UPPSALA UNIVERSITET

UPTEC F 21028

Examensarbete 30 hp

Juni 2021

Creation of a Next-Generation Standardized Drug Grouping for QT Prolonging Reactions using Machine Learning Techniques

Elsa Rådahl, Jacob Tiensuu

Civilingenjörsprogrammet i teknisk fysik

Creation of a Next-Generation Standardized Drug Grouping for QT Prolonging Reactions using Machine Learning Techniques

Elsa Rådahl, Jacob Tiensuu

Abstract

This project aims to support pharmacovigilance, the science and activities relating to drug-safety and prevention of adverse drug reactions (ADRs). We focus on a specific ADR called QT prolongation, a serious reaction affecting the heartbeat. Our main goal is to group medicinal ingredients that might cause QT prolongation. This grouping can be used in safety analysis and for exclusion lists in clinical studies. It should preferably be ranked according to level of suspected correlation. We wished to create an automated and standardised process.

Drug safety-related reports describing patients' experienced ADRs and what medicinal products they have taken are collected in a database called VigiBase, that we have used as source for ingredient extraction. The ADRs are described in free-texts and coded using an international standardised terminology. This helps us to process the data and filter ingredients included in a report that describes QT prolongation. To broaden our project scope to include uncoded data, we extended the process to use free-text verbatims describing the ADR as input. By processing and filtering the free-text data and training a classification model for natural language processing released by Google on VigiBase data, we were able to predict if a free-text verbatim is describing QT prolongation. The classification resulted in an F1-score of 98%.

For the ingredients extracted from VigiBase, we wanted to validate if there is a known connection to QT prolongation. The VigiBase occurrences is a parameter to consider, but it might be misleading since a report can include several drugs, and a drug can include several ingredients, making it hard to validate the cause. For validation, we used product labels connected to each ingredient of interest. We used a tool to download, scan and code product labels in order to see which ones mention QT prolongation. To rank our final list of ingredients according to level of suspected QT prolongation correlation, we used a multinomial logistic regression model. As training data, we used a data subset manually labeled by pharmacists. Used on unlabeled validation data, the model accuracy was 68%. Analyzing the training data showed that it was not easily separated linearly explaining the limited classification performance. The final ranked list of ingredients suspected to cause QT prolongation consists of 1086 ingredients.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala/Visby

Handledare: Klas Östlund, Kerstin Ersson, Emma Rofors Ämnesgranskare: Niklas Wahlström

Examinator: Tomas Nyberg

Populärvetenskaplig sammanfattning

Det här projektet har i syfte att främja farmakovigilans, vilket handlar om läkemedelssäkerhet och att upptäcka, analysera och arbeta för att förhindra oönskade biverkningar. Vi har valt att fokusera på biverkningen QT-förlängning, vilket innebär en rubbning i hjärtfrekvensen som kan ge allvarliga följder. Vårt mål med projektet är att gruppera och rangordna medicinska substanser som tros ge upphov till QT-förlängning. Denna typ av gruppering kan användas för att exkludera personer som tar vissa substanser från kliniska studier, samt för säkerhetsanalyser. Tidigare grupperingar har gjorts med avseende på vissa samband, t.ex. önskad effekt, men att gruppera med avseende på gemensam biverkning är ett relativt outforskat område. Till skillnad från den manuella gruppering som används idag ville vi skapa en automatiserad process som använder maskininlärningsmetoder för utvinning, validering och klassificering.

För att hitta ingredienser som kan ge upphov till QT-förlängning har vi använt oss av databasen VigiBase, som innehåller miljontals rapporter innehållande beskrivningar av patienters upplevda biverkningar samt vilka läkemedel patienten i fråga har tagit. Rapporterna i VigiBase är kodade enligt en internationell standardiserad medicinsk terminologi; Medical Dictionary for Regulatory Activities (MedDRA), vilket innebär att de beskrivna biverkningarna är kodade till standardiserade termer. Det finns en gruppering av dessa termer för just QT-förlängning. Denna har vi använt för att filtrera ut intressanta substanser ur VigiBase, det vill säga substanser som ingår i ett läkemedel som en patient har tagit medan denne haft en påvisad QT-förlängning.

