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

By Gemma Milman

March 2023

Table of contents

01 Introduction

Orphanet: a knowledge base for rare diseases

What is a rare disease?

- Less than 1 in 2000 people
- Cover all medical specialities
- Rare disease example: cystic fibrosis

What is orphanet?

- Defines rare disease nomenclature and classification
- Collection of data for each and every rare disease

Orpha nomencalture

What is a the orpha nomenclature?

- Standardised naming convention
- Permits coding of rare disease patients

Consists of :

- Disease name
- Orpha code
- Definition
- Extact synonyms

Disease name	**Cystic fibrosis**
Orpha code	ORPHA:586
Disease definition	A rare, genetic pulmonary disorder characterized by sweat, thick mucus secretions causing multisystem disease, chronic infections of the lungs, bulky diarrhea and short stature.
Synonyms	CF Mucoviscidosis

Orpha clinical annotations

Clinical signs and symptoms
Very frequent
Absent vas deferens HP:0012873
Airway obstruction HP:0006536
Bronchiectasis HP:0002110
Elevated sweat chloride HP:0012236
Exocrine pancreatic insufficiency HP:0001738
Malabsorption HP:0002024
Recurrent respiratory infections HP:0002205

What is a the orpha clinical annotations?

- List of clinical signs or symptoms that occur in a disease
- Categorized by frequency
- Curated manually by an MD, based on medical literature
- Uses HPO terms, standardised medical terms

human phenotype ontology

All

cardiac anomaly

Abnormal heart morphology HP:0001627

Any structural anomaly of the heart.

Synonyms: *Abnormality of the heart, Heart defect, Abnormally anomalies, Cardiac anomaly, Congenital heart defect, Congenital*

Cross References: *MSH:D006330, SNOMEDCT_US:13213009,*

No. Descendants | Hierarchy

- Abnormality of cardiovascular system morphology
  - **Abnormal heart morphology**
    - Abnormal heart valve morphology
    - Abnormal cardiac ventricle morphology
    - Abnormal myocardium

02

Business case & objective

Business case & objective

Commitment to quality

- Data is manually curated
- Pre- and post release quality assurances in place
- But not for the definitions...

Objective

- Perform a quality control on the disease definitions
- Comparing the defining clinical terms with the clinical annotations

The process

Disease definition

A rare severe, X-linked, neurodevelopmental disorder characterized by rapid developmental regression in infancy, partial or complete loss of purposeful hand movements, loss of speech, gait abnormalities, and stereotypic hand movements, commonly associated with deceleration of head growth, severe intellectual disability, seizures, and breathing abnormalities. The disorder has a progressive clinical course and may associate various comorbidities including gastrointestinal diseases, scoliosis, and behavioral disorders.

HPO mapping

Clinical signs and symptoms

Very frequent

- Abnormal pattern of respiration HP:0002793
- Absent speech HP:0001344
- Developmental regression HP:0002376
- Gait disturbance HP:0001288
- Global developmental delay HP:0001263
- High-pitched cry HP:0025430
- Progressive microcephaly HP:0000253
- Stereotypical hand wringing HP:0012171
- Stereotypy HP:0000733

03

Data collection & processing

Data collection

Orphanet	Text mining	Human phenotye project
• Orphanet diseases • Orphanet definitions • Clinical annotations	• Extraction of the clinical terms from the Orphanet disease definitions	• HPO ontology
Data format: XML files Size: (7,241, 6 columns), (10,675 rows, 8 cols), (112,689 rows, 5 cols)	Imported XML data manipulated in python	Data format: obo file Size: Size: 16,874 rows, 5 columns

API available for both websites, but no access to Orphanet API and the HPO API not really adapted for the purposes of this project

Import XML files via Elmtree

```python
def import_clinical_annotations():

    tree = ET.parse(r'C:\Users\gemma\Documents\IronHack\FINAL PROJECT\raw_data\en_signs.xml')
    root = tree.getroot()
    data = []

    # loop over the <Disorder> elements and extract the values
    for disorder in root.findall('.//Disorder'):
        orpha_code = disorder.find('OrphaCode').text
        name = disorder.find('Name').text
        hpo_list = disorder.findall('.//HPO')
        for hpo_elem in hpo_list:
            hpo_id = hpo_elem.find('HPOId').text
            hpo_term = hpo_elem.find('HPOTerm').text
            hpo_frequency = disorder.find('.//HPOFrequency/Name').text
            row = {'OrphaCode': orpha_code, 'Name': name, 'HPO_id': hpo_id, 'HPO_term': hpo_term,
                   'HPO_frequency': hpo_frequency}
            data.append(row)

    # create a DataFrame from the data
    df = pd.DataFrame(data)
    df.to_csv('processed_data\clinical_annotations.csv', index=False)
    display(df.head())
    return df
import_clinical_annotations()
```

Import obo file via pyobo

```
obo_path =  r'C:\Users\gemma\Documents\IronHack\FINAL PROJECT\raw_data\HPO.OBO'
graph=pyobo.from_obo_path(obo_path, prefix='HP')

# Empty lists for the terms and their attributes
terms = []
ids = []
definitions = []
synonyms = []

# Iterate over each term in the OBO file
for term in graph.iter_terms():
    terms.append(term.name)  # append term name
    ids.append(term.identifier) # append HPO ID
    definitions.append(term.definition) # append definitin
    # Make seperate list for synonymse, then append list above    synonym_list = []
    for syn in term.synonyms:
        if syn.specificity== 'EXACT':
            synonym_list.append(syn.name)
    synonyms.append(', '.join(synonym_list))

# export list to pd dataframe
df = pd.DataFrame({'Term': terms, 'ID': ids, 'Definition': definitions, 'Synonyms': synonyms})
```

Data cleaning

Null values

- Non in Orphanet data
- Definitions & synonyms of HPO – not critical

Kept

Duplicates

- 66 in the clinical annotations
- 34 in the HPO ontology

Drop

Irrelevant data

- Nomenclature Categories: category, particular clinical situations, biological anomaly
- Non-rare entities

Structural fixes

- Format of HPO ID in HPO and Clinical annotations : HPO:0000256 – » 256

Fixed

Text extraction

Disease definition

A rare severe, X-linked, neurodevelopmental disorder characterized by rapid developmental regression in infancy, partial or complete loss of purposeful hand movements, loss of speech, gait abnormalities, and stereotypic hand movements, commonly associated with deceleration of head growth, severe intellectual disability, seizures, and breathing abnormalities. The disorder has a progressive clinical course and may associate various comorbidities including gastrointestinal diseases, scoliosis, and behavioral disorders.

