The SURROGATOR Framework for Context-Aware Surrogation of Privacy Sensitive Information – Technical Description

Christina Lohr[1*], Marvin Seiferling[2], Philipp Wiesenbach[2],
Jakob Faller[3], Christoph Dieterich[2,4*]

[1*]Institute for Medical Informatics, Statistics and Epidemiology (IMISE), University Leipzig, Härtelstraße 16-18, Leipzig, 04107, Saxony, Germany.
[2]Klaus Tschira Institute for Integrative Computational Cardiology, University Hospital Heidelberg, Im Neuenheimer Feld 669, Heidelberg, 69120, Baden-Württemberg, Germany.
[3]Medical Center for Information and Communication Technology, Universitätsklinikum Erlangen, Krankenhausstraße 22, Erlangen, 91054, Bavaria, Germany.
[4]Partner Site Heidelberg/Mannheim, German Centre for Cardiovascular Research (DZHK), Im Neuenheimer Feld 669, Heidelberg, 69120, Baden-Württemberg, Germany.

*Corresponding author(s). E-mail(s): christina.lohr@imise.uni-leipzig.de; christoph.dieterich@med.uni-heidelberg.de;
Contributing authors: marvinseiferling@web.de; Philipp.Wiesenbach@med.uni-heidelberg.de; jakob.faller@uk-erlangen.de;

This document is a technical description of the logic of the SURROGATOR framework.

Contents

1 Workflow

The generation of high-fidelity, synthetic surrogates is not an isolated process but rather the final step in a meticulously designed, multi-stage workflow. This chapter describes the foundation upon which the Surrogator is built: an established process that extends from manual annotation through rigorous quality assurance to the final surrogate replacement.

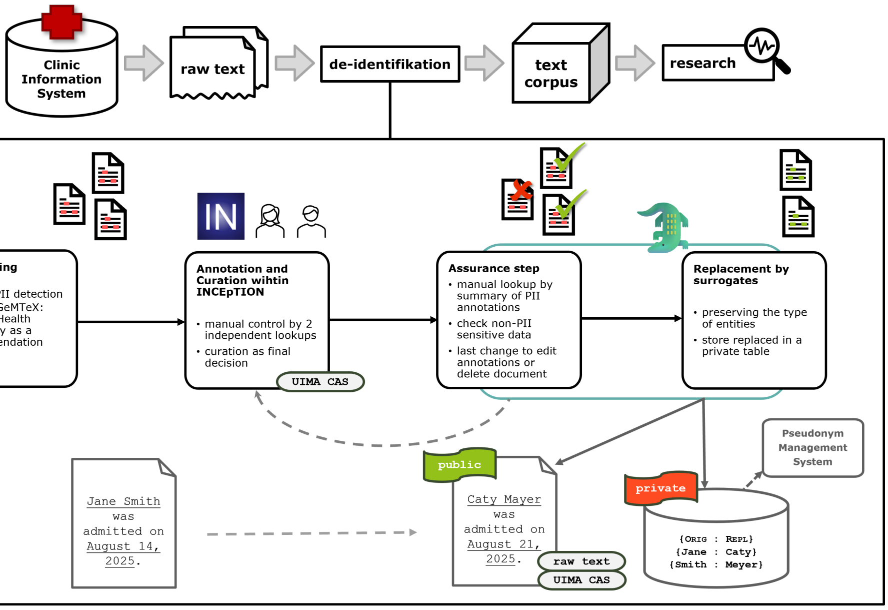

Clinic Information System
raw text
de-identifikation
text corpus
research
IN
Pre-Tagging
• autom. PII detection
• used in GeMTeX: Averbis Health Disvovery as a recommendation system-
Annotation and Curation wihtin INCEpTION
• manual control by 2 independent lookups
• curation as final decision
UIMA CAS
Assurance step
• manual lookup by summary of PII annotations
• check non-PII sensitive data
• last change to edit annotations or delete document
Replacement by surrogates
• preserving the type of entities
• store replaced in a private table
Pseudonym Management System
Jane Smith was admitted on August 14, 2025.
public
Caty Mayer was admitted on August 21, 2025.
raw text
UIMA CAS
private
{Orig : Repl}
{Jane : Caty}
{Smith : Meyer}

Fig. 1 The multi-stage de-identification and surrogation workflow in the GeMTeX project. The process begins with curated annotation in INCEpTION, proceeds through an automated quality control step, and culminates in pseudonymization, which generates the de-identified texts and a separate key-value mapping file.