Utöver att gruppera substanser utifrån rapporter som är kodade enligt MedDRA så strävade vi efter en utökning av processen för att inkludera biverkningsbeskrivningar i fritext. Fritexterna kommer från rapporter som inte nödvändigtvis är MedDRA-kodade. Vi utvann dessa ur VigiBase samt förbehandlade och filtrerade dem för att endast inkludera engelskspråkiga texter. Genom att använda en modell för språkteknologi utvecklad av Google; BERT, tränade vi modellen för att avgöra om en rapports fritext beskriver en QT-förlängning. Denna klassificering arbetade vi med som ett fristående projekt från substansgrupperingen, dock i avseende att kunna kombinera dem i framtiden. Modellen visade mycket goda resultat med en F_1-score på dryga 98%, vilken innebär att endast en mycket liten del av valideringsdatan har klassats inkorrekt.

De läkemedel som är inkluderade i en rapport kan beskrivas enligt en internationell klassificering som kallas WHODrug Global, där inkluderade substanser beskrivs som koder. För substansgrupperingen använde vi dessa koder för att hitta substansens namn och variant i WHODrugs lexikon. För att validera om substanserna har en känd koppling till QT-förlängning undersökte vi de bipacksedlar som innefattar vardera utvunnen substans. Genom att använda ett verktyg som laddar ner, läser och kodar innehållet i bipacksedlarna till MedDRA-termer kunde vi validera substanserna.

För att avgöra till vilken grad en substans rapporteras tillsammans med en beskrivning av QT-förlängning räknade vi ut hur stor procentsats av alla rapporter som berör en viss substans som beskriver en QT-förlängning. En potentiell felkälla i den procentsatsen är det faktum att en rapport kan nämna en eller flera läkemedel, som i sin tur kan innehålla en eller flera substanser. Därför är det svårt att dra slutsatser om vilken av flera substanser som orsakat reaktionen. För att förbättra den nämnda procentsatsen så modifierade vi den genom att ta hänsyn till substanser vi starkt misstänker vara QT-förlängande efter att ha validerat mot bipacksedlar.

Vår slutgiltiga lista innehåller 1086 substanser som vi ville rangordna efter misstänkt korrelation till QT-förlängning. Det gjorde vi genom att träna en klassificeringsmodell som använder logistisk regression för att klassa varje ingrediens efter grad av misstänkt QT-förlängande effekt. Som träningsdata användes en mängd substanser som två farmaceuter fick klassa oberoende av varandra. Efter korsvalidering presterade klassificeringsmodellen en pricksäkerhet på 68% och ett kvadratiskt medelfel på 47% på valideringsdatan. Det är en relativt osäker klassificering vilket delvis kan förklaras av en liten mängd samt relativt spridda träningsdata. Vi anser dock att det fungerar väl för en grövre sortering.

Slutprodukten består av de rangordnade substanserna samt tillhörande information som är till hjälp för en farmaceut att avgränsa vilka substanser som bör ingå i substansgrupperingen. Vi har arbetat aktivt för att inkludera även substanser med mycket svag indikation på koppling till QT-förlängning för att inte missa att inkludera relevanta substanser. Genom att sätta gränser för olika parametrar kan man minska mängden substanser genom att utesluta dem med mycket svag koppling till QT-förlängning.

Acknowledgements

We would like to thank several people without whom we would not have been able to accomplish the results of this thesis project. Our brilliant supervisors Klas Östlund, Kerstin Ersson and Emma Rofors, a.k.a Team Shrek, for your daily support, encouragement and ideas. Even in a time of pandemic and remote working, you have never felt far away. The rest of Team Tarzan for showing interest and providing really bad puns, seeing your faces has been the best way to start our days. Niklas Wahlström for your great knowledge of data-driven methods and your continuous feedback. Jessica Nilsson for your VigiBase expertise and your stubbornness to never let a question go unanswered. Eva-Lisa Meldau for your spot-on questions and endless ideas on machine-learning methods. Shachi Bista for your invaluable help with the SPC Mining. Vanja Wallner for your thorough work with NLP methods, and for sharing your knowledge and ideas. Tommy Dzus and Sofia Fors for hours spent labeling ingredients. Denis Krylov at OpenFDA, for providing expert API queries, and Ray Woosley at AzCERT for sending us CredibleMeds' QT drug list. Lastly, thank you to everyone at UMC for showing interest and support and for making us feel welcome, as well as our friends and families for supporting us through these 5 years of study, making it a time of our lives we will never forget.