Expectation

- Use named entity recognition via pretrained models for clinical text

Options explored:

- ClinicalBERT
- MedCAT
- SciSpacy
- MetaMap

Due to time constrains could not implement one of these models

Text extraction: Plan B

- Use regex pattern recognition to cut up texts
- Clean and lemmatize strings
- Map the terms to the corresponding HPO term using Python module Fuzzy wuzzy

```
def extract_clinical_terms(df_codes,df_text, lemmatize=False):

    stop_words=list(STOP_WORDS) # from nltk library

    token_split=[]
    token_clean=[]
    token_list_clean=[]
    orpha_tokens={}
    lmtzr = WordNetLemmatizer() # from nltk library

    if lemmatize == True:
        for orpha, row in zip(df_codes,df_text):
            pattern=',|\s+and+\s|includ(?:es|e|ing)\
            |associat(?:es|ed|e|ing)|\.|show(?:s|ing)|marked|\
            resulting|\s+or+\s|present(?:\s|s|ing)|with onset|\
            such as|linked to|combined with|with worsening|with\s'
            row= re.sub(r'\([^)]*\)', '',row) # remove text in ()
            token_list=re.split(pattern, row)
            # strip
            token_list=[token.strip() for token in token_list]

            #remove stop words & lemmatize to improve search and mathcing later
            lemmatized = [[lmtzr.lemmatize(word) for word in \
                           word_tokenize(w) if word not in stop_words]\
                          for w in token_list]
            token_clean=[' '.join(lem) for lem in lemmatized]
            token_clean = [ele for ele in token_clean if ele.strip()]
            token_list_clean.append(token_clean)

            orpha_tokens[orpha]=token_clean #dictionary:orpha code +
extraction...
    else:... # code withou lemmatization
    return orpha_tokens
```

Mapping to HPO terms

- Fuzzy wuzzy gives a similarity ratio between two strings
- Assessing the output, cut off of 90 determined
- **11,915 terms mapped** (out of total 53,000 extracted terms)

Extracted term	Matched term
exaggerated lumbar lordosis	[('exaggerated startle response', 65), ('lumbar hyperlordosis', 64)]
severely impaired color discrimination	[('impaired two-point discrimination', 73), ('abnormal speech discrimination', 62)]
cardiac anomalies	[('cardiac sarcoma', 75), ('cardiac hemangioma', 74)]
midface hypoplasia	[('hemifacial hypoplasia', 82), ('biceps hypoplasia', 80)]
low visual acuity	[('very low visual acuity', 87), ('reduced visual acuity', 74)]
immune deficiency	[('immunodeficiency', 91)
nystagmus	[('nystagmus', 100), ('rotary nystagmus', 72)]

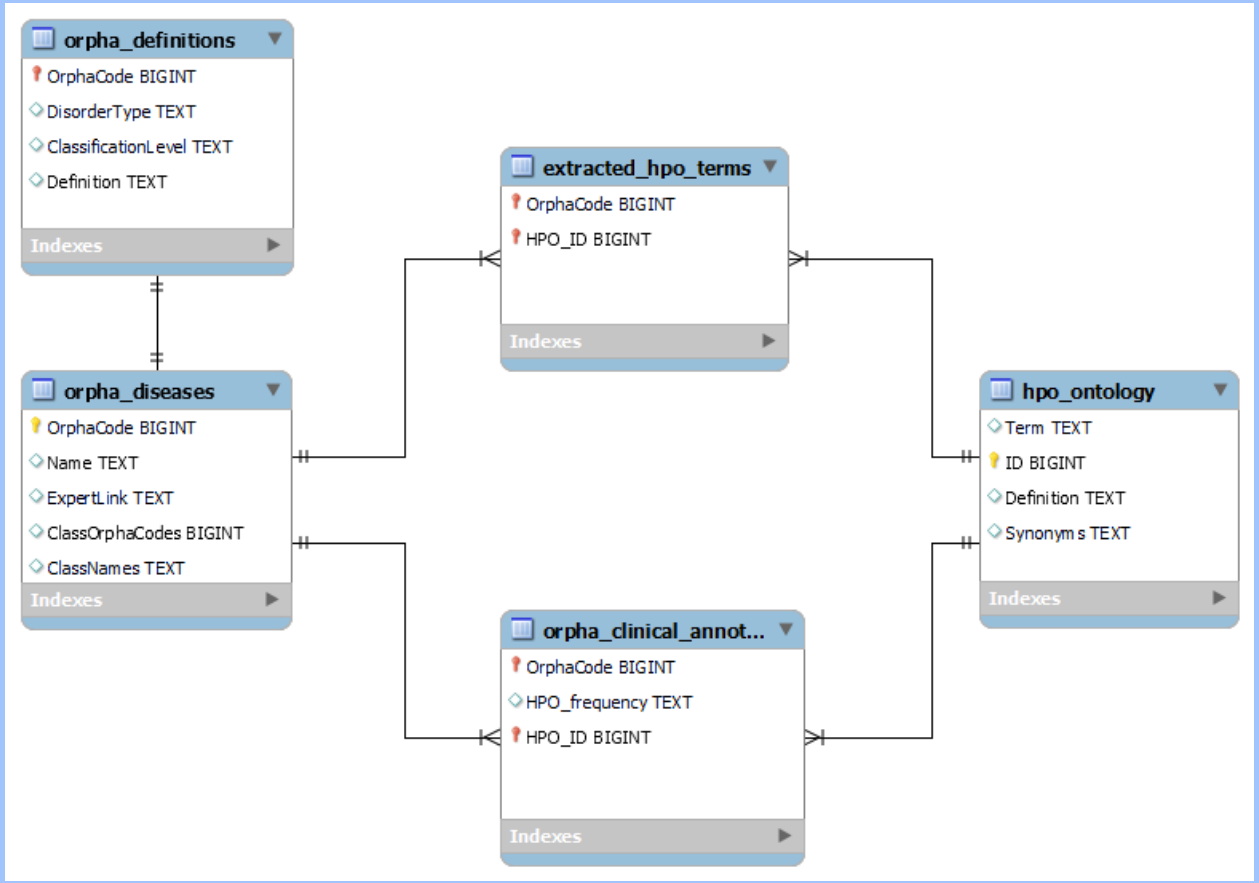
04 SQL database

Chosen database

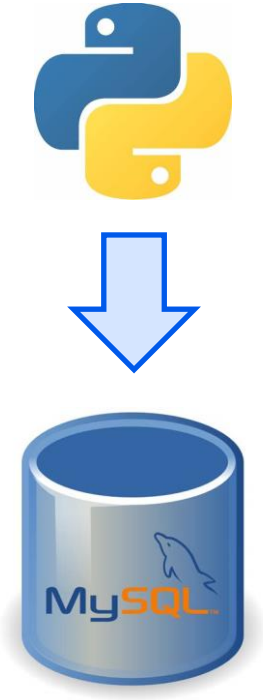
SQL database used for this project

- Small project with one user
- Exploit relationships between tables
- Knowledge of SQL
- Structured data

