TECHNICAL UNIVERSITY OF MUNICH

TUM Data Innovation Lab

"Building and Applying an Ontology-Based Medical Graph Database"

Authors	Philipp Hausenblas, Nikolas Pfeiffer, Manal Hamdi, Chris Emezue
Mentor(s)	Su Hwan Kim, Francisco Pinto, Alvaro Sanchez
	Smart Reporting
Co-Mentor	Philippe Sünnen, Özge Sahin
Project Lead	Dr. Ricardo Acevedo Cabra (Department of Mathematics)
Supervisor	Prof. Dr. Massimo Fornasier (Department of Mathematics)

Jul 2021

Abstract

Background: Radiologists, medical doctors who interpret medical images for diagnosis, go through the strenuous stages of perusing large image datasets, evaluating and interpreting their findings, and writing radiology reports based on their findings. Secondly, these doctors have to memorize a huge, ever-growing amount of medical knowledge. Thirdly, the amount of electronic health records (EHRs) and clinical data keep increasing exponentially, but around 80% of them are unstructured (free-form writing) and large proportions are not even digitized. Given the importance of EHRs and clinical data to the provision of effective treatment to patients, is it paramount to 1) make medical knowledge easily accessible for medical doctors and 2) structure the report data so that the reports (and by extension - EHRs) generated are structured. This structured reporting will foster digitization of reports, sharing of medical knowledge and automation of some of the medical tasks.

The Project: The TUM Data Innovation Lab project titled 'Building and Applying an Ontology-Based Medical Graph Database' which took place in cooperation with Smart Reporting, was created to develop and design a proof-of-concept structured, growing medical knowledge graph database. Furthermore, some report analyses were conducted and a recommendation system (with a user interface) was built to enable medical doctors leverage the power of the knowledge graph database.

Contents

1 Project Overview

1.1 Smart Reporting

Smart Reporting is a digital health start up, based in Munich, founded in 2014 by Prof. Dr. Wieland Sommer. Today, the company brings together a large interdisciplinary team of physicians and software engineers to develop software that is rooted in a deep understanding of clinical workflows. Its multilingual, cloud-based software for structured reporting in radiology and pathology is used by more than 10,000 physicians in more than 90 countries.

Diagnostic reports serve an important role in medicine. Physicians may base their diagnosis and treatment decision on such findings reports. The completeness and quality of such diagnostic reports is critical for subsequent clinical decision-making. Today, these reports are mostly narrative text documents created without structure or guidance.

Smart Reporting aims to make up-to-date medical knowledge accessible for clinical decisions and structure the report data so that reports generated are machine-readable and computerizable. The company has been working on approaches to make the creation of new reporting templates more scalable by improving reusability of existing content. This involved the introduction of reusable, modular content units, among others. Today, Smart Reporting offers leading structured reporting solutions that help customers become more productive, improve quality and streamline communication. International collaborations and partnerships with leading players in the field help ensure that the software reaches clinical routine and an exciting innovation pipeline [1].

1.2 Problem Definition and Goals of the Project

Problem Definition: The Project aims to solve the problem radiologists face in interpreting large medical data and making reports. The large, ever-growing amount of medical knowledge needed to make decisions, as well as the unstructured format of the medical reports are the two major issues we investigated during the scope of the project.

Goals of the Project: The (sub)goals of the project were:

1. Researching the application of medical knowledge graphs for structured medical reporting.
2. Designing, developing and implementing a proof-of-concept radiology knowledge graph.
3. Clinically relevant analytic queries of synthetic report data.
4. Building a recommender system for report template creation based on the knowledge graph, as well as a user interface to test the recommender system.

The data we used for the project came from:

1. SNOMED CT - a publicly available ontology of medical terms.
2. Smart Reporting's structured medical reporting template database. For the report analysis, synthetic report data was created based on Smart Reporting's templates.

1.3 Project Timeline

In the following table, we give a short overview of our project timeline. The detailed timeline can be found in Table 2.

Table 1: Timeline of our DI-LAB Project

Week(s)	**Plan**
Week 1-3	Clinical Reporting, Knowledge Representation and Graph Databases
Week 4-5	Template-Based Graph Creation
Week 6-14	Graph-Based Data Analysis of Medical Reports
Week 6-14	Recommender system for Graph-Based Template Creation

Furthermore the roadmap for report analysis creation can be found in Figure 25 while that of the recommender system can be found in Figure 24.

2 Structured Medical Reporting and Knowledge Representation

Radiology is a field of medicine in which images of the body's organs are interpreted in order to diagnose disease [18]. Radiologists are medical doctors who interpret medical images for diagnosis. The job of the radiologist is two-fold: identifying and interpreting the information available from diagnostic imaging studies and communicating that interpretation meaningfully to the referring clinician [6].

The complexity of medical imaging has increased dramatically over the past few decades, providing radiologists with an ever-larger number of images to interpret and more imaging modalities to compare. Important clinical information is often recorded in unstructured free text, and converting it to a structured format can be a time-consuming task that may not successfully capture all facets of the information. [15, 14]. Most reports still contain free-form text dictated or typed by the radiologist, with an introductory section (summarizing the examination technique and clinical history), a main body (consisting of a paragraph or more describing the findings), and a brief overall impression section [15, 10]. Given the growing complexity of the information radiologists are charged with interpreting, it is important to develop a standard method of writing medical reports. This is where structured reporting comes into play. Structured reporting enables the capture of radiology report information so it later can be retrieved and reused [16].

Studies [15, 16, 14, 6] have shown that structured medical reporting is a promising tool in radiology reporting. The benefits of structured reporting include making data easily retrievable for both clinicians and research purposes, improving quality of care on the basis of standardized language used among all patients and clinicians, and reducing the misses of incidental findings on studies done for other purposes by having a standard way of reporting medical findings [12].

2.1 Knowledge Representations and Ontologies

In order to make medical definitions, procedures, diseases and diagnostics machine-readable, it is necessary to introduce a well-defined terminology (or ontology). "Terminology" and "ontology" are often used interchangeably in this context. One of the most popular terminologies in clinical documentation is SNOMED CT [2]. It provides more than 350,000 unique concepts. Every concept is identified by an `node id` and contains multiple attributes, such as its *name*, possible *synonyms* and a *description*. Concepts are organized in hierarchies (top level domains) and connected to one another via relationships. The relationships explain the connection between any two nodes. A typical relationship is the `is a` relationship. For example, the concept `Neurological finding (finding)` is connected via `is a` to the concept `Clinical finding (finding)` which means the former is a subtype of the latter (or the former is a child and the latter is a parent). In total, SNOMED CT has over $1,360,000$ relationships and provides the means to represent medical knowledge in a machine interpretable way. An overview of SNOMED CT is shown in Figure 2. An important aspect of structured ontologies such as SNOMED CT is that they allow semantic inference and reasoning, thus, representing more knowledge than directly encoded. For example, as the `is a` relation is known to be transitive, a lot more concepts are connected via the `is a` relation than directly shown.

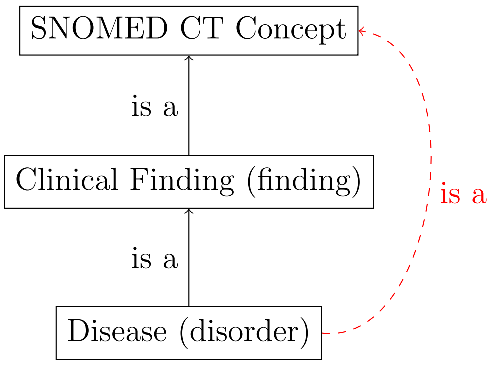

Figure 1: Although there exists no direct edge between `SNOMED CT Concept` and `Disease (disorder)` it can be inferred that they are connected via `is a` by its transitiveness. Thus, we know that `Disease (disorder)` is a type of `SNOMED CT Concept`.