1.1 Data Protection Concept and GeMTeX' PII

GeMTeX is the largest effort to collect a text corpus from multiple clinical sites in Germany. Here, we explain how the surrogator tool fits into a concept using the GeMTeX data protection concept as an example. The following process is part of the GeMTeX data protection concept and part of the study protocol, approved by six Ethics Committees. Table 1.1 lists GeMTeX's PII. This list is the foundation of our processing and part of the data protection concept.

The basis for any surrogation is the precise and complete annotation of personally identifiable information (PII). In the GeMTeX project, this is ensured by an established annotation process that has been described in detail elsewhere Lohr *et al* [1]

PII	Description
NAME_PATIENT NAME_DOCTOR NAME_RELATIVE NAME_EXTERN	all types of personal names, first and last names annotated together in 1 span, social and functional roles characterized; user names and titles are treated separately;
NAME_USERNAME	Username, secretary's initials, artificial names for system logins
NAME_TITLE	academic title designations
DATE DATE_BIRTH DATA_DEATH	covers all possible types of uniquely identifying dates patient's data of birth and the date of death (if specified) should be considered separately, only absolute dates, no relational dates
AGE	age of patient in years
LOCATION_STREET LOCATION_ZIP LOCATION_CITY LOCATION_COUNTRY	all types of address information
LOCATION_HOSPITAL	Facility with clinical and medical relevance in treatment process chain
LOCATION_ORGANIZATION	Naming an organization without clinical relevance (e.g., insurance companies)
LOCATION_OTHER	Other addresses, clearly identifiable local entities, locations without clinical functions
ID	identifiers in form of a sequence of characters, including numbers, digits, or alphanumeric combinations of numbers and letters, including patient IDs and case IDs, IDs from medical subsystems, insurance numbers, account numbers
CONTACT_PHONE CONTACT_FAX CONTACT_EMAIL CONTACT_URL	contact information
PROFESSION	information about a patient's profession, to clarify in a subsequent step whether the job title provides information that allows conclusions to be drawn about a patient.
OTHER	all other information that should be de-identified

Table 1 GeMTeX's PII with their description.

(Annotation guideline: [2]). The process (see Figure 1, left) can be summarized as follows:

1. **Pre-Tagging:** Clinical documents are first pre-annotated by an automated system marking potential PII.
2. **Manual Annotation and Curation:** A 2+1 review process is applied within the **INCEpTION** annotation platform. Two annotators independently review and correct the pre-annotations. Subsequently, a curator harmonizes the two versions, resolves discrepancies, and ensures the final, quality-assured annotation.
3. **Export:** The completed annotation projects are exported from INCEpTION in the `UIMA CAS` format. This format contains both the original text and the associated PII annotations with their types (e.g., `NAME_PATIENT`, `DATE_BIRTH`) and exact positions in the text.

Please note that besides the fact of strong project correlation to GeMTeX, the presented surrogation process is unique in the context of german clinical language. Therefore we claim that the surrogation procedure can be adapted to any other project task by adjusting the surrogation properties.

1.2 Quality Assurance: An Upstream Review Step

Before an irreversible replacement of PII with surrogates occurs, a critical intermediate step for quality assurance (QA) should be done. Certain PII categories are inherently high-risk or too semantically complex to be adequately prepared for automated replacement by annotation alone. Therefore, the Surrogator first conducts a QA process (see Figure 1, center).

This step summarizes all curated annotations and automatically generates aggregated reports that focus on four particularly sensitive categories:

- `AGE`: The age distribution can provide clues about identity. For example, US regulations under Hipaa [3] exclude patients older than 89. We do not exclude documents with an age older than 89 or an other age annotation.
- `PROFESSION`: Rare or very specific job titles (e.g., "university professor", "minister") can make an individual identifiable. We also recommend checking documents and their job titles.
- `OTHER`: This category serves as a catch-all for highly identifying information that does not fit into any other category. Documents with this annotation are excluded from further processing by default.