Entity relationship diagram



Importing to SQL



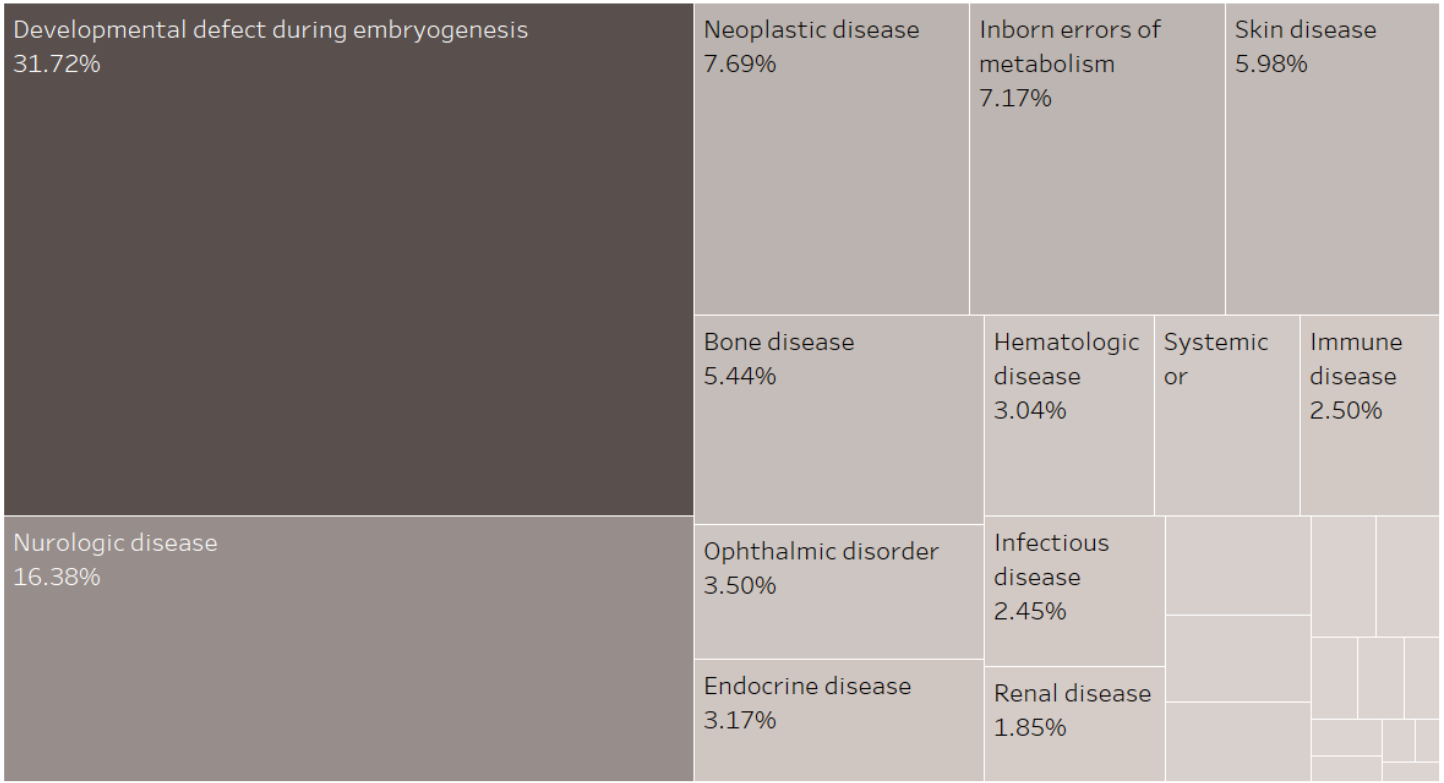

```
def
convert_pd_df_tosql(file_name,table_name,schema='orphanet'):
    import config
    pw=config.pw
    connection_string = 'mysql+pymysql://root:' + pw +\
    '@127.0.0.1:3306/'
    engine = create_engine(connection_string)
    df = pd.read_csv(file_name, index_col=False)
    df.to_sql(table_name,engine,schema,index=False,\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

Define primary, composite primary & foreign keys

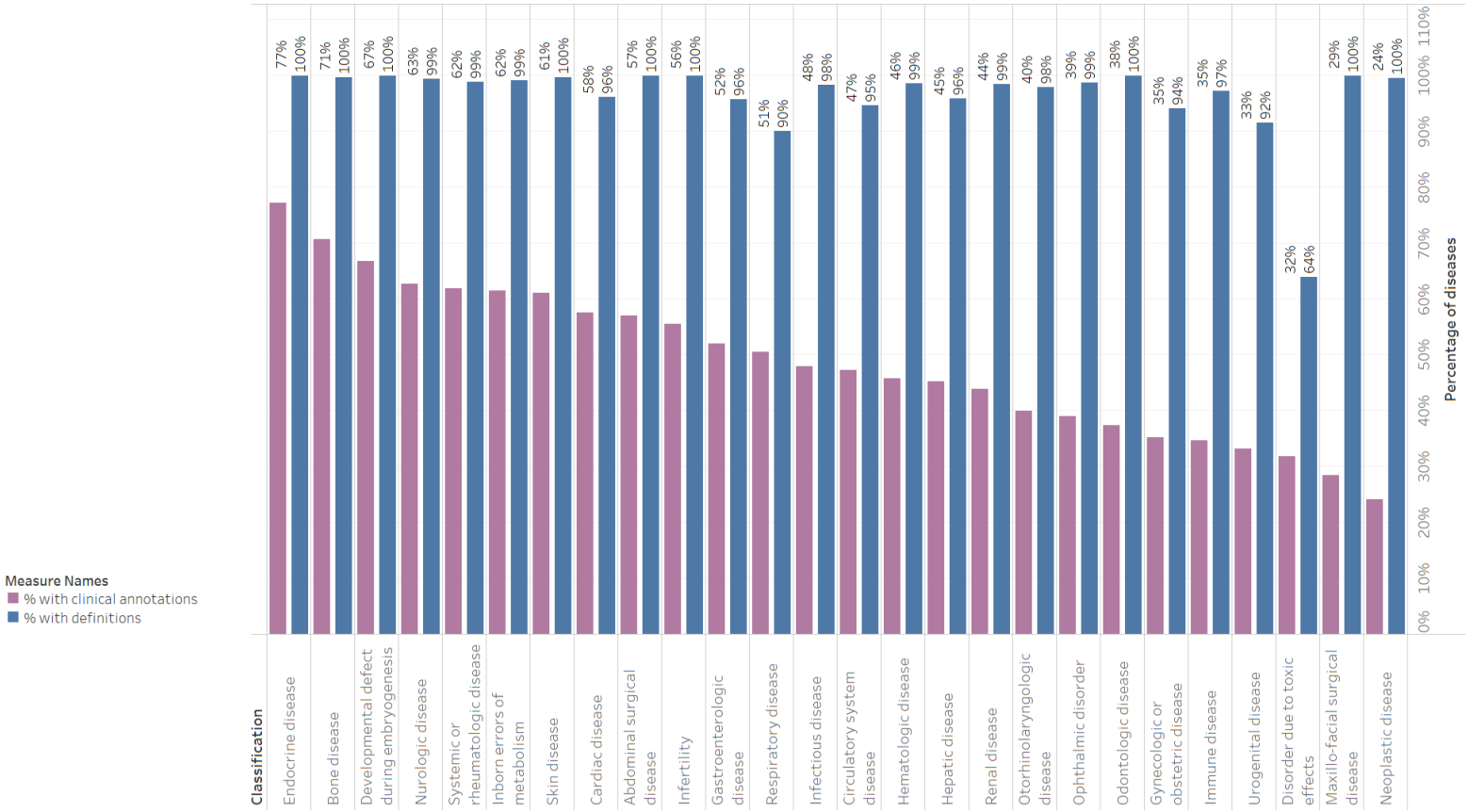
```
def define_PK(table, col_name):