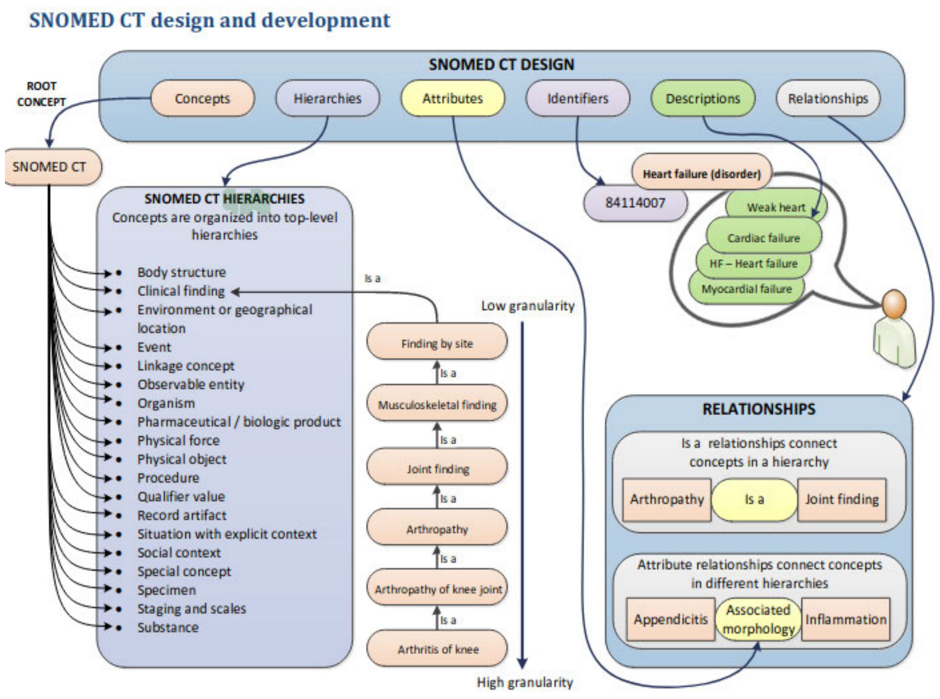

Figure 2: Design of SNOMED CT.

2.2 Templates and Reports

A template can be understood as a structure that guides the medical doctor through all the relevant clinical parameters that need to be evaluated while making reports or publishing findings. It is an essential component of structured reporting because it enables the methodical creation of structured medical content. Smart Reporting models templates as a graph that consists of different concepts that are connected via edges (relationships). These concepts are mostly taken from ontologies like SNOMED CT. Templates can, however, be generated independent of ontologies.

A radiologist can use such a template to describe the results of diagnostics, fill in the patient data (e.g. `Tumor:` Present, `Location:` Right Lung, `Diameter:` 3 cm, etc.) and write a conclusion. A template, with patient data and conclusions, is then called a structured report. In contrast to classical prose reports, structured reports are comparable and more importantly can be analyzed, as we will discuss in more detail in section 4. Since template creation is an exhausting and complicated task, we created a recommender system to assist in template creation (section 5). Figure 3 shows an instance of a template and its corresponding graph structure.

3 Knowledge Graphs for Medical Ontologies

One of the project's main goals was to develop a medical knowledge graph. Most simply put, a knowledge graph is knowledge (or knowledge representation) organized in a manner that a machine can easily understand and extract information from. The advantages of using knowledge graphs in the medical domain are abound: they pave the way for

[redacted]

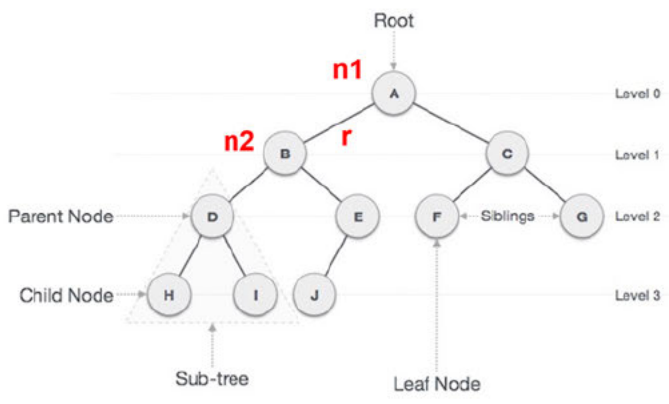

Figure 3: A template and its corresponding graph structure on the right. Every concept corresponds to a node in the graph.

automation of many medical processes, as well as the implementation of machine learning to aid in the medical domain [7].

Studies[7, 14] have shown that the best way to realize a knowledge graph is through a graph database, as opposed to relational databases(like Oracle Database, MySQL, etc). This is because knowledge graphs have very complex structures that relational databases cannot handle. Taking SNOMED CT as an example of a medical knowledge representation, [7] showed that due to the "ontologic and polyhierarchical" nature of SNOMED CT, a relational database was not able to fully utilize the whole knowledge embedded in the ontology. The author(s) showed the power of graph databases over relational databases through a series of medical queries on the SNOMED CT ontology.

Selection of graph database model: Having established that a graph database was best suited to design, develop and implement an incrementally growing medical knowledge graph, we studied in depth the two existing graph model approaches, Resource Description Framework (RDF) and Labeled Property Graphs (LPG) [9, 4, 5], in order to find the better one for our task. While both RDF and LPG share a lot of similarities, LPG was chosen mainly because it supported the uncertainty of the graph schema and it allowed duplicate edges. With RDF the structure of nodes and edges must be predefined (which was an issue since our graph schema was undefined at the early stage) and duplicate edges or edge properties were not possible [9, 4].

Selection of graph database: After deciding to go with labeled property graphs (LPGs), an appropriate graph database had to be chosen. Neo4j[1] is currently the industry-leading graph database in the world. The next in line is Grakn[2]. We therefore limited our

[1]https://neo4j.com/
[2]https://vaticle.com/

choices to Grakn or Neo4j (it is important to add that we explored other graph databases too).

Grakn seemed better suited at first because of the ability to create semantic rules, which would make querying easier. On the down side, Grakn is more strict, which makes it less adaptable to future changes in the schema. Also there was very little to no open documentation on loading SNOMED CT data into the database, in contrast to Neo4j which had several public documentations.

Neo4j is schema-free which makes implementation easier. If a schema is needed in the future, it could be defined in the form of constraints. From the performance perspective Neo4j is faster.

All the above factors inspired our choice of Neo4j.

[redacted]

[redacted]

- [redacted]
- [redacted]

  [redacted]

  [redacted]
- [redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

4 Report Creation and Analytical Queries

4.1 Data Generation

In order to generate the report dataset, we used the Python programming language[3] as it provides powerful libraries. Performance time was not a priority issue for us as we only needed to generate the report data once, then use it to make queries. We generated the following in CSV format, and loaded into Neo4j:

1. Patient metadata
2. Report metadata
3. Reports

4.1.1 Patient Metadata

Having patient data in our synthetic report dataset would increase the domain of our report analysis (which is discussed in section 4.2). For this proof of concept we generated 150 patient data. The metadata relevant for each patient were `id`, `name`, `date of birth` and `gender`. The patient ids were integers between 0 and 149. Regarding the gender, half of the patients were chosen to be male - and the other half female. We generated their names according to the gender. And for the date of birth, we decided to have patients between 18 and 100 years old.

4.1.2 Reports Metadata

Before generating the report metadata it was important to first ascertain the total number of reports, R, given that we had 150 patients. It was important that some patients had more than 1 report in order to run queries on the development of a medical condition over time for that particular patient. Consequently we decided that each patient could have from 1 to 10 reports, but in average should have 2 reports. Therefore we generated the number of reports assigned to a patient according to a normal distribution bounded between 1 and 10, with a mean of 3 and a standard deviation of 1. The distribution generated doubles which were converted to integers by taking the floor. In this way, we generated 450 reports.

[redacted]

[3]https://www.python.org/

4.1.3 Reports

At this stage we generated the metadata of 450 reports along with the metadata of 150 patients they belonged to. What we had left was to generate a CSV file with the content of the report from the template it corresponded to. To do that we needed input data files which described the medical data we wanted to generate, as well as some additional information that would make our synthetic data more realistic.
The input data files we needed were:

1. [redacted]

2. [redacted]

[redacted]

3. Sections file
   A node was said to be a section if it should always be active. Therefore this file lists which nodes were section.

