. 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

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	$\text{UK}_{\text{location}}$

centric view shown in Table 4, where the textual attribute *text* is omitted for simplification.

First, we need to 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 textual value is "My name is Pedro, I'm a 36 years old engineer from Mexico". The sensitive terms in this example are $Pedro_{\text{person}}$, $36\ years\ old_{\text{age}}$, $engineer_{\text{job}}$, and $Mexico_{\text{location}}$. This analysis of texts needs to be 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 require to find and mark redundant sensitive information using the results of the prior steps. Therefore, we perform row-wise analysis of relational values with sensitive terms to find links, which actually represent the same information. In our previous example, the sensitive term 36 $\text{years old}_{\text{age}}$ depicts the same information as the value *36* in the 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 need to build a person centric view to have a condensed view on all information available for each individual. Therefore, as described in Section 3.1, we will 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 will collect all sensitive terms mentioned in texts of one individual by performing **union** on the sets of sensitive terms.

Table 4 shows the final preprocessed version of our initial example. The preprocessed version depicts a person centric view 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′*.

3.2.2 Anonymization Algorithm. 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. However, in addition to a modified Mondrian partitioning implementation, we propose Global Document Frequency (GDF) partitioning as a new 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. The first step of the algorithm is to find partitions of records with a partition size of at least k. This process is also called partitioning. 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 shown in Algorithm 1. 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. In case of numerical attributes, the normalized range is defined as minimum to maximum. For categorical attributes, the range can be defined as the number of distinct categories observable given a partition. Sensitive terms are treated as categorical attributes.

In addition to this heuristic, we introduce a relational weight parameter λ. λ can be a value between 0 and 1 and 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* - Greedy strict top-down partitioning for relational attributes adapted from LeFevre et al. [31].

Input : Partition P, relational 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`)`
    $P_l = (r \in P | r.A <$ `find_median(`F`))`
    $P_r = P \setminus P_l$
    return `mondrian_partitioning(`P_l`)` $\cup$ `mondrian_partitioning(`P_r`)`
  end

Global Document Frequency Partitioning. Using the idea of a top-down strict partitioning algorithm, we propose a new greedy partitioning algorithm using the presence and absence of sensitive terms. Thereby, 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. However, 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, and therefore build partitions with records which would prevent other partitions from being k-anonymous.

Algorithm 2: *GDF Partitioning* - Strict top-down document-frequency-based 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)
    $P_l = (r \in P | x \in r.X')$
    $P_r = P \setminus P_l$
    $F = F \setminus \{(x, f)\}$ `// remove considered term`
    return `gdf_partitioning` $(P_l, F) \cup$ `gdf_partitioning` (P_r, F)
  end

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 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 $P_{rl} = \{4, 6\}$ 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 a 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, 58]. 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.

Moreover, depending on the scale of the attribute, multiple recoding schemes are available. Nominal and ordinal values are usually recoded using Domain Generalization Hierarchies (DGHs) as introduced by Sweeney [51] and used in multiple other work such as [17, 39, 41, 58]. 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] proposed 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. [11] proposed 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 which are achieved by suppressing information on each level (i.e. day on the first, and month on the second level).

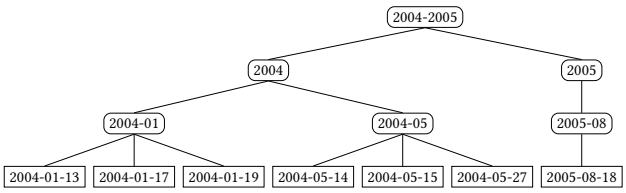


Figure 1: Domain Generalization Hierarchy for date attributes. The leaves depict actual values appearing in the original dataset. The first level of generalization involves suppressing the day. The second level of generalization suppresses the month. Finally, the root can be automatically generated as 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. [11] 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.

After equivalence classes have been determined, relational attributes can be recoded as previously described. Table 5 shows how those recoding schemes are applied to our initial example.

In addition, a k-anonymous representation of the text attribute X' has to be created. Terms which are marked as redundant sensitive information will be 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 $\not{k}$-anonymity for set-valued attributes from Section 3.1, and therefore 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 $\not{k}$-anonymity and therefore 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.

4 EXPERIMENTAL APPARATUS

We evaluate our framework by running experiments using data from two real world applications and collect state-of-the-art metrics. In order to evaluate our anonymization approach, we implemented our framework in Python, using well-known libraries such as Pandas and NumPy for data manipulation as well as spaCy for NLP related tasks. Within spaCy, we use a transformer-based language model for English (Version 3.0.0a0) with an F1-score for NER tasks of 89.41[3]. We implement well-known Mondrian partitioning as well as our new GDF partitioning and compare them with regard to statistics on resulting partitions as well as information loss. In addition, we use λ to manipulate splitting decisions in Mondrian as discussed in Section 3.2. Even though our anonymization framework runs on commodity hardware, we run our experiments on a multi-core server to parallelize execution of experiments.

4.1 Datasets

To run our experiments, we have a strong requirement for a dataset to 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 3.1). A dataset which fits these requirements is the *Blog Authorship Corpus*[4], a dataset which was originally used to create profiles from authors [48]. We also used the schema of the Blog Authorship Corpus throughout our running examples in Tables 1, 4, and 5. This corpus has also been used in the field of privacy research with respect to author re-identification [28].

After cleaning the input data, 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*, *age*, *topic*, and *zodiac sign* of a blogger as well as the *date* and the *text* of the published blog entry. Each record 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.

Texts within the Blog Authorship Corpus contained unreadable characters as well as HTML-characters and unnecessary spaces. To ensure proper NLP processing, we cleaned the texts of the dataset by removing HTML-tags as well as non-ASCII characters and unnecessary spaces. Data cleansing improved entity detection significantly.

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 12 astrological signs. The textual attribute has been processed as previously described.

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 which is available on Kaggle[5]. We refer to this dataset as the *Hotel Reviews Dataset*. This dataset contains 17 attributes, where 15 attributes are considered relational, and two attributes are considered textual. The textual attributes are *positive* and *negative reviews* of users. Both textual attributes are preprocessed and cleaned as previously described for the Blog Authorship Corpus.

Among the relational attributes, we treat *hotel name* and *hotel address* as direct identifiers. *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.

4.2 Procedure

As a baseline, we assume 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. However, with our experiments we want to show that we can improve (i.e. reduce) the information loss in texts under our $\not{k}$-anonymity model. Moreover, we want to optimize the trade-off between relational and textual information loss.

We use both the Blog Authorship Corpus and the Hotel Reviews Dataset for our experiments. The intermediate results of the NLP preprocessing are saved to disk to speed up consecutive runs of our anonymization tool, since texts and their recognized entities always remain the same for all runs. Similar to experiments conducted in prior work [17, 18, 41], we run our anonymization tool for different values of k (2, 3, 4, 5, 10, 20, 50). Sensitive entity types considered in texts are all entity types in

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

[4] Accessed from https://www.kaggle.com/rtatman/blog-authorship-corpus on September 30, 2020.

[5] Accessed from https://www.kaggle.com/jiashenliu/515k-hotel-reviews-data-in-europe on September 15, 2020.

Table 5: Anonymized version of the illustrative example with k = 2. Redundant information as well as 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!

Table 12 and Table 13 except the unspecific types QUANTITY, ORDINAL, and CARDINAL. 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 also vary the weight parameter λ to tune Mondrian. We vary λ from 0 to 1 in 0.1 steps. 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 its information loss and the corresponding partitioning sizes, as described in the following section.

In addition to the previous experiment setup, we repeat the identical experiments just considering location entities with entity type GPE. We use those experiments to showcase an anonymization task with reduced complexity. We chose to only consider location based entities since they are present in blog posts as well as in hotel reviews and therefore allow for comparison of both datasets.

4.3 Measures and Metrics

In order to evaluate our anonymization approach and compare results of partitioning, we introduce metrics measured for each run of the experiments. In particular, we compare statistics on partitions as well as relational and textual information loss.

4.3.1 Statistics on Partitions. We are interested in the resulting partitions of the anonymized dataset. First, we evaluate how partitions are created and how λ influences splitting decisions. Therefore, in the case of Mondrian partitioning, we evaluate the number of splits of partitions based on relational attributes versus textual attributes. We expect that for $\lambda < 0.5$ we observe more splits on textual attributes and for $\lambda > 0.5$ more splits on relational attributes.

In addition to the number of splits, we want to evaluate 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.

4.3.2 Information Loss. Measuring the information loss of an anonymized dataset is well-known practice for evaluating the amount of utility remaining for a published dataset. We will use Normalized Certainty Penalty (NCP) introduced by Xu et al. [58] 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 adapt the definitions of NCP introduced in [58] such that for one record r we calculate the information loss as

$$NCP(r) = \frac{w_R \cdot NCP_A(r) + w_X \cdot NCP_X(r)}{w_A + w_X}, \tag{2}$$

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. Similarly we define w_X and $NCP_X(r)$ for the textual attribute. For our evaluation, we set both w_A and w_X to 1.

For **relational attributes** $A = \{A_1, ..., A_n\}$ we can define the information loss as

$$NCP_A(r) = \frac{\sum_{A_i \in A} NCP_{A_i}(r)}{|A|}, \tag{3}$$

where $|A|$ denotes the number of relational attributes and NCP_{A_i} is the information loss for a single attribute and can be calculated either using NCP_{num} for numerical attributes or NCP_{cat} for categorical attributes. NCP_{num} for **numerical values** is defined as

$$NCP_{num}(r) = \frac{z_i - y_i}{|A_i|}, \tag{4}$$

with z_i being the upper and y_i being the lower boundary of the recoded interval and $|A_i| = max_{r \in D^*}(r.A_i) - min_{r \in D^*}(r.A_i)$. Moreover, for **categorical values**, NCP_{cat} can be calculated as

$$NCP_{cat}(r) = \begin{cases} 0 & |u| = 1 \\ \frac{|u|}{|A_i|} & \text{otherwise,} \end{cases} \tag{5}$$

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. Moreover, for date attributes, $|u|$ denotes the number of leaves of the subtree below the recoded value.

Similarly to the relational attributes, we can define NCP_X for the **textual attribute** as

$$NCP_X(r) = \frac{\sum_{x \in r.X'} NCP_x(x)}{|r.X'|}, \tag{6}$$

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

$$NCP_x(x) = \begin{cases} 1 & x \text{ is suppressed} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Finally, we can calculate the **total information loss** for the RX-Dataset D^* as

$$NCP(D^*) = \frac{\sum_{r \in D^*} NCP(r)}{|D^*|}, \tag{8}$$

where for each record r the information loss $NCP(r)$ is calculated and afterwards divided by the number of records $|D^*|$.

5 RESULTS

Given the experimental setup from Section 4, we evaluate the results based on partition statistics and information loss. While we focus on results of experiments run on the Blog Authorship Corpus, we also compare and validate our findings with results of experiments run on the Hotel Reviews Dataset. For a complete overview of the experiment results refer to Section C.

5.1 Partitions

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. To 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. In addition to assessing the share of splitting decisions for relational and textual attributes, we also evaluate the number of resulting partitions and statistics with respect to their sizes.

5.1.1 Partition Splits. Figure 2 shows the distribution of splitting decisions for experiments run on the Blog Authorship Corpus considering all entities for $k = 5$. Partition splits are only evaluated for Mondrian partitioning, since λ is not available for GDF partitioning. As λ was designed, $\lambda = 1$ results in only splits on relational attributes whereas $\lambda = 0$ results in splits only on sensitive terms. 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 $\lambda > 0.5$, there is no relevant change since relational attributes are considered almost every time throughout the partitioning phase. However, for $\lambda < 0.5$, we can control the share of splitting decisions of Mondrian between relational and textual attributes.

Figure 7 compares partition splits for all experiments run on the Blog Authorship Corpus. 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. Another interesting observation is that if only locations (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.

We can verify our observations on splitting decisions observing results of the Hotel Reviews Dataset shown in Figure 8. The number of splits is generally lower, since the Hotel Reviews Dataset contains less records. 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.

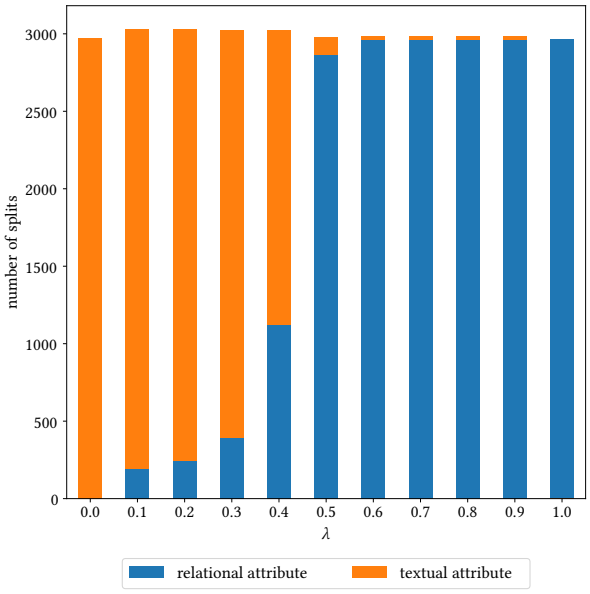
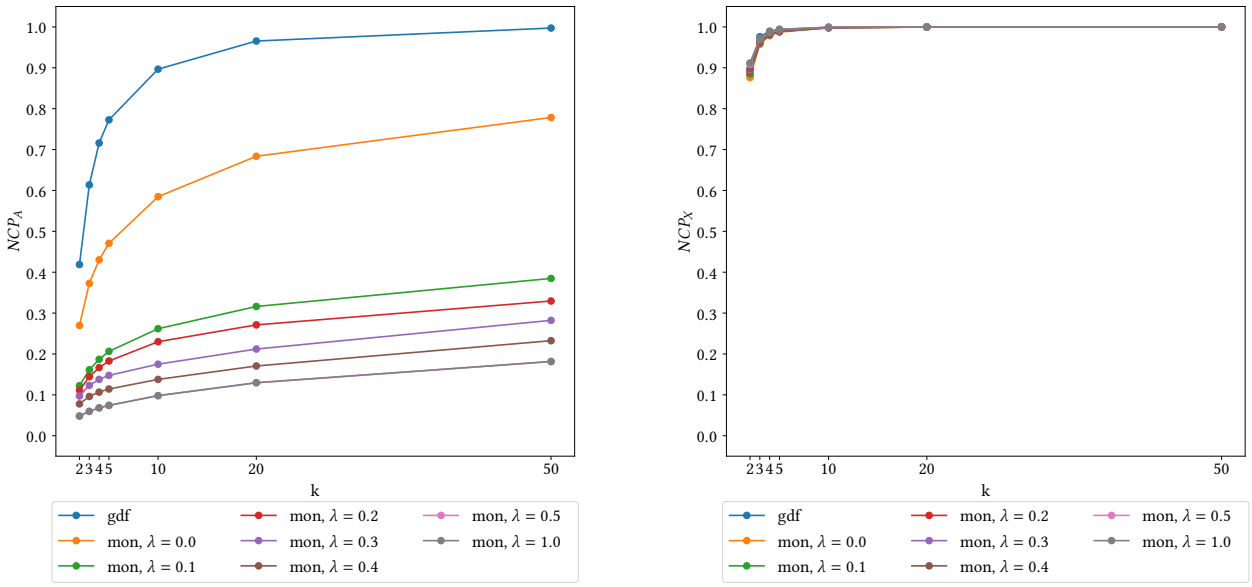


Figure 2: Distribution of splitting decisions using Mondrian partitioning with varying relational weight λ for k = 5 running on the Blog Authorship Corpus considering all entities.