    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});')
    ...
    with engine.connect() as con:
        #con.execute(f'ALTER TABLE {table} DROP PRIMARY KEY;')
        con.execute(f'ALTER TABLE {table} ADD PRIMARY KEY  ({col_name},{col_name_2});')
    ...
    with engine.connect() as con:
        con.execute(f'ALTER TABLE {table} ADD FOREIGN KEY  ({col_name_fk}) REFERENCES
{table_pk}({col_name_pk});')
```

05 Exploratory data analysis

Repartion of diseases by Orphanet classifications



Proportion of diseases that have definitions or clinical annotations

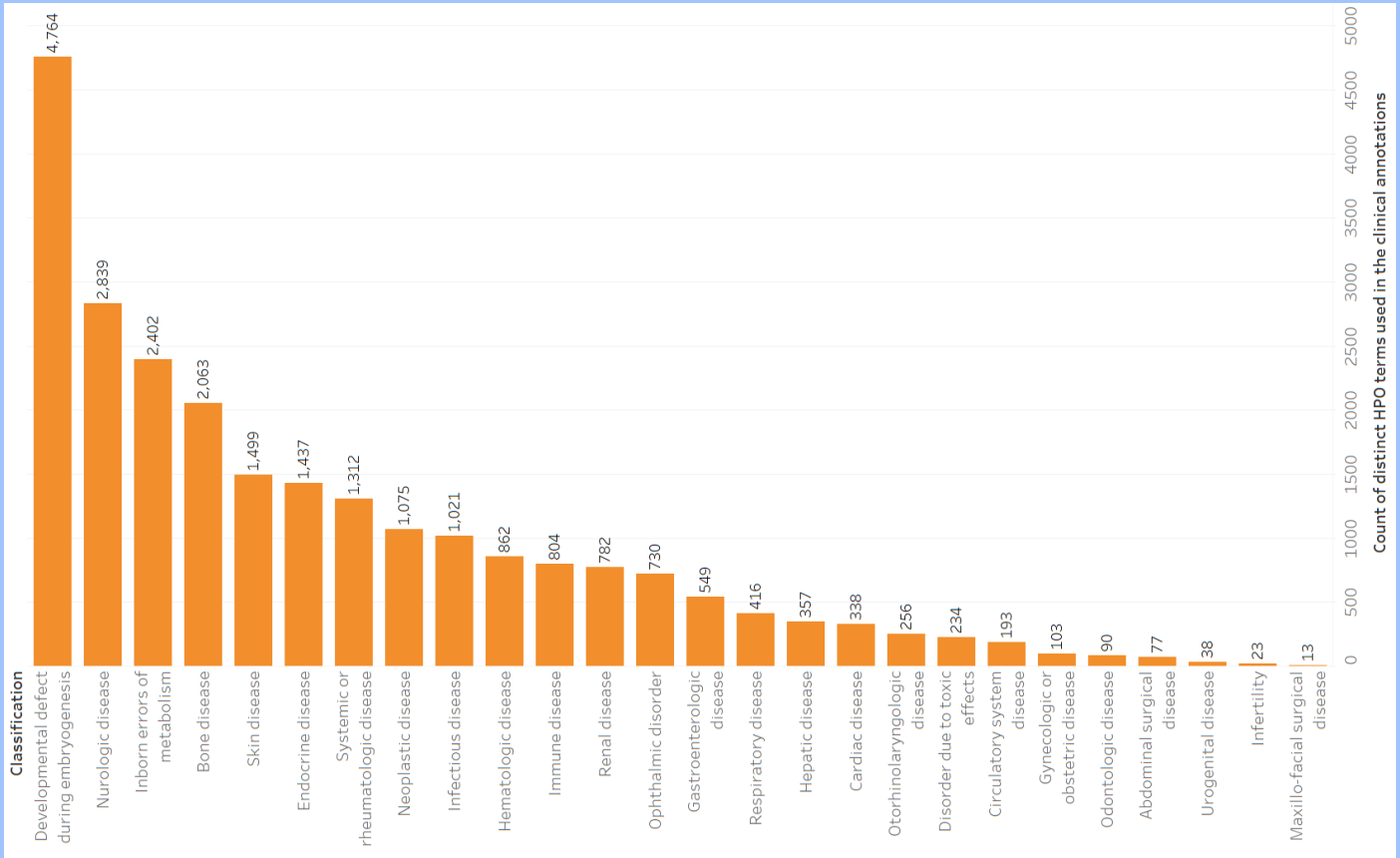


SQL Query: number of unique HPO terms used by the clinical annotations by classification

```sql
select ClassNames as Classification,
    Count_HPO
from
    (select
        ClassNames,
        count(distinct(HPO_ID)) as Count_HPO
    from clinical_annotations c
    left join orpha_diseases d
        on c.OrphaCode = d.OrphaCode
    group by ClassNames) as subQ;
```

Classification	count
Rare developmental defect during embryogenesis	4839
Rare neurologic disease	2875
Rare inborn errors of metabolism	2424
Rare bone disease	2084
Rare skin disease	1517
Rare endocrine disease	1464
Rare systemic or rheumatologic disease	1329
Rare neoplastic disease	1084
Rare infectious disease	1027
Rare hematologic disease	866
Rare immune disease	807
…	…

Distinct HPO terms per classification for the clinical annotations

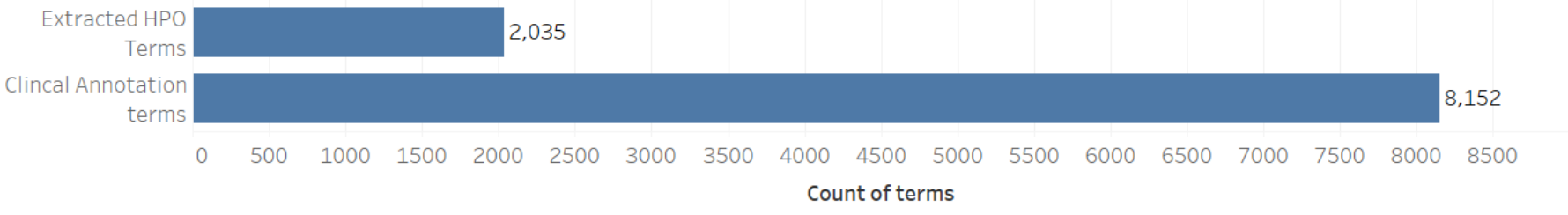


Query: Total HPO terms & percentage of all HPO terms used by clinical annotations and extracted terms

```
select
    count(distinct(c.HPO_ID)) as Clinical_Annotations_Total_terms,
    count(distinct(e.HPO_ID)) as Extracted_Terms_Total,
    count(distinct(ID)) as Total_HPO_Terms,
    count(distinct(c.HPO_ID)) / count(distinct(ID))  * 100
        as  Clinical_annotations_as_percentage_Total_HPO,
    count(distinct(e.HPO_ID)) / count(distinct(ID))  * 100
        as Extracted_Terms_as_percentage_Total_HPO,
    count(distinct(e.HPO_ID)) / count(distinct(c.HPO_ID))  * 100
        as Extracted_Terms_as_percentage_Clinical_Annotations
from clinical_annotations c
right join hpo_ontology h on  c.HPO_ID=ID
left join extracted_hpo_terms e on e.HPO_ID=ID;
```

Clinical_Annotations_Total_terms	Extracted_Terms_Total	Total_HPO_Terms	Clinical_annotations_as_percentage_Total_HPO	Extracted_Terms_as_percentage_Total_HPO	Extracted_Terms_as_percentage_Clinical_Annotations
8152	2035	16873	48.3139	12.0607	24.9632

Analysis of the text extraction methods



8,125 distinct HPO terms used in the clinical annotations

2,035 distinct terms extracted using fuzzy wuzzy

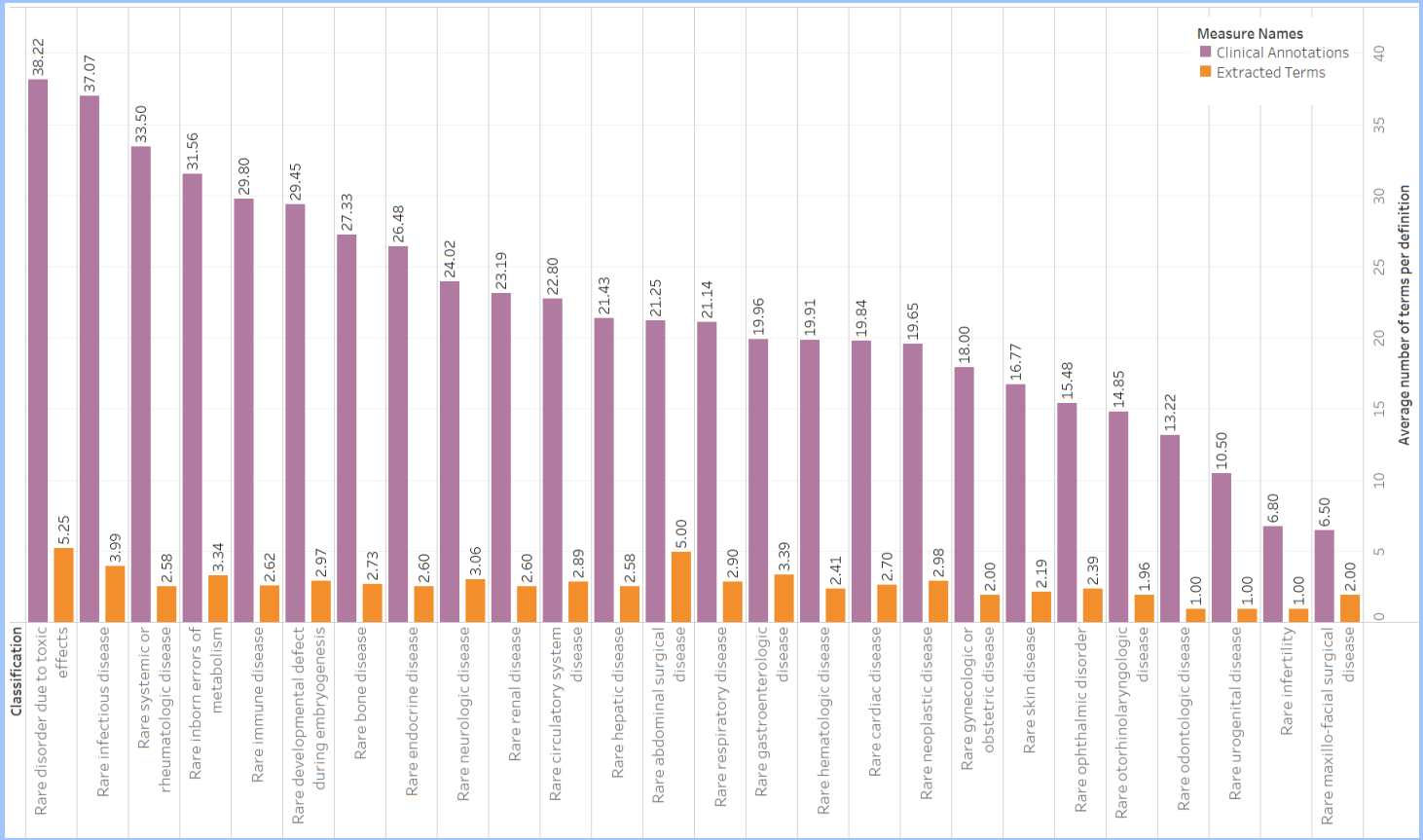
Extraction rate: Approximately 25%

Query: proportion of extracted hpo terms for each orpha code, that match the very frequent HPO terms for the same disease in the Orphanet clinical annotations

```
select
    subq2.ClassNames,
    Avg_Extracted_Term_per_Def ,
    Avg_Clinical_Annotations_per_Def
from
    (select
        ClassNames,
        sum(HPO_count)/count(Extracted_Orpha_Codes)
                        as Avg_Extracted_Term_per_Def
    from
        (select
            e.OrphaCode as Extracted_Orpha_Codes,
            count(e.HPO_ID) as HPO_count,
            ClassNames
        from extracted_hpo_terms e
        join orpha_diseases d on e.OrphaCode=d.OrphaCode
        group by e.OrphaCode, ClassNames) as subq1
        group by ClassNames) as subq2

    join
        (select
            d2.ClassNames,
            sum(HPO_count_2)/count(Clinical_Orpha_Codes)
                            as Avg_Clinical_Annotations_per_Def
        from
            (select
                c.OrphaCode as Clinical_Orpha_codes,
                count(c.HPO_ID) as HPO_count_2,
                d.ClassNames
            from clinical_annotations c
            join orpha_diseases d on c.OrphaCode=d.OrphaCode
            group by d.ClassNames, c.OrphaCode) as subq3
            join orpha_diseases d2
                on subq3.Clinical_Orpha_codes = d2.OrphaCode
            group by d2.ClassNames) as subq4
        on subq4.ClassNames=subq2.ClassNames;
```

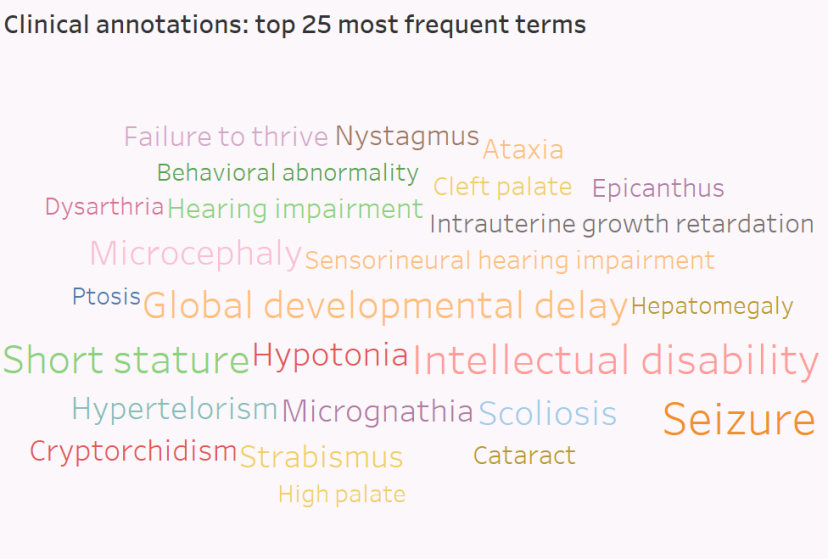
OrphaCode	Total_extracted_terms	Matched_Extracted_Terms	Total_Clinical_Annotations
61	2	1	42
812	2	2	42
584	2	2	26
881	1	1	111
126	4	2	9
14	3	1	63
1716	5	1	30
2773	2	2	8
236	2	1	27
1065	2	1	11
147	5	1	7
1538	3	2	11
1488	8	6	26
1369	3	2	10
1770	1	1	26

Average HPO terms per definition for each classificiation



Small fraction of terms being extracted for each definition

Word cloud of most frequent HPO terms



Some good overlap but some anomalies

05 Supervised ML Models

Model to classify diseases by speciality based on input text

Disease definition

A rare severe, X-linked, neurodevelopmental disorder characterized by rapid developmental regression in infancy, partial or complete loss of purposeful hand movements, loss of speech, gait abnormalities, and stereotypic hand movements, commonly associated with...

Neurological Classification

?

Developmental defect Classification

Models

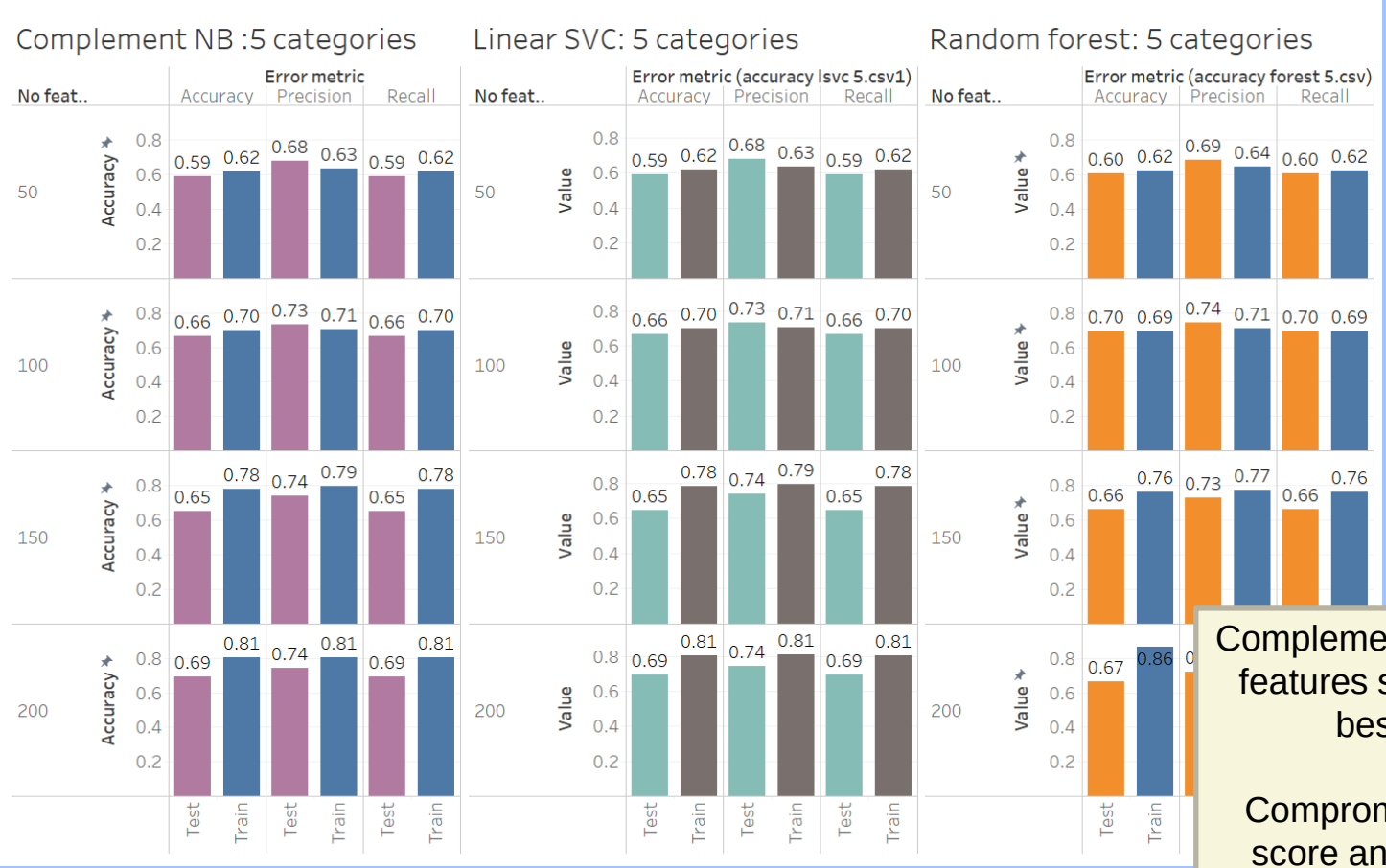
Complement Naive Bayes

Linear SVC

Random Forest

- Some hyperparameter tuning
- Varying number of features
- Feature engineering

Comparison of models with variable features



Complement NB with 100 features seems to give best result

Compromise between score and over-fitting

Overall scores for 5 categories

	F1-Score	Precision	Recall
accuracy	0.6563	0.6563	0.6563
macro avg	0.5749	0.5753	0.6484
weighted avg	0.6823	0.7603	0.6563

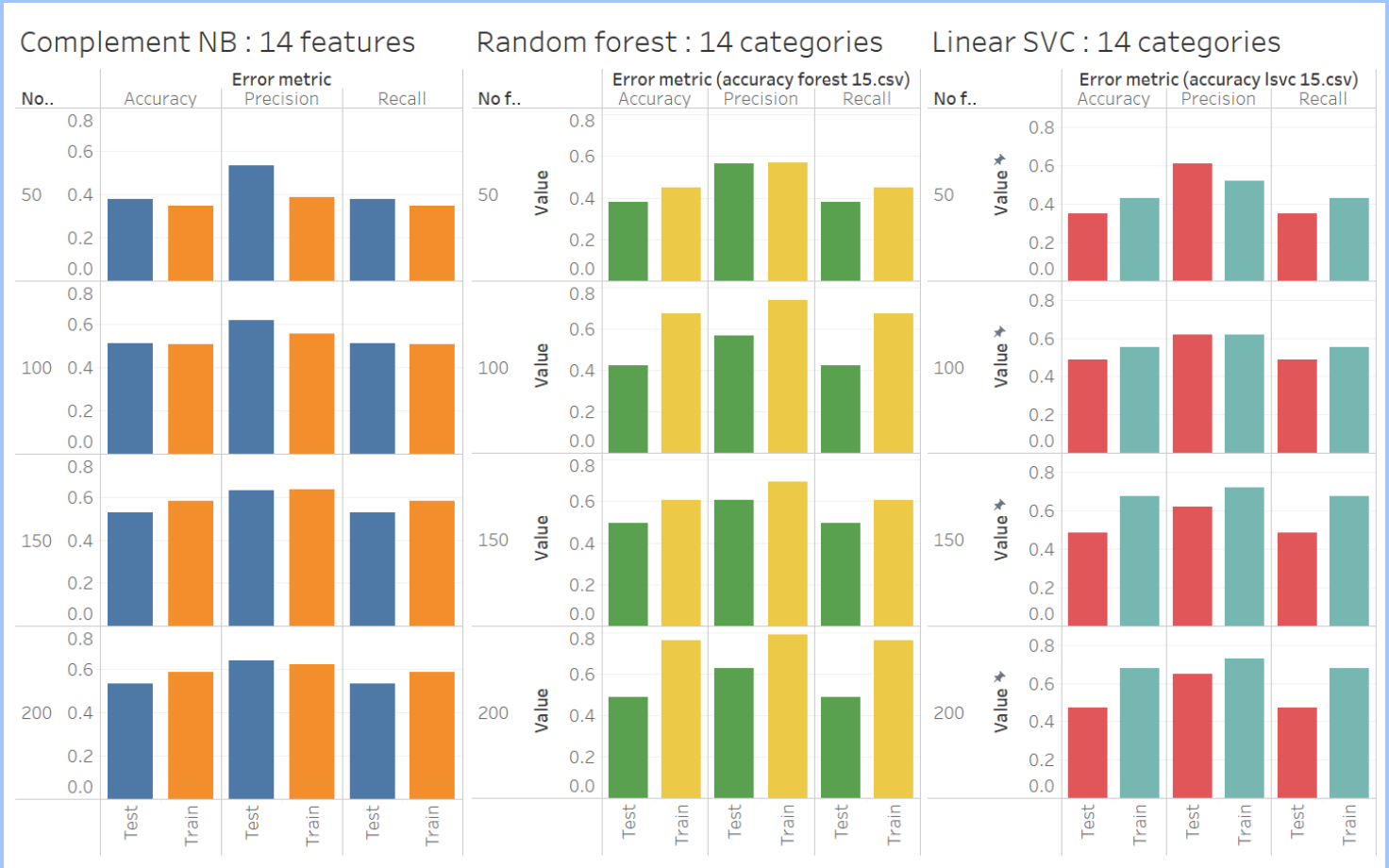
•Accuracy: Accuracy measures the overall percentage of correct predictions made by the model across all classes.

•Macro average: The macro average is simply the average of the precision, recall, and F1-score across all classes in the dataset.

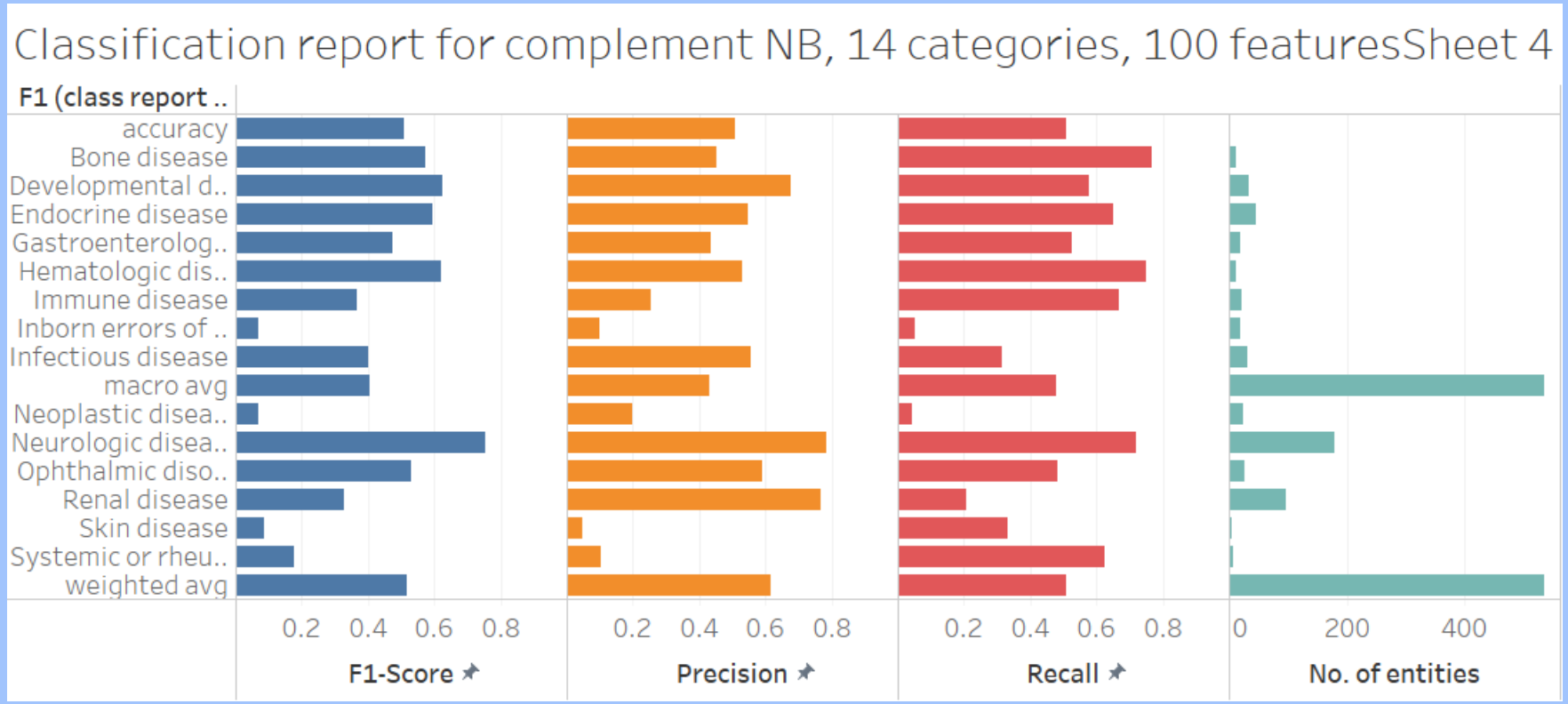
•Weighted average: takes into account the number of instances of each class

Classification report complement NB, 4 categories, 100 features

And if we add more categories (14)?



Classification report for 14 categories: Complement NB, 100 features



Overall scores for 14 categories

	F1-Score	Precision	Recall
accuracy	0.5103	0.5103	0.5103
macro avg	0.4045	0.4322	0.4797
weighted avg	0.5190	0.6160	0.5103

•Accuracy: Accuracy measures the overall percentage of correct predictions made by the model across all classes.

•Macro average: The macro average is simply the average of the precision, recall, and F1-score across all classes in the dataset.

•Weighted average: takes into account the number of instances of each class

Data preprocessing

1. Parse the beginning of each definition (up to ‘characterized by’)
2. Drop any definition that doesn’t start with ‘A rare’ (as means has disease name in) > 2000 entries
3. Clean text and tokenize
4. Create list of useful terms for modelling
    1. Manually by selecting from most frequent
    2. Using TF IDF tokenized
5. Create dataframe with terms and count for each disease
6. Split training and test data
7. Balance the data (initially taking top 5 categories

Challenges

- Text extraction

Intial idea for the model : to improve HPO mapping for the extracted text

- Requires manual curation of HPO terms to extracted text
- Good input data (i.e. better extraction methods)
- Expert input for the annotations

Conclusions

- With current methods not possible to compare the disease definitions with the Orphanet clinical annotations
- Better text extraction and mapping to the HPO terms required
  - Potential to provides indicator of quality
  - Could help with providing annotations by mining the literature
  - Help improve coverage of clinical annotations
- Complement naïve baye appeared to be the best model here
- Model needs more data and probably better feature engineering
  - Possible problem with the lemmatization

Thanks!

Special acknowledgement to the TA s Andy and Angela for all their help and Rafa for his great teaching and guidance.

Extra-special thanks to Etienne for supporting me through this course, and taking on more than his fair share of the child care.