Display	Node Code	System	Node Type
Finding of Abdomen	609624008	SNOMED CT	Section

Figure 6: Header and example of section nodes file

4. Groupings file
   Each row of the file contained a node and a type of outgoing edge from it with a label `is_mandatory` (indicating whether the selection was mandatory or optional) and `is_single` (indicating whether it was a single or multiple selection) (see Figure 7).

Parent code	Parent display	Relationship code	Relationship display	is mandatory
46621007	Atelectasis	363698007	Finding site	Mandatory

Figure 7: Header and example of groupings file

[redacted]

[redacted]

We ended up with a CSV file where each row had the following format: `parent node`, `parent activation status`, `edge`, `child node`, `child activation status`.

4.2 Analytical Queries

We loaded the generated report data into the Neo4j knowledge graph by running appropriate Cypher queries in a Jupyter notebook. First, all the corresponding nodes of each report had to be loaded and merged. This means that if duplicate nodes existed in the report data, they would not occur multiple times in the graph database within each report. Then the patient and report-overview nodes were loaded. [redacted]

After carefully loading all the nodes and relationships from the reports and the templates into the knowledge graph, analytical queries were performed to show the usability of the report data. First of all, medical questions were formulated together with the medical employees of Smart Reporting. Then, corresponding queries were created and run. Some of the clinical questions asked were:

- How many of the reports are with clinical procedure CT and how many with X-ray?
- Which risk factors occur in patients with Pulmonary Embolism?

[4] https://en.wikipedia.org/wiki/Breadth-first_search

[redacted]

- What percentage of pneumonia cases are accompanied by pleural effusions?
- What is the severity distribution of pneumonia cases with pleural effusions (mild/moderate/severe)?
- What is the distribution of pleural effusions that are unilateral (right or left) or bilateral (right and left)?
- What are the most common locations of pulmonary embolism?
- What is the amount of acute and chronic pulmonary embolism cases?
- How often are heart size increases observed in PE cases?
- How are the locations of pulmonary embolism distributed?
- What findings most commonly coexist with pulmonary embolism?
- How are patients with pulmonary embolism distributed per gender/per age?
- Pulmonary Nodules and Lung Lesions: What is the average tumor size of all patients with this diagnose?
- Pulmonary Nodules and Lung Lesions: How did tumor size evolve per patient?
- What is the frequency of suspicious lymph node by regions?

- Which are the patients with a specific finding in a specific time period?

In the following paragraphs, we provide a visualization of the results to some clinical queries. Further visualizations of the results of the clinical queries stated above are shown in the Appendix.

The distribution of the reports per method can be seen in fig. 9, which are 300 reports using Computer Tomography as procedure and 150 with X-ray.

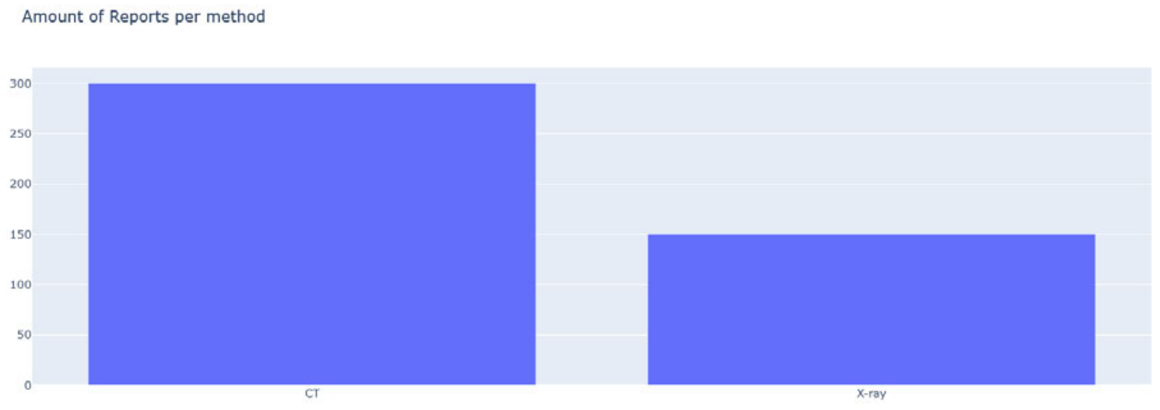

Figure 9: Amount of reports per method

Figure 10 shows the distribution of reports with pleural effusions within all the reports with pneumonia cases. In this example 47,7% of the pneumonia cases are accompanied by pleural effusion.

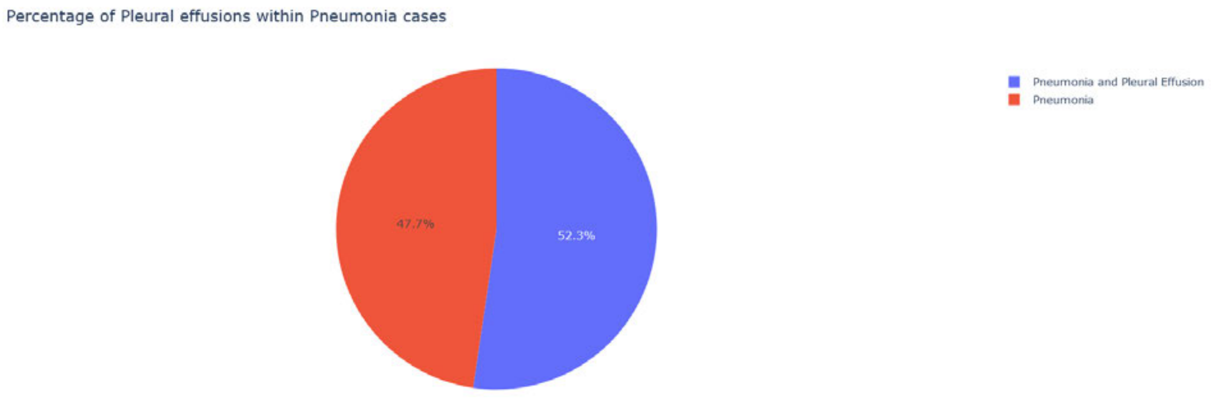

Figure 10: Percentage of Pleural effusions within Pneumonia cases

The evolution of a parameter can be visualized like in Figure 11. The tumor size in *cm* is shown over time for patient 10, 22, 71, 109 and 146. With this visualization existing trends can be seen directly.

4.3 Limitations

During the creation of report data and the medically relevant queries, some limitations were faced. For the proof of concept it was not yet relevant, but for a production-ready

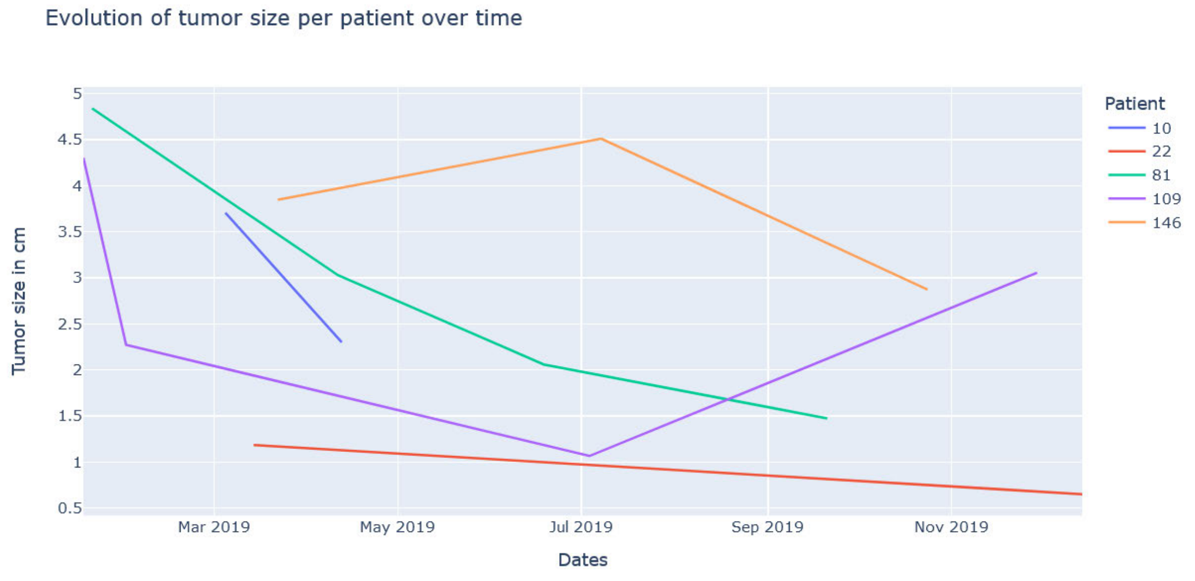

Figure 11: Evolution of tumor size per patient over time

implementation the performance had to be evaluated and improved. High performance was also necessary to make live queries possible and use those results (for example for a live dashboard giving an overview of all patients with a diagnosis of cancer). During the report creation no duplicates of one instance were created, though in reality multiple instances of one finding can occur (for example multiple instances of pulmonary nodules). This issue only occurred because we created synthetic data, instead of using real-world data. Our graph model can handle multiple instances of one finding nevertheless. Finally, during the project only templates from the field of *Chest Radiology* were used. Though there exists other medical disciplines (for example Cardiology, Oncology, Pathology etc) where data can be used for analysis purposes. Combining data of multiple medical disciplines will lead to more advanced queries and analyses.

4.4 Future Work

Our current proof of concept can be improved in the following ways:

- Exploring the capabilities of the created knowledge graph in the future with real clinical data.
- Creating a user interface for the report analyses. One option could be to create a dashboard where some parameters (for example specific clinical findings, gender and age of patients or time intervals of the report creation dates) can be selected. Those parameters could then be added to a predefined query and the results of this query shown within the dashboard. Using this approach means that users, with no skills in the use of Cypher query language, could also make report analyses and visualize the results.

This project is a proof of concept to showcase the power of representing medical data in a knowledge graph, and so we aligned our priorities with the aim to provide a working prototype. Therefore, in order for our work to be used in production, there are additional considerations to have in place:

- Adapt the knowledge graph model to support 'modules': some subtrees should be identifiable to be able to automatically duplicate them if requested by the user.

- Generate more data from additional templates.

5 Recommender System for Template Creation

The second major part of the project was to implement a recommender system for the creation of templates. Creating radiology reporting templates is an important part of Smart Reporting's work. Currently, template creation is exhaustive.When creating a template, the creator is confronted with the question of which concepts should be included. Furthermore, the creator needs to know their names, ontologies and ontology IDs and therefore must browse the different ontologies in order to obtain this information. This is a very time-consuming, error-prone and repetitive process. To fix this problem, we designed a recommender system from scratch for template creation. Given a current specification of the template, the recommender system should suggest which concept(s) should be added to the template, in the specific context of the template tree.

5.1 Overview of Recommender Systems

With the rapid development of the internet, the volume of data has grown exponentially. Due to the large amount of information and choices, users find it increasingly difficult to make selections. To improve the user experience, recommender systems have been developed for scenarios such as music recommendation, movie recommendation, and online shopping [11]. In our project, we extended the field of recommendations to medical concepts used in radiology reporting templates.

Recommender systems are mainly categorized into collaborative filtering (CF)-based recommender systems, content-based recommender systems, and hybrid recommender systems [3]. CF-based recommendation models user preference based on the similarity of users or items, while content-based recommendation utilizes the content of the items (i.e. almost no information about users is used) [11]. In agreement with the Smart Reporting mentors, we decided to stay user agnostic (i.e. we do not assume any knowledge about the creator of the template). Thus, it is only natural to use content-based methods of medical concepts in our implementation of the recommender system.

Introducing a knowledge graph (KG) into the recommender system has been shown to greatly improve the recommendations. A knowledge graph (KG), also known as a semantic network, represents a network of real-world entities (objects, events, situations, or concepts) and illustrates the relationship between them [17]. A knowledge graph is made up of three main components: nodes, edges, and labels. Any object, place, or person can be a node. An edge defines the relationship between the nodes. In other words, KGs are a way to store both information and it's meaning.

5.2 Implementation

Here we explain the mechanisms behind our recommender system and highlight some issues faced during implementation. The knowledge graph used for our recommendations is composed of the annotated templates (described in detail in Section 4.1.3), and the SNOMED CT ontology. We loaded them into the KG using the Snomed dataloader which is available for free on GitHub. As stated in Section 3, we used the labeled property graph model with the Neo4J database.

5.2.1 Recommender System

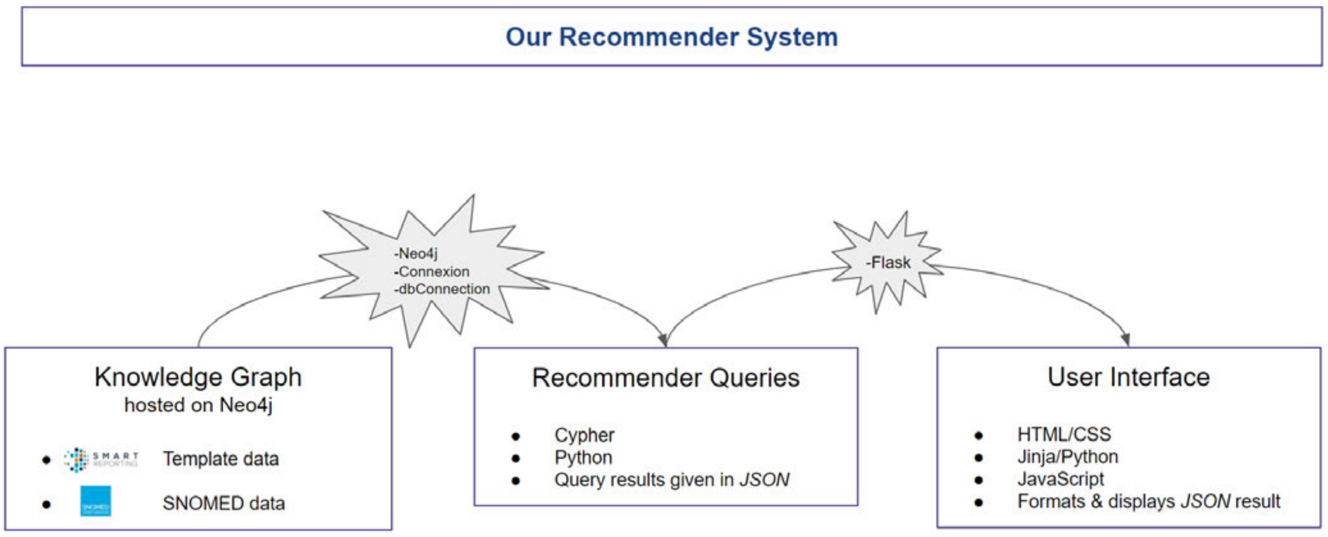

Figure 12: Flowchart of Implementation of our Recommender System

Figure 12 gives a flowchart of the implementation of our recommender system. The database queries were written in Cypher (the native query language for NEO4J). Using Flask, we connected the codebase with the queries: making queries from our database in Cypher and connecting the results to an API via Python packages and Flask.

As we started this project from scratch (no prior infrastructure existed when we began), the time scope of the project was very limited. Therefore, we decided not to exploit advanced machine learning techniques for knowledge graphs, but rather to implement some handcrafted recommendation methods that we discussed with the Smart Reporting team.

Knowledge Graph (KG) The knowledge graph is made up of:

- Template Graph: This refers to the template files described in Section 4.1.3.
- SNOMED CT Graph: This refers to the SNOMED CT database (specifically the SNOMED release from January 2021).

In the future, this knowledge graph could be made more powerful by incorporating other ontologies like RADLEX to further increase the amount of available concepts, and subsequently the power of the KG.