Acronyms

MedDRA	Medical Dictionary for Regulatory Activities
ICSR	Individual Case Safety Report
BERT	Bidirectional Encoder Representations from Transformers
UMC	Uppsala Monitoring Center
WHO	World Health Organization
EMA	European Medicines Agency
ADR	adverse drug reaction
ECG	electrocardiogram
TdP	Torsades de Pointes
NLM	the U.S. National Library of Medicine
SPC	summary of product characteristics
FDA	Food and Drug Administration
SDG	Standardised Drug Grouping
SMQ	Standardised MedDRA Query
ICH	International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use
AzCERT	Arizona Center for Education and Research on Therapeutics
IAA	Inter-Annotator Agreement
API	Application Programming Interface
NLP	Natural Language Processing
PT	Preferred Term
LLT	Lowest Level Term
SGD	Stochastic Gradient Descent
MSE	mean squared error
UNII	Unique Ingredient Identifier

Contents

1 Introduction

This master thesis has been conducted in collaboration with Uppsala Monitoring Center (UMC). UMC is a non-profit foundation working alongside the World Health Organization (WHO) with international drug monitoring. Their goal is to promote safer use of medication globally, by working in specialized teams to support member countries of the WHO Programme for International Drug Monitoring and supporting patient protection. UMC also develops and distributes the Global Drug Dictionary; WHODrug Global, which is used to standardize pharmaceutical information.[1] Supervisors from UMC are system developers Kerstin Ersson and Klas Östlund and the subject reviewer is Niklas Wahlström from the Department of Information Technology at Uppsala University.

1.1 Problem background

This project aims to support pharmacovigilance, the collective name for what WHO describes as ”the science and activities relating to the detection, assessment, understanding and prevention of adverse effects or any other drug-related problem”. It serves as a key public health function where the main objective is to present reliable information and taking action in order to improve patient care and safety.[10] UMC works with pharmacovigilance on a global scale, with the mission of safer and more effective use of medicine worldwide.[11]. A part of UMC's work with pharmacovigilance is to maintain VigiBase, a database containing millions of drug safety-related reports. These reports contain valuable information about post-marketing patient experiences of drugs and play a key role in the monitoring of drugs post-marketing.

Medicinal ingredients can be grouped based on one or several shared properties. These listings are referred to as Standardised Drug Groupings (SDGs). UMC creates and manages several SDGs where medicinal products and active ingredients are grouped, often according to type or effect. Some examples of existing SDGs are Vaccines, Antihistamines and Cancer therapies listings. By customer requests, the idea arose to group ingredients based on a shared adverse drug reaction (ADR), an unwanted side effect. This new kind of grouping is called next-generation SDGs.

1.2 Problem formulation

The main goal of this thesis is to create an SDG basis for QT prolongation. To further extend the process to include free-text information, we have also set an aim regarding free-text processing.

1.2.1 Creation of SDG basis

The aim is to create an automated process for finding and presenting information that a pharmacist can use to decide whether a medicinal ingredient should be included in an SDG listing ingredients that might cause QT prolongation. QT prolongation is an ADR affecting the heartbeat. The list of potential ingredients can be long, so the ingredients should be listed according to the level of suspected QT prolongation connection.

1.2.2 Free-text processing

To decide if a free-text description of an experienced ADR describes QT prolongation, we also want to predict if a free-text verbatim describes QT prolongation. If so, the report should be included in the SDG creation process.

1.3 Purpose

In this project we focus on the ADR QT prolongation since it is a serious and occasionally life-threatening reaction and therefore important to track. By creating the process as general as possible, it can be used as groundwork for other types of SDGs focusing on different ADRs.

The SDG can be used in various types of safety analyses, supporting investigation regarding which substances or concomitant medication (used at the same time) that are known or suspected to cause QT prolongation. For example, if a patient experiences a QT-related reaction this could be investigated by examining if he/she has been taking a substance listed in the QT prolongation SDG. It can also be used in the specification of exclusion criteria in clinical trials, meaning that subjects taking a medicinal product listed in the QT SDG are excluded from the study. This is to avoid interference in the study results, as well as to ensure patient safety.

