Data Analytics

Text mining as a quality assurance method for the Orphanet nomenclature, can it be done?

Gemma Milman

March, 2023

Table of Contents

1. Glossary of terms

Below is a glossary of terms to help you navigate through the rare disease terminology used in this project.

Term	Definition
Clinical annotations	A set of clinical phenotypes that are observed in specific disease. At orphanet, the phenotypes are listed for each disease in order of frequency that they are observed in the patient population. The annotations are with the HPO terms.
Disease definition	A short description the clinical signs and symptoms (phenotypes) that characterise the disease, and distinguish it form other similar diseases.
HPO	Shorthand for the Human Phenotype Ontology project, which provides an ontology of medically relevant phenotypes and the relationship between related terms.
HPO ID	Each HPO term has a unique HPO identifier
HPO term	Standardised medical term to be used for each clinical phenotype (i.e., clinical sign or symptom). Each official term is accompanied by a definition and the synonyms
Nomenclature	A nomenclature provides a standardised naming convention and a set of rules for this naming convention. The Orphanet nomenclature specifically provides standardised names for every rare disease, an Orpha code, a disease definition and a list of exact synonyms for the disease.
Ontology	An ontology provides standardised names and definitions for entities and the relationship between the entities in a particular domain. Here, ontology specifically refers to the HPO ontology which provides standardized vocabulary for phenotypic abnormalities.
Orpha code	A unique identifier for every rare disease in the Orphanet knowledge base. The term 'orpha' is derived from 'orphan' which is commonly used to designate rare diseases (Orphan diseases). Orpha is used as a prefix in the tables for this project where the data comes from Orphanet.
Orphanet	A knowledge base for rare diseases, assimilating lots of different information about each rare disease entity. At the centre of this is the Orphanet nomenclature, which defines each disease by name, orpha code, definition and it's synonyms. Accompanying each disease is clinically relevant information, the associated orphan drugs, medical institutes/centres, and patient organisations.
Phenotype(s)	Any observable characteristic or trait of a disease, such as morphology, development, biochemical or physiological properties, or behaviour, without any implication of a mechanism.

2. Introduction

Orphanet is an international rare disease knowledge base, assimilating all aspects of data associated with all rare diseases. This ranges from orphan drugs, expert resources, scientific and medical knowledge of each disease, as well as a nomenclature and classification system.

So, what is a rare disease? A disease is considered rare if it affects less than 1 in 2,000 people. Whilst a rare disease on its own is uncommon, rare diseases as a whole are fairly frequent and the chances are that you know someone or have met someone with a rare disease (although you may not know they have a disease).

The rare diseases field touches all organ systems: a rare disease may involve the heart, blood, or the brain for example. They can even involve multiple different systems in the same disease (e.g. Brittle cornea syndrome involves the eyes, ears, joints), these disorders are considered syndromic. The distinction between diseases and syndromes is not the focus here, and for ease, all diseases and syndromes are referred to as disease (but disease and disorder may be used interchangeably).

The Orphanet nomenclature is pivotal to the Orphanet knowledgebase and consists of the disease name, orpha code (unique and stable identifier), synonyms and a disease definition. This nomenclature is beginning to be implemented in clinical settings across the globe for the coding of rare disease patients. Having a code allows rare disease patients to be easily identifiable in the hospital systems and should thereby help improve their care by facilitating access to information on their specific rare disease. This is particularly important outside specialist centres where knowledge of rare diseases may be limited.

As a former employee at Orphanet, I though it might be interesting to employ my newly developed coding and data analysis skills to address an issue that we had previously been unable to tackle.

2.1. Business case

Orphanet is committed to providing quality data. The data is currently manually curated and expert validated. Pre- and post-release procedures are in place to assure the quality of the data. Due to the technical constraints, no post-release, automated quality control is currently in place for the disease definitions.

The definitions characterise a disease in terms of its defining clinical characteristics, and should be stable, withstanding changes in evolutions in knowledge or medical developments (e.g., treatments that increase life expectancy). It is thus important to assure the quality of these definitions, as the nomenclature is now being implemented for coding of rare disease patients.

Given the recent developments in text mining and named entity recognition, the aim here is to exploit these techniques in order to provide a quality indicator for the Orphanet disease definitions.

2.2. Objective 1

Develop an indicator of quality for the Orphanet disease definitions by comparing with the Orphanet clinical annotations which are provided for a high proportion of diseases in the Orphanet knowledgebase.

The clinical annotations are a list of standardised medical terms (HPO terms), listing the signs and symptoms for each disease, and categorising them by frequency. The principal idea for the quality indicator is to extract the clinical terms from the disease definitions (via text mining) and compare them to the most frequent signs and symptoms listed in the Orphanet clinical annotations.

Below is a diagram to help illustrate the idea for developing a quality indicator for the definitions.

Diagram 1. Illustration of the text extraction from the disease definitions and comparing to the Orphanet clinical annotations. In the top of the diagram we have the definition for Rett disease and at the bottom we have a list of the very frequent clinical signs and symptoms (known as the Orphanet clinical annotations) for this same disease. The aim of the text extraction is to identify the clinical entities in the definitions (some examples are circled in red), map these terms to the standardised HPO term and then compare the resulting HPO terms with the terms in the clinical annotations. The proportion of matched terms would act as an indicator of definition quality.

2.3. Objective 2:

Develop a model to improve mapping between the extracted clinical terms and the standardised medical terms (HPO terms).

The standardised medical terms come from the Human Phenotype Ontology project. The extracted terms will need to be mapped to the HPO terms in order to subsequently compare with the clinical annotations.

It is not expected that the mapping will be very reliable, thus there will be a need to train a model to improve mapping between the extracted raw text and the standardized HPO terms.

The first objective is the preliminary step to the modelling aspect of this IronHack Project and will be the focus of this report. Whilst the second objective is out of scope for this RNCP report, it will be covered in the presentation.

3. Data collection

As a not-for-profit with a mission to Contribute to generating knowledge on rare diseases, Orphanet allows access to much of the core data via Orphadata.com. Data is provided as an XML file or is available via API on request. As I did not get a response to my API request, I proceeded with the download of the XML files.

The XML files were imported and parsed using the python Elmtree module, and the resulting list of dictionaries were read into a Pandas dataframe before being saved to csv. There are three data sets that have been extracted and include:

- Clinical annotations for Orphanet diseases (112689 rows × 5 columns)
    - Lists the HPO terms linked with each disease, and the frequency for which the occur in the disease.
- Orphanet nomenclature (10,675 rows x 8 columns)
    - Dataset containing the disease definitions linked to each disease
- Linearised Orphanet file (7,241 rows x 6 columns)
    - The complete set of diseases that exist in the database, which includes active and inactive entities. For this project, the data was limited to active entries.

The Human Phenotype Ontology (HPO) project provides an ontology of medically relevant phenotypes, disease-phenotype annotations, and the algorithms that operate on these. These standardised medical terms (HPO terms) are used to annotate the diseases in the Orphanet knowledge base. The HPO project permits free access to their ontology via download of an obo file. This file was imported and parsed via the pyobo module in python. A list was created for each attribute (e.g. term name, id, definition, synonyms) which subsequently converted into a single Pandas dataframe before being exported to csv.

- HPO onotology file (16874 rows x 5 columns)

The final data set was created by text mining the definitions and then mapping to the HPO terms in the HPO ontology file. This is described in more detail in the Data wrangling section.

4. Metadata

Below is the metadata for the four imported files, mentioned in the section above and the 'wrangled' data obtained from text mining of the disease definitions.