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

The type of recommendation needed determines if some of the above inputs are required or optional. For example, if a user wants a *Name Search* recommendation: that is, a user begins typing `lun...` and wants immediate recommendations of concepts containing the keyword `lun`. In this case, the parent node is not required, because we are only looking at the KG and getting concepts containing the keyword `lun`. The modality will help provide better recommendations (maybe the user prefers concepts connected more to X-ray than CT), so it will be required.

Every recommendation consists of two queries, one in which the template graph is queried, one in which the SNOMED CT graph is queried. Results from the template graph are always prioritized over results from the SNOMED CT graph. SNOMED CT offers many granular concepts, but many of them are not relevant for clinical documentation and image diagnostics.

Types of Queries Implemented: In the scope of the DI-LAB project, we were able to successfully implement four query types:

1. **Name Search (with Fuzzy capabilities):**
   This type of query is when a user wants to get recommendations based on a complete or incomplete name. For example, if a user is looking for recommendations based on *lung finding* (a complete name), once the user starts typing `lu...` (which is incomplete), the recommendation system already starts giving live look-ahead suggestions. For this type of query, the required input is the keyword (whether complete or incomplete) as well as the modality.

   For every concept node in our template graph we check whether its name matches the key phrase in a fuzzy manner. That means a word in the concept name matches a word in the key phrase within a Levenshtein distance[8] of 2. The Levenshtein distance between two words is the number of single character operations that must be performed to transform one into another. It is a way of quantifying how dissimilar two strings (e.g., words) are to one another. For example, "Pulminaru Artery" would match "Pulmonary Artery". For the implementation of this name search, we used Neo4j's Full-Text search together with Lucene for advanced querying. When querying the template graph, we can only make use of the *name* attribute, but when querying the SNOMED CT graph, we can make use of all *synonyms* that are available for a concept. Searching for the key phrase `pulmonary nodule` would, for example, also return the concept "Nodule of lung" as the key phrase, *pulmonary nodule* is one of its synonyms.

2. **Children Filter:**
   Here, a user wants to find concept children suggestions under a parent concept. For this query, we take as input details of the parent concept (like the *node id* and *node ontology*) and return only concepts that are direct children of the node with the given id and ontology. These concepts naturally form good candidates for being added to the template next. Children nodes from the template graph are always prioritized over those from the SNOMED CT graph. We also take care not to output duplicate nodes: nodes that were already in our template tree should not be considered when getting suggestions from the SNOMED CT graph. Also, nodes that are already in our user interface should not be repeated in our current suggestions.

3. **Children Relationship Filter:**
   In our knowledge graph, the concepts are connected to each other by different types of relationships (for example `is a`, `finding_site`, `causative_agent`). The relationships explain the connection between any two nodes. For example, $A -$ [`finding_site`] $\rightarrow B$ means that B is a finding location of A.

   In many cases, the template creator wants to first know the available relationships between the parent node and suggested child nodes before going further to choose the child nodes. Therefore, we want to 1) query which relationship types are available for the children of the currently selected parent node, 2) let the creator choose the relationship wanted, and 3) return all children that are connected to the parent node via the selected relationship.

4. **Children Filter with Name Search (Fuzzy):**
   This query type combines the children filter with name search. This query type is called when a user wants to find children concepts based on a specific given keyword. For example, a user wants to get children suggestions for *Clinical Finding*, which contain the keyword `lung`. This also takes into consideration the relationship capabilities explained in *Children Relationship Filter.*

[redacted]

[redacted]

Multiple Graph Querying: It is important to note that the SNOMED CT graph is enormous. Also, for technical reasons, we stored the template graph and the SNOMED CT graph in two different sub-databases. This greatly increased the time overhead. Since the query time is an important key performance indicator in a recommendation system (we do not want to wait for a long time just to get suggestions), we had to figure out a way to greatly reduce the query time taken. For that we implemented a combined graph querying approach using Neo4j's Fabric platform [13]. Neo4j Fabric supports the storage, processing, analysis and management of data distributed and stored in multiple Neo4j databases. It enables data distribution and parallel querying over multiple graph (sub)databases. In integrating multiple graph querying, we were able to greatly reduce the query time for recommendations, For example, we reduced the time for *Children Filter* **from 8.276 seconds to 30.4 milliseconds.**

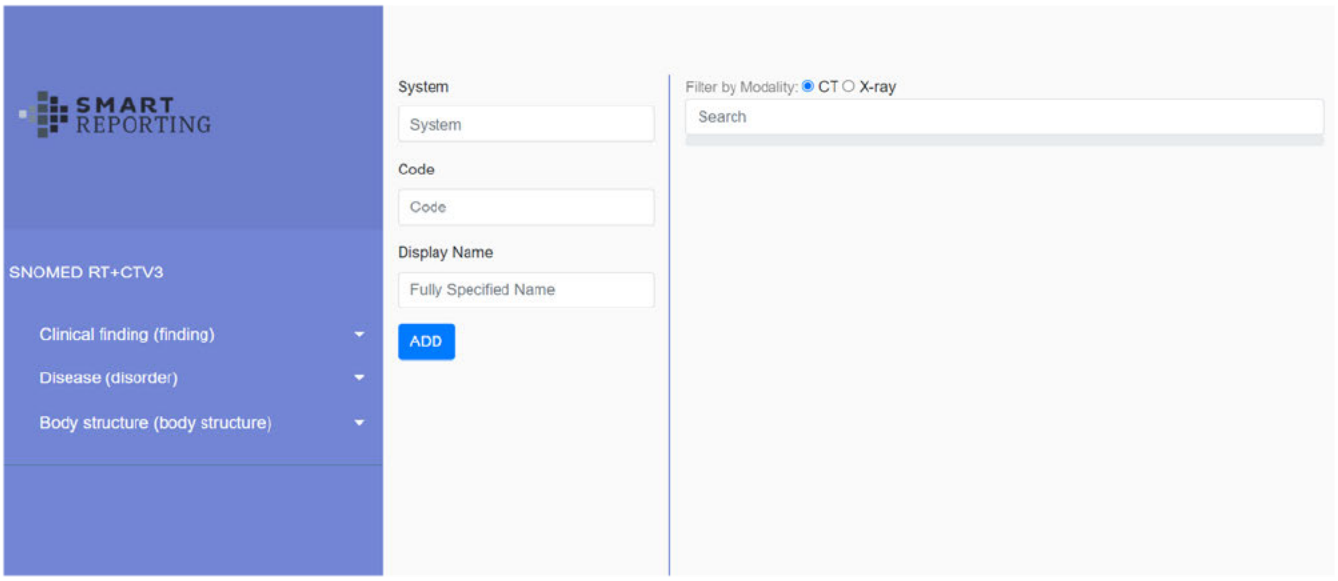

Figure 13: Homepage of our UI

5.2.2 User Interface (UI)

A user interface is necessary for users to actually try our recommendation queries and for us to know how good the suggestions from our recommender systems are. Therefore, it was imperative to design a user interface for the recommender system. [redacted]

[redacted] Therefore we (the team and Smart Reporting) decided it would be better to build a separate Template creation platform (using the programming languages we are most familiar with) along with the recommendation system. This led to us creating our user interface from scratch. The homepage of our UI is show in Figure 13.

Our user interface is a dynamic web platform. The front-end was written in HTML, CSS with Jinja and JavaScript to connect to the back-end, which was written in Python. Using Flask, we connected to an API where we sent the queries and received output in JSON format. This output was then formatted and displayed in our UI.

5.3 Future Work for Recommendation System

Despite the excellent performance of our current recommender system, there exists several ideas to further improve the recommendation quality:

1. Using Report data for recommendation: The number of reports that a concept is part of could also be a property for additionally ordering our recommendations. As all our report data is randomly generated, we decided that it is not meaningful to implement this approach within the scope of our DI-LAB project.

2. NLP (Natural Language Processing) for name search: Using the fuzzy name search showed good performance, but it is in no way capable of distinguishing between more meaningful and less meaningful words in a sentence. Some of the *names* of the concepts are long and consists of stop words (words that are not important to the query like ‘the’, ‘a’, ‘of’, ‘is’, etc). We think that using advanced machine learning techniques (with focus on medical terms) could be very beneficial to filtering and ordering the results.

3. Translation incorporation: Implementing searches in different languages would enable searches and search results to be made in other languages besides English. Ingesting additional language versions of SNOMED CT could be a feasible approach. We were not able to implement this due to time constraints.

4. Other ideas include user related recommendations (using knowledge about the user creating the template while making recommendations) and module recommendation (a powerful tool would be to recommend whole subtrees).

6 Conclusion

Our DI-LAB project, titled “Building and Applying an Ontology-Based Medical Graph Database”, was the design of a proof-of-concept, structured, growing medical knowledge graph. In the scope of this project we

1) designed a model for the medical knowledge graph.
2) generated synthetic medical reports from the annotated templates. In addition to that, we generated patient data.
3) created the knowledge graph using the generated report/patient data and incorporating the SNOMED CT medical ontology.
4) performed medical report analyses using the knowledge graph.
5) created a recommender system and a user interface for template creation, leveraging the knowledge graph.

As shown through the sample queries in Section 4.2, we were able to extract good insight from our knowledge graph - like visualizing the tumor size over time or getting coexisting findings to pulmonary embolism. We also showed (through test cases) that the recommender system, by using insights from the knowledge graph, was able to provide useful recommendations which aid medical template creators. We were able to greatly reduce the query time of the recommender system (section 5.2.1), which is one of if not the most important key performance indicators.

References

[1] URL: https://www.smart-reporting.com/en/company/about (visited on 06/30/2021).

[2] URL: https://www.nlm.nih.gov/healthit/index.html (visited on 07/10/2021).

[3] Mehdi Hosseinzadeh Aghdam, Morteza Analoui, and Peyman Kabiri. “Modelling trust networks using resistive circuits for trust-aware recommender systems”. In: *Journal of Information Science* 43.1 (2017), pp. 135–144. DOI: 10.1177/0165551516628733. eprint: https://doi.org/10.1177/0165551516628733. URL: https://doi.org/10.1177/0165551516628733.

[4] *AllegroGraph vs. Apache Jena - TDB vs. GraphDB Comparison*. URL: https://db-engines.com/en/system/AllegroGraph%3BApache+Jena+-+TDB%3BGraphDB (visited on 07/15/2021).

[5] Duygu ALTINOK. *Neo4j vs GRAKN Part II: Semantics*. en. Feb. 2020. URL: https://towardsdatascience.com/neo4j-vs-grakn-part-ii-semantics-11a0847ae7a2 (visited on 07/15/2021).

[6] Adrian Brady. “Radiology reporting: from Hemingway to HAL?” In: *Insights into Imaging* 9 (Mar. 2018). DOI: 10.1007/s13244-018-0596-3.

[7] W Scott Campbell et al. “An alternative database approach for management of SNOMED CT and improved patient data queries”. In: *Journal of biomedical informatics* 57 (2015), pp. 350–357.

[8] Fred J. Damerau. “A Technique for Computer Detection and Correction of Spelling Errors”. In: *Commun. ACM* 7.3 (Mar. 1964), 171â€“176. ISSN: 0001-0782. DOI: 10.1145/363958.363994. URL: https://doi.org/10.1145/363958.363994.

[9] Alex Donkers, Dujuan Yang, and Nico Baken. “Linked Data for Smart Homes: Comparing RDF and Labeled Property Graphs”. In: June 2020.

[10] R. A. Gagliardi. “The evolution of the X-ray report.” In: *American Journal of Roentgenology* 164.2 (Feb. 1995), pp. 501–502. ISSN: 0361-803X. DOI: 10.2214/ajr.164.2.7839998. URL: https://doi.org/10.2214/ajr.164.2.7839998.

[11] Qingyu Guo et al. *A Survey on Knowledge Graph-Based Recommender Systems*. 2020. arXiv: 2003.00911 [cs.IR].

[12] Reham R. Haroun, Maysoon M. Al-Hihi, and Hani H. Abujudeh. “The Pros and Cons of Structured Reports”. In: *Current Radiology Reports* 7.11 (Oct. 2019), p. 31. ISSN: 2167-4825. DOI: 10.1007/s40134-019-0342-8. URL: https://doi.org/10.1007/s40134-019-0342-8.

[13] *Introduction - Chapter 7 Fabric*. URL: https://neo4j.com/docs/operations-manual/4.0-preview/fabric/introduction/ (visited on 07/15/2021).

[14] Kory Kreimeyer et al. “Natural Language Processing Systems for Capturing and Standardizing Unstructured Clinical Information: a systematic review”. In: *Journal of Biomedical Informatics* 73 (July 2017). DOI: 10.1016/j.jbi.2017.07.012.

[15] Lawrence H. Schwartz et al. "Improving Communication of Diagnostic Radiology Findings through Structured Reporting". In: *Radiology* 260.1 (2011). PMID: 21518775, pp. 174–181. DOI: 10.1148/radiol.11101913. eprint: https://doi.org/10.1148/radiol.11101913. URL: https://doi.org/10.1148/radiol.11101913.

[16] David L Weiss and Curtis P Langlotz. "Structured reporting: patient care enhancement or productivity nightmare?" In: *Radiology* 249.3 (Dec. 2008), 739â€"747. ISSN: 0033-8419. DOI: 10.1148/radiol.2493080988. URL: http://intl-radiology.rsnajnls.org/cgi/content/full/249/3/739.

[17] *What is a Knowledge Graph?* en-us. URL: https://www.ibm.com/cloud/learn/knowledge-graph (visited on 07/14/2021).

[18] *What is radiology?* URL: https://rad-aid.org/resource-center/radiology-serving-the-world/what-is-radiology (visited on 07/15/2021).

Appendix

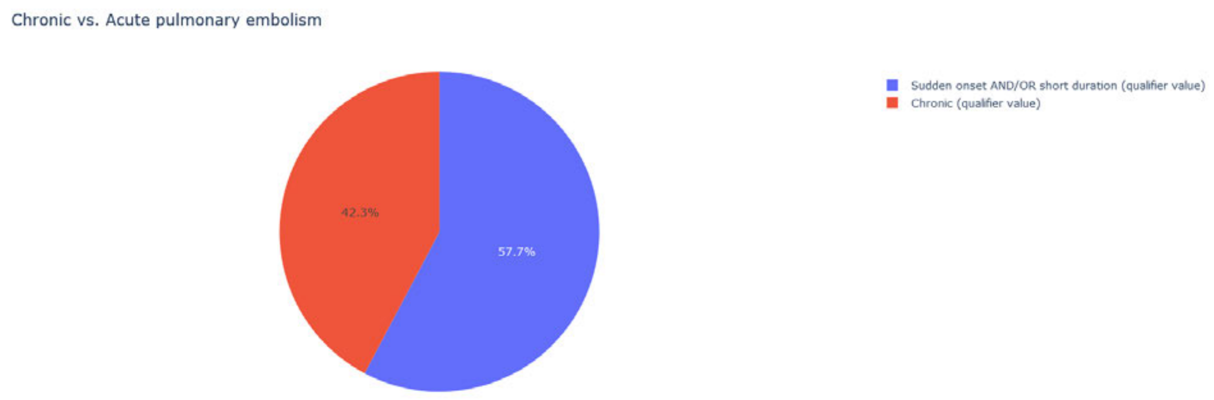

Figure 14: Chronic vs. Acute pulmonary embolism

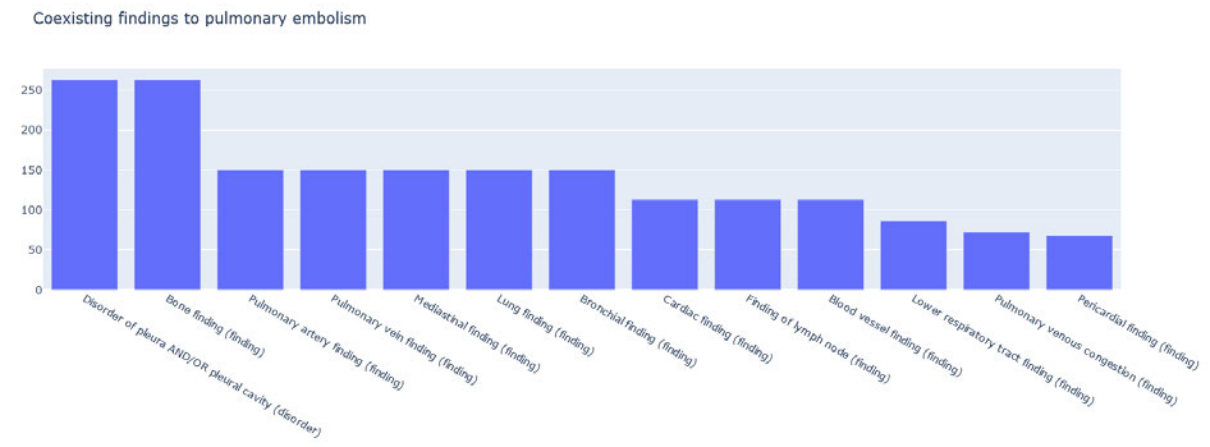

Figure 15: Coexisting findings to pulmonary embolism

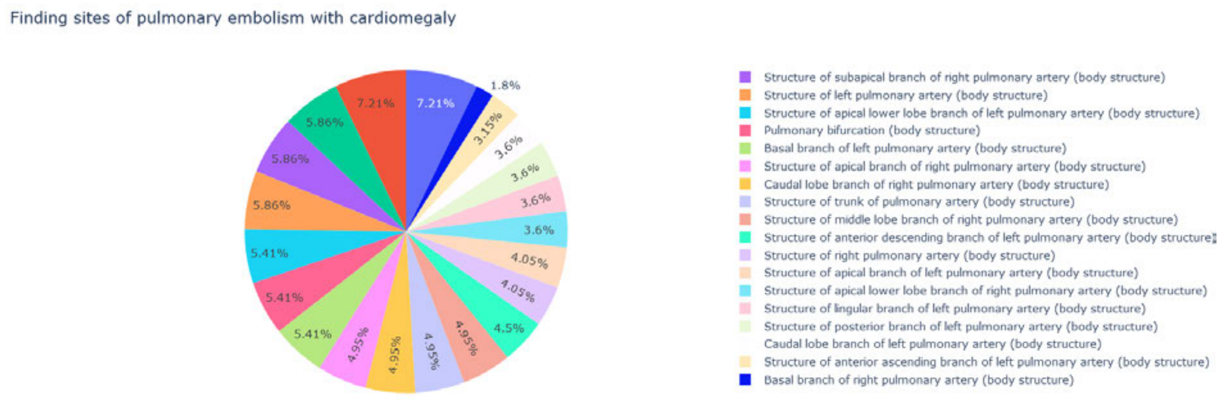

Figure 16: Finding sites of pulmonary embolism with cardiomegaly

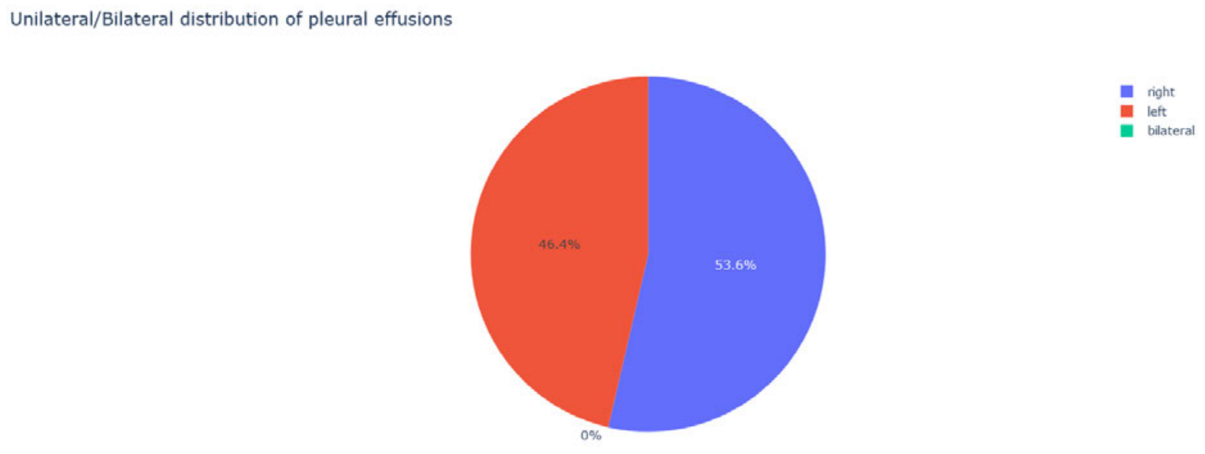

Figure 17: Unilateral/Bilateral distribution of pleural effusions

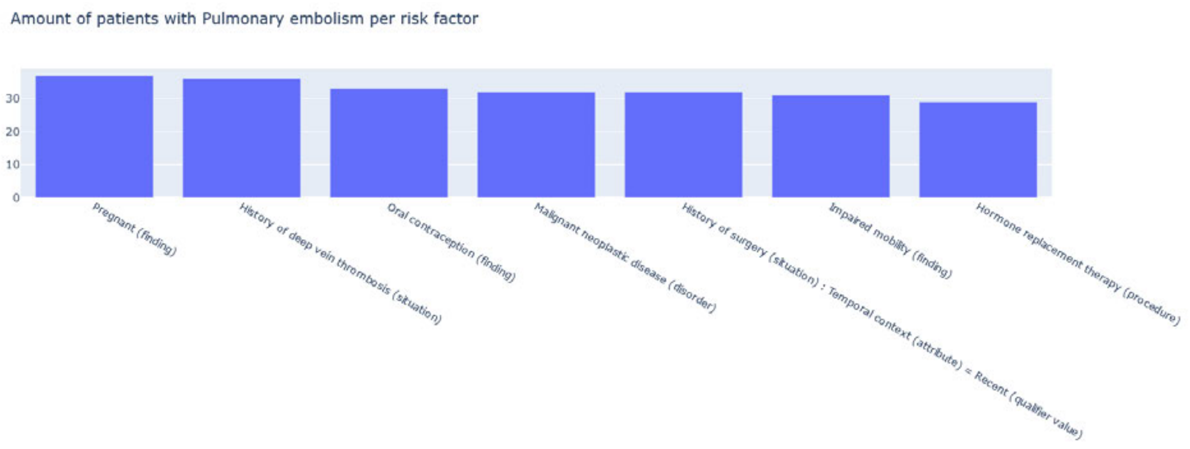

Figure 18: Amount of patients with pulmonary embolism per risk factor

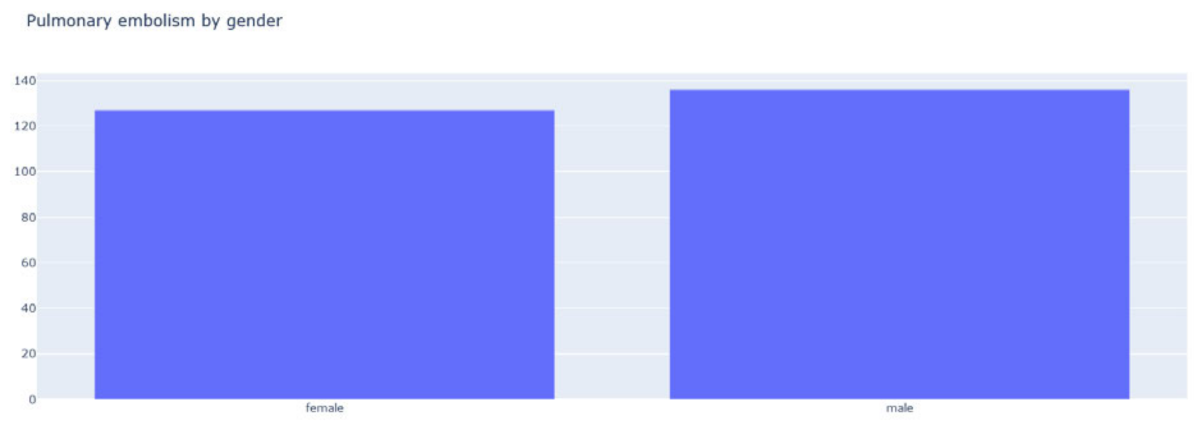

Figure 19: Pulmonary embolism by gender

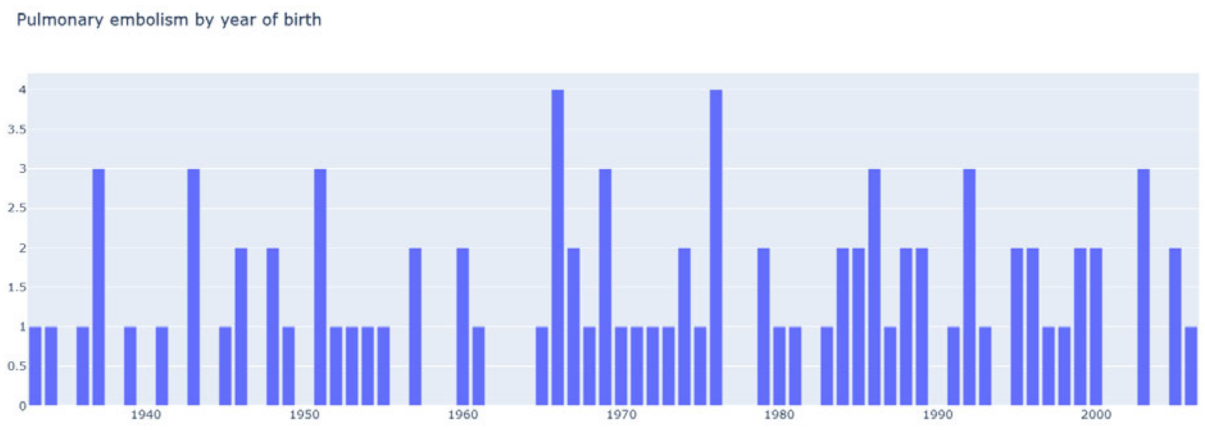

Figure 20: Pulmonary embolism by year of birth

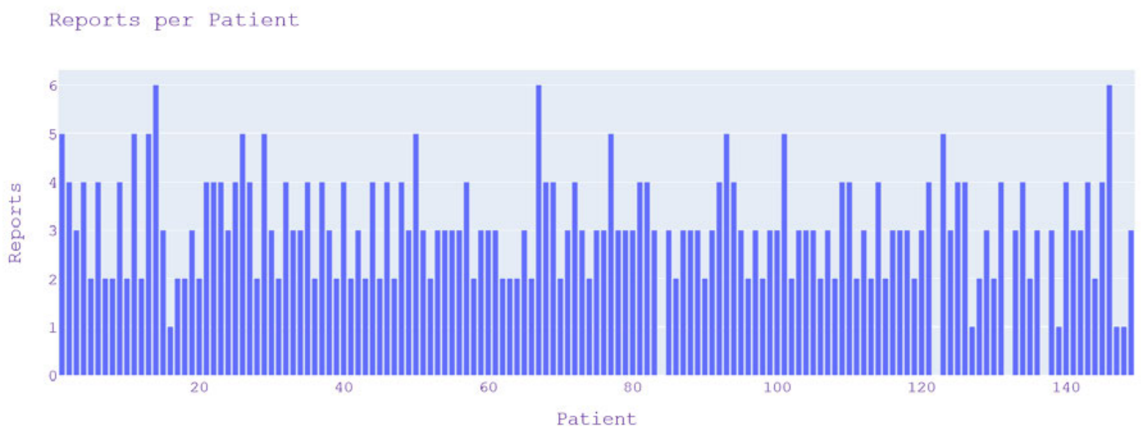

Figure 21: Amount of reports per patient

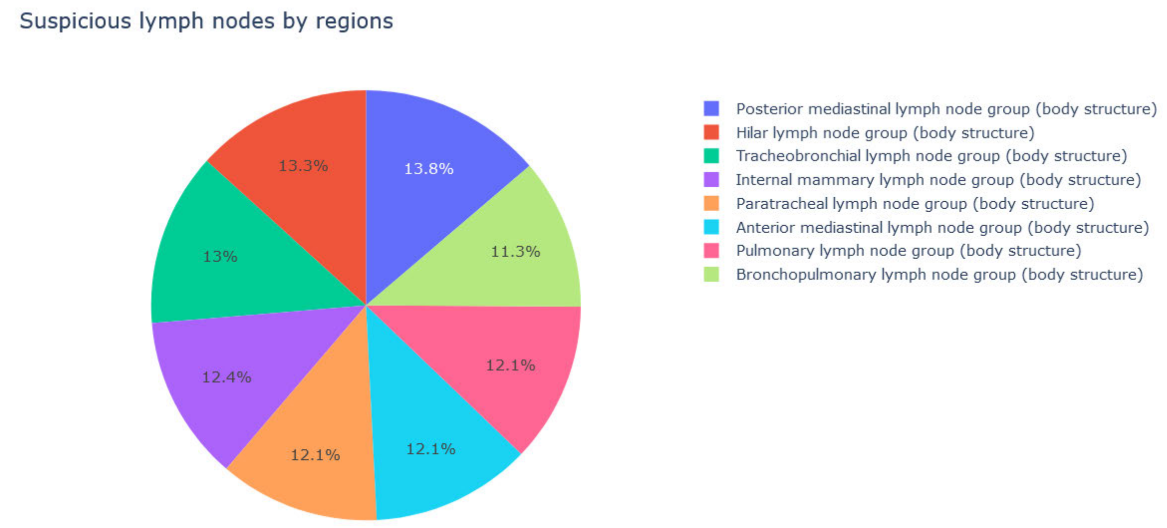

Figure 22: Suspicious lymph nodes distribution by regions

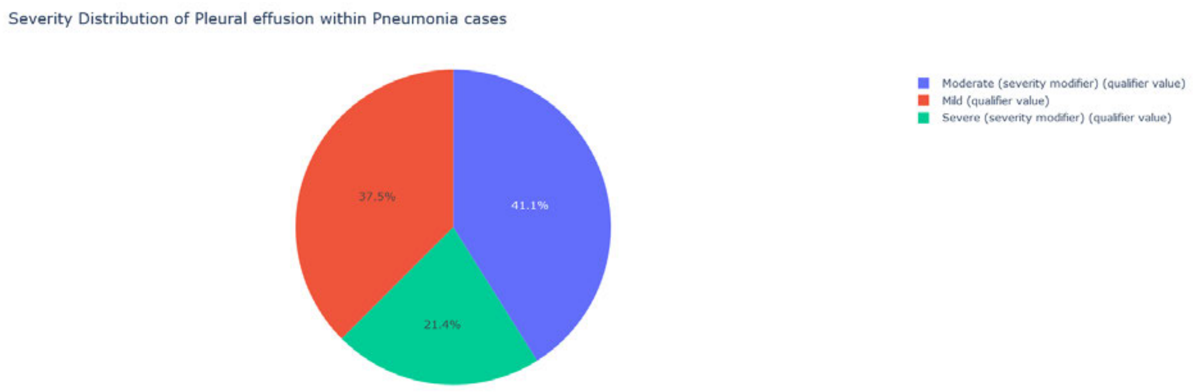

Figure 23: Severity distribution of pleural effusion within pneumonia cases

[redacted]

Table 2: Detailed timeline of our DI-LAB Project

Week(s)	**Plan**	**Detailed Explanation**

Week 1-3	Clinical Reporting, Knowledge Representation and Graph Databases	[redacted]

Week 4-5	Template-Based Graph Creation	[redacted]
Week 6-14	Graph-Based Data Analysis of Medical Reports	[redacted]

Week 6-14	Recommender system for Graph-Based Template Creation	[redacted]