5.1.2 Partition Count and Size. Table 8 and Table 9 provide statistics on partitions using GDF partitioning as well as Mondrian partitioning with varying λ for the Blog Authorship Corpus considering all entities and only GPE entities respectively. One observation is that GDF partitioning is not able to generate partition sizes close to k, compared to Mondrian partitioning. Moreover, we can state that in case of Mondrian partitioning, the partitioning algorithm ends up with the same partitioning layout for values of λ between 0.6 and 0.9. This observation matches statistics on partition splits, since for those values of λ the algorithm decides on the same attributes to split on.



(a) Relational information loss NCP_A

(b) Textual information loss NCP_X

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

If only location entities are considered, $\lambda = 0$ leads to bigger and fewer partitions compared to other settings for λ. If GDF is compared with Mondrian using $\lambda = 0$, 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.

Looking at results of experiments run on the Hotel Reviews Dataset in Table 10 and Table 11, we can confirm our observations. However, due to the lower number of records in the Hotel Reviews Dataset, the count of partitions ends up comparatively lower.

5.2 Information Loss

In addition to statistics on partitions, we are interested in how GDF partitioning performs against our tunable Mondrian implementation with respect to the introduced information loss in the released dataset. Figure 3 provides an overview on relational and textual information loss for experiments run on the Blog Authorship Corpus considering all entities. We omit results for values of λ between 0.6 and 0.9, since results are almost identical with runs using $\lambda = 0.5$. For a full comparison of information loss for experiments run on the Blog Authorship Corpus, see Figure 9 as well as Figure 11 and Figure 12 for zoomed versions of plots showing textual information loss. Similarly, Figure 10 provides an overview of the information loss for experiments run on the Hotel Reviews Dataset. Zoomed plots for textual information loss are available in Figure 13 and Figure 14.

5.2.1 Relational Information Loss. We can state that the information loss in all categories and throughout all experiments increases with larger numbers for k. Higher information loss is caused by having larger partitions and therefore higher efforts in recoding (ref. to previous section). Furthermore, we can state that information loss in the relational attributes increases if the tuning parameter λ decreases (see 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 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. 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).

For the Hotel Reviews Dataset, we can state that relational information loss appears to be higher in general compared to the Blog Authorship Corpus. However, we can still observe the same behavior where higher values of λ result in relatively lower relational information loss. Therefore, we can confirm our observations for relational information loss.

5.2.2 Textual Information Loss. Analyzing the information loss in the textual attribute, 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.

Experiments on the Hotel Reviews Dataset considering all entities confirm these observations. Textual information loss for

$k \leq 5$ tends to be slightly lower. However, if only locations are considered, textual information loss in hotel reviews can significantly be reduced. 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 $k \leq 10$.

To get a deep understanding of textual attributes, we analyzed textual information loss on an attribute based level. Figure 4 provides an overview of information loss per entity type of the attribute *text* in the Blog Authorship Corpus for $\lambda = 0.2$. Results indicate that information loss for sensitive terms of type LANGUAGE can significantly be reduced for values of $k \leq 5$. Since the number of distinct entities with type LANGUAGE is significantly lower compared to other entity types in the Blog Authorship Corpus (cf. Table 6), information (i.e. number of sensitive terms) for entities with type LANGUAGE can be better preserved. We can confirm this observation for Mondrian partitioning with $\lambda \leq 0.4$. For a full comparison on information loss per entity type of the attribute *text* in the Blog Authorship Corpus, refer to Figure 15.

We can also confirm this observation with the Hotel Reviews Dataset (see Figure 16 and Figure 17). In addition to LANGUAGE entities, sensitive locations (GPE) can also be preserved for both textual attributes.

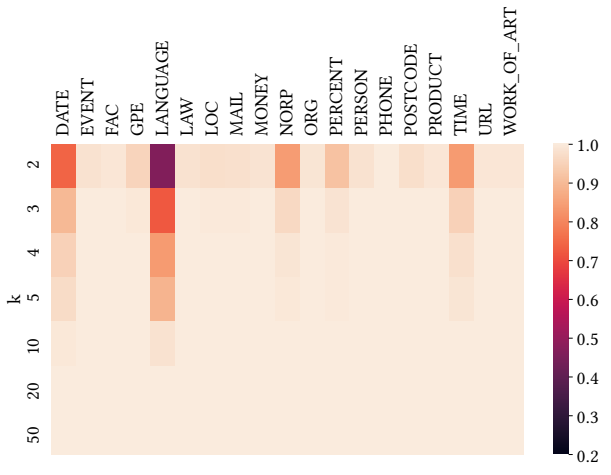


Figure 4: Detailed textual information loss of the attribute *text* per entity type for experiments run on the Blog Authorship Corpus with Mondrian partitioning and $\lambda = 0.2$.

6 DISCUSSION

In this work, we presented a novel approach on de-identification of heterogeneous data consisting of traditional relational as well as textual data. We introduced the concept of redundant sensitive information being the same information appearing in relational and textual attributes and used it to produce a consistent and privacy preserved dataset. Moreover, we showed that sensitive terms within texts of textual attributes can be treated as a structured anonymization problem which we solved using $\mathcal{k}$-anonymity as introduced by Sweeney [52]. We implemented our anonymization approach introduced in Section 3.2 and evaluated it on two real-world datasets by measuring statistics on partitions as well as information loss.

We used Mondrian as introduced by LeFevre et al. [31] to find equivalence classes with sizes $\geq k$. Mondrian was designed to automatically partition a multi-dimensional dataset using heuristics to deduce the next attribute to split a partition on. Due to heterogeneity of sensitive terms in texts, by default, they were less likely considered to split on. By introducing the tuning parameter λ, we were able to control Mondrian to preserve more information in either relational, or textual attributes. Experiments showed that for heterogeneous data it may be required to tune partitioning in order to favor information preservation in textual attributes over relational attributes.

Moreover, looking at the information loss, our anonymization approach allowed us to reduce the information loss in texts under the $\mathcal{k}$-anonymity privacy model. In recent work [8, 33, 49] sensitive terms have been completely suppressed. However, our experiments verified that for $k \leq 5$ not all sensitive terms require to be suppressed. In particular, for entities of type LANGUAGE our approach could preserve up to 80 % of terms for $k = 2$ in the Hotel Reviews Dataset and about 60 % for $k = 2$ in the Blog Authorship Corpus. Based on these results, we can infer that applying $\mathcal{k}$-anonymity on sensitive terms works best for homogeneous texts from closed domain, since the diversity of sensitive terms limits the applicability of $\mathcal{k}$-anonymity to text attributes.

6.1 Threads to Validity

While our approach presents a general framework to anonymize heterogeneous data, our choices on detecting and comparing sensitive terms have an impact on experiments outcome. First, we consider all sensitive terms in 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.

Moreover, we can state that the accuracy of detecting sensitive terms impacts anonymization results. Those impacts can be categorized in over-anonymization and under-anonymization. While the former depicts the case where sensitive terms are falsely suppressed, the latter describes a case where sensitive terms are falsely kept.

Over-anonymization is caused by low precision. First, if terms which do not pose any risk of identity disclosure are detected and anonymized, the text might lose utility since important structures within the text might be missing. Moreover, if sensitive terms are labeled with false entity types, they might also falsely be anonymized, since our strict definition of $\mathcal{k}$-anonymity requires also entity types to be equal. In addition, over-anonymization can also be caused by differently written sensitive information with the same meaning. Imagine that one text refers to Berlin by mentioning the city's actual name "Berlin" while another text refers to Berlin by the term "the capital of Germany". While both references refer to the same entity, the city Berlin in Germany, it is not possible for our system to resolve such linkage. We refer to such cases as false negative matches. False negative matches can be reduced by introducing synonym tables, semantic rules, and by using different metrics to compare terms such as Levenshtein distance to cope with spelling mistakes.

In contrast, under-anonymization 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. Another reason for under-anonymization are identical terms which actually have different semantics. Suppose that the phrases "I live in Berlin" and "I love Berlin" appear in two different records which happen to be grouped into the same partition. In this case, our approach would treat both appearances of "Berlin" the same way even though the term in the first case is referring to a place of residence while in the second case a preference is being expressed. We refer to such a scenario as false positive matches. To cope with false positive matches, one suggestion is to also consider the surrounding context by comparing Part-of-Speech-Tags and dependencies of terms within and across sentences.

Both, 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.

6.2 Generalizability

Our anonymization approach shows that anonymization of heterogeneous data can be beneficial if a consistent anonymized version of data is required. In particular, if there is a high overlap between relational and textual data, a combined anonymization approach is favorable. Our work has multiple implications which can be beneficial for other work.

First, we showed that anonymizing unstructured data like free texts can be achieved by extracting and converting sensitive information into a structured anonymization problem. Since our approach aims to combine structured and unstructured data, we can generalize the concept also for semi-structured data (e.g. JSON). Therefore, 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 data.

In addition, tuning 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 (e.g. age), and less information in others.

6.3 Future Work

Our work enables several paths for future work. 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.

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.

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.

We used $\mathscr{k}$-anonymity as the privacy model to prevent identity disclosure. Even though $\mathscr{k}$-anonymity establishes guarantees on privacy, it does not guard against attacks where adversaries have access to background knowledge. Differential privacy introduced by Dwork [9] resists such attacks by adding noise to data and could be an alternative privacy model applicable to anonymize *RX*-datasets. 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 [12, 59].

7 CONCLUSION

In this work, we have made the first step towards a combined framework for anonymizing heterogeneous datasets consisting of traditional relational as well as textual attributes using the privacy model $\mathscr{k}$-anonymity. We have formulated the problem by transferring sensitive terms in texts to an anonymization task of structured data. Experiments on two real-world datasets have shown that our anonymization approach can be successfully used to anonymize heterogeneous data. Compared to recent approaches, we have offered anonymization of textual attributes retaining sensitive terms under the $\mathscr{k}$-anonymity model. Furthermore, we have shown that by introducing a tuning parameter λ in Mondrian partitioning, we were able to control and prioritize information loss in relational and textual attributes.

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.

Supplementary Materials

A EXTENDED RELATED WORK

In addition to Section 2 we present related work for anonymization of other types of data. Moreover, we present an overview of regulations and their view on PII. Finally, we present a non-exhaustive overview of anonymization tools and frameworks available.

A.1 Additional Efforts in Data Anonymization

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. [14] focused on live de-identification of faces in video streams.

Finally, McDonald et al. [36] 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.

A.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.

A.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 [55]. 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 [55]. 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

A.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].

A.2.3 Guidelines by NIST. In contrast to the GDPR, the National Institute of Standards and Technology (NIST) provides guidance on protecting PII [35]. 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 ..." [35]. 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 [35]. Examples for linked or linkable attributes are:

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

A.3 Existing Anonymization Tools and Frameworks

Multiple publicly available tools and frameworks for anonymization of data have been released. *ARX*[6] is an open source comprehensive software providing a graphical interface for anonymizing structured datasets [42, 43]. 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*[7] 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 containing set-valued data fields. Finally, *Privacy Analytics*[8] 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*[9] is a Python implementation of the Datafly algorithm introduced by Sweeney [51] as one of the first algorithms to transfer structured data to match k-anonymity. Additionally, *Crowds*[10] 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. [11] to achieve k-anonymity. Finally, an example for an implementation of the Mondrian algorithm [31] is available for Python[11] 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*[12] 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)*[13] is a suite of tools for identifying and redacting PII in free-text medical records [26]. *deid*[14] is a tool which allows anonymization of free texts within the medical domain. Finally, *deidentify*[15] 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 [56].

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

B ANONYMIZATION GUIDELINES

This section builds a fundamental idea on how anonymization of any data works, which steps are involved, and what should be considered throughout this process. The task of anonymizing a dataset, regardless of the type of data, is a complex multi-step process. The typical process of de-identifying a dataset involves the following steps [23]:

(1) Assess the intended audience to decide on the release model. There exist public, semi-public, and non-public release models which have an impact on the risk of re-identification.
(2) Specify and name direct-identifiers, quasi-identifiers (also called semi-identifiers), and sensitive attributes. For structured data, each column of a dataset can directly be labeled as either one of the three previously mentioned types, or as an insensitive attribute. For unstructured free text, manual or automatic recognition of attribute types has to be done.
(3) Assess the risk of re-identification by analyzing the adversary and possible re-identification attacks.
(4) Calculate the amount of de-identification needed. This also includes the evaluation of de-identification techniques which should be applied (refer to Section 2).
(5) De-identify the dataset.
(6) Verify the de-identified dataset on privacy, but also data utility.
(7) Document the process.