Orpha diseases (Orpha linearized file)	
Attribute	**Definition**
Orpha code (integer)	Unique identifier for each disease
Name (text)	Disease name
Expert link (text)	URL to webpage for the disease (kept because it can be useful to see disease page)
Class orpha codes (integer	Unique identifier for the classification
Class Names	Disease classification name

Table 1. metadata for import Orphanet linearized disease file

Orpha definitions (Orpha nomenclature file)	
Attribute	**Definition**
Orpha code (integer)	Unique identifier for each disease
Disorder Type	Disorder entities can be classed as either a category, clinical group, disease, malformation syndrome, or a disease subtype (clinical, etiological, histopathological). This information is kept because it is useful to see what type on entity it is.
Classification level	The above classification system is simplified here by classifying entities as either clinical group, disorder or subtype.
Definition	The textual description of the disease

Table 2. Metadata for the imported Orphanet nomenclature data

HPO ontology	
Attribute	**Definition**
Term (text)	Phenotype name (i.e. name of the clinical sign or symptom)
ID (integer)	Unique identifier for each HPO term, referred to as HPO ID
Definition (text)	Definition of the phenotype
Synonyms (text)	Synonyms of the phenotype

Table 3. Metadata for the imported HPO ontology data

Orpha clinical annotations	
Attribute	**Definition**
Orpha code (integer)	Unique disease identifier
HPO frequency (text)	Frequency that the HPO identifier occurs in the disease, categorical options : obligate, very frequent, frequent, occasional and rare.
HPO ID	Unique identifier for each HPO term, referred to as HPO ID

Table 4. Metadata for the imported Orphanet clinical annotations

Extracted HPO terms	
Attribute	**Definition**
Orpha code	Disease identifier
Phenotype	HPO phenotype extracted from the disease definition
HPO ID	Unique identifier of the above HPO phenotype

Table 5. Meta date for the terms extracted from the disease definitions.

5. Data cleaning and wrangling

Data cleaning is an important first step towards data analysis and includes assessing and addressing missing and duplicate values, removing irrelevant data, fixing structural errors and assessing outliers. Since the data here consists of text and identifier fields, outliers were not as issue. Instead, I will briefly address the other data cleaning methods applied in this project.

5.1. Removing missing data

There were no null values in the nomenclature, clinical annotations or linearized disease list. The HPO ontology had missing values in the definition and synonym fields, but as the corresponding HPO IDs and HPO Term names are required, the data entries were kept, and the null values replaced with blanks.

5.2. Removing duplicates

On checking duplicated rows, the clinical annotations data contained 33 duplicated entries which were removed. At the stage of setting up the database in SQL, it became apparent that there were still duplicates when assigning primary keys to my tables. On re-checking the tables for duplicates using only the identifier columns (OrphaCode and HPO_ID) 33 and 34 duplicates were identified respectively in the tables orpha_clinical_annotations and extracted_HPO_terms. These were subsequently dropped.

5.3. Removal of irrelevant data

The nomenclature data (definitions) contains certain categories that are irrelevant to this project. In particular, the disorders are classed by type, e.g. disorder subtype, group of disorders and disorder. However there are a few other categories that pertain to the classification system and are not relevant to this project, these include the category 'Category', 'Biological anomaly" and 'Particular clinical situation'. In addition, the data included some non-rare entities which were removed.

Some additional columns were removed such as Disorder ID (from the orpha_disease and orpha_definition data) and Expert link (from the definition data). The Disorder ID is and internal, legacy ID, for the diseases but since the Orpha code is unique for each disease, keeping the Disorder ID is unnecessary. The Expert link provides a URL to the disease page on the Orphanet website, this has been kept in the Disease table for referencing purposes, but a duplication in the definitions table is unnecessary.

5.4. Fixing structural errors

The main problem to fix was the format of the HPO identifier which differed between the HPO ontology and the Clinical annotations. The official structure is 'HP:#######' where the hash tag represents a digit. The 'HP:' characters were removed in the clinical annotations, leaving only the digits. Having matching formats is essential for creating relationships between data sets later on.

The definitions and Clinical Annotations both contained diseases (Orpha Codes) which were not in the main disease (orpha_diseases) table. Thus, these extra entries (1356 for the definitions, 44 for the clinical annotations) were removed from the respective tables.

With regards to the HPO ontology data, the Orphanet clinical annotations contained data for 122 HPO identifiers which were not in the HPO ontology file (possibly due to an update of the ontology since the clinical annotations were last made). These were equally dropped.

5.5. Wrangling: extraction of clinical terms from the Orphanet disease definitions

Extraction of the clinical terms from the Orphanet definitions proved quite challenging. Clinical text has formulations and characteristics that are distinct to everyday prose (e.g. prose found in blogs). Initially, the plan was to use Named Entity Recognition (NER) to extract the terms, but this requires a pretrained model. Several options were explored (ClinicalBERT, MedCAT, SciSpacy, MetaMap[1]) but due to time constraints this could not be explored any further. Instead, as a first iteration, the

[1] Useful information : https://gweissman.github.io/post/using-metamap-with-python-to-access-the-umls-metathesaurus-a-quick-start-guide/

disease definitions were split using regex patterns and cleaned to remove stop words. Lemmatization of each term was tested but did not seem to impact the HPO matching later.

5.6. Wrangling: Align the extracted terms with the HPO terms

The next step was to match these terms to their corresponding HPO term. This was done using the module FuzzyWuzzy[2], iterating over each clinical term and comparing with a list of all HPO terms. FuzzyWuzzy has four different ratio calculations that can be used for the comparison; these were tested to see which returned the best matches by assessing the number of correct matches. Using my knowledge of the domain, the ratio cut-off of 90 was determined; the ratios below this threshold were very mixed, giving a mix of incorrect and correct matches, and would thus need manual curation to determine correct matches.

Once the terms were matched based on the textual closeness, the corresponding HPO identifiers were mapped back to the terms using the Pandas map function.
In total, approximately 12,000 extracted, raw terms were matched to the HPO terms from and input of 53,000.

[2] Documentation: https://pypi.org/project/fuzzywuzzy/)

Data exploration

5.7. Exploration of the Orphanet knowledge base

In the 'linearized disease' file, diseases are organized according to their primary classification (for the most part, the classification corresponds to the primary medical speciality for that disease). Given this information, let us start by looking at the repartition of diseases in the Orphanet knowledgebase according to the disease calssification. The tree map below (created in Tableau) shows the proportion of all diseases in each disease classification.

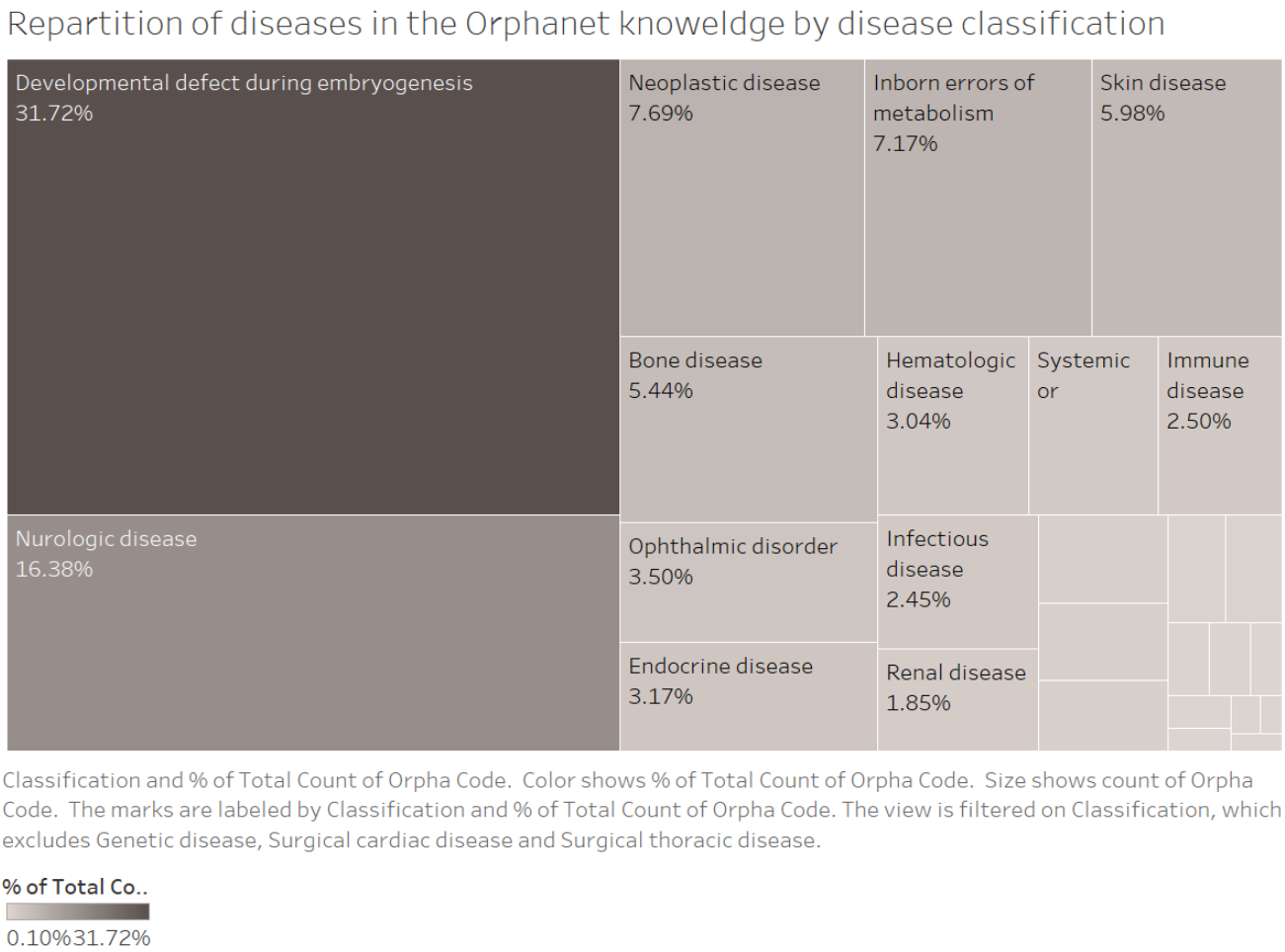

Figure 1.