The output of this QA step is a tabular summary that allows curators to make a final release decision for each document. An *"Inclusion Toggle"* list (`part_of_corpus`) is used to specify whether a document proceeds to surrogation (`1`) or is excluded (`0`). Only after this explicit manual review and approval are the documents passed on to the next step. This ensures that only sufficiently quality-assured annotations serve as the basis for surrogate generation.

1.3 Surrogate Generation with Surrogator

After quality control, the Surrogator performs the actual pseudonymization. This process involves two core tasks: replacing the PII in the text and creating a a mapping file, containing PII and surrogats.

2 Technical Details

Our tool is based on previously published work ClinicalSurrogateGeneration [4, 5] and is designed to process surrogates in a German-language clinical text document. Surrogator is implemented in Python and takes pre-processed text data with annotated PII as input.

The **input** format is Uima Cas format [6], which is used by the INCEpTION annotation platform [7]. Surrogator offers a command line interface to batch process individual projects, which were exported from INCEpTION with the curated documents or individual annotation files. Alternatively, Surrogator offers a direct connection to the INCEpTION plattform via a Web interface. The **output** consists of new text documents with replaced text spans.

In the following sections, we outline the implementation details of fictive surrogate creation. Our focus is limited to processing raw text and its associated PII annotations. It is important to note that hospital file names, which often include case numbers and patient identifiers, are not processed by this system.

We introduce in our supported modes and details methods used for the fictitious surrogate process, beginning with the straightforward rule-based replacements, progressing to syntactic-aware name generation, and culminating in the advanced semantic and hierarchical methods developed for location-based entities.

The Surrogator does not support pseudonym management, as offered by gICS [8] (part of the Greifswald tools Mosaic) [9]. However, a Json file is provided to link the systems in other ways.

2.1 Supported Modes for Replacement

In the following, we present the implemented modes for replacing annotated de-identified documents.

(1): Masking by X

- Replacing identified PII with generic placeholder XXX.
- Example: `"Patient XXXX was admitted on XXX."`

(2): Masking by Entity Types

- Replacing identified PII with generic placeholder of annotation type.
- Example: `Patient NAME_PATIENT was admitted on DATE.`

(3): Advanced Masking by Entity Types and a key (”GeMTeX mode”)

- Replace identified PII with a combination of the annotation type and a unique, random key[1] (e.g., `[**NAME_PATIENT FR7CR8**]`, `[**DATE DT9AS1**]`). Each instance of the same original PII receives the same key in the document.
- Used as an intermediate step in GeMTeX, it preserves the type information and the association of similar entities. It enables theoretical reversibility.

[1] This mode was created as an interim solution because the tool was under development in the GeMTeX project while documents already needed to be de-identified.

- Example: `Patient [** NAME_PATIENT FR7CR8 **] was admitted on [** DATE DT9AS1 **].`

(4): Masking by surrogates

- Replacement of identified PII with fictitious, yet plausible, type-consistent data. The goal is to generate expressions that look like real information.
- High data utility & Naturalness: aims to maintain the most natural text structure and readability possible to minimize loss of utility compared to original data and maximize performance of downstream NLP tasks.
- ”Hiding in Plain Sight” principle: Argumentation that this approach potentially even increases data protection (difficulty detecting replacements, masking of non-annotated PII). Complexity: Requires more complex generation logic.
- Example: `Patient Tina Smith was admitted on February 1, 2024.`

2.2 Details on the generation of Synthetic Fictitious Surrogates

The primary objective of the Surrogator is to replace annotated personally identifiable information (PII) with fictitious, but plausible, and type-consistent data. The generation logic depends on entity type and employs methods that range from simple rule-based substitutions to complex machine learning models. Our tool set encompasses spaCy's German language models [10–12], a Sentence Transformers model (*paraphrase-multilingual-MiniLM-L12-v2*) [13], and reference data from OpenStreetMap [14].

2.2.1 Rule-Based and Pattern-Preserving Surrogates

For several entity types, a direct, rule-based substitution is sufficient to ensure anonymization while preserving the necessary structural format.