B.1 Audience

In order for data to be anonymized, it is important to know who actually will have access to the anonymized data. Within the terminology of PPDP, the audience which will have access to this data is also called data recipient(s) [13]. An anonymized dataset is also called release. A release is always associated with a release model, which determines who will actually be accessing the data. Release models can be categorized into *public*, *semi-public*, and *non-public* [23]. For public releases of data, no assumptions on the audience or the purpose of the released data can be made. In some sense, this relates to the *Release-and-Forget* model introduced by Ohm [40], which states that after data is released (e.g. in the internet), the publishing process can actually not be reversed. Non-public release models refer to releases where only a selected audience is able to access the released data and each data recipient needs to agree on the terms of conditions which apply. Finally, semi-public releases also require the audience to accept the conditions of accessing and using the data, but the audience could be the public.

B.2 What should be anonymized?

In order for data to be processed, one must understand the dataset. The same applies when it comes to data anonymization, where it is important to determine what data attributes actually represent and how they contribute to the risk of identity disclosure. Therefore, we provide a list with aspects to consider while working with data and creating a privacy-preserved version of this data.

B.2.1 Cardinality of relation. Given an individual and corresponding fields within a dataset, attributes can be distinguished by their directed cardinality of the relation to the individual. If an attribute relates to exactly one person and only this person appears to have this attribute, then the relation is called one-to-one. If a person relates to multiple measurements of an attribute, but a value of this attribute relates to only one individual, then this relation is called one-to-many. The opposite, namely many-to-one describes that one individual has exactly one value of this attribute, but this attribute can appear for multiple individuals. Finally, many-to-many are relations where one individual can be related to multiple values of an attribute while one value of an attribute can also happen to be measured for multiple individuals.

Suppose we want to protect an individual identity disclosure. Then the name of the individual depicts an one-to-one attribute, since one name exactly matches one individual (ignoring the distribution of names and possible duplicates). Moreover, a list of the credit card numbers this individual possesses (could be multiple) is a one-to-many relation. In contrast, the age of the person is a many-to-one relation, since an individual has only one age, but one specific age value appears most-likely for multiple individuals. Finally, a many-to-many attribute can be a list of hobbies, where one individual favors more than one hobby, but other individuals might also favor the same hobby.

B.2.2 Anonymization specific classes. When it comes to the task of anonymization of a dataset, the cardinality of relations provides valuable insights for their risk of violating privacy [16]. Therefore, literature has come up with four categories which can be used to determine the importance for the anonymization task. *Direct identifiers* are one or more attributes which can directly be used to identify an individual. Their cardinalities of relation are one-to-one and one-to-many. Examples are names, email addresses, credit card numbers, or social security numbers. In contrast, *quasi-identifiers* are attributes which can not directly be used to identify an individual, but pose potential to help identifying a person if the adversary can use other public datasets or has background knowledge. Examples are age, gender, ethnic origin, or profession. They usually have many-to-one or many-to-many cardinalities. *Sensitive attributes* are attributes which do not reveal an individual's identity, but depict critical information which should not leak to the adversary since it could harm data subjects. Examples are salary numbers or medical conditions. Finally, *insensitive attributes* are attributes which do not pose a risk to neither identity disclosure nor attribute disclosure.

B.2.3 Regulations. In many cases, regulations from official authorities state which data is personal and therefore needs to be anonymized (also refer to Section A.2). The GDPR introduces the term *personal data* and defines it as any information relating to an identified or identifiable natural person [7]. Furthermore, the NIST defines PIIs as any information that can be used to distinguish or trace an individual's identity (direct identifiers) and any information that is linked or linkable to an individual (quasi-identifiers) [35]. The Safe Harbor Method of the HIPAA defines 18 attribute classes, which should be completely removed. Among others, those are names, phone numbers, geographic locations, email addresses, medical record numbers, and dates (for a full list see Section A.2). However, there are also cases where dates have not to be removed, e.g. if only years remain. Similarly, only geographic entities which are smaller than states and which contain less than 20,000 people need to be removed.

The following enumeration provides examples of attributes discussed in the previously mentioned regulations and maps them to the classes direct identifiers, quasi-identifiers, and sensitive attributes. Note that quasi-identifiers and sensitive attributes might be interchangeable depending on whether we assume that the adversary might have background knowledge (quasi-identifier) or not (sensitive attribute).

- Direct Identifiers
  - Name
  - Social security numbers
  - Passport numbers
  - Email addresses
- Quasi-Identifiers
  - ZIP
  - Age
  - Place of birth
  - Sex
- Sensitive Attributes
  - Diseases
  - Salary figures

B.3 Determine the risk model

In PPDP one assumption is that the released dataset might be accessed by an adversary, who wants to use the data in a misbehavioral manner. Therefore, depending on the audience, assumptions on the background knowledge, and possible attacks executed by the adversary, a risk model needs to be defined which guards against specific attacks.

There exists a variety of attacks on released data. In any attack, the motivation for the adversary is to gain information using the released data as well as any background knowledge available to him. Attacks can be grouped into linkage attacks and probabilistic attacks, where the former category contains attacks where we assume the attacker knows the quasi-identifier of a person while the latter category are attacks where an adversary changes his beliefs [13].

First, the *record linkage attack* describes an attack where an adversary uses additional data which he might observe or is publicly available to identify an individual's record within a released dataset. Sweeney [52] showed that by using a publicly available voter's list which she bought for $20, she could identify individuals in a medical dataset by linking both sets of data on the attributes sex, birth date, and ZIP.

In contrast, the *attribute linkage attack* does not aim on revealing an identity within a dataset, but instead aims on gaining sensible information about an individual. This attack requires for a released dataset to contain at least one sensitive attribute, which an adversary might learn about. This attack works if an individual can be linked to a group of records within the released data, and this group shares some sensitive information about an individual. In particular, if sensitive attributes within a group of individuals are homogeneous, someone can easily learn some sensitive information about an individual with certain confidence. If an adversary uses the homogeneity of sensitive attributes, this attack is called *homogeneity attack*. In addition, Machanavajjhala et al. [34] introduced the idea of *background knowledge* attacks where sensitive information might be deduced by limiting the number of possibilities within a group of people using publicly available background knowledge such as statistics on diseases.

Moreover, the *table linkage attack* is an attack on a released dataset where an adversary tries to know whether a particular individual is present in a dataset or not. Harm might already be

made to an individual if an adversary knows that this individual appears in a database (e.g. from a hospital).

Finally, *probabilistic attacks* are attacks where the posterior belief of an attacker has significantly changed from the prior belief due to the released data. In other words, if the adversary cannot gain information from the released data and therefore cannot adapt his beliefs, the probabilistic attack fails. Compared to linkage attacks, the group of privacy models protecting against such attacks do not differentiate quasi-identifiers and sensitive attributes, but make probabilistic statements on how much information can be learned by a released dataset.

B.4 Amount of anonymization needed

Determining the amount of anonymization needed depends on multiple factors, namely the type of properties with respect to the de-identification task, who the intended audience for the release is, and what assumptions on the adversary can be made. Suppose that we make the unreal assumption that an adversary has no background knowledge. Then, in order to guard for identity disclosure, it is enough to remove obvious direct identifiers (such as names) and keep all quasi-identifiers and sensitive values to preserve high utility. However, this assumption does not hold in reality, since due to publicly or semi-publicly available datasets, some background knowledge is available to any attacker.

Moreover, the number of quasi-identifiers available in a released dataset and the context of the released data has an impact on the amount of anonymization needed. Suppose that a released dataset only contains the quasi-identifiers age and gender. Then, without having additional knowledge available within this dataset, we can assume that a combination of age and gender relates to a high number of individuals in the world. By introducing a third quasi-identifier, e.g. ZIP, we make the individuals which appear in the data more specific. In particular Sweeney [50] showed that by using the ZIP, birth date, and sex of individuals available in a semi-public voter registration list, she could match records of a medical dataset to individuals by linking the tables. In addition, she states that 87.1 % of the population can be uniquely identified by just having the quasi-identifiers ZIP, gender, and date of birth. Therefore, bigger efforts on anonymizing data with several quasi-identifiers has to be undertaken since more information about individuals is present.

Additionally, if you put the release into context, say that the released data is generated from an employee list of a company, the number of individuals having the combination age, gender, and "employee of this company" relates to a significantly smaller group of people. Therefore, the amount of anonymization needs to be carefully determined prior to releasing an anonymized version of the data.

B.5 De-identification mechanisms

After determining the release model, identifying critical attributes with respect to identity, and choosing attribute disclosure and the risk model, the de-identification mechanism which should be applied on the dataset can be determined.

B.5.1 Strategies. The strategy which should be used to generate an anonymized version of a dataset with respect to predetermined criteria is critical and has impact on the released dataset. Sweeney [52] introduced a group-based anonymization method called k-anonymity, which establishes equivalence classes within a dataset with a size of at least k, which share their quasi-identifiers. Therefore, after removing all direct identifiers, a record linkage attack is not possible, since at least k records are not indistinguishable from each other with respect to their quasi-identifiers. However, as Machanavajjhala et al. [34] showed, a k-anonymous dataset is still vulnerable against homogeneity or background knowledge attacks. In contrast, probabilistic approaches exist which make use of perturbation of data items in order to limit the additional knowledge an adversary can gain from a dataset. The most prominent definition of such an anonymization strategy is differential privacy introduced by Dwork [9]. Differential-privacy is a concept where no assumptions about background knowledge of adversaries are made. Instead, a differential private release of a dataset does not "differ" if the raw dataset contains a specific record or not. This is achieved by making probabilistic statements about data items. However, due to the noise added to data items, values might differ from reality.

B.5.2 Scale of data. When it comes to applying transformations (grouping, adding noise, etc.) to attributes, it is important to consider their actual data types to determine allowed operations. Given an arbitrary dataset, attributes can be distinguished by their actual data types and therefore their corresponding scale. Atomic attributes which appear in relational datasets can be grouped into four main groups, namely nominal, ordinal, interval, and ratio. Nominal values are qualitative and can only be used to compare for equality. Ordinal values are also qualitative, but values can be ordered. The interval level of measurement contains quantitative values and allows to add and subtract values. Finally, ratio scales allow for multiplication and division. Moreover, quantitative data can be either discrete or continuous. If nothing is known about an attribute type, attributes can always be treated as a nominal by default.

A nominal attribute can be compared for equality. This allows to find duplicates, or group those attributes into a set of attributes. This set however is not ordered since nominal attributes do not have a natural ordering. An example for a nominal attribute would be gender, where values can be compared to each other and be grouped in sets, but there is no order applicable. If we introduce ordering of qualitative attributes, we can now rank those values and group them such that close values are within one set. Levels of education (e.g. high school, Bachelors, Masters, PhD) is an example for an ordinal attribute, since they can be naturally ordered. This allows us to form sets where we can group neighboring values together to reduce the spans of those sets. Attributes of type interval can be added and subtracted, which enables those attributes not only to be grouped into sets, but also for a mean to be calculated. Moreover, ratio attributes can also be multiplied and divided. However, for anonymization purposes the distinction between interval and ratio is not of relevance. Therefore, we will refer to those types as numerical. An example for a numerical attribute is age, where the mean of the age characteristic of a group of people can be calculated.

B.5.3 Domain Generalization Hierarchies. Sweeney [51] introduced Domain Generalization Hierarchies (DGHs) as an useful concept to generate more general versions of a specific value. A generalization hierarchy can be seen as a tree where more general terms are closer to the root while values get more specific towards the leaves. In other words, the domain of nodes at each level decreases towards the root. Figure 5 shows a hierarchy for categorical attributes, where each ancestor of a node within the hierarchy is a more general term for the node itself. This concept can also be applied to semi-structured categorical values such as postcodes, where characters within the values represent

some kind of grouping (refer to Figure 6a) as well as to numerical data, where larger groups of values depict a generalized representation (refer to Figure 6b). Moreover, within a group of leaves, the domain might be finite (e.g. there exists a limited number of countries or companies) while there can also be domains with an infinite number values (e.g. for person names).

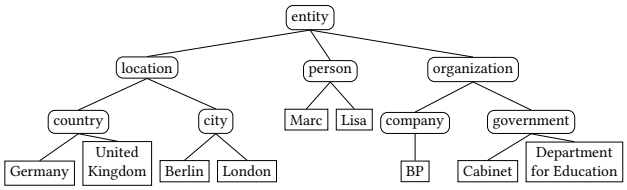
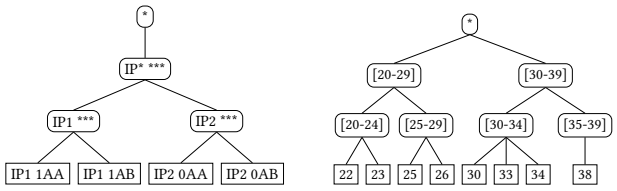


Figure 5: Domain Generalization Hierarchy for nominal attributes. Entity is used as root.



(a) DGH for postcodes in Ipswich, UK

(b) DGH for the age of persons

Figure 6: Automatically generated Domain Generalization Hierarchies.

B.5.4 Suppression. Suppression is the process of replacing a specific value with a placeholder to indicate that this entry is not disclosed. Furthermore, suppression can also be used on a whole data entry which then will be removed from the set of entries of the data [45]. By using suppression, statistical outliers which would disclose an entity in a dataset, are removed in order to not reveal identities. However, suppression alters the distribution within data and therefore reduces its quality, which might limit the usefulness of a release [52].

B.5.5 Generalization. An alternative approach to suppression is generalization. Generalization tries to reduce a specific characteristics of a data record to a more general one by applying a generalization function [51]. The generalization function uses DGHs and maps the input (a node in the DGH) to its ancestor (also a node in the DGH) as the output. Given the DGH in Figure 5, if "Berlin" is the input, the generalization function would return "city" as a more general representation of "Berlin". The advantages of using generalization over suppression are that data items do not get removed, but still remain in a generalized form within the released dataset. However, generalization always requires DGHs which cover the domains of all quasi-identifiers. While DGHs can be automatically generated for numerical values (Figure 6b) and semi-structured qualitative values (Figure 6a), manual effort is needed in order to define DGHs for qualitative values which cannot easily be reduced by suppressing parts of their values.

B.5.6 Generalization including suppression. Sweeney [51] extended the idea of generalization by introducing suppression as a last generalization step and refers to this as *generalization including suppressing*. To achieve this, DGHs are extended such that a new root representing the suppressed value (e.g. "*") is introduced. Both DGHs in Figure 6 introduce "*" as the root. This allows the generalization function to eventually reach the suppressed value, if prior generalization steps fail to fulfill requirements of a privacy model.

B.5.7 Recoding strategies. When it comes to changing a representation of a value using either suppression or generalization, different approaches on recoding exist. LeFevre et al. [30] states that recoding models also can be used to categorize anonymization models. *Global recoding* describes the process of transforming all occurrences of a particular value to one consistent representation. This means that either all appearances get suppressed, or none. Furthermore, with generalization this means that all occurrences happen to be transformed to exactly one generalized value using the DGH. An extension to global recoding is *full-domain generalization* which performs generalization not only on a partition level, but generalizes all values of an attribute such that they are within the same domain (the same level in the DGH) [45]. In contrast, *local recoding* describes that occurrences are considered individually and therefore also might be recoded differently. In particular, this states that some occurrences of a particular value might get suppressed, while others do not. Moreover, multiple occurrences of a single value could be generalized to different values.

Suppose that "Berlin" appears for two records within a specific attribute. A global recoding of "Berlin" would mean that it could be generalized to "city", "location", or "entity", but all occurrences have to be generalized equally. In contrast, by applying a local recoding scheme "Berlin" could be generalized to "city" at one occurrence, and to "location" on another, depending on the level of generalization needed for a particular attribute of a particular record.

B.5.8 Partitioning strategies. k-anonymity requires to split a dataset into partitions with size of at least k. Therefore, in addition to recoding strategies, there also exist two distinct classes of partitioning strategies. *Single-dimensional* is a partitioning scheme where partitions are created and mapped for each attribute independently. This approach is straightforward, but lacks flexibility [17]. Therefore, LeFevre et al. [31] introduced a *multi-dimensional* partitioning scheme which instead of looking at each attribute individually, uses the cartesian product of attributes to form partitions.

B.6 Evaluate the usefulness of the anonymized data

Usually releasing a dataset implies that an anonymized dataset should be shared either with some closed party, or the public. To ensure that the anonymized dataset still is valuable for a given task, the utility of the anonymized dataset should be evaluated. While levels of privacy and utility in general contradict each other, both requirements are necessary for a meaningful release of data.

If there are certain requirements for utility known prior to releasing data, those requirements can already be incorporated by choosing the right de-identification mechanisms (cf. previous section). Depending on the nature of the data to be anonymized, suppression might be favorable compared to generalization [23]. One example where suppression might be favorable compared to generalization is if datasets are homogeneous. In this case,

outliers can easily be suppressed. However, if the party using the anonymized dataset requires all records and data items to be within the release, generalization remains the only option.

Moreover, partitioning and recoding strategies have a direct impact on the utility of data. Therefore, it might be reasonable to assess whether improvements of utility of anonymized data can be achieved, by varying partitioning algorithms and recoding schemes.

B.7 Document the process

Finally, after the dataset has been evaluated and its utility has been confirmed, a transparent and complete documentation of the anonymization process should be created in order to build trust in the anonymization process and show evidence that compliance and regulatory rules have been taken into account. In detail, this could mean that parameters of algorithms, classes of attributes and all additional assumptions with respect to the dataset or the anonymization process are documented.

C EXTENDED EXPERIMENT 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.

C.1 Distinct Terms

Table 6 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 6: 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

C.2 Performance

Table 7 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 7: 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

C.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.

C.3.1 Partition Splits. Figure 7 provides an overview of the distribution of splitting decisions between relational and textual attributes for experiments run on the Blog Authorship Corpus. The left column includes experiments considering all entities, while the right column presents results for experiments run only considering location (GPE) entities. Similarly, Figure 8 highlights the impact of λ on partition splits for experiments run on the Hotel Reviews Dataset.

C.3.2 Partition Count and Size. We present the results regarding partition statistics. Table 8 provides valuable insights on the number of partitions as well as the mean and standard deviation regarding partition sizes for the experiments on the Blog Authorship Corpus considering all entities. Similarly, Table 9 provides an overview of the same metrics for the Blog Authorship Corpus only considering GPE entities (locations). Table 10 and Table 11 share insights on partition statistics for the Hotel Reviews dataset respectively.

C.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 9 provides an overview of information loss for experiments run on the Blog Authorship Corpus. In particular, Figure 9a visualizes relational and Figure 9b textual information loss for experiments considering all entity types. Similarly, Figure 9c and Figure 9d provide an overview of information loss if only GPE entities are considered. Figure 10 provides the same statistics for the Hotel Reviews Dataset.

Moreover, Figure 11 and Figure 12 provide a zoomed version of Figure 9b and Figure 9d 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 13 and Figure 14.

In addition to high-level charts on information loss, Figure 15 provides an detailed analysis of information loss per entity type for the attribute *text* in the Blog Authorship Corpus. Additionally, Figure 16 and Figure 17 visualize the textual information loss per entity type for the attributes *negative review* and *positive review* respectively.

Figure 7: Splitting statistics for 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 8: 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 8: 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
		mean	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
		mean	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
		mean	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
		mean	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
		mean	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
		mean	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
		mean	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
		mean	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
		mean	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 9: Statistics on resulting partitions for Blog Authorship Corpus considering only GPE entities

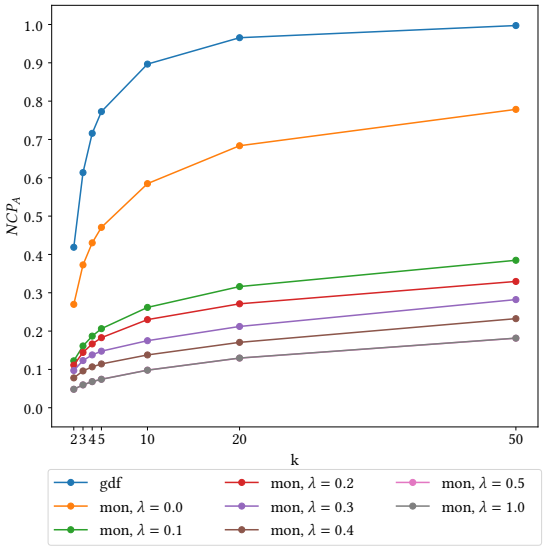
	λ	k	2	3	4	5	10	20	50
GDF	-	count	2140	737	301	129	2	1	1
		mean	9.03	26.21	64.18	149.76	9659.50	19319	19319
		std	575.34	1016.03	1629.78	13645.04	0	0	
Mondrian	0	count	1447	1164	828	703	392	242	121
		mean	13.35	16.60	23.33	27.48	49.28	79.83	159.66
		std	113.84	126.77	150.14	162.72	217.34	275.39	383.25
	0.1	count	8137	5265	3790	3019	1460	704	281
		mean	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
		mean	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
		mean	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
		mean	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
		mean	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
		mean	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
		mean	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 10: Statistics on resulting partitions for Hotel Reviews Dataset considering all entity types.

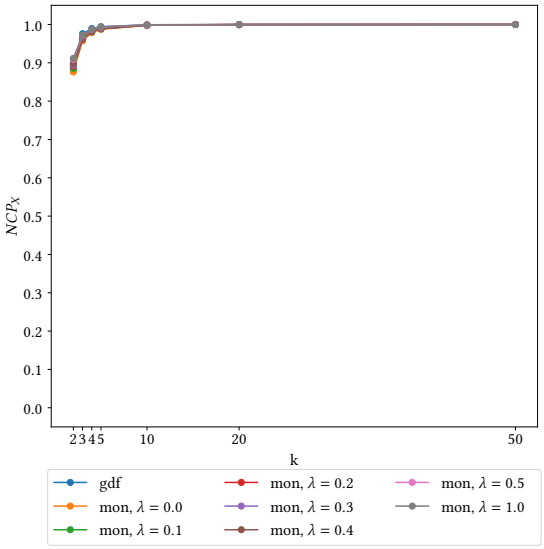
	λ	k	2	3	4	5	10	20	50
GDF	-	count	523	272	163	127	43	16	1
		mean	2.82	5.42	9.05	11.61	34.30	92.19	1475
		std	3.45	11.51	24.56	35.47	112.32	264.30	0
Mondrian	0	count	645	398	293	226	117	54	21
		mean	2.29	3.71	5.03	6.53	12.61	27.31	70.24
		std	0.47	0.77	1.11	1.45	2.60	5.59	15.67
	0.1	count	635	401	294	231	115	56	22
		mean	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
		mean	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
		mean	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
		mean	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
		mean	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
		mean	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 11: Statistics on resulting partitions for Hotel Reviews Dataset considering only GPE entities.

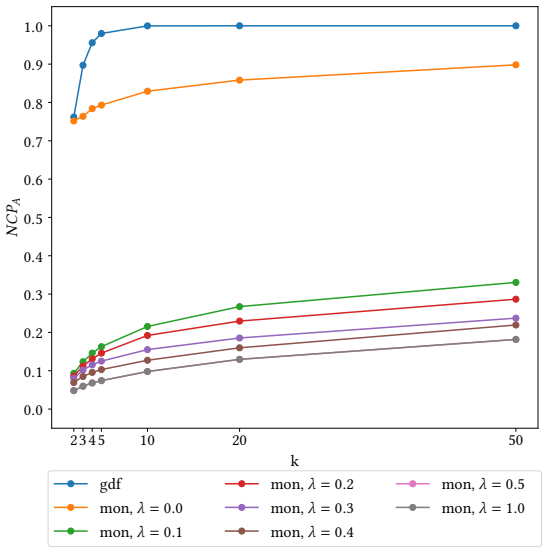
	λ	k	2	3	4	5	10	20	50
GDF	-	count	303	107	47	15	1	1	1
		mean	4.87	13.79	31.38	98.33	1475	1475	1475
		std	18.59	90.92	179.98	358.71	0	0	0
Mondrian	0	count	74	49	35	32	17	13	7
		mean	19.93	30.10	42.14	46.09	86.76	113.46	210.71
		std	61.71	74.77	86.30	89.59	112.62	121.77	135.59
	0.1	count	603	408	276	236	117	59	18
		mean	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
		mean	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
		mean	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
		mean	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
		mean	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



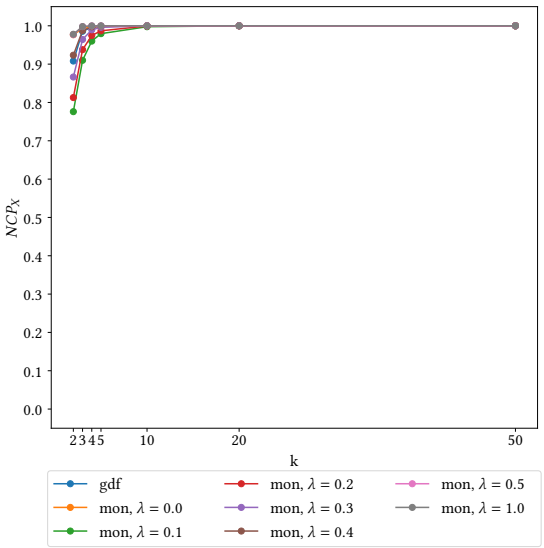
(a) Relational information loss NCP_A, all entity types



(b) Textual information loss NCP_X, all entity types

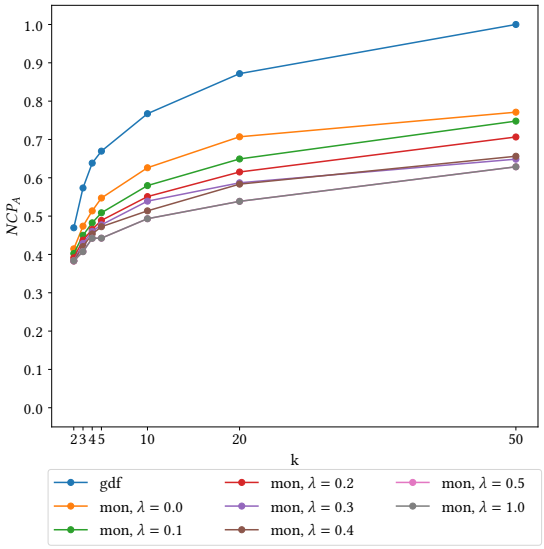


(c) Relational information loss NCP_A, only GPE entities

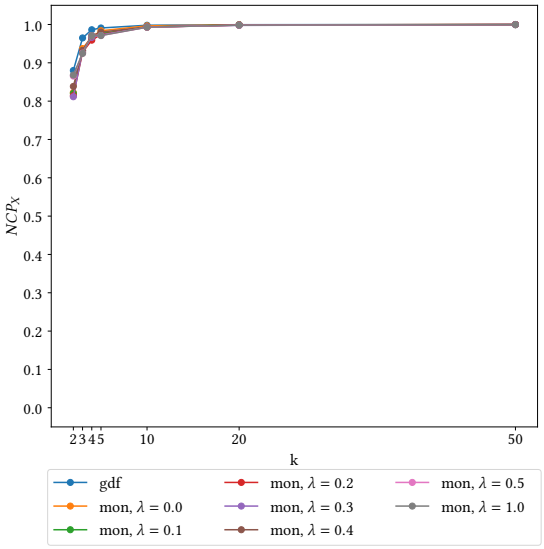


(d) Textual information loss NCP_X, only GPE entities

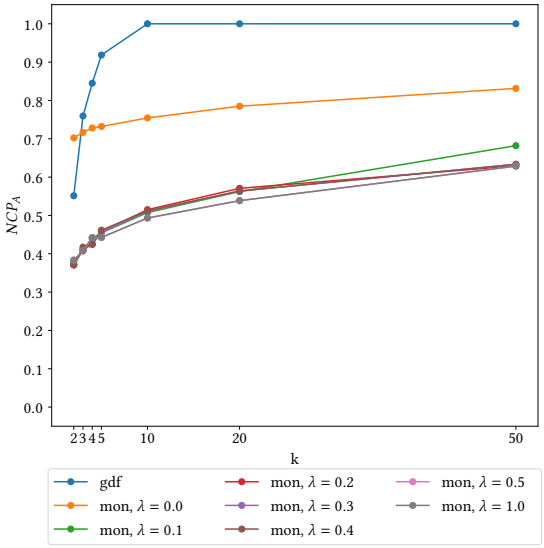
Figure 9: Relational and textual information loss for experiments run on the Blog Authorship Corpus. Results for relational (a) and textual information loss (b) for experiments considering all entities. Results for relational (c) and textual information loss (d) for experiments considering only GPE entities.