The classification with the most diseases is the Developmental defect during embryogenesis. This classification is very broad, and includes diseases that could touch many different organs systems (e.g. intellectual disabilities are neurological by nature, and a cardiac malformation is cardiovascular by nature, but falls into this classification because it forms during development). With this information, it makes sense that this would be the largest classification. However, it is surprising that there is not equal repartition of diseases in other specialities. Neurological is the next biggest classification, and then there is a gradual decrease in proportion for all other classifications.

Now we know the repartition of diseases according to classification, let's look at the proportion of diseases per classification that have a definition or clinical annotations. In figure 2, we can see that there is fairly complete coverage of definitions in each classification (i.e. almost every disease has a definitions). In contrast there in variable coverage of clinical annotations; the classifications with the most diseases clinically annotated include Endocrine diseases (77% of diseases covered), Bone disease (71%), Development defect during embryogenesis (67%) and Neurological diseases (63%). Excluding Surgical cardiac disease and Genetic disease (for which there are very few diseases), the least clinically annotated diseases include Neoplastic diseases (24%), Maxillo-facial surgical disease (29%), Disorder due to toxic effects (32%) and Urogenital disease (33%). Based on this graph, I will focus on the developing an eventual model using terms from the most clinically annotated classifications (at least as a first iteration).

Proportions of diseases per classification that have definitions and clinical annotations

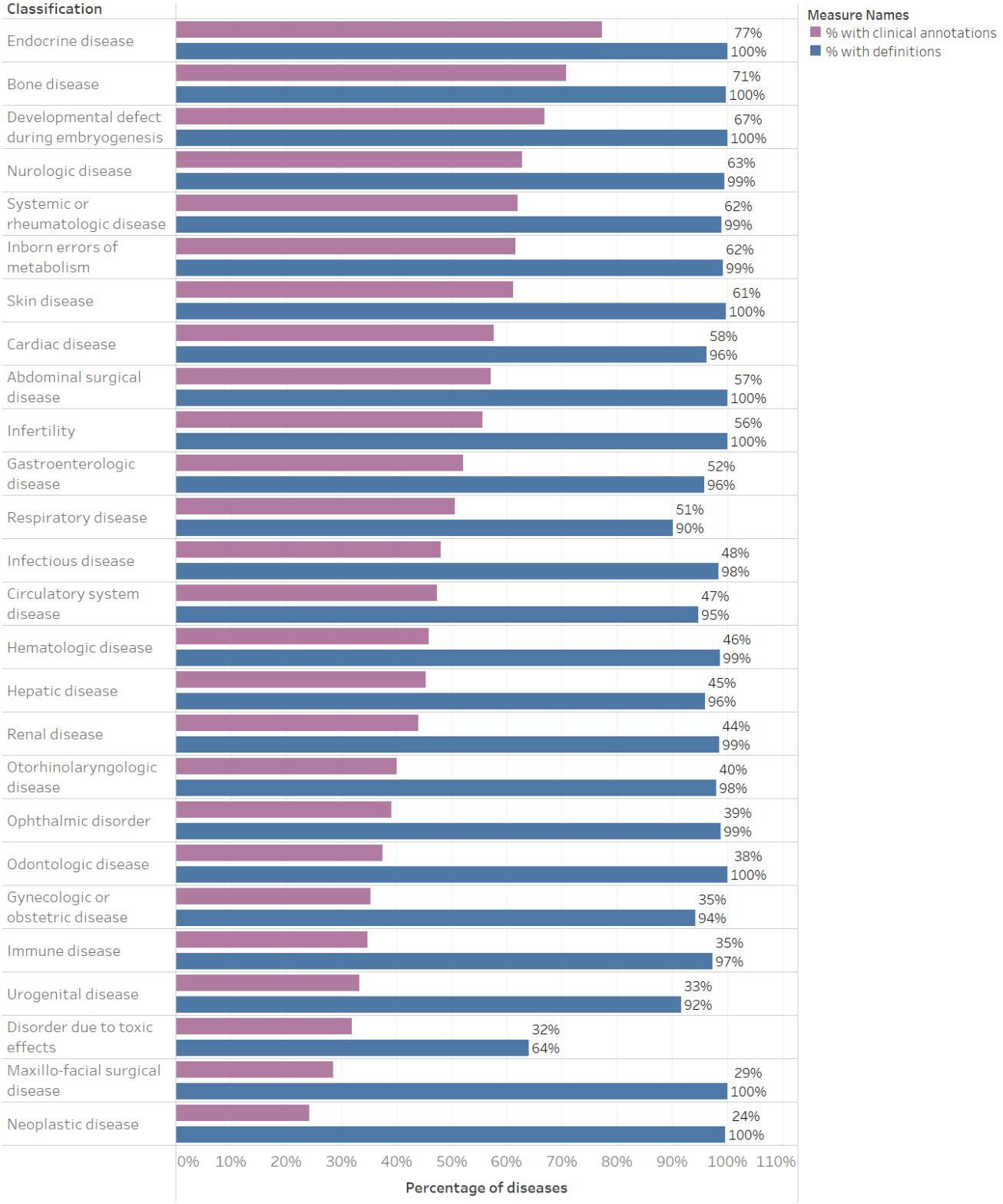

Figure 2

5.8. Orphanet medical terminology

The Orphanet clinical annotations use standardized medical terms called HPO terms. As this project aims to extract these terms from the disease definitions, lets look at the number of distinct HPO terms per classification (figure 3).