- **`ID` and `CONTACT` (Phone, Fax, Email, URL):** These entities are processed using pattern-preservation logic. The surrogate string replicates the character-level structure of the original: uppercase letters are replaced with random uppercase letters, digits with random digits, and all other characters (such as hyphens) are retained. This approach ensures that the surrogate remains format-compliant. For special cases like validated German IBANs and BICs, the `schwifty` library is used to generate entirely new, valid random identifiers.

  Examples:

  - **ID (Pattern-Preserving):**
    `A-202344102` → `V-012195586`
    Letters and digits are replaced; special characters are retained.
  - **IBAN (Special Case):**
    `DE89 3704 0044 0532 0130 00` → `DE21 5001 0517 8361 5471 23`
    A completely new, but mathematically valid, German IBAN is generated.

- **DATE:** The surrogation logic for dates is context-dependent. GeMTeX mode: To mitigate re-identification risks, birth (DATE_BIRTH) and death dates (DATE_DEATH) are generalized by rounding them to the first day of their respective quarter (e.g., "20.05.1950" becomes "01.04.1950"). All other DATE annotations, which often provide critical clinical context, are preserved. The Mode Fictive Generation includes an additional capability for applying a consistent date-shift across a document.

  Examples:

  - **DATE (Consistent Shift):**
    03.07.2023 → 07.08.2023
    21. Juli 2022 → 25. August 2022
    05/2025 → 06/2025
    A consistent shift (+35 days) is applied while preserving the original format.

2.2.2 Syntactic and Gender-Aware Name Surrogation (NAME_*)

Replacing personal names requires a more sophisticated approach to maintain linguistic realism. The goal is to substitute names while preserving their gender and syntactic structure (i.e., first vs. last names). This is achieved through a three-step process:

1. **Gender Detection:** The gender of a name is inferred using a combination of the gender_guesser library and a set of German-specific heuristics. These rules check for preceding salutations (e.g., *Frau*, *Herr*) and gendered suffixes (e.g., *-in*) to improve accuracy.
2. **Name Classification** is analyzed by a rule-based approach of the structure of the full name string to classify its components as either a first name (FN) or a last name (LN). It handles various formats, including those with commas (e.g., "Meier, Hans"), salutations, and multi-word family names.
3. **Surrogate Sampling** is based on the classification, the system randomly samples replacement names from pre-compiled lists of male, female, and family names. The name resources are adapted from publicly available lists provided in the ClinicalSurrogateGeneration repository [15].

Examples:

- **NAME (Syntactic and Gender-Aware):**
  Frau Beate Albers → Frau Julia Schmidt
  Patientin Chris Wolf → Patientin Tina Koch
  Meier, Hans → Müller, Max

The NAME_USERNAME entity is handled with the pattern-preserving logic described earlier, while NAME_TITLE is replaced by a randomly sampled title from a curated list.

2.3 Semantic Surrogation for Standalone Locations

For location entities that are not part of a structured address a simple rule-based approach is insufficient. These names carry significant semantic meaning that must be preserved. To address this, we developed a novel surrogate generation method based on instance-based learning and semantic similarity.

2.3.1 Data Foundation: Curated Location Datasets from OpenStreetMap

A prerequisite for generating realistic location surrogates is a large, diverse, and high-quality candidate pool. To this end, we created custom datasets of German locations [14] to ensure plausibility [16].by querying the OpenStreetMap (OSM) database via its Overpass API [17]. The complete data collection and cleaning methodology is documented and reproducible in a public GitHub repository [16].

The datasets were compiled as follows:

- **`LOCATION_HOSPITAL`:** A list of 30,000 healthcare facilities was generated by querying OSM for all entities tagged with `healthcare=*`. The raw data was extensively cleaned by filtering out entries classified as person names using NER and regex, while using a keyword list (e.g., *Klinik*, *Praxis*) to prevent the incorrect removal of legitimate facilities.
- **`LOCATION_ORGANIZATION`:** A list of organizations was created by querying OSM nodes with tags such as `office=*`, `craft=*`, `club=*`, and `industrial=*`.
- **`LOCATION_OTHER`:** To ensure broad coverage for miscellaneous locations, a diverse dataset was built by querying a wide array of OSM's primary map feature tags, including `amenity`, `shop`, `tourism`, and `leisure`.

The 5-Stage Algorithm for `LOCATION_HOSPITAL` Surrogates

We used a multi-step approach to generate surrogates for hospitals, as these names often contain a complex mixture of functional descriptions (e.g., Kardiologie), specific proper nouns (e.g., a founder's name like in Robert-Bosch-Krankenhaus), and geographic identifiers (e.g., Universitätsklinikum Leipzig), necessitating a multi-faceted approach to balance semantic relevance and privacy.

1. **Stage: Query Normalization:** The original hospital name is cleaned (e.g., punctuation removal, lowercasing) and common German abbreviations are expanded (e.g., *"UK"* → *"Universitätsklinikum"*) to create a standardized query.
2. **Stage: Adaptive Similarity Search:** The normalized query is encoded into a vector embedding using a sentence-transformer model. An adaptive k-Nearest Neighbors (k-NN) search is performed against the pre-embedded OSM hospital dataset to retrieve an initial pool of semantically similar candidates. The search window (k) is widened iteratively if not enough safe candidates are found in the next step.
3. **Stage: Sensitive Term Filtering:** To prevent data leakage, a spaCy NER and POS pipeline identifies any proper nouns (Person, Org, etc.) in the original

query. Any candidate from the pool that contains sensitive terms found in the original is discarded.

4. **Stage: Domain-Specific Similarity Ranking:** The remaining safe candidates are re-ranked based on their functional similarity. This is achieved by calculating the normalized Levenshtein distance between the candidate names and any healthcare-specific keywords (e.g., *kardio*, *reha*) present in the original query.
5. **Stage: Probabilistic Sampling:** Instead of deterministically choosing the top-ranked candidate, the final surrogate is selected via weighted random sampling. A probability distribution is created from the similarity scores (using temperature scaling), ensuring that the process is non-reversible while still favoring the best matches.

Examples:

- **LOCATION_HOSPITAL:**
  Krankenhaus der Samariter Holzhausen → Johanniter-Krankenhaus Geesthacht
  Städt. Klinikum Neustadt → Klinikum Kassel
  Universitätsklinikum Klagenfurt → Universitätsklinikum des Saarlandes

Surrogation for `LOCATION_ORGANIZATION` and `LOCATION_OTHER`

These entities leverage a simplified version of the same semantic framework. The process relies primarily on the sentence-embedding similarity search (see above *2. Stage*) and sensitive-term filtering (see above *3. Stage*) to find a plausible replacement, but forgoes the intensive domain-specific ranking, as the goal is general realism rather than strict functional equivalence.
Examples:

- **LOCATION_ORGANIZATION:**
  Schlachhof Schlacht-Gut → Dachdeckerei Schöler&Groth
  LIFE Management Cluster → TÜV SÜD Life Service

- **LOCATION_OTHER:**
  Rosensäle → Die blaue Rose
  Botanischer Garten → Garten Bräunlein

Hierarchical Surrogation for Geographic Addresses

Structured addresses, composed of entities like `LOCATION_STREET`, `LOCATION_CITY`, and `LOCATION_ZIP`, present a unique challenge: maintaining real-world geographic consistency. Independently sampling a surrogate for each address component would lead to nonsensical combinations, such as a city from Bavaria being placed in the state of Saxony. To maintain realism, the real-world administrative relationships between location entities within a document must be modeled and preserved.

1. **Step: Hierarchical Clustering of Annotations** For each document, the Surrogator first constructs a *"location tree"* representing the relationships between

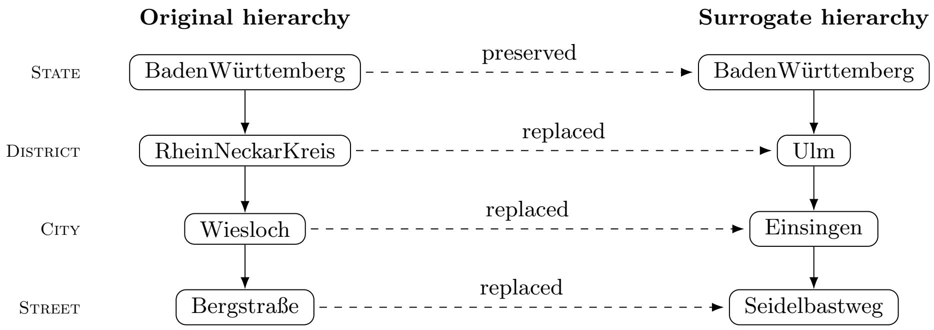


Fig. 2 Illustration of hierarchical location surrogation with level descriptors. The state is preserved, while lower-level entities are sampled consistently within the administrative boundary of the surrogate parent.

all annotated location entities. It queries the OSM Overpass API to fetch the administrative level of each location and determine its parent-child relationships. For example, it identifies that *"Heidelberg"* is a city within the state of *"Baden-Württemberg"*. This step effectively clusters the document's scattered location annotations into a coherent hierarchical structure.