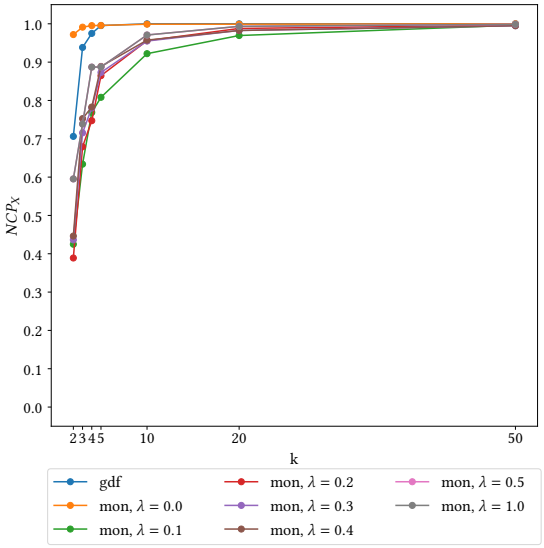
(a) Relational information loss NCP_A, all entity types



(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 10: 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.

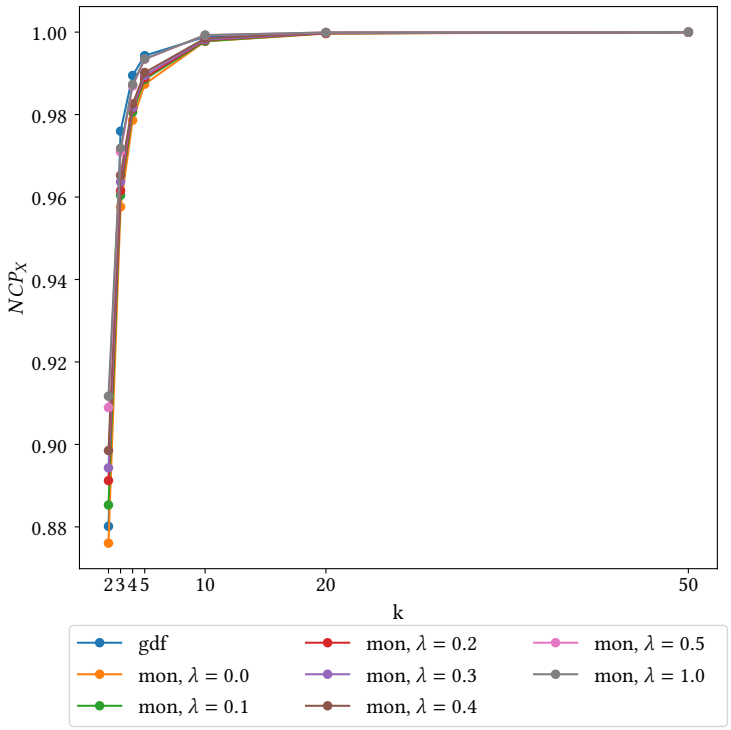


Figure 11: Zoomed textual information loss for experiments run on the Blog Authorship Corpus considering all entities.

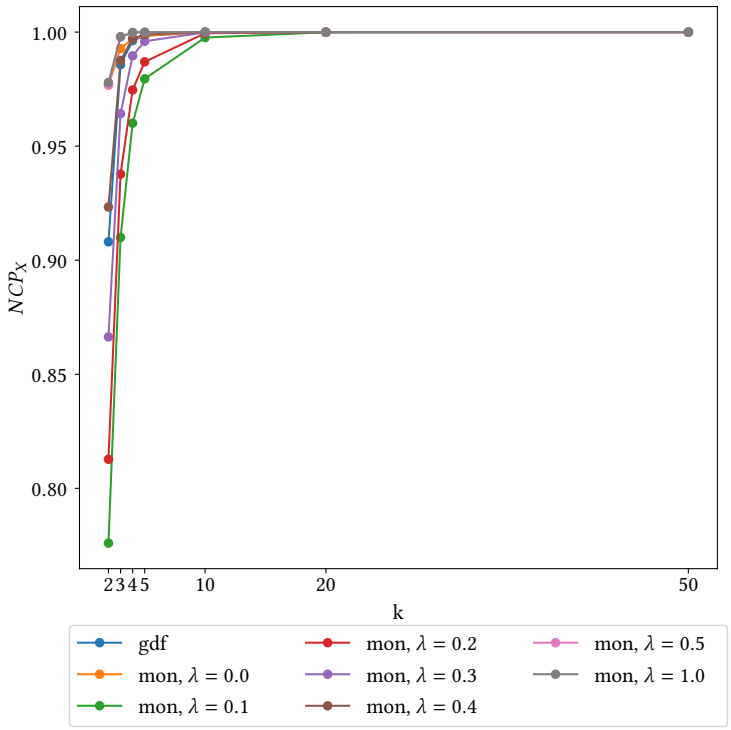


Figure 12: Zoomed textual information loss for experiments run on the Blog Authorship Corpus considering only GPE entities.

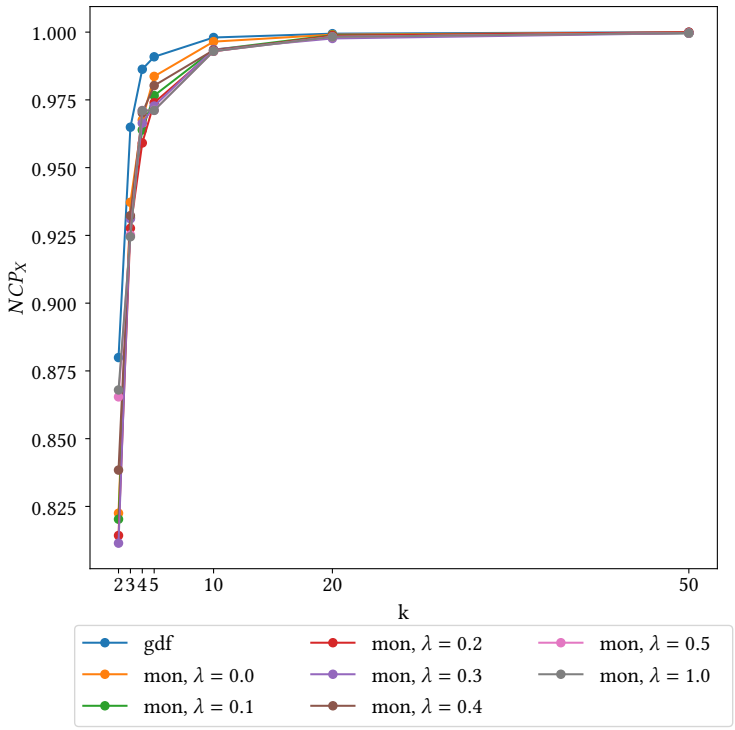


Figure 13: Zoomed textual information loss for experiments run on the Hotel Reviews Dataset considering all entities.

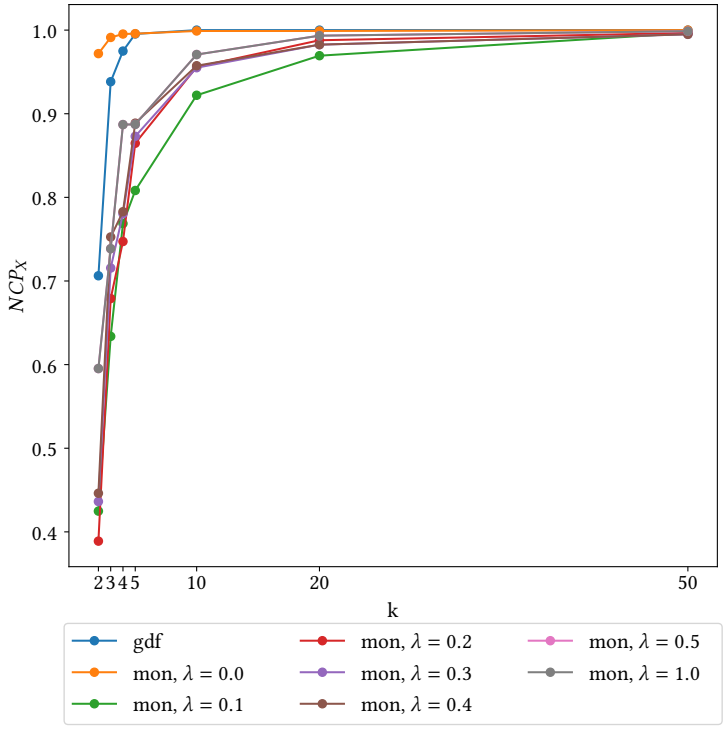


Figure 14: Zoomed textual information loss for experiments run on the Hotel Reviews Dataset considering only GPE entities.

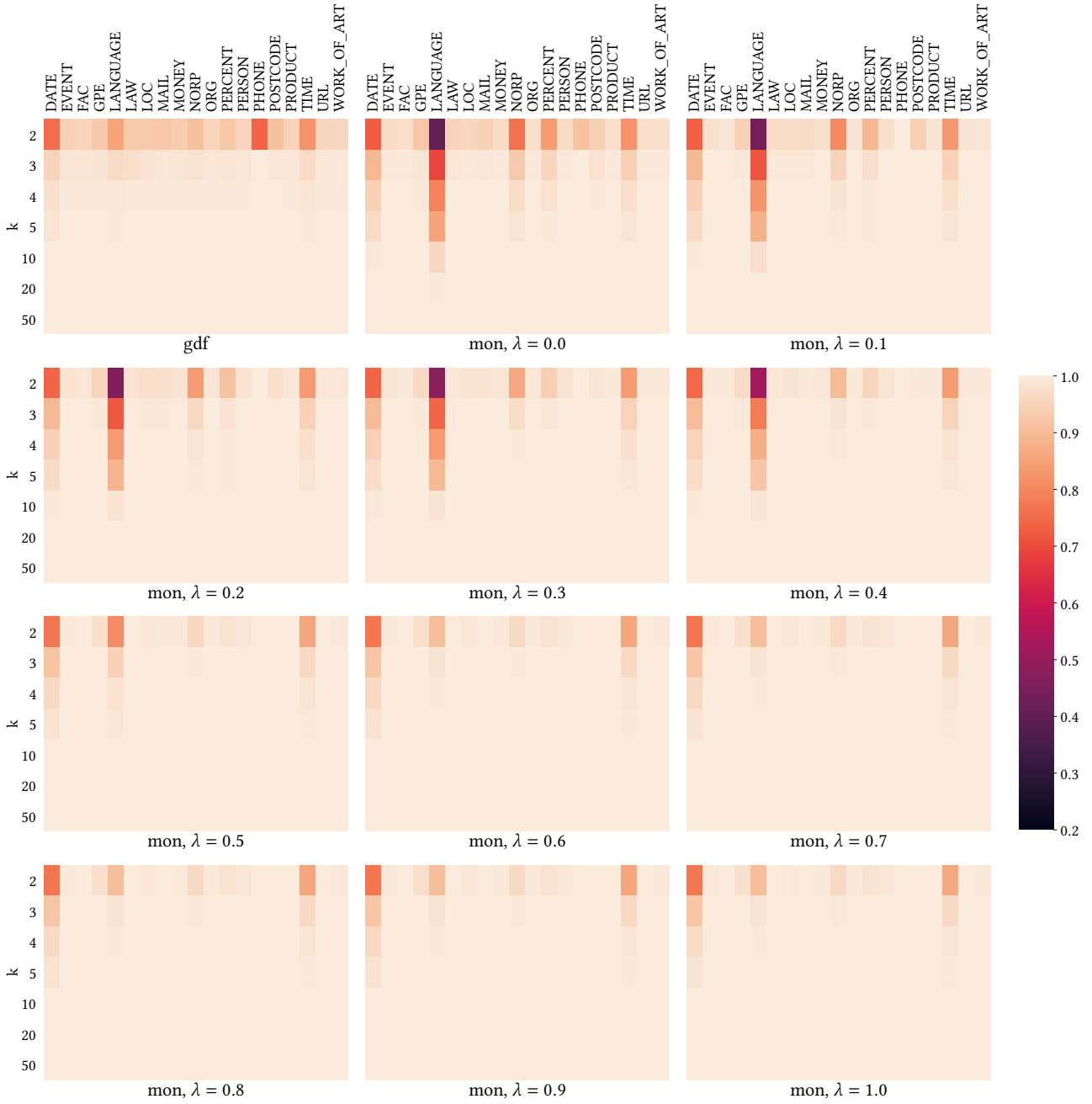


Figure 15: Textual information loss of the attribute *text* per entity types for experiments run on the Blog Authorship Corpus. Information loss is visualized for GDF and Mondrian partitioning with varying λ.

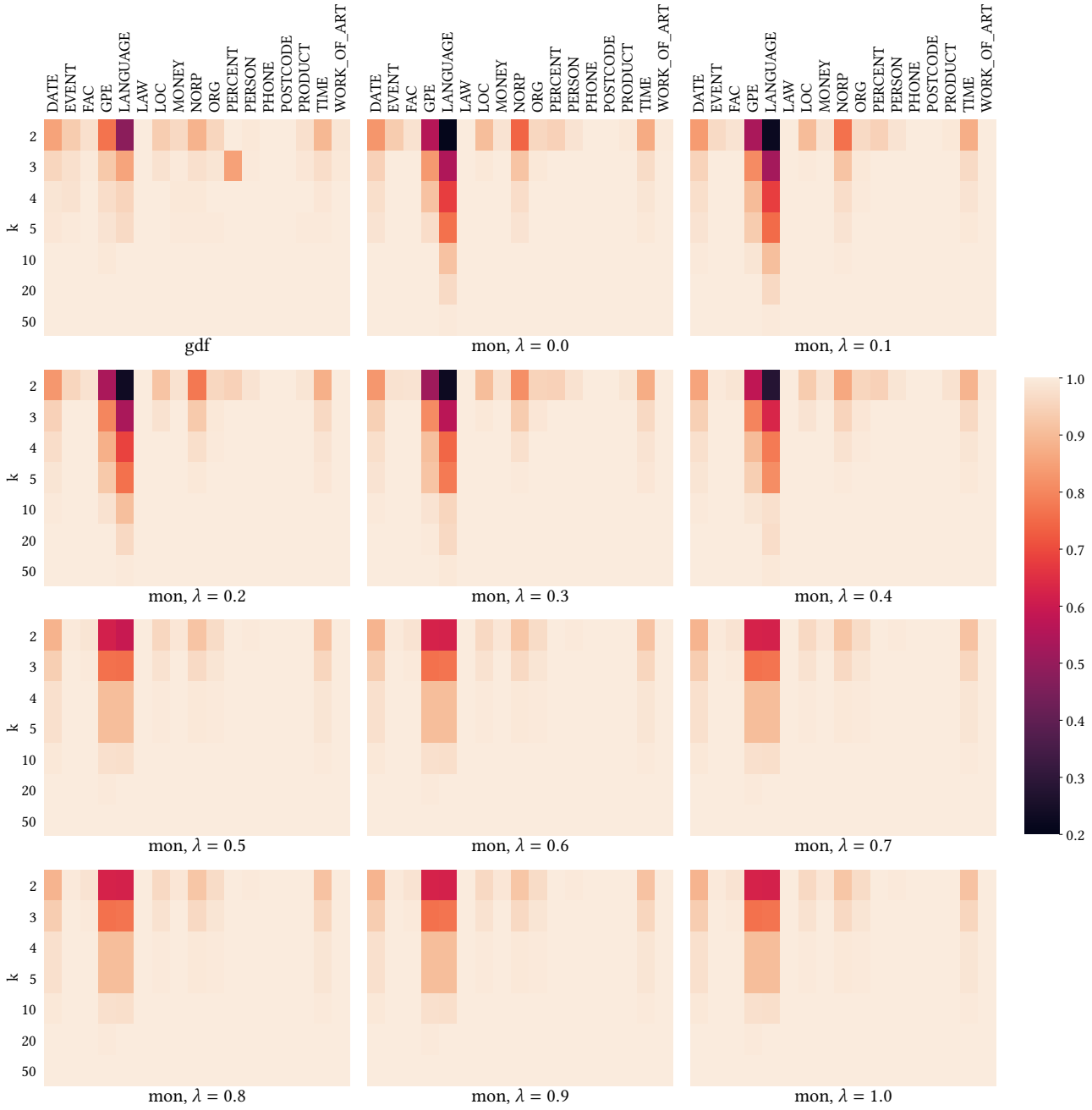


Figure 16: Textual information loss of the attribute *negative review* per entity type for experiments run on the Hotel Reviews Dataset. Information loss is visualized for GDF and Mondrian partitioning with varying λ.

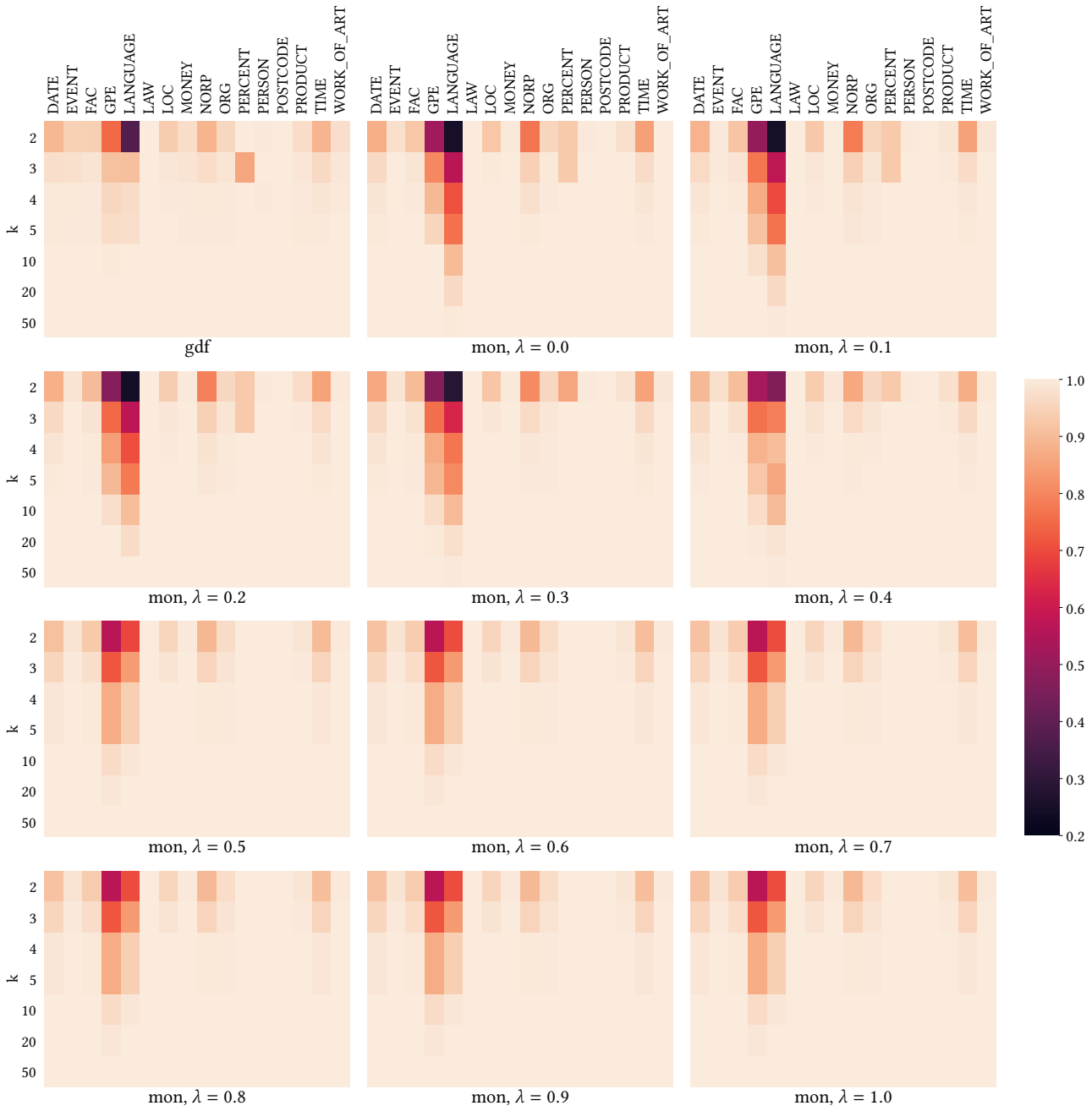


Figure 17: Textual information loss of the attribute *positive review* per entity type for experiments run on the Hotel Reviews Dataset. Information loss is visualized for GDF and Mondrian partitioning with varying λ.

D USER'S GUIDE

This section describes how to use our anonymization framework. In particular, we highlight the general features and requirements, depict how to install our tool, and provide guidance on usage and configuration. Moreover, we highlight another executable which we used to run our experiments.

D.1 Features

The key features of this anonymization tool are the following:

- Anonymization of heterogeneous datasets using the approach introduced in Section 3.2
- Comfortable configuration using a single configuration file
- Easy and ready to use software available as Docker image
- Modular software design to easily extend the anonymization framework

D.2 Dependencies

In order for the anonymization tool to work, `python3` must be available. Moreover, the following packages are required.

- `tqdm`[16]
- `pyyaml`[17]
- `numpy`[18]
- `pandas`[19]
- `bs4`[20]
- `anytree`[21]
- `matplotlib`[22]
- `spacy-nightly`[23]
- `en_core_web_trf`[24]

D.3 Installation

The anonymization tool is available in a public Docker repository on DockerHub as a pre-build image[25]. Moreover, all source code is available on GitHub[26].

D.3.1 Installation using Docker Image. The preferred way of using the tool is to use the provided Docker image and use `docker-compose` to run the container. The corresponding compose file contains the correct name of the image, and information on volumes. Please adjust the volumes according to your preferences. In order to import and export files to and from the container, you need to use the mounted volume and ensure that mounted folders provide read, write, and execute rights for others. The content of the compose file looks like follows.

```
version: "3.8"
services:
  app:
    image: fabiansinghoferuniulm/anon:latest
    container_name: anon_instance
    volumes:
      - ./data:/home/anon_user/data
    stdin_open: true
    tty: true
```

After creating a file with the name `docker-compose.yml` on your machine you need to pull the required image first by running

```
$ docker-compose pull
```

After successfully pulling the image, you can run the container using the next command. The option *–rm* tells Docker to automatically remove the Docker container after you exited it.

```
$ docker-compose run --rm app
```

After running the previous command, you end up within a bash in the container. All required packages are already installed. If you do not want to use `docker-compose`, you can use the Docker CLI directly to pull and run the image. To pull the image without `docker-compose`, run

```
$ docker pull fabiansinghoferuniulm/anon:latest
```

Finally, run the container and mount a folder to the container to retrieve the results by running

```
$ docker run -it --rm \
-v "$(pwd)"/data:/home/anon_user/data \
fabiansinghoferuniulm/anon:latest
```

D.3.2 Manual Installation. If you prefer to install the tool manually, you need to have `python3` and `pip` installed. Preferably, you should work within a virtual environment to ensure to not influence your system dependencies. After cloning or downloading the repository, you need to change into the root directory of the project.

```
$ cd 2020ss-thesis-fabian
```

Install the required dependencies. This might take a while since spaCy models are downloaded and built from scratch.

```
$ pip install -U -r requirements/base.txt
```

After installing the requirements with pip, you are all set.

D.4 Basic Usage

To use the tool, you basically just need to provide the path to the input file, the configuration file, and the output file. The command to run the tool could look like the following statement.

```
$ python anon/main.py \
    -i data/datasets/paper_example.csv \
    -c data/configurations/blog_authorship_corpus.yaml \
    -o data/results/paper_example_anonymized.csv
```

The options available for the tool are listed below.

Required options

- `-i`, `--input` Path to the input file containing the data to be anonymized.
- `-c`, `--config` Path to the configuration file containing the settings for the anonymization process.
- `-o`, `--output` Path where the anonymized file should be stored.

Optional options

- `-s`, `--use_cached_docs` Flag indicating whether cached documents if available should be used.
- `-v`, `--verbose` Flag indicating that verbose logging should be used.

[16] https://tqdm.github.io/
[17] https://pyyaml.org/wiki/PyYAMLDocumentation
[18] https://numpy.org/
[19] https://pandas.pydata.org/
[20] https://www.crummy.com/software/BeautifulSoup/
[21] https://anytree.readthedocs.io/en/latest/
[22] https://matplotlib.org/
[23] https://pypi.org/project/spacy-nightly/
[24] https://github.com/explosion/spacy-models/releases//tag/en_core_web_trf-3.0.0a0
[25] https://hub.docker.com/repository/docker/fabiansinghoferuniulm/anon
[26] https://github.com/Serpinex3/rx-anon

D.5 Configuration

To configure the anonymization tool, a configuration file should be provided using the `-c` option. Configuration files are composed of four sections, namely parameters, nlp, attributes, and entities.

D.5.1 Parameters. Within the **parameters** section, important settings for the anonymization algorithm can be configured. In particular, the anonymization parameter `k` can be set. `k` defaults to 10 if omitted. Moreover, the partitioning strategy can be configured. By default, our proposed partitioning strategy `gdf` is used (see Section 3.2.2). If you choose to use `mondrian` as the partitioning strategy of choice, you might also want to specify `relational_weight` to tune the importance of relational attributes compared to textual attributes throughout the partitioning phase. The parameter `relational_weight` corresponds to λ in Section 3.2. The following snippet describes the parameters part of a configuration file. In particular, we set k to 2, choose Mondrian partitioning, and set λ to 0.1.

```yaml
parameters:
  k: 2
  strategy: mondrian
  relational_weight: 0.1
```

D.5.2 NLP. Within the **nlp** section, settings for the Natural Language Processing module can be set. Currently, the options available are the model and the path to the cache. The default model used to analyze textual attributes is `en_core_web_trf`. However, any language model locally installed (even custom ones) can be used. The cache setting determines the default folder where processed documents should be stored for later usage. By default, the tool will use `data/cached_docs` and will create the folder if it does not exist. The following configuration snippet configures `en_core_web_trf` as the model to use and `data/cached_docs/blog_authorship_corpus` as the folder to store preprocessed results.

```yaml
nlp:
  model: en_core_web_trf
  cache: data/cached_docs/blog_authorship_corpus
```

D.5.3 Attributes. The **attributes** section contains all information on the attributes appearing in the dataset to be anonymized. Within this section, each attribute appearing in the dataset is listed together with its properties. The `anonymization_type` describes the attributes relevance for the anonymization task. Possible values are `direct_identifier`, `quasi_identifier`, and `insensitive_attribute`, and `text`. By default attributes are considered insensitive. Moreover, the data type of the attribute can be defined. Supported data types are `nominal`, `ordinal`, `numerical`, `date`, and `text`. Date attributes should also have their format configured. Moreover, each attribute can have a parameter called `entities`, which contains a list of entities this attribute is related to. During the anonymization process, the tool will look for redundant information using the assigned entity types and compare attribute values with sensitive terms of those entity types. The first direct identifying attribute is considered the key attribute which is used to compress the dataset for the anonymization process.

If `mondrian` is chosen as partitioning strategy, relational attributes can also have a bias which marks their importance in splitting partitions on a specific attribute. The default bias is 0 and viable options are between 0 and 1.