The purpose of the free-text processing is that it complements the creation of the SDG basis. This by allowing free-text data as input that does not need to have been manually reviewed and coded. By automating the whole process, we aim to streamline and time optimize the SDG creation, which would otherwise be performed manually. We also wish to standardize the process such that every ingredient's correlation to an ADR is based on the same parameters. Using these as a decision basis allows us to avoid human error and bias.

1.4 Delimitations

To limit the scope of this project, some delimitations have been set. For the free text classification, we will filter for reports written mainly in English. This is because tools and systems used in further processing use English data, and for evaluation purposes.

Coding conventions for medical reports change over time, so to keep the coding consistent throughout the data a date restriction has been set such that only reports submitted after the 1st of January 2018 are considered for the ingredient extraction. Since the VigiBase database includes medicinal products prescribed for humans only, we will discard information about drugs for veterinary use in our validation stage.

1.5 Related work

Next-generation SDGs are a new concept and have not been previously developed, thus it is a relatively unexplored area. An important part of the SDG creation process is to automatically scan and code product label information available online. To do this, we have used a UMC-created pipeline based on Shachi Bista's paper ”Extracting Adverse Drug Reactions from Product Labels using Deep Learning and Natural Language Processing”[17], where the free-text coding is based on Vanja Vallner's paper ”Extracting Adverse Drug Reactions from Product Labels using DeepLearning and Natural Language Processing”[18]. The latter has also been of great influence on our Free-text processing, from which we have based parts of our implementation. Regarding listing drugs with a connection to QT prolongation, similar work has been done by the Arizona Center for Education and Research on Therapeutics (AzCERT), maintaining the CredibleMeds database that contains a list of QT drugs.[20]

2 Drug-safety related background

This section describes the drug-related components, systems and tools that have been used in the SDG creation process. These relations are illustrated in Figure 1.

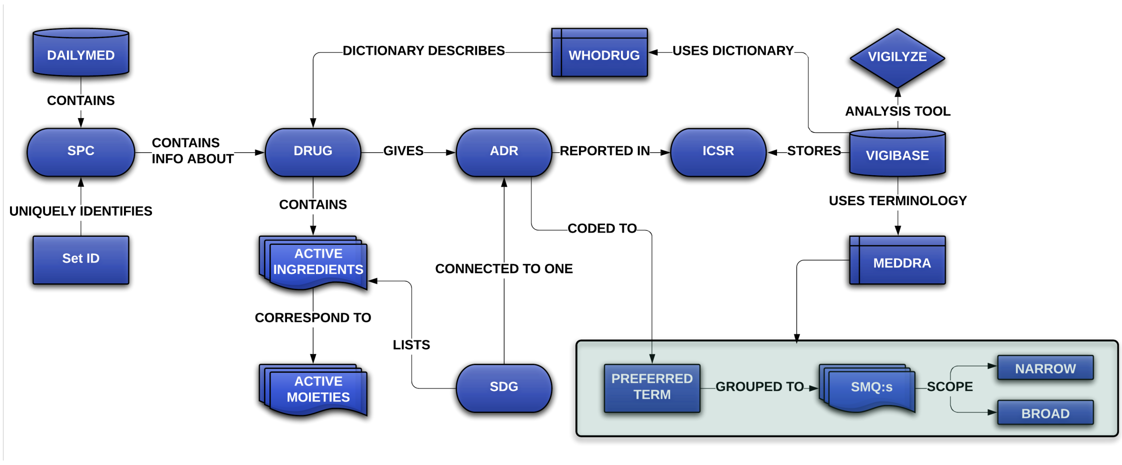


Figure 1: Concept map describing component relations

2.1 Standardised Drug Grouping

SDGs are collections of ingredients having one or more properties in common. The individual grouping can be based on indication, chemical properties, pharmacodynamic properties or pharmacokinetic properties, as well as any other property of interest.[2] A so-called next-generation SDG has the purpose of grouping ingredients with a common ADR. SDGs can be used whenever there is a need to group drugs e.g. in clinical trials where they need to make sure to exclude patients taking certain medications. The SDGs can also be used during signal detection, a core activity at UMC that involves identifying and describing suspected harm caused by a patient's use of medicine. A signal in this context is described as ”a hypothesis of a risk with a medicine with data and arguments that support it, derived from data from one or more of many possible sources”. The objective is to find new and unknown ADRs and to see group effects.[3] In this project, we aim to construct an SDG listing ingredients suspected to cause the ADR QT prolongation.