2. **Step: Constrained Hierarchical Sampling** Once the location tree is built, surrogate generation proceeds top-down. By default, the `LOCATION_COUNTRY` and `LOCATION_STATE` are preserved to maintain high-level geospatial utility.
   (a) A surrogate is first sampled for the highest-level entity being replaced (e.g., a city). The system queries OSM for a random city within the preserved state.
   (b) This choice constrains all subsequent sampling. A surrogate for a street, for example, will then be sampled exclusively from within the boundaries of the newly chosen surrogate city.

   This constrained, top-down process guarantees that the final set of address surrogates is geographically consistent and realistic, as illustrated in Figure 2.

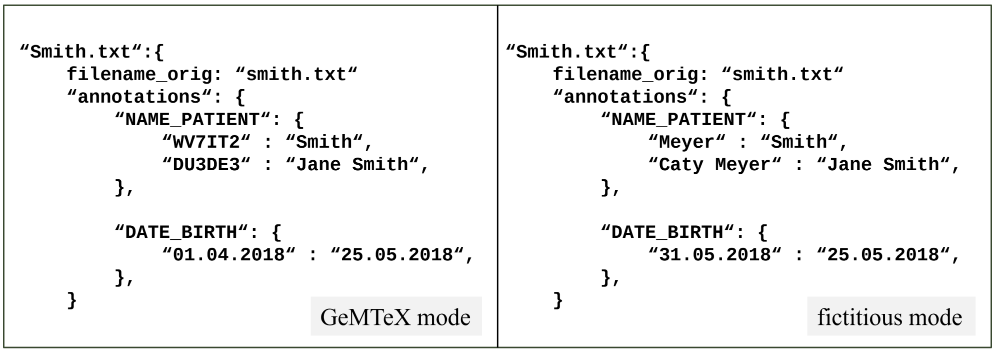
3 Handling of Non-Replaced Entities

Finally, not all PII categories are automatically replaced. The entities `PROFESSION` and `OTHER` are deliberately excluded from the surrogation process. The information they contain is often too unique, nuanced, or high-risk to allow for a safe and meaningful automated replacement. This deliberate exclusion underscores a core 'safety-first' design principle of the surrogator: when automated replacement carries an unacceptably high risk of failing to obscure a unique identifier, the safest course of action is to flag the document for human review and probable exclusion from the final corpus to ensure maximum patient safety.

4 Consistency and Mapping

A central feature of the surrogator is **consistency within a document**: every instance of the same entity is replaced by the same surrogate. If the name *Jane Smith* appears multiple times in a doctor’s letter, it is consistently replaced with the same fictitious name (e.g., *Caty Meyer*) each time.

Simultaneously, a mapping between the original value and the surrogate is stored for each replacement. This is accomplished via a **key-based mapping file** in JSON format (see Figure 3, bottom right).



```
“Smith.txt“:{
    filename_orig: “smith.txt“
    “annotations“: {
        “NAME_PATIENT“: {
            “WV7IT2“ : “Smith“,
            “DU3DE3“ : “Jane Smith“,
        },

        “DATE_BIRTH“: {
            “01.04.2018“ : “25.05.2018“,
        },
    }
```

GeMTeX mode

```
“Smith.txt“:{
    filename_orig: “smith.txt“
    “annotations“: {
        “NAME_PATIENT“: {
            “Meyer“ : “Smith“,
            “Caty Meyer“ : “Jane Smith“,
        },

        “DATE_BIRTH“: {
            “31.05.2018“ : “25.05.2018“,
        },
    }
```

fictitious mode

Fig. 3 Example of a mapping table of all key-based entries, left site: example snippet for GeMTeX mode, right site example snippet for fictitious mode.