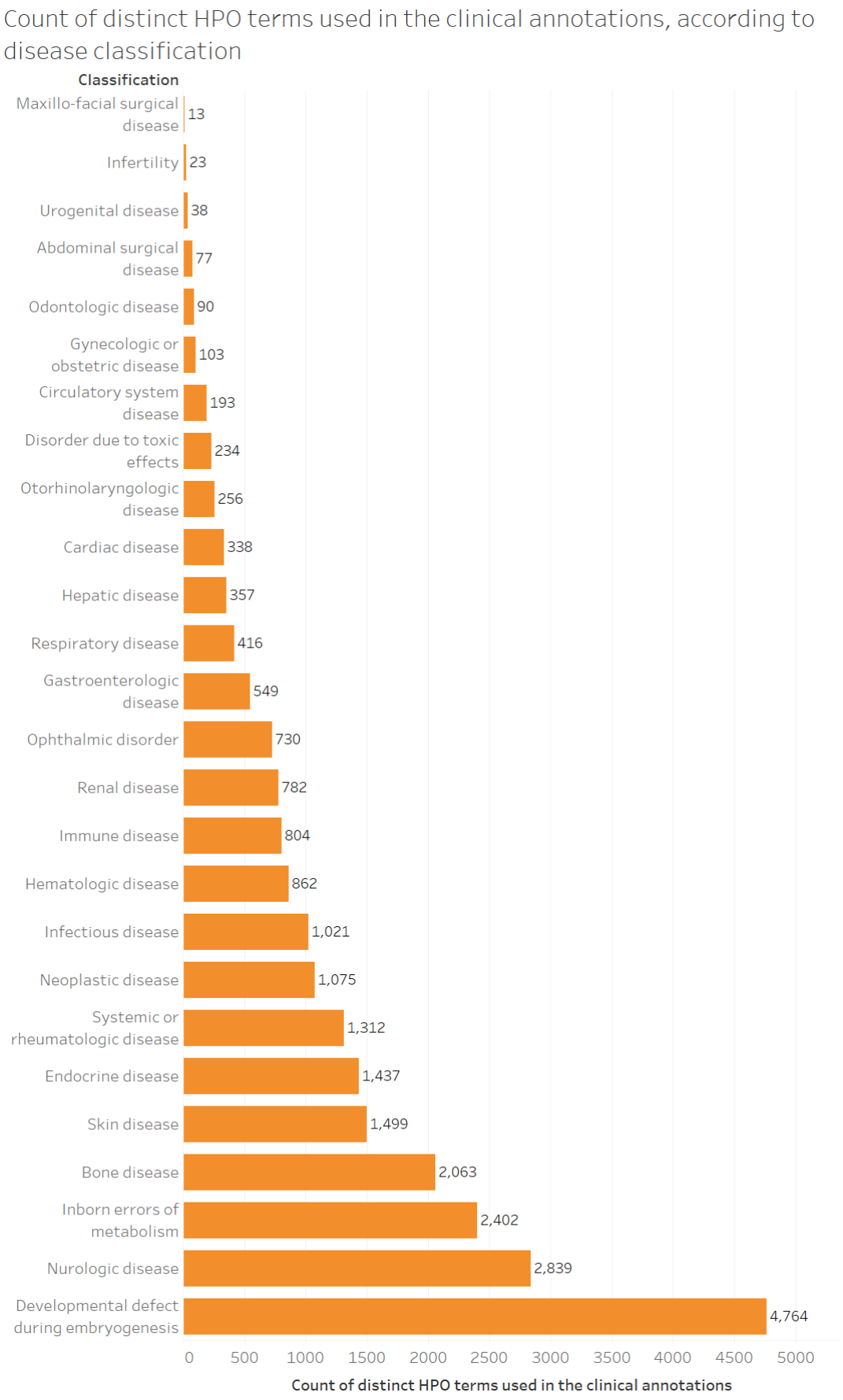

Figure 3.

Unsurprisingly, the classification with the most distinct HPO terms is the Development defect during embryogenesis. This means that there is a wide range of terms used, with almost 5,000 distinct terms, which makes sense given the different medical specialties that can be involved. It also means, that this might be the most challenging for the initial development of a model since there is not much time to pull together the training data. Instead, the diseases from the Bone or Endocrine classification might be a good starting point for the model, as there are fewer unique terms used (2,063 and 1,437, respectively) and have good coverage in terms of the clinical annotations available (see figure 2).

Now let us look to see if certain terms correlate with specific specialities (figure 4). From initial data exploration looking at all the HPO terms used in the clinical annotations, it appeared as if there was no correlation between the terms and specialities. In reality, there were too many terms to see the detail. It is possible that many of the terms do not correlate with specific specialities, but I was certain that certain terms did.

By taking the top 20 most frequent terms for each classification, it is much clearer to see the correlation between the terms and the specific classifications. Figure 4 shows that there are certain terms that correlate specifically with a classification, not appearing in any other classification. Other terms do appear in other classifications but generally not in all other classifications. The same is still true if we look at the top 100 terms (figure 5), although there is obviously more overlap between classifications.

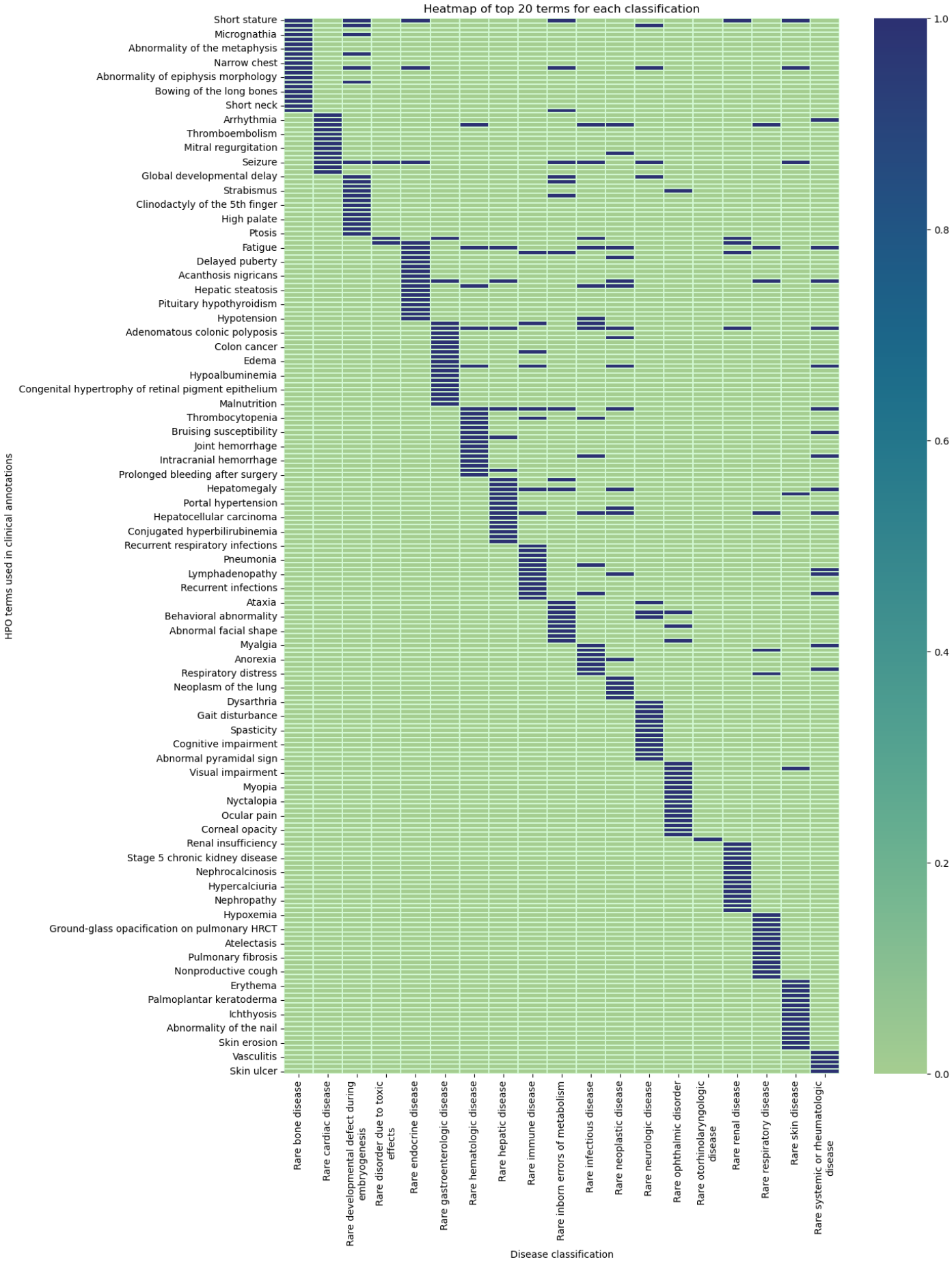

Figure 4. Heatmap showing the correlation between the top 20 terms for each speciality. For reasons of readability, note that not all HPO terms are displayed on the heatmap. Grid lines were added for readability.

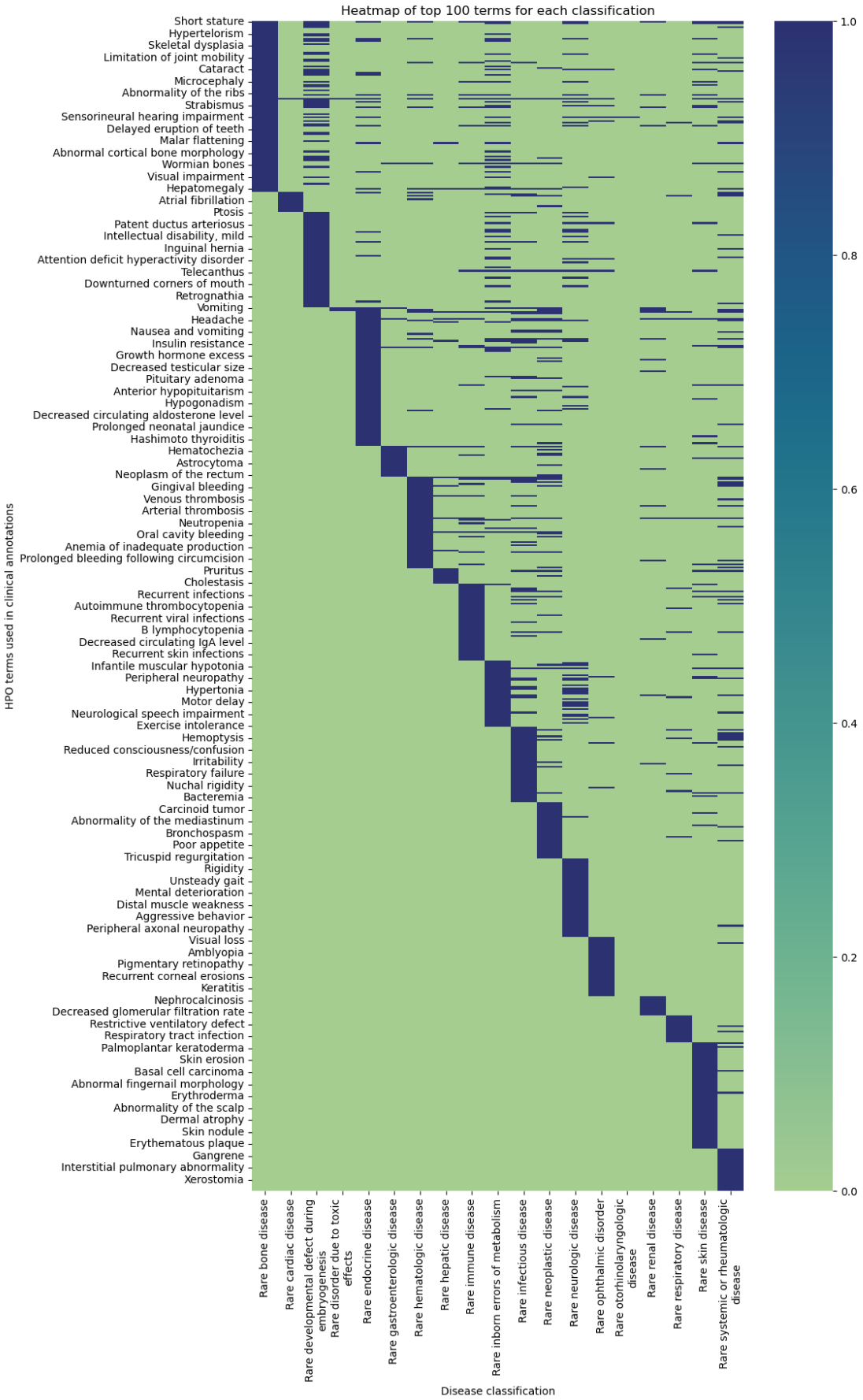

Figure 5. Heatmap for the top 100 terms, showing that the correlation is generalisable. Since there are too many terms to show, gridlines have not been added in this case. Again, not all term names are shown on the heatmap.

5.9. Analysis of the text extraction method

Looking at the number of distinct HPO terms extracted from the definitions versus the number of distinct HPO terms used in the clinical annotations, would suggest that we extracted roughly 25% of all the terms used, as 2000 unique terms were extracted versus the 8000 in the clinical annotations.

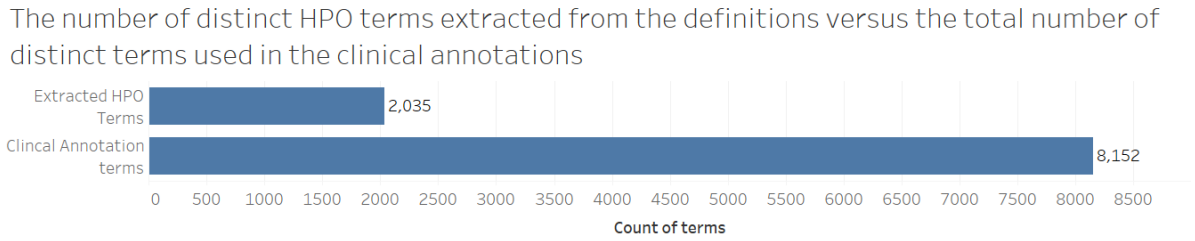

Figure 6

However, if we look at the average number of terms extracted per definition, we get a very different story. From figure 7, we can see that the number of HPO terms extracted per definition is very low compared to the number of terms used to annotate each disease in the clinical annotations. The extracted terms ranged from 1 to 5.25, ignoring the fact that many definitions had 0 terms extracted as these entries were dropped from the data. In contrast, the average number of clinical annotations per disease ranges from 38.22 to 6.5.

To explore this further, I decided to look at the extraction of terms according to the number of words per term, maybe the method is good for single word terms? By comparing the number of terms with 1 - 8 words for the clinical annotations, the extracted raw terms and the mapped HPO terms, we can see the relative proportions of each are the same at each word count. That is for all except 6 to 8 words per HPO term, which starts to plateau for the clinical annotations but continues to decrease for the extracted raw and mapped terms. This could imply that the methods employed here are not good for extracting and mapping terms with 6 to 8 words. For the most part, this data would suggest it is not the term length that is affecting the extraction and mapping.

Average number of HPO terms per definition per classification

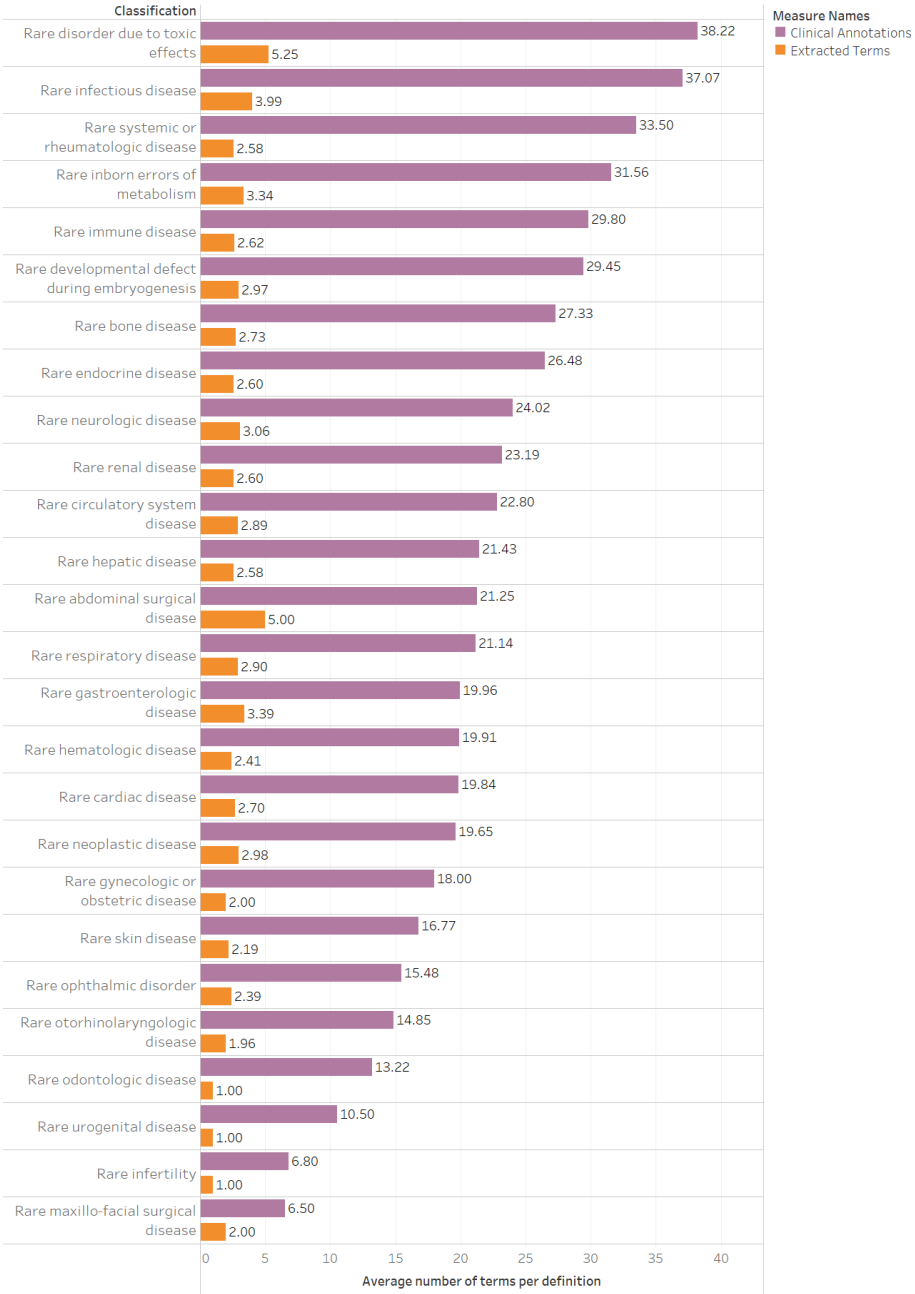

Figure 7

Comparison of the number of words per term (HPO term or extracted term) in the clinical annotations, the raw extracted text and the HPO mapped terms