2.2 Drug-induced QT Prolongation and Torsades de Pointes

QT prolongation is a serious cardiac ADR of delayed ventricular repolarization, i.e. when the time it takes for the heart to recharge between beats is longer than usual. QT prolongation can be congenital or drug-induced (our focus), and it is important to discover in clinical trials and post-authorization safety studies (studies conducted after a drug has been approved to further analyze safety and effectiveness). QT describes a specific interval, see Figure 2, that can be observed when measuring the electrical activity of the heart in an electrocardiogram (ECG). To investigate the reason behind the reaction, it is important to know which medications are known to cause the reaction. Since there are many potential medications (and by extension, ingredients) correlated to QT prolongation that might not cause it, there is a need to review the available evidence of causation to decide whether or not to include the medication in the SDG.

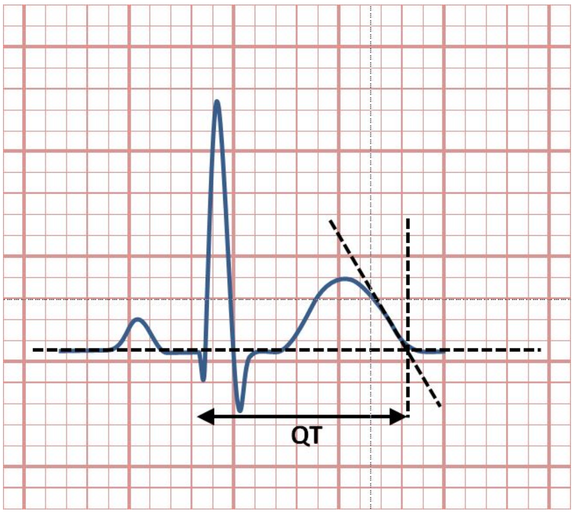


Figure 2: The QT interval. ”File:QT interval.jpg” by PeaBrainC is licensed under CC BY-SA 4.0

A prolonged QT-interval combined with a certain form of ventricular tachycardia is the definition of Torsades de Pointes (TdP). Drug-induced QT prolongation increases the risk for but does not always progress to TdP. TdP can lead to ventricular fibrillation and/or sudden cardiac death.[5]

2.3 Individual Case Safety Report

To document and analyze the ADRs experienced by drug users, Individual Case Safety Reports (ICSRs) are collected. An ICSR is described by the European Medicines Agency (EMA) as a ”document providing information related to an individual case of a suspected side effect due to a medicine”.[4] It contains information needed to track and report ADRs and medicinal product problems. The report must contain the medicinal product taken and perceived adverse event (ADRs are a subset of adverse events where a causal relationship is suspected), which could be a fatal outcome. The patient and reporter (e.g. a healthcare professional) must be identifiable in the original report. Optional included information could be for example medical history, patient characteristics and health-related test results. The collection of ADRs is handled by the national centre for pharmacovigilance for each country participating in the WHO Programme for International Drug Monitoring.[12] The current number of fully participating countries in the programme is 142 (November 2020).[13]

2.4 Medical Dictionary for Regulatory Activities

With all these reports submitted from across the globe, written by different professions and in different languages, it is a challenge to correctly translate and interpret the reports. Thus arose the need for a standardized terminology.

Medical Dictionary for Regulatory Activities (MedDRA) is an international standardized medical terminology developed by the International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use (ICH). The MedDRA standard terms are used to facilitate the exchange of clinical information such as registration, documentation and monitoring of clinical substances. MedDRA is originally in English but the terminology has been translated to 13 additional languages [6]. A new MedDRA version is released every six months and the current version is 23.1 (February 2021).

MedDRA is based on a hierarchical structure consisting of five levels. The free text information about an ADR from an ICSR is coded into MedDRA standard terms. The coding, which is often done by national centres for pharmacovigilance, is done at the lowest and most specific level, Lowest Level Term (LLT). The LLTs can be described as synonyms or different ways to formulate a Preferred Term (PT), where a PT is the ”correct” term to describe an ADR. When a matching LLT is found for a free text described ADR, it is coded to the corresponding PT, which is a distinct description for each ADR. For example, ”Dizziness” is a PT. If the report describes an ADR as for example ”Light-headed”, ”Woozy” or ”Swaying feeling”, which are all LLTs under ”Dizziness”, it will be coded to that PT. The PTs in turn belong to more general levels where the most general is System Organ Class. The hierarchical structure can be seen in Figure 3, as well as an example of what the hierarchy looks like for the LLT ”Long QT” specifically.