This file is of critical importance: It contains the original PII and is the key to theoretical reversibility. It must therefore remain within the secure environment and must not be distributed under any circumstances. Storing these mappings enables complete documentation and auditability of the de-identification process.

At the end of the workflow, two separate artifacts are produced:

1. `public` directory: a set of **de-identified text documents** in which PII has been replaced with realistic surrogates.
2. `private` directory: set of **private files** containing the sensitive mapping information and the QA reports.

Only the de-identified text files leave the immediate control environment for subsequent research steps, such as semantic annotation.

References

[1] Lohr, C., Faller, J., Riedel, A., Nguyen, H.M., Wolfien, M., Hofenbitzer, J., Modersohn, L., Romberg, J., Prasser, F., Omeirat, J., Wen, Y., Galusch, O., Hahn, U., Seiferling, M., Dieterich, C., Klügl, P., Matthies, F., Kind, J., Boeker, M., Löffler, M., Meineke, F.: GeMTeX's De-Identification in Action: Lessons Learned & Devil's Details. Netherlands (2025). https://doi.org/10.3233/SHTI251406

[2] Lohr, C.: GraSCCo_PII_V2 - Graz Synthetic Clinical text Corpus with PII Annotations. Zenodo (2025). https://doi.org/10.5281/zenodo.15747389 . https://doi.org/10.5281/zenodo.15747389

[3] Health, U.S.D., (HHS), H.S.: Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule. https://www.hhs.gov/hipaa/for-professionals/special-topics/de-identification/index.html

[4] Lohr, C., Eder, E., Hahn, U.: Pseudonymization of PHI items in german clinical reports. Stud Health Technol Inform **281**, 273–277 (2021)

[5] Eder, E., Krieg-Holz, U., Hahn, U.: De-identification of emails: Pseudonymizing privacy-sensitive data in a German email corpus. In: Mitkov, R., Angelova, G. (eds.) Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), pp. 259–269. INCOMA Ltd., Varna, Bulgaria (2019). https://doi.org/10.26615/978-954-452-056-4_030 . https://aclanthology.org/R19-1030/

[6] Foundation, T.A.S.: Apache UIMA. https://uima.apache.org/

[7] Eckart De Castilho, R., Klie, J.-C., Gurevych, I.: Integrating INCEpTION into larger annotation processes. In: Hernandez Farias, D.I., Hope, T., Li, M. (eds.) Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 110–121. Association for Computational Linguistics, Miami, Florida, USA (2024). https://doi.org/10.18653/v1/2024.emnlp-demo.12

[8] Stahl, D.: Our consent management. gICS. https://www.ths-greifswald.de/en/researchers-general-public/gics/

[9] Bialke, M., Stahl, D., Leddig, T., Hoffmann, W.: The university medicine greifswald's trusted third party dispatcher: State-of-the-art perspective into comprehensive architectures and complex research workflows. JMIR Med Inform **12**, 65784 (2024) https://doi.org/10.2196/65784

[10] Honnibal, M., Johnson, M.: An improved non-monotonic transition system for dependency parsing. In: Màrquez, L., Callison-Burch, C., Su, J. (eds.) Proceedings

of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1373–1378. Association for Computational Linguistics, Lisbon, Portugal (2015). https://doi.org/10.18653/v1/D15-1162

[11] GmbH, E.: spaCy, Industrial-Strength Natural Language Processing in Python. https://spacy.io/

[12] GmbH, E.: spaCy, German Available trained pipelines for German. https://spacy.io/models/de

[13] Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, ??? (2019). http://arxiv.org/abs/1908.10084

[14] Stiftung, O.: OpenStreetMap. https://www.openstreetmap.org

[15] Lohr, C.: JULIELab/ClinicalSurrogateGeneration: Release. https://doi.org/10.5281/zenodo.4884304 . https://doi.org/10.5281/zenodo.4884304

[16] Seiferling, M.: OSM_Location_data_collection: OSM-Based Location Entity Datasets. https://github.com/dieterich-lab/OSM_Location_data_collection. README file, commit 8a1ac18 (2025)

[17] Olbricht, R.: Overpass API. https://wiki.openstreetmap.org/wiki/Overpass_API