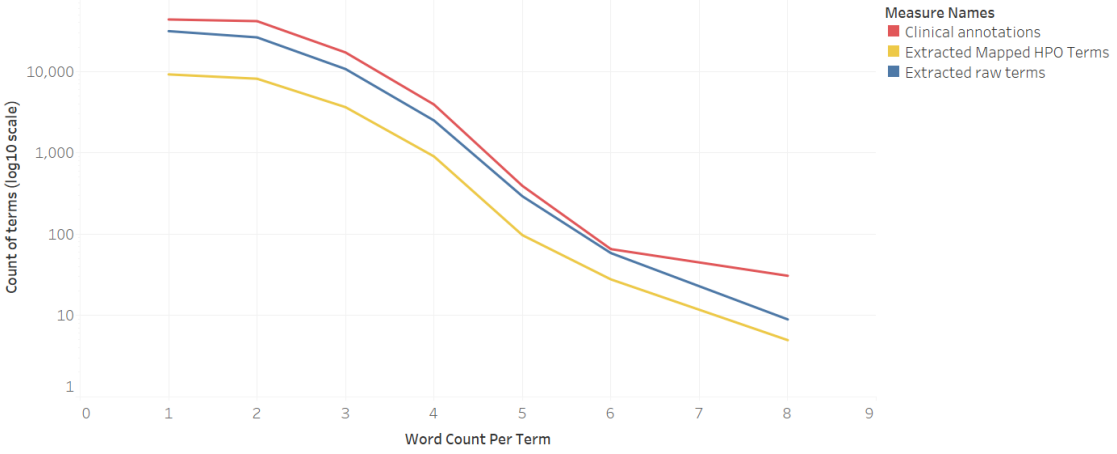

Figure 8

Finally, let us look at the most common words used in the clinical annotations as well as the most common extracted, mapped HPO terms. There is some good overlap between the top terms from both categories (clinical annotations vs extracted terms). From looking at them they appear to mostly come from the Developmental defect during embryogenesis and Neurological classifications, which is consistent with the fact that these are the two biggest classifications. There are, however, a few anomalies in the extracted terms, there are single modifier words, such as 'severe', 'unilateral' and 'bilateral', which are being presented as HPO terms themselves. In reality, these terms would never be used by Orphanet to clinically annotate a disease.

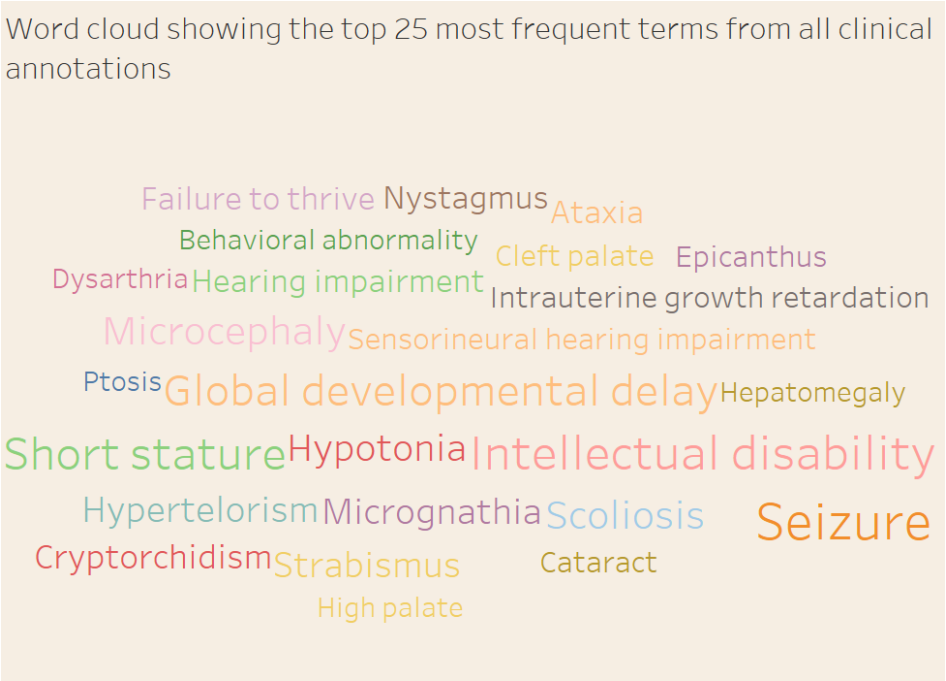

Figure 9. Word cloud for the most popular HPO terms used in the Orphanet clinical annotations

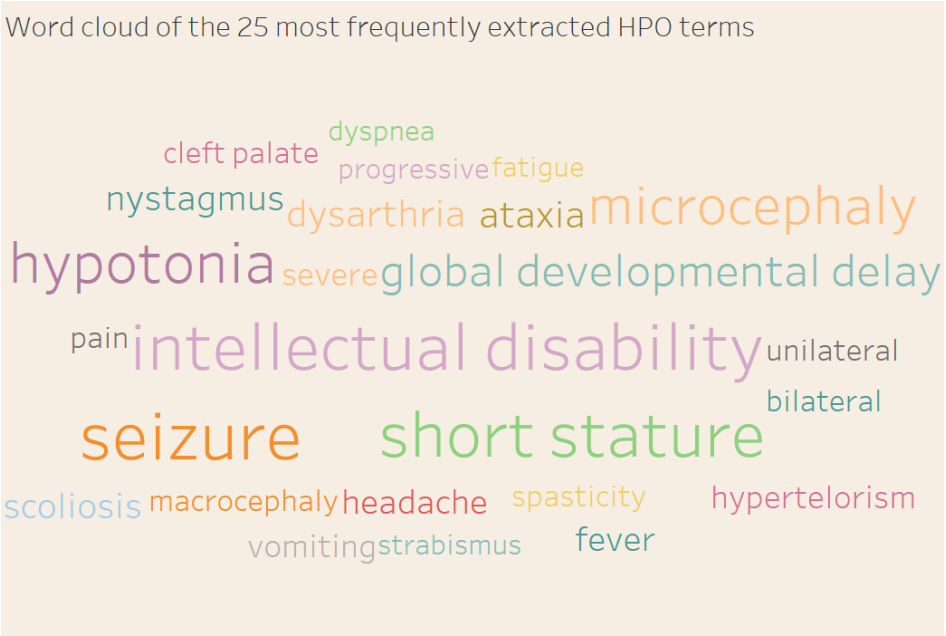

Figure 10. Word cloud for the most frequent HPO mapped terms extracted from the disease definitions.

6. SQL vs no SQL databases

SQL databases have been around since the 70's, and are structured, relational databases. There as been a recent growth in non-SQL databases which can take on many different formats such as graph, key-value, column store and document store. Here I am going to briefly describe the characteristics of each and then explain why I have chosen an SQL database.

SQL databases:

1. Data in SQL databases are organised into structured tables, the structure and constraints of which is determined by the administrator. This ensures that data is consistent, and that new data follows this structure.
2. Relationships can be established between different tables by implementing primary and foreign keys. This means that data does not need to be duplicated as the primary data sources can exploited via their key. This saves space but also provides a level of security for sensitive information as access to this can be restricted whilst the related data can still be exploited.
3. SQL databases use a Structured query language that are standard across different platforms and are very flexible in terms of the queries that can be passed.
4. SQL databases are typically held on a single server, and thus need to be scaled vertically (e.g. adding more computing power) when the computational demands increase.
5. For this above reason, SQL databases can be limiting when there are many users and queries operating at the same time or when large data storage is required.

No-SQL databases:

1. Whilst they are extremely flexible, they are not structured and thus data consistency is not enforced.
2. There are no, or very few, relations between data sets/collections.
3. Data is typically nested or merged in a few collections, with a collection serving a particular purpose. In this way, the data may not be easily exploited for other purposes.
4. Both horizontal and vertical scaling are permitted, this offers great performance particularly when the large data storage is required or when there are lots of users and queries operating at the same time.
5. The languages are platform specific and not necessarily as flexible as in SQL

For this project, MySQL relational database management system, which operates locally on my machine, was used because it is a small-scale project (only one user) and the relationships between tables permits exploitation of the 5 different related tables. The structured query language (SQL) permits flexible queries and is well documented, thus adapted for the purposes of the project.

7. Entity relation diagram

The entity relation diagram for the SQL database set up for this project can be found in figure 11.

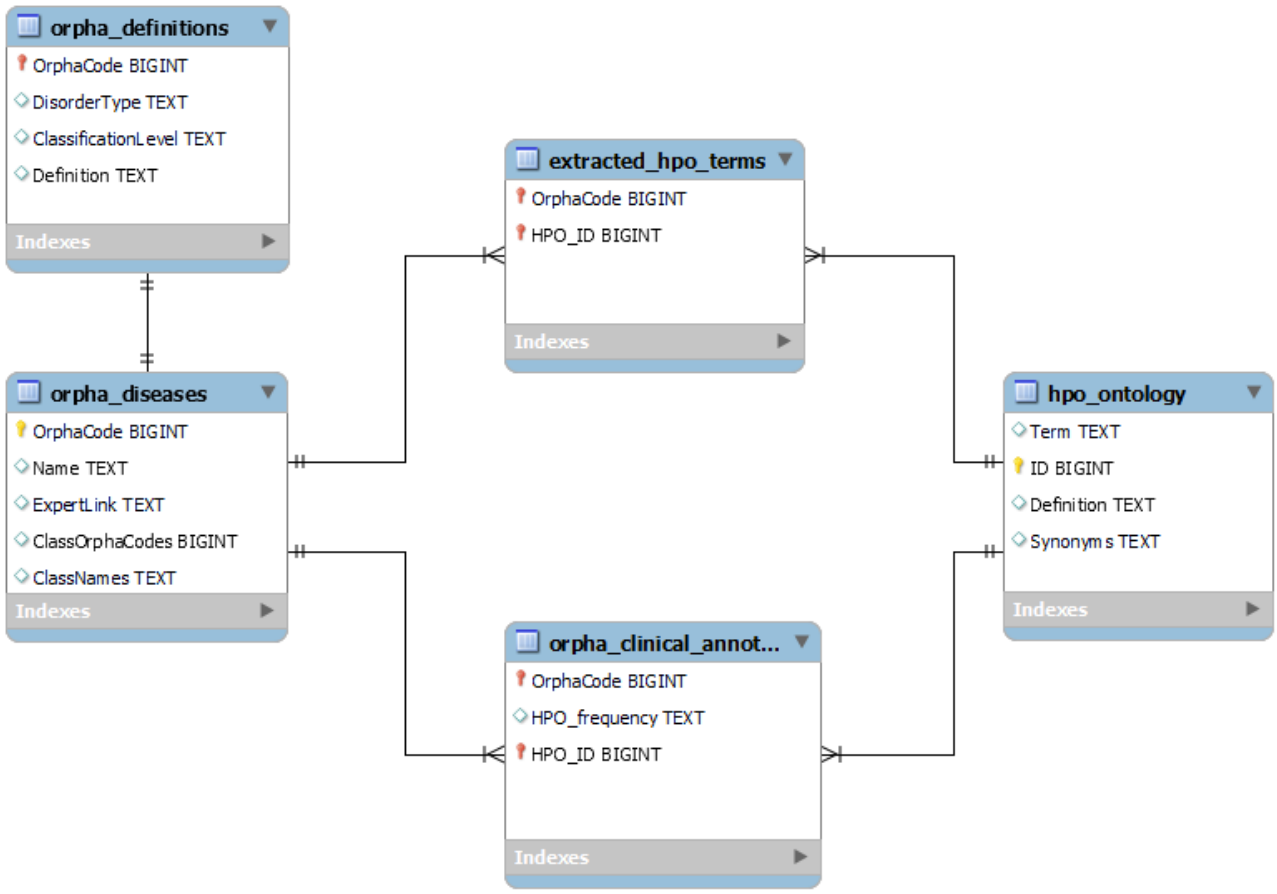

Figure 11

There are five tables in total: the three primary tables consist of the orpha_diseases, the orpha_definitions and the HPO_ontology tables. There are two intermediary tables, the orpha_clinical_annotations and the extracted_HPO_terms, which both contain multiple diseases (identifiable by their OrphaCode) and multiple HPO phenotypes (identifiable via their respective HPO_IDs). In these two intermediary tables, each disease can be listed multiple times, but each disease will be associated with a different HPO_ID. The combination of the Orpha code and HPO ID is unique such that it forms a composite primary key, with each individual code acting as the foreign keys. The relationship between the intermediary tables and primary tables, orpha_disease and HPO_ontology, is one-to-many.

The relationship between the orpha_disease and orpha_definitions table is one-to-one: the maximum number of unique diseases that can be in the definitions table is one, the minimum number is zero.

8. Data import and creation of tables

Since the raw data had been cleaned and wrangled in python, it made sense it export the data directly to MySQL using python. The following libraries were used to do this export: pymysql.cursors, sqlalchemy (create_engine, text) and pandas. In addition, a config file was imported containing the MySql password. The function used to do this is below:

```
def convert_pd_df_tosql(file_name,table_name,schema='orphanet'):
    import config
    pw=config.pw
    connection_string = 'mysql+pymysql://root:' + pw + '@127.0.0.1:3306/'
    engine = create_engine(connection_string)
    df = pd.read_csv(file_name, index_col=False)
    df.to_sql(table_name,engine,schema,index=False, chunksize=5000,if_exists='replace')
    return 'Created table'
```

Whilst, you can specify the data formats specifically for each column, I chose to simply check the formats after importing. Since, the data is either text or integers, the default import options seem to work well.

Of course the table can be created in MySql using the CREATE TABLE commands and then specifying the columns and there input characteristics. However, the above method was much quicker.

The primary and foreign keys were also set using python. This permitted me to easily debug some issues, specifically some OrphaCodes in the definitions table did not exist in the primary diseases table, and HPO terms in the clinical annotations table that were not in the HPO ontology. The incoherent entries were dropped from the respective tables.

The primary keys were set with the following function:

```
#Function to define primary key
def define_PK(table, col_name):
    import config
    pw=config.pw


    connection_string = 'mysql+pymysql://root:' + pw + '@127.0.0.1:3306/orphanet'
    engine = create_engine(connection_string)
    with engine.connect() as con:
        con.execute(f'ALTER TABLE {table} ADD PRIMARY KEY  ({col_name});')
```

Before the foreign keys could be set, the incoherences in the clinical annotations and disease definition data sets (mentioned above) needed to be fixed. The function to do this is below (note that the incoherences were visualised before removing):

```python
def remove_code_anomalies(table_1,table_2, id_1,id_2):

    connection_string = 'mysql+pymysql://root:' + pw + '@127.0.0.1:3306/orphanet'
    engine = create_engine(connection_string)

    # load the clinical_annotations table and orpha disease table into a dataframe
    query = f'SELECT * FROM {table_1}'
    df_1 = pd.read_sql_query(query, engine)

    query = f'SELECT * FROM {table_2}'
    df_2 = pd.read_sql_query(query, engine)

    # get the OrphaCodes that are missing from the orpha_diseases table
    missing_codes = set(df_1[id_1]) - set(df_2[id_2])

    # drop the rows from the clinical_annotations table that have missing OrphaCodes
    df_1 = df_1[~df_1[id_1].isin(missing_codes)]

    #to check that codes removed
    missing_codes = set(df_1[id_1]) - set(df_2[id_2])

    #update the clinical_annotations table in the database
    df_1.to_sql(table_1, engine, if_exists='replace', index=False)

    return df_1[id_1].dtypes, df_2[id_2].dtypes, missing_codes

remove_code_anomalies('orpha_definitions', 'orpha_diseases', 'OrphaCode','OrphaCode')
```