2.4.1 Standardized MedDRA Query

Even though all PTs are grouped according to the hierarchy explained, a need arose for a different set of groupings to easier identify all MedDRA terms related to a specific medical condition (where the terms may belong to different System Organ Classes). A Standardised MedDRA Query (SMQ) is the product of this separate way to group ADRs outside of the hierarchical levels[8]. The grouping is done at the PT level, as seen in Figure 3. The SMQs are a tool for investigation of drug safety issues and can be for example ”Drug abuse, dependence and withdrawal”, ”COVID-19” or ”Taste and smell disorders”. As seen in the example in Figure 3, one PT can belong to several SMQs. Since we aim to group and sort substances related to QT Prolongation, the SMQ of interest is the QT prolongation/TdP SMQ. We will use this SMQ to sort out relevant data (suspected relation to TdP/QT prolongation) for the SDG creation.

Each SMQ is divided into two subgroups: narrow scope and broad scope. The narrow scope includes the PTs that are most likely to correspond to the SMQ characteristic, whereas the broad scope is estimated to have a lower correlation. The QT Prolongation/TdP SMQ includes the following PTs in the narrow scope (February 2021):

- Long QT syndrome

- Long QT syndrome congenital
- TdP
- Electrocardiogram QT interval abnormal
- Electrocardiogram QT prolonged
- Ventricular tachycardia

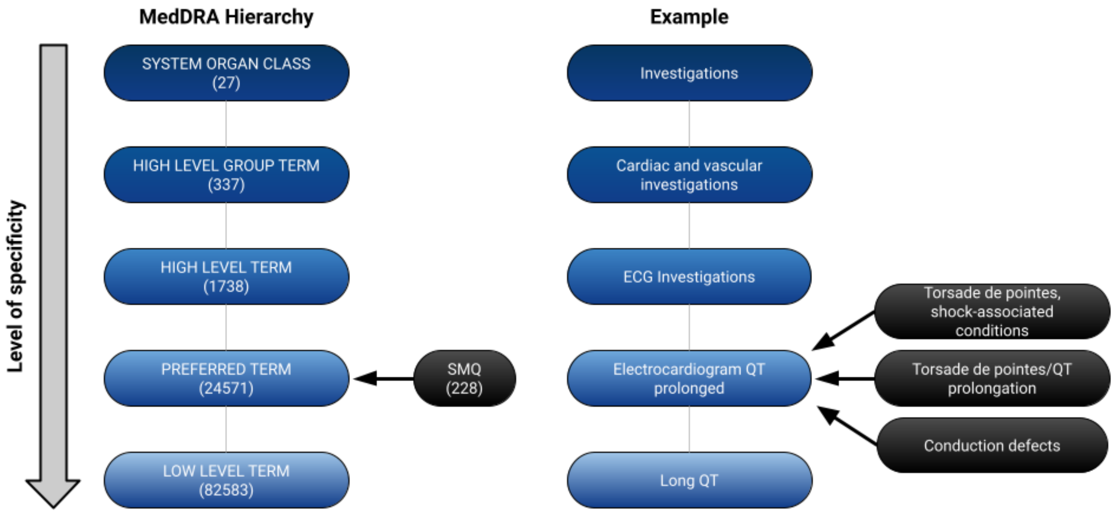


Figure 3: MedDRA v23.1 hierarchy, including the number of terms for each level[14]

2.4.2 Drug Characterization ID

MedDRA uses the label ”Drug Characterization ID” to describe the presumed connection between a drug and a reported ADR. The different values and their descriptions are:

1. Suspected (Drug is suspected to have caused the ADR)
2. Interacting (The ADR is suspected to be caused by several interacting drugs)
3. Concomitant (The drug has been taken when the ADR occurred, but a relation is not suspected)

2.5 WHODrug Global

While MedDRA is used to code and describe the ADRs, we also need a system to code and describe the drugs mentioned in the reports. For this purpose we use WHODrug.