When it comes to recoding, there are also options available to configure. For numerical attributes, hierarchies can be set. Those hierarchies are then considered during the recoding phase and numerical values are replaced by the specified intervals. To use hierarchical recoding, `recoding_strategy` needs to be set to `hierarchy` and a hierarchy needs to be defined. Moreover, nominal and ordinal attributes are recoded as groups of values by default. You can change their recoding strategy to string reduction by using `string_reduction`. This allows for attributes such as postcodes to be automatically generalized by suppressing characters starting from the back (like shown in Figure 6a). Recoding strategies for date and text attributes cannot be changed. The following snippet shows a possible configuration for anonymizing the Blog Authorship Corpus.

```yaml
attributes:
  id:
    anonymization_type: direct_identifier
  gender:
    type: nominal
    anonymization_type: quasi_identifier
    bias: 0.5
  age:
    type: numerical
    anonymization_type: quasi_identifier
    recoding_strategy: hierarchy
    hierarchy:
      name: '1-100'
      children:
      - name: '1-20'
        children:
          - name: '1-10'
          - name: '11-20'
      - name: '21-40'
        children:
          - name: '21-30'
            children:
              - name: '21-25'
              - name: '26-30'
          - name: '31-40'
            children:
              - name: '31-35'
              - name: '36-40'
      - name: '41-60'
        children:
          - name: '41-50'
          - name: '51-60'
      - name: '61-100'
    entities:
      - DATE
  topic:
    type: nominal
    anonymization_type: quasi_identifier
    entities:
      - TOPIC
  sign:
    type: nominal
    anonymization_type: quasi_identifier
    entities:
      - SIGN
  date:
    type: date
    anonymization_type: quasi_identifier
```

```
    format: "%d,%B,%Y"
    entities:
      - DATE
  text:
    type: text
    anonymization_type: text
```

D.5.4 Entities. The last section in the configuration file is called **entities** and contains all entity types which should be considered for anonymization. This section is divided into native (built-in) entities and custom entities. Native entities are entities which are either available by the configured language model from spaCy, or entities which have built-in rules set up within the tool. For a complete list of available entity types refer to Table 12 and Table 13. In contrast, custom entities are entities which can be defined by users themselves. To introduce custom entities, types and corresponding sensitive terms need to be added. By default, all entities are disabled. **Note:** Currently only custom entities consisting of single words are supported.

```
entities:
  native:
    - PERSON
    - NORP
    - FAC
    - ORG
    - GPE
    - LOC
    - PRODUCT
    - EVENT
    - WORK_OF_ART
    - LAW
    - LANGUAGE
    - DATE
    - TIME
    - PERCENT
    - MONEY
    - MAIL
    - URL
    - PHONE
    - POSTCODE
  custom:
    JOB: engineer|scientist|biologist|...
    SIGN: aries|taurus|gemini|cancer|leo|virgo|...
    TOPIC: science|...
```

D.6 Experiment Runner

In addition to the anonymization tool, we provide a simple way to run experiments. You can simply re-run our experiments by running the provided shell script `run_experiment.sh` within the main directory of the project.

```
$ ./run_experiment.sh
```

Experiments can be run using `experiment_runner.py`, a Python program taking care of running experiments given some particular settings. Chosen values of k as well as metrics to measure are configured within the Python script. To run an experiment with Mondrian partitioning and $\lambda = 0.3$ on the Blog Authorship Corpus and verbose logging, you would need to start the experiment as follows.

```
$ python anon/experiment_runner.py \
    -i "blog_authorship_corpus.csv" \
    -c "blog_authorship_corpus.yaml" \
    -r "blog_authorship_corpus_mondrian_0_3" \
    -w 0.3 -v
```

All options available for running experiments are listed below.

Required options

- `-i`, `--input` Path to the input file containing the data to be anonymized.
- `-c`, `--config` Path to the configuration file containing the settings for the anonymization process. **Note:** The parameters section is ignored since we control parameters using the experiment runner.

Optional options

- `-w`, `--weight` Option used to determine the relational attribute weight for Mondrian partitioning. If no weight is provided, GDF partitioning is used as partitioning strategy.
- `-r`, `--result_dir` Name of the directory where experiment results should be stored in. Defaults to the dataset name.
- `-v`, `--verbose` Flag indicating that verbose logging should be used.

To evaluate and generate plots for the experiment results, you can run the provided evaluation script. The script will use results within the `experiment_results` folder and merge experiment results. Moreover, statistics and plots on partition splits, partition sizes, and information loss are generated.

```
$ python anon/evaluate_experiment_results.py
```

E DEVELOPER'S GUIDE

In this section, important details on the implementation are given. The implementation of the anonymization approach discussed in Section 3.2 is used to perform experiments and poses as a proof of concept implementation.

E.1 General Anonymization Pipeline

In general, our anonymization pipeline requires two inputs, and produces one output. Figure 18 shows the modular framework. Inputs are raw input data, and the configuration. The output which is produced is an anonymized version of the input data. Within the anonymization framework, there are five main components, which will be discussed below.

E.1.1 Preprocessor. Within the preprocessing component, multiple data manipulations are performed on the raw input data in order to proceed with the anonymization process. First, input data is cleansed. This involves converting attributes to their right types (according to configuration) and dropping of any rows which fail to be parsed. Moreover, texts of the configured textual attributes are cleaned by removing non-printable characters, HTML-Tags, as well as unnecessary spaces. In the next step, texts are analyzed and sensitive terms with their entity types are recognized by the NLP module and transferred to structured data. Afterwards, redundant sensitive information is detected by facilitating matching functions of the similarity module. Finally, the cleaned and modified data gets compressed to a person centric view using direct identifying attributes. The preprocessing module is stateful. To receive the current state of the dataframe, you need to call `get_df()`. Moreover, you can also get frequencies of sensitive terms and their ids to their appearances by calling `get_sensitive_terms()`. In addition, the preprocessor also

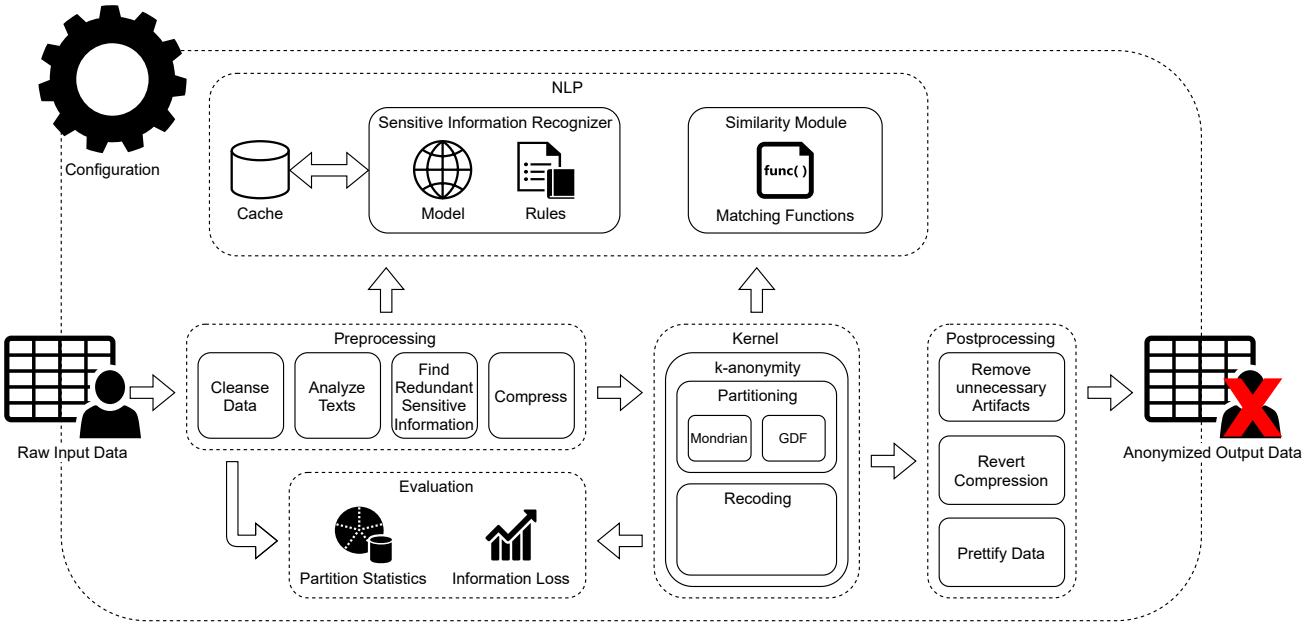


Figure 18: Anonymization framework including its sub-modules. The main components of the framework are the preprocessing module, the anonymization kernel, an evaluation component, the NLP module, and the postprocessing component.

holds the mapping of textual attributes and their helping attributes, which contain only sensitive terms. This mapping can be requested by calling `get_textual_attribute_mapping()`.

The following code snippet shows all functions used to clean up textual data.

```
def remove_html_tags(text: str) -> str:
    text = BeautifulSoup(text, 'html.parser').get_text()
    text = text.replace(u'\xa0', '␣')
    return text

def remove_non_printable_characters(text: str) -> str:
    return re.sub(r'[^\x00-\x7F]+', '', text)

def remove_unnecessary_spaces(text: str) -> str:
    return re.sub('␣+', '␣', text).strip()
```

E.1.2 NLP module. Since we are anonymizing a dataset which contains structured data as well as free text, we have the need for an NLP framework which allows us to detect and categorize entities. One popular framework is spaCy[27] which is also heavily used in industry by partners like *airbnb*, *Uber*, and *Microsoft*. spaCy offers pretrained core models for multiple languages, namely English, French, and German, among others. Those models are aimed for general purpose use and can be used to predict named entities within unstructured texts or do part-of-speech tagging. Moreover, spaCy also allows for custom models to be trained from scratch or retrained based on another model to be able to fulfill more specialized tasks.

Since we depend on detecting entities within texts, the most important feature we require from spaCy is the detection of entities and their entity types. By default, the tool supports the transformer-based model `en_core_web_trf` as well as other models installed on the system. The English transformer model is able to recognize 18 entity types (see Table 12) and has been trained on the OntoNotes5 corpus, which allows a fine-grained entity distinction compared to the Wikipedia scheme[28].

In addition to the default types, we integrated four rules to cope with entities which can easily be detected using patterns (see Table 13). For the actual patterns, please refer to the source code directly. In total, users can choose from 22 entity types (18 built-in to the model and four built-in using rules) for the anonymization process and extend the list of recognized entities with custom entity types.

Within the NLP module, there are two sub-modules. First, we implemented the sensitive information recognizer. This module takes care of recognizing sensitive terms and later recode them using the results provided by the kernel. Moreover, we also implemented a similarity module which provides matching rules which can be facilitated throughout the pipeline to compare sensitive terms in texts with relational attributes and their values.

E.1.3 Kernel. The anonymization kernel contains algorithms and methods to transform preprocessed data to an anonymized version. Within the kernel, we currently support one privacy model, namely k-anonymity, which implements concepts elaborated in Section 3.1 and Section 3.2. A k-anonymous version of the data is generated by using a two step process. First, the dataset is divided into partitions of size $|P| \geq k$ using a partitioning strategy. Partitioning strategies available include Mondrian (ref. to Algorithm 1) and GDF (ref. to Algorithm 2). Mondrian uses a weight parameter λ, describing the importance of relational attributes against textual attributes during the partitioning phase. 0.5 describes a neutral setting, while 1 only takes relational, and 0 only textual attributes into consideration. As a second step, values within partitions are recoded to match k-anonymity. Moreover, redundant and non-redundant sensitive information is recoded using the NLP module as described in Section 3.2. The output of

[27] https://spacy.io/

[28] https://spacy.io/api/annotation#named-entities

Table 12: Entities detected by spaCy's English models trained on the OntoNotes5 corpus. Taken from the documentation of spaCy[28].

Type	Description
PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.
LOC	Non-GPE locations, mountain ranges, bodies of water.
PRODUCT	Objects, vehicles, foods, etc. (Not services.)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK_OF_ART	Titles of books, songs, etc.
LAW	Named documents made into laws.
LANGUAGE	Any named language.
DATE	Absolute or relative dates or periods.
TIME	Times smaller than a day.
PERCENT	Percentage, including ”%“.
MONEY	Monetary values, including units.
QUANTITY	Measurements, as of weight or distance.
ORDINAL	“first”, “second”, etc.
CARDINAL	Numerals that do not fall under another type.

Table 13: Rule-based entities added to the NLP module.

Type	Description
MAIL	E-Mail addresses. Uses built-in pattern.
URL	Internet addresses. Uses built-in pattern.
PHONE	UK phone numbers. Uses custom pattern.
POSTCODE	UK postcodes. Uses custom pattern.

the anonymization kernel depicts an anonymized version of its input data.

E.1.4 Evaluation Module. The evaluation module contains two main components, namely a component to calculate the information loss as well as another component providing statistics on partitions. The information loss is calculated using Normalized Certainty Penalty (NCP) as described in Section 4.3. We calculate NCP_A for relational and NCP_X for textual attributes. Moreover, we provide methods to calculate NCP for each data type used in our implementation (numerical, categorical, date).

For partitions, we provide methods to calculate the mean size as well as the standard deviation given some partitions. Moreover, we also provide functionality to transform statistics on partition splits (result from kernel) to general statistics on the share of splits between relational and textual attributes.

E.1.5 Postprocessor. Within the postprocessing module, the data is modified such that it can be released. In particular, unnecessary artifacts such as temporary attributes containing sensitive terms from texts are removed. Moreover, the postprocessor performs the inverse of the compression performed by the preprocessor. By uncompressing the data, the release will have the same number of rows as the original dataset. Finally, recoded values will be prettyfied. The code snippet below shows how recoded data types are converted to their release representatives.

```
def convert_to_pretty(value, date_format="%Y-%m-%d"):
    if isinstance(value, AnyNode):
        return convert_to_pretty(value.range)
    elif isinstance(value, range):
        return "[{}-{}]".format(value.start, value.stop -
            1)
    elif isinstance(value, (pd.Timestamp, datetime.
         datetime)):
        return value.strftime(date_format)
    elif isinstance(value, pd.Period):
        return str(value)
    elif isinstance(value, (set, frozenset)):
        sorted_values = sorted(list(value), key=str.lower)
        res = ','.join(str(e) for e in sorted_values)
        return "({})".format(res)
    elif isinstance(value, list):
        sorted_values = sorted(value, str.lower)
        res = ','.join(str(e) for e in sorted_values)
        return "[{}]".format(res)
    elif pd.isnull(value):
        return ''
    else:
        return str(value)
```

E.2 Expandability

Our implementation offers various possibilities for extensions. Please also refer to Section 6 for the discussion and possible future work.

E.2.1 NLP Model. Our implementation uses spaCy for processing the textual attributes. Our modular framework allows us to easily specify another language model. This allows for other researchers to train custom models on the task of recognizing sensitive terms. The anonymization process can remain the same. However, improved models with higher precision and recall will improve privacy and utility.