The foreign keys were determined for three tables (clinical annotations, orpha_definitions and extracted_hpo_terms) using the following function:

```python
#Function to define froeign key

def define_FK(table,col_name_fk,  table_pk,col_name_pk):
    import config
    pw=config.pw

    connection_string = 'mysql+pymysql://root:' + pw + '@127.0.0.1:3306/orphanet'
    engine = create_engine(connection_string)
    with engine.connect() as con:
        con.execute(f'ALTER TABLE {table} ADD FOREIGN KEY  ({col_name_fk}) REFERENCES {table_pk}({col_name_pk});')

define_FK('orpha_definitions','OrphaCode','orphanet.orpha_diseases','OrphaCode')
```

9. SQL Queries and insights

9.1. Query 1

Let's look at the repartition of diseases in the Orphanet database by speciality:

```sql
select ClassNames as Speciality, count(Distinct(d.OrphaCode)) as DiseaseCount
from orpha_diseases d
left join orpha_definitions def on d.OrphaCode = def.OrphaCode
group by ClassNames
order by DiseaseCount desc;
```

Speciality	DiseaseCount
Rare developmental defect during embryogenesis	2293
Rare neurologic disease	1184
Rare neoplastic disease	556
Rare inborn errors of metabolism	518
Rare skin disease	432
Rare bone disease	393
Rare ophthalmic disorder	253
Rare endocrine disease	229
Rare hematologic disease	220
Rare systemic or rheumatologic disease	187
Rare immune disease	181
Rare infectious disease	177
Rare renal disease	134
Rare gastroenterologic disease	94
Rare respiratory disease	81
Rare hepatic disease	75
Rare cardiac disease	52
Rare otorhinolaryngologic disease	50

Whilst I would expect the most frequent classification to be Rare developmental defect during embryogenesis, as this covers all specialities from intellectual disability to physical malformations, I think it's surprising to see that there is not and equal spread between all other specialities. However, the rare diseases are classed into multiple classifications (secondary classifications) depending on the medical specialities they fall into, so this does not mean that these other systems aren't as involved (e.g. a syndromic disease could have neurology as the primary speciality but bone and endocrine systems could also be involved).

For this project, the results of this query gives me an idea for which speciality to tackle for the modelling part of the project, as there will not be time to address all terms for all specialities in this week long project.

9.2. Query 2

Query to look at the number of definitions by disorder type.

```
4 •   select DisorderType, count(DisorderType) as count from orpha_definitions
5     group by DisorderType with rollup;
```

DisorderType	count
Clinical subtype	771
Clinical syndrome	45
Disease	3950
Etiological subtype	193
Histopathological subtype	50
Malformation syndrome	1748
Morphological anomaly	414
NULL	7171

It is no surprise that the highest number of definitions is for the disease level, followed by malformation syndromes and then the subtype level (clinical, etiological and histopathological). Clinical syndromes are very particular, and not very frequent in the Orphanet database, so it is not unexpected that there are only 45. In total, there are 7171 definitions in the database.

9.3. Query 3

Query to look the number of clinical annotations by disease type:

```
select COALESCE (ClassificationLevel, 'TOTAL') as classification, count(Distinct(c.OrphaCode)) as disease_count
from clinical_annotations c
left join orpha_definitions d on c.OrphaCode=d.OrphaCode
group by ClassificationLevel with rollup;
select * from clinical_annotations c
left join orpha_definitions d on c.OrphaCode=d.OrphaCode
where ClassificationLevel is null;
```

classification	disease_count
Disorder	3832
Subtype of disorder	365
TOTAL	4197

In contrast to the query 2, there are fewer different disorder types that have clinical annotations, specifically they are restricted to Disorder and Subtype of disorder. This means that the malformation syndromes are missing clinical annotations, which is something that could be addressed the Orphanet team responsible. In terms of this project, this means we can discard the morphological syndromes and the morphological anomalies, as well as any HPO terms specifically associated to these types of disorders. In total, there are clinical annotations for 4197 diseases.

9.4. Query 4

For each diseases, let's look at the number of HPO terms extracted from the definitions, the number of these terms that match the clinical annotations for each disease, and the total number of HPO terms in clinical annotations for each disease:

```
select
    e1.OrphaCode, count(e1.HPO_ID) as Total_extracted_terms,
    subq.Matched_Extracted_Terms as Matched_Extracted_Terms,
    subq.Total_Clinical_Annotations as Total_Clinical_Annotations
from  extracted_hpo_terms e1
inner join          -- query to count the number of matched terms, against total number of very frequent clinical
    (select e.OrphaCode,  count(e.HPO_ID) as Matched_Extracted_Terms,
            sub_clin.Total_Clinical_Annotations as Total_Clinical_Annotations
        from extracted_hpo_terms e
            inner join clinical_annotations c
            ON e.OrphaCode = c.OrphaCode AND e.HPO_ID = c.HPO_ID AND c.HPO_frequency = 'Very frequent (99-80%)'
            inner join  -- query to count the number of very frequent annotations
                (select OrphaCode, count(distinct(HPO_ID)) as Total_Clinical_Annotations
                from clinical_annotations
                where HPO_frequency = 'Very frequent (99-80%)'
                group by OrphaCode) AS sub_clin
            on e.OrphaCode = sub_clin.OrphaCode
        group by e.OrphaCode, sub_clin.Total_Clinical_Annotations) AS subq
        ON e1.OrphaCode = subq.OrphaCode
group by e1.OrphaCode, subq.Matched_Extracted_Terms, subq.Total_Clinical_Annotations;
```

OrphaCode	Total_extracted_terms	Matched_Extracted_Terms	Total_Clinical_Annotations
61	2	1	42
812	2	2	42
584	2	2	26
881	1	1	111
126	4	2	9
14	3	1	63
1716	5	1	30
2773	2	2	8
236	2	1	27
1065	2	1	11
147	5	1	7
1538	3	2	11
1488	8	6	26
1369	3	2	10
1770	1	1	26
2233	2	2	13
2135	7	6	52
351	2	1	9

...

The table above is a small extract of the results, as there are thousands of diseases. Looking through the results table, the general trend is that very few terms were extracted from the definitions in general. In contrast, there seems to be a good proportion of the extracted terms that matched the terms in the clinical annotations for each disease.

9.5. Query 5

Following on from the above query, lets look at the average number of HPO terms per disease in the clinical annotations and the extracted terms:

This query shows there are on average 26 terms in the clinical annotations per disease. In contrast, only 3 extracted-mapped HPO were retrieved per definition.

```
81  select Avg_Extracted_Term_per_Def, Avg_Clinical_Annotations_per_Def
82  from
83      (select sum(HPO_count)/count(Extracted_Orpha_Codes) as Avg_Extracted_Term_per_Def
84      from
85          (select e.OrphaCode as Extracted_Orpha_Codes,
86          count(e.HPO_ID) as HPO_count
87          from extracted_hpo_terms e
88          group by e.OrphaCode) as subq1) as subq2
89      cross join
90          (select sum(HPO_count_2)/count(Clinical_Orpha_Codes) as Avg_Clinical_Annotations_per_Def
91          from
92              (select c.OrphaCode as Clinical_Orpha_codes,
93              count(c.HPO_ID) as HPO_count_2
94              from clinical_annotations c
95              group by c.OrphaCode) as subq3) as subq4;
96
```

Avg_Extracted_Term_per_Def	Avg_Clinical_Annotations_per_Def
2.9119	26.1982

10. Conclusions

Unfortunately, the methods employed here to extract the clinical terms from the disease definitions and subsequently map them to the standardised HPO terms, was not sufficient to create an indicator of quality for the definitions. This is because the text extraction only returned a few HPO terms per definitions and on average a disease should have 26 HPO terms, according to the clinical annotations.

Whilst this could be due to the definitions not containing the necessary terms, I think that this is highly unlikely and rather a reflection of the methods employed. In addition, the ratio threshold was set to minimum 90 for the mapping, and from a total of almost 56,000 terms entered only 12,000 were mapped to HPO terms (of which only 2000 were unique HPO terms).

My conclusion is therefore that a model is needed in order to better map the HPO terms to the extracted terms from the definition. If there is time, improvement to the initial term extraction using Named entity recognition could also be employed to get better quality input data.

11. Annexe

11.1. Project plan:

The project plan was created in Trello and can be accessed here.

11.2. Github link

https://github.com/gmilman/Orphanet