Towards Generating Synthetic EHR Knowledge Graphs – a Probabilistic Approach

Milos Jovanovik[1,2][0000−0001−7360−8015], Eva Milenkova[2],
Maxime Jakubowski[1][0000−0002−7420−1337], and Katja Hose[1][0000−0001−7025−8099]

[1] Institute of Logic and Computation, TU Wien, Austria
{name.surname}@tuwien.ac.at
[2] Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University in Skopje, N. Macedonia
{name.surname}@finki.ukim.mk

Abstract. Advances in medical AI and data analytics require large amounts of patient data. Due to privacy concerns, such data is not always available. Synthetic data generation promises a solution to provide the required data despite privacy restrictions. In this paper, we therefore introduce SynMedRDF, an open-source tool to generate synthetic Electronic Health Records. It ensures clinical accuracy by using real-world probabilities and correlations. The data is output as an RDF knowledge graph, enabling structure- and semantics-aware sharing, linking, and analysis.

1 Introduction

In today's era of data-driven healthcare, developing AI models and analytic systems depends on access to large, diverse, and high-quality medical datasets. However, obtaining and sharing real-world Electronic Health Records (EHRs) poses ethical, legal, and practical challenges, particularly due to privacy concerns and regulations such as GDPR and HIPAA [11]. Synthetic medical data has emerged as a viable solution, enabling work with realistic data without compromising confidentiality. However, ensuring clinical plausibility and structural richness remains challenging, especially for semantically informed AI and interoperable analysis.

In the past couple of years, several methods [3,4,8,12,13,15,17] have been proposed to synthetically generate medical data to overcome privacy concerns and provide valuable datasets for research, testing, and training purposes. Typically, however, they rely on access to real medical data. EHR-Safe [16], for example, uses a two-stage model combining sequential encoder-decoder networks and generative adversarial networks to address the inherent challenges of EHR data. The data is generated in tabular format though, not in RDF.

To address these shortcomings, we introduce SynMedRDF [6], an open-source tool for generating synthetic EHRs based on real-world probability distributions and correlations between medical variables, ensuring clinical accuracy and internal consistency without relying on the availability of real medical data. The data

is serialized as an RDF Knowledge Graph (KG) to leverage semantic technologies for data linking, integration, and analysis. SynMedRDF allows configurable parameters to tailor datasets for specific use cases and research goals.

2 Data Preparation

Our goal is to generate complete EHRs with patient information such as name, surname, age, and gender, together with a diagnosis, including an ICD-10 code [5], relevant medical history with probable precursor diagnoses, prescribed associated medications with their ATC code [1] and detailed drug information.

Diagnoses. To encode diagnoses, we apply the International Classification of Diseases (ICD-10) [5], which uses alphanumeric codes, such as "A00: Cholera" and "C50: Malignant neoplasm of the breast". This standardized system helps recording, analyzing, and sharing health data between different healthcare systems.

Medication. The ATC classification system [1] is an international standard for encoding medication, which organizes drugs into a five-level hierarchical structure. We use ATC codes to denote the ID of the medication as part of the generated EHR, in order to ensure data compatibility.

Integrating ATC codes of medications with ICD-10 diagnoses is essential for creating precise EHRs, e.g., by linking patient diagnoses to plausible prescribed medications. To consider this correlation, we make use of the results of López-Rodríguez et. al. [9].

Statistical Distributions. Creating realistic synthetic EHRs requires that our data adheres to the demographic characteristics of patients and the prevalence of specific diagnoses and medications. Using uniformly random distributions is not a good approach because it skews results in data analytics and benchmarks. Instead, it is preferable to use statistical models and distributions based on real-world data. A key resource we identified for this process is a study conducted on a large dataset from a hospital in Xuzhou, China, which involves over 144,000 patients and over 1,550,000 diagnostic records [10]. The statistical data from this study provides the foundation for generating realistic patient records, mapping diagnoses to groups of patients, and mapping medications to specific diagnoses. More specifically, we use the findings from this large-scale study to define the probability that a synthetic patient belongs to a given age group, as well as the probability of their gender, the probability of their diagnosis based on the age group, gender, and pregnancy status, and the probability of a medication being prescribed based on the diagnosis.

Correlation of Diagnoses. The correlation between one or more diagnoses (comorbidities) in a single patient has been studied in the past [10]. We use these insights to generate realistic plausible medical histories for the generated synthetic patients.

3 The SynMedRDF Generator

We developed our synthetic EHR generator as a Python tool, which is open-source and publicly available on GitHub[3] [6].

Input Data and Configuration. Our synthetic EHR generator uses the following information as input: diseases and their ICD codes, medications and their ATC codes, probabilistic distributions of patients and diseases per age group, gender and pregnancy status, diagnosis-diagnosis and diagnosis-medication interrelationships. These information points are stored in collections in the tool, and are open-source. To support more specific use cases, the generator can be configured (parameters are stored in a YAML file `config.yml`). The default parameters include `faker_locale`, which specifies the locale for generating randomized data for the patient; `filter_icd_groups`, which allows exclusion of specific ICD-10 code groups; `result_format`, which determines the output format (e.g., JSON, RDF, XML); `maternity`, which defines the probability of a patient being assigned a pregnancy-related diagnosis; and `fhir_format`, which determines whether an additional FHIR format [14] result is generated alongside the primary output.

Creating an Electronic Health Record. The synthetic data generation process starts by generating an artificial patient. Personal patient information includes a patient identification number, name and surname, gender, address, mobile phone, and age. For data points that do not affect the diagnosis decision, we use the Faker library[4], which can generate artificial data of a large variety, including names, addresses, emails, phone numbers, dates, job titles, company names, etc. The selection of a diagnosis for a generated patient is based on gender, age, and specific conditions like pregnancy, with diagnoses filtered according to these factors and any exclusions specified in the `config.yml` file. Once selecting a diagnosis is done, the generator selects a medication associated with it, based on the probabilities defined in the dictionary. Next, based on the rules for determining possible preconditions and comorbidities of a given disease, the generator can select previous diagnoses and add them to the EHR being generated. Additionally, to ensure speed and efficiency in the data generation process, the generator supports parallel execution of tasks across multiple processes, significantly increasing the performance when generating large data volumes.

Output. The generator outputs records in various formats, including JSON, XML, and RDF (serialized as Turtle, RDF/XML or JSON-LD). We chose RDF as default to enable interoperability with other healthcare datasets available as knowledge graphs – either publicly on the Web or in private contexts. The generator also uses the `Schema.org` vocabulary, as the most commonly used RDFS vocabulary on the Web. Below we show an example EHR for a single patient in RDF (Turtle):

[3] SynMedRDF Generator: `https://github.com/etnc/synmed-ehr-generator`
[4] Faker Library: `https://pypi.org/project/Faker/`

```
<https://synmed.org/patient/1223dd17-9> a schema:Patient ;
    schema:address "51 Aguascalientes, San Nelly de la Montaña, Mexico" ;
    schema:age 37 ; schema:birthDate "1987-01-02" ;
    schema:diagnosis <https://synmed.org/diagnosis/D61.9> ;
    schema:gender "Male" ;
    schema:name "Porfirio Griego" ;
    schema:telephone "(101) 467-4034" ;
    schema:usesDrug <https://synmed.org/medication/H02AB04> .

<https://synmed.org/diagnosis/D61.9> a schema:MedicalCondition ;
    schema:identifier "D61.9" ;
    schema:name "Aplastic anemia, unspecified" .

<https://synmed.org/medication/H02AB04> a schema:Drug ;
    schema:administrationRoute "Oral" ;
    schema:description "Dosage: 7.5 mg per day" ;
    schema:identifier "H02AB04" ;
    schema:name "methylprednisolone" ;
```

Usage. Using SynMedRDF is straight-forward; if parameters are not provided explicitly (e.g., `records`, `result_format`, etc.), the generator can fall back to pre-configured default values (`config.yml`). The example below shows a command to instruct the generator to generate 100 EHRs in RDF Turtle format:

```
python generate_ehr.py --records 100 --result_format turtle
```

4 Discussion and Conclusion

In this paper, we have presented SynMedRDF, a ready-to-use tool for generating synthetic EHR data as RDF knowledge graphs. The main design decision was to ground the generator in real-world probabilistic distributions. Another major design decision was the configurability of the tool — the distributions can be explicitly configured (e.g. 100% pregnant, female patients for an OB/GYN dataset). The third major design decision was to make the tool open-source, so that any interested party can join in the development and extension of the generator. We will continue to work on the tool by further improving performance. We also plan to extend the expressivity of the generated knowledge graphs by including other medical entities, such as procedures, observations, etc. Another direction for future work is investigating the possibility of more detailed control over the structure of the generated knowledge graph using SHACL shapes [2, 7].

Acknowledgments. This publication is based upon work from COST Action CA23147 GOBLIN - Global Network on Large-Scale, Cross-domain and Multilingual Open Knowledge Graphs, supported by COST (European Cooperation in Science and Technology, `https://www.cost.eu`). This research was partially supported by the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje. This research was partially funded by the European Union's Horizon Europe research and innovation program under grant agreement no 101136244 (TARGET).

References

1. ATC/DDD Index 2025, https://atcddd.fhi.no/atc_ddd_index/
2. Shapes Constraint Language (SHACL), https://www.w3.org/TR/shacl/
3. Baowaly, M.K., Lin, C.C., Liu, C.L., Chen, K.T.: Synthesizing Electronic Health Records Using Improved Generative Adversarial Networks. Journal of the American Medical Informatics Association **26**(3), 228–241 (2019)
4. Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W.F., Sun, J.: Generating Multi-Label Discrete Patient Records Using Generative Adversarial Networks. In: Machine Learning for Healthcare Conference. pp. 286–305. PMLR (2017)
5. Hirsch, J., Nicola, G., McGinty, G., Liu, R., Barr, R., Chittle, M., Manchikanti, L.: ICD-10: History and Context. American Journal of Neuroradiology **37**(4), 596–599 (2016)
6. Jovanovik, M., Milenkova, E., Jakubowski, M., Hose, K.: SynMedRDF on GitHub (2025), https://github.com/etnc/synmed-ehr-generator
7. Jovanovik, M., Vecovska, M., Jakubowski, M., Hose, K.: RDFGraphGen: An RDF Graph Generator based on SHACL Shapes. arXiv preprint arXiv:2407.17941 (2025), https://arxiv.org/abs/2407.17941
8. Kapenekakis, A., Dell'Aglio, D., Vesteghem, C., Poulsen, L., Bøgsted, M., Garofalakis, M., Hose, K.: Synthesizing Accurate Relational Data under Differential Privacy. In: BigData. pp. 433–439 (2024)
9. López-Rodríguez, I., Reyes-Manzano, C.F., Reyes-Ramírez, I., Contreras-Uribe, T.J., Guzmán-Vargas, L.: Drugs, Active Ingredients and Diseases Database in Spanish. Augmenting the Resources for Analyses on Drug–Illness Interactions. Data **6**(1), 3 (2021)
10. Ma, H., Ding, J., Liu, M., Liu, Y.: Connections Between Various Disorders: Combination Pattern Mining Using Apriori Algorithm Based on Diagnosis Information From Electronic Medical Records. BioMed Research International **2022**(1), 2199317 (2022)
11. Mittal, S., Thakral, K., Singh, R., Vatsa, M., Glaser, T., Canton Ferrer, C., Hassner, T.: On Responsible Machine Learning Datasets Emphasizing Fairness, Privacy and Regulatory Norms With Examples in Biometrics and Healthcare. Nature Machine Intelligence **6**(8), 936–949 (2024)
12. Rashidian, S., Wang, F., Moffitt, R., Garcia, V., Dutt, A., Chang, W., Pandya, V., Hajagos, J., Saltz, M., Saltz, J.: SMOOTH-GAN: Towards Sharp and Smooth Synthetic EHR Data Generation. In: AIME 2020. pp. 37–48 (2020)
13. Torfi, A., Fox, E.A.: CorGAN: Correlation-Capturing Convolutional Generative Adversarial Networks for Generating Synthetic Healthcare Records. In: FLAIRS. pp. 335–340 (2020)
14. Vorisek, C.N., Lehne, M., Klopfenstein, S.A.I., Mayer, P.J., Bartschke, A., Haese, T., Thun, S.: Fast Healthcare Interoperability Resources (FHIR) for Interoperability in Health Research: Systematic Review. JMIR Medical Informatics **10**(7), e35724 (2022)
15. Xie, L., Lin, K., Wang, S., Wang, F., Zhou, J.: Differentially Private Generative Adversarial Network. arXiv preprint arXiv:1802.06739 (2018)
16. Yoon, J., Mizrahi, M., Ghalaty, N.: EHR-Safe: Generating High-Fidelity and Privacy-Preserving Synthetic Electronic Health Records. npj Digital Medicine **6**, 429 (2023)
17. Zhang, Z., Yan, C., Mesa, D.A., Sun, J., Malin, B.A.: Ensuring Electronic Medical Record Simulation Through Better Training, Modeling, and Evaluation. Journal of the American Medical Informatics Association **27**(1), 99–108 (2020)