WHODrug is a WHO global drug dictionary for medicinal products managed by UMC which describes the purpose as ”The dictionary is used to identify drug names and evaluate medicinal product information, including active ingredients and products’ anatomical and therapeutic classifications, from nearly 150 countries". The dictionary standardizes the data using a unique drug code hierarchy and terminology and facilitates pharmacovigilance by allowing easier identification and evaluation of drug-related issues.[9] The drug code consists of three parts:

- Drug Record Number: Describes an active moiety, regardless of variations.
- Sequence number 1: Identifies variations such as salts, plant parts and extraction methods, hence describing an active substance (or a combination of several).
- Sequence number 2: Identifies the WHODrug record name.

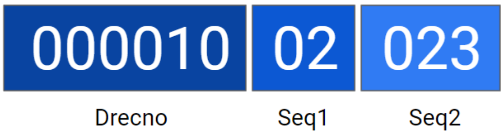


Figure 4: Drug code for Aralen Phosphate which is used in the treatment of malaria

2.5.1 Insight

WHODrug Insight is an online search engine for easy access to all WHODrug data managed by UMC [15]. We used Insight to get access to a small part of data that was missing in the database tables used.

2.6 VigiBase

All the submitted reports need to be stored for easy access and analysis. VigiBase is a WHO global database managed by UMC containing 24 812 310 ICSRs (February 2021), initially described in free text and electronically transferred from the national centres for pharmacovigilance.

The transferred data is anonymized in the way that the data can not be linked to a name or social security number, although patient initials may be included in the reports. The patient demographics are included on a country level. VigiBase supports pharmacovigilance by providing structured data for analysis. It is linked to several terminologies, including MedDRA and WHODrug. The reported adverse events are codes allocated according to the latest versions of the used terminology, currently MedDRA 23.1 (February 2021).[12]

2.6.1 VigiLyze

The national centres can access and analyze the data via the platform VigiLyze. It allows easy access and overview of the VigiBase data through graphs and listings, as well as information and results from investigations. VigiLyze was a useful tool for us to validate that the correct amount of data was covered/extracted in SQL queries.

2.7 Summary of Product Characteristics

Each medicinal drug on the market must come with a summary of product characteristics (SPC). It is a legal document containing medicinal product information directed towards health professionals about how to use the product safely and effectively. The SPC includes information about benefits, risks, composition, dosage, storage and information for individualized care, among others. The product label included with the medicinal product is based on the SPC information, written in a way suited for users. In this project, we have used SPCs to validate connections between QT prolongation and drugs containing a specific ingredient.

2.8 DailyMed

As our source of SPCs we have used DailyMed, a website and database operated by the U.S. National Library of Medicine (NLM). It contains SPC information produced and updated by pharmaceutical companies based on their knowledge and research regarding the product, where the products and product labels are approved by the U.S. Food and Drug Administration (FDA).

2.9 CredibleMeds

For SDG ingredient validation and improvement purposes, we searched for reliable sources with listed QT-related drugs (although not based on VigiBase ICSRs). CredibleMeds is an online database containing drug safety-related information. The database is created and managed by the University of AzCERT, a non-profit organization located in the US with close ties to the FDA.[20] A list of ingredients with a connection to QT-prolongation/TdP is accessible on their website for registered users (free registration).

3 Data-driven methods

In this section, we will go through the theoretical explanations of the supervised machine learning models used in this project. The goal of supervised machine learning is to predict the outcome for given input data. This is done by allowing the model to learn from labeled training data containing information on how the input variables relate to the output variables. The models used in this project are classification models, meaning that they predict which class (e.g. positive or negative in a binary case) a data point belongs to. The models are:

- Ingredient classifier for SDG presentation order. Multinomial logistic regression was used, with Stochastic Gradient Descent (SGD) as training algorithm.
- Binary classifier of free text verbatims using the deep learning language model Bidirectional Encoder Representations from Transformers (BERT).

3.1 Logistic Regression

Logistic regression is a linear classification algorithm in the sense that the data is separated by linear hyperplanes. The regular and most basic form of logistic regression is the binary classification model where each data point is classified with one of two labels.