E.2.2 Similarity Module. Currently, we use string-matching as a straight-forward approach to find redundant information from the relational values in the textual fields. We can think of more extensive strategies based on synonym tables, rules, or even word embeddings which can lead to more sophisticated similarity algorithms to detect redundant information.

E.2.3 Partitioning. The quality of the anonymization process highly depends on the layout of partitions. Our implementation uses GDF partitioning as well as Mondrian partitioning as described in Section 3.2. Due to the modularity of the code, someone can easily introduce a new partitioning strategy and compare it against our proposed partitioning algorithms using the metrics on partitions and information loss we calculate in the evaluation module.

E.2.4 Recoding. Currently we recode groups by using a local recoding scheme. Numerical values are recoded into intervals, nominal and ordinal attributes are recoded as sets of those attributes. We see two possible extensions in the recoding module.

First, instead of using a local recoding scheme, a global recoding scheme might be used. In order to introduce global recoding, the recoding has to be performed on the complete dataset using results from the partitioning algorithm.

Second, we see the possibility of extending recoding schemes. Numericals could be recoded by calculating some summary statistic like mean, or median. Moreover, for nominal and ordinal values, manual or semi-automatic generalization hierarchies might be applicable. The benefit of using generalization hierarchies is that released data would not need to use sets anymore. Ultimately, also sensitive terms could be recoded using generalization hierarchies.

E.2.5 Parallelization. Finally, we see potential in improving the performance of the anonymization framework. Our current implementation is not optimized for parallel work. However, parts of the anonymization pipeline such as recoding of partitions could be run in parallel. Parallelization will likely improve performance and reduce execution times.

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 Transactions on Circuits and Systems for Video Technology* 21, 3 (2011), 299–310. https://doi.org/10.1109/TCSVT.2011.2105551

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

[4] Roberto J. Bayardo and Rakesh Agrawal. 2005. Data Privacy through Optimal k-Anonymization. In *Proceedings of the 21st International Conference on Data Engineering*. IEEE, Tokyo, Japan, 217–228. https://doi.org/10.1109/ICDE.2005.42

[5] Venkatesan T. Chakaravarthy, Himanshu Gupta, Prasan Roy, and Mukesh K. Mohania. 2008. Efficient techniques for document sanitization. In *Proceedings of the 17th ACM conference on Information and knowledge mining - CIKM '08*. ACM, Napa Valley, California, USA, 843–852. https://doi.org/10.1145/1458082.1458194

[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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 2. Association for Computational Linguistics, Minneapolis, MN, USA, 197–204. https://doi.org/10.18653/v1/N19-2025 arXiv:1903.07037

[7] Council of European Union. 2016. EU General Data Protection Regulation (GDPR). https://doi.org/10.1308/rcsfdj.2018.54

[8] Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. 2017. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association* 24, 3 (2017), 596–606. https://doi.org/10.1093/jamia/ocw156 arXiv:1606.03475

[9] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming, 33rd International Colloquium*, Vol. 4052. Springer, Venice, Italy, 1–12. https://doi.org/10.1007/11787006_1

[10] Elisabeth Eder, Ulrike Krieg-Holz, and Udo Hahn. 2019. De-Identification of Emails: Pseudonymizing Privacy-Sensitive Data in a German Email Corpus. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. Incoma Ltd., Varna, Bulgaria, 259–269. https://doi.org/10.26615/978-954-452-056-4_030

[11] Khaled El Emam, Fida Kamal Dankar, Romeo Issa, Elizabeth Jonker, Daniel Amyot, Elise Cogo, Jean-Pierre Corriveau, Mark Walker, Sadrul Chowdhury, Regis Vaillancourt, Tyson Roffey, and Jim Bottomley. 2009. A Globally Optimal k-Anonymity Method for the De-Identification of Health Data. *Journal of the American Medical Informatics Association* 16, 5 (2009), 670–682. https://doi.org/10.1197/jamia.M3144

[12] Natasha Fernandes, Mark Dras, and Annabelle McIver. 2019. Generalised Differential Privacy for Text Document Processing. In *Principles of Security and Trust - 8th International Conference (Lecture Notes in Computer Science, Vol. 11426)*, Flemming Nielson and David Sands (Eds.). Springer, Cham, 123–148. https://doi.org/10.1007/978-3-030-17138-4

[13] 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. https://doi.org/10.1145/1749603.1749605

[14] Oran Gafni, Lior Wolf, and Yaniv Taigman. 2019. Live Face De-Identification in Video. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Seoul, South Korea, 9377–9386. https://doi.org/10.1109/ICCV.2019.00947

[15] James Gardner and Li Xiong. 2008. HIDE: An Integrated System for Health Information DE-identification. In *Proceedings of the Twenty-First {IEEE} International Symposium on Computer-Based Medical Systems*. IEEE, Jyvaskyla, Finland, 254–259. https://doi.org/10.1109/CBMS.2008.129

[16] Simson L. Garfinkel. 2015. *De-identification of personal information.* Technical Report. National Institute of Standards and Technology. 1–46 pages. https://doi.org/10.6028/NIST.IR.8053

[17] Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. 2007. Fast data anonymization with low information loss. In *Proceedings of the 33rd International Conference on Very Large Data Bases*. ACM, Vienna, Austria, 758–769. https://doi.org/10.1.1.138.3217

[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. https://doi.org/10.1016/j.knosys.2016.10.012

[19] Ralph Gross, Latanya Sweeney, F. de la Torre, and Simon Baker. 2006. Model-Based Face De-Identification. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, New York, NY, USA, 161–161. https://doi.org/10.1109/CVPRW.2006.125

[20] Oktie Hassanzadeh, Lipyeow Lim, Anastasios Kementsietsidis, and Min Wang. 2009. A declarative framework for semantic link discovery over relational data. In *WWW'09 - Proceedings of the 18th International World Wide Web Conference*. ACM, Madrid, Spain, 1101–1102. https://doi.org/10.1145/1526709.1526876

[21] Yeye He and Jeffrey F. Naughton. 2009. Anonymization of Set-Valued Data via Top-Down, Local Generalization. *Proceedings of the VLDB Endowment* 2, 1 (2009), 934–945. https://doi.org/10.14778/1687627.1687733

[22] Håkon Hukkelås, Rudolf Mester, and Frank Lindseth. 2019. DeepPrivacy: A Generative Adversarial Network for Face Anonymization. In *Advances in Visual Computing - 14th International Symposium on Visual Computing*, George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Daniela Ushizima, Sek Chai, Shinjiro Sueda, Xin Lin, Aidong Lu, Daniel Thalmann, Chaoli Wang, and Panpan Xu (Eds.), Vol. 11844. Springer, Lake Tahoe, NV, USA, 565–578. https://doi.org/10.1007/978-3-030-33720-9_44

[23] Information and Privacy Commissioner of Ontario. 2016. *De-identification Guidelines for Structured Data.* Technical Report June. Information and Privacy Commissioner of Ontario. 1–28 pages. 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 *Proceedings of the ACM Conference on Health, Inference, and Learning*. ACM, Toronto, Ontario, Canada, 214–221. https://doi.org/10.1145/3368555.3384455

[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 International Conference and Workshops on Automatic Face and Gesture Recognition*. IEEE, Ljubljana, Slovenia, 1–7. https://doi.org/10.1109/FG.2015.7285021

[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, Washington, DC, USA, 767–776. http://www.ncbi.nlm.nih.gov/pubmed/25954383http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4419982

[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 http://arxiv.org/abs/2001.08904

[28] Moshe Koppel, Jonathan Schler, Shlomo Argamon, and Eran Messeri. 2006. Authorship attribution with thousands of candidate authors. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Vol. 2006. ACM, Seattle, Washington, USA, 659–660. https://doi.org/10.1145/1148170.1148304

[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. https://doi.org/10.1016/j.eswa.2007.08.042

[30] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. 2005. Incognito: efficient full-domain K-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, Baltimore, Maryland, USA, 49–60. https://doi.org/10.1145/1066157.1066164

[31] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. 2006. Mondrian Multidimensional K-Anonymity. In *Proceedings of the 22nd International Conference on Data Engineering*. IEEE, Atlanta, GA, USA, 25–25. https://doi.org/10.1109/ICDE.2006.101

[32] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In *Proceedings of the 23rd International Conference on Data Engineering*. IEEE, Istanbul, Turkey, 106–115. https://doi.org/10.1109/ICDE.2007.367856

[33] Zengjian Liu, Buzhou Tang, Xiaolong Wang, and Qingcai Chen. 2017. De-identification of clinical notes via recurrent neural network and conditional random field. *Journal of Biomedical Informatics* 75 (2017), S34–S42. https://doi.org/10.1016/j.jbi.2017.05.023

[34] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. 2006. L-diversity: privacy beyond k-anonymity. In *Proceedings of the 22nd International Conference on Data Engineering*. IEEE, Atlanta, GA, USA, 24–24. https://doi.org/10.1109/ICDE.2006.1

[35] 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, Gaithersburg,

MD. 1–59 pages. https://doi.org/10.6028/NIST.SP.800-122

[36] 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 - 12th International Symposium, PETS 2012, Vigo, Spain, July 11-13, 2012. Proceedings*, Vol. 7384. Springer, Vigo, Spain, 299–318. https://doi.org/10.1007/978-3-642-31680-7_16

[37] Adam Meyerson and Ryan Williams. 2004. On the complexity of optimal k-anonymity. In *Proceedings of the Twenty-third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. ACM, Paris, France, 223–228. https://doi.org/10.1145/1055558.1055591

[38] Ishna Neamatullah, Margaret M. Douglass, Li-wei H Lehman, Andrew Reisner, Mauricio Villarroel, William J. Long, Peter Szolovits, George B. Moody, Roger G. Mark, and Gari D. Clifford. 2008. Automated de-identification of free-text medical records. *BMC Medical Informatics and Decision Making* 8 (2008), 32. https://doi.org/10.1186/1472-6947-8-32

[39] Mehmet Ercan Nergiz, Christopher Clifton, and Ahmet Erhan Nergiz. 2007. MultiRelational k-Anonymity. In *Proceedings of the 23rd International Conference on Data Engineering*, Vol. 21. IEEE, Istanbul, Turkey, 1417–1421. https://doi.org/10.1109/ICDE.2007.369025

[40] Paul Ohm. 2010. Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA Law Review* 57 (2010), 1701–1777.

[41] 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 - European Conference*, Vol. 8190. Springer, Prague, Czech Republic, 353–369. https://doi.org/10.1007/978-3-642-40994-3_23

[42] 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. https://doi.org/10.1002/spe.2812

[43] 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, Washington, DC, USA, 984–993. http://www.ncbi.nlm.nih.gov/pubmed/25954407http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4419984

[44] 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, Los Angeles, CA, USA, 729–733. http://www.ncbi.nlm.nih.gov/pubmed/11079980http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2244050

[45] Pierangela Samarati. 2001. Protecting respondents identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering* 13, 6 (2001), 1010–1027. https://doi.org/10.1109/69.971193

[46] David Sánchez, Montserrat Batet, and Alexandre Viejo. 2013. Automatic general-purpose sanitization of textual documents. *IEEE Transactions on Information Forensics and Security* 8, 6 (2013), 853–862. https://doi.org/10.1109/TIFS.2013.2239641

[47] Yücel Saygin, Dilek Hakkani-Tür, and Gökhan Tür. 2009. Sanitization and Anonymization of Document Repositories. In *Database Technologies: Concepts, Methodologies, Tools, and Applications*, John Erickson (Ed.). IGI Global, 2129–2139.

[48] Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James Pennebaker. 2006. Effects of age and gender on blogging. In *Computational Approaches to Analyzing Weblogs, Papers from the 2006 AAAI Spring Symposium*. AAAI, Stanford, California, USA, 199–205.

[49] Latanya Sweeney. 1996. Replacing personally-identifying information in medical records, the Scrub system. In *Proceedings : a conference of the American Medical Informatics Association. AMIA Fall Symposium*. AMIA, Washington, DC, USA, 333–337. http://www.ncbi.nlm.nih.gov/pubmed/8947683http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC2233179

[50] Latanya Sweeney. 2000. Simple Demographics Often Identify People Uniquely. (2000).

[51] Latanya Sweeney. 2002. Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 5 (2002), 571–588. https://doi.org/10.1142/S021848850200165X

[52] Latanya Sweeney. 2002. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 5 (2002), 557–570. https://doi.org/10.1142/S0218488502001648

[53] 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 - 14th International Conference*, Vol. 9850. Springer, Rio de Janeiro, Brazil, 401–417. https://doi.org/10.1007/978-3-319-45348-4_23

[54] Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. 2008. Privacy-preserving anonymization of set-valued data. *Proceedings of the VLDB Endowment* 1, 1 (2008), 115–125. https://doi.org/10.14778/1453856.1453874

[55] 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. 1–32 pages. https://www.hhs.gov/sites/default/files/ocr/privacy/hipaa/understanding/coveredentities/De-identification/hhs_deid_guidance.pdf

[56] 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 *Proceedings of the ACM WSDM 2020 Health Search and Data Mining Workshop*, Vol. 2551. CEUR-WS.org, Houston, TX, USA, 3–11. arXiv:2001.05714

[57] 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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*. Curran Associates, Inc., Long Beach, CA, USA, 5998–6008. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[58] Jian Xu, Wei Wang, Jian Pei, Xiaoyuan Wang, Baile Shi, and Ada Wai-Chee Fu. 2006. Utility-based anonymization using local recoding. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, Philadelphia, PA, USA, 785–790. https://doi.org/10.1145/1150402.1150504

[59] Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. 2020. A Differentially Private Text Perturbation Method Using Regularized Mahalanobis Metric. (2020), 7–17 pages. https://doi.org/10.18653/v1/2020.privatenlp-1.2 arXiv:2010.11947

[60] Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: Adapting Transformer Encoder for Named Entity Recognition. (2019). arXiv:1911.04474 http://arxiv.org/abs/1911.04474

[61] Ying Zhao and Charles C. Zhou. 2020. Link Analysis to Discover Insights from Structured and Unstructured Data on COVID-19. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. ACM, Virtual Event, USA, 1–8. https://doi.org/10.1145/3388440.3415990