The input parameters that are used to train the model are called features. Each input data point consists of a set of features, here described as $\vec{x} = \{x_1, ..., x_N\}$ for N number of features, as well as the label y_m given M number of classes. The binary labels (where $M = 2$) would typically be set as $y_1 = 1$ and $y_2 = -1$. To predict the label of a data point given the set of features, we calculate the pseudo probability p that the data belongs to each class. The probability that the data point belongs to class y_m given features $\vec{x}$ is given as $p_m = p(y = y_m|\vec{x})$. The class probabilities always add to 1:

$$\sum_{m=1}^{M} p(y = y_m|\vec{x}) = p(y = y_1|\vec{x}) + ... + p(y = y_M|\vec{x}) = 1$$

To calculate the probabilities, we initiate a set of weights $\vec{\omega} = \{\omega_1, ..., \omega_N\}$ corresponding to the features, as well as a bias b. The weights and biases are constants. Combining these using linear regression for a reference class (one of the two classes) in the binary case results in a logit z:

$$z = \vec{\omega}^T \vec{x} + b = \omega_1 x_1 + \omega_2 x_2 + ... + \omega_N x_N + b$$

To transform this logit to a probability p_1 (assuming that class 1 is the reference class), we apply the logistic function:

$$p_1(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

The probability for the second class is set so that p_1 and p_2 add to 1:

$$p_2(z) = 1 - p_1(z)$$

The class prediction in the binary case will be the class y_m with the highest probability, i.e. where $p(y = y_m|\vec{x}) > 0.5$.

3.1.1 Multinomial Extension

For $M > 2$ number of classes, a vector of logits, $\vec{z} = \{z_1, ..., z_M\}$, is used:

$$z_m = \vec{\omega}_m{}^T\vec{x} + b_m = \omega_{1,m}x_1 + \omega_{2,m}x_2 + ... + \omega_{N,m}x_N + b_m, \qquad m \in \{1, M\} \tag{2}$$

To extend the algorithm to allow multiple classes, we use a generalization of Equation 1 known as the Softmax function:

$$p_m(\vec{z}) = \frac{e^{z_m}}{\sum_{j=1}^{M} e^{z_j}}, \qquad m \in \{1, M\} \tag{3}$$

This results in a number of probabilities corresponding to the number of classes. The class prediction will still be the class y_m with the highest probability.

3.1.2 Training

The training consists of updating weights and biases to best predict the training data. We initiate a matrix containing the weights and biases on the following form:

$$\begin{bmatrix} \vec{\omega}_1 \\ \vdots \\ \vec{\omega}_N \\ \vec{\omega}_{N+1} = \vec{b} \end{bmatrix} = \begin{bmatrix} \omega_{1,1} & \omega_{1,2} & \cdots & \omega_{1,M} \\ \vdots & \ddots & \ddots & \vdots \\ \omega_{N,1} & \ddots & \ddots & \omega_{N,M} \\ b_1 & b_2 & \cdots & b_M \end{bmatrix}$$

Initially, the matrix is filled with random values. For the training, we declare the following variables:

- Target vector $\vec{\tau}$
- Output probabilities p_m
- Learning rate γ

The target vector $\vec{\tau}$ is a label y encoded to a binary vector using one-hot scheme. The length of $\vec{\tau}$ corresponds to the number of possible classes. Its elements are 0 except for the element with an index representing the label y, $\tau_{m=y}$, where the value is 1. M classes would be represented by following target vectors:

$$\begin{aligned} &y = 1 \ \rightarrow \vec{\tau} = [1,\ 0,\ 0, \cdots,\ \tau_M = 0] \\ &y = 2 \ \rightarrow \vec{\tau} = [0,\ 1,\ 0, \cdots,\ \tau_M = 0] \\ &\vdots \\ &y = M \rightarrow \vec{\tau} = [0,\ 0,\ 0, \cdots,\ \tau_M = 1] \end{aligned}$$

The output probabilities p_m corresponding to each class is calculated by the Softmax function, see Equation 3.

3.1.3 Stochastic Gradient Descent

As training algorithm, we have used Stochastic Gradient Descent (SGD). For each epoch, which is an iteration over the whole training data set, the indices of the training set data points are shuffled to process the data in random order. For each index *i*, the weights and biases are updated in order to minimize the differences between probabilities and targets. Consider the loss function $L(\omega)$ that we wish to minimize:

$$L(\omega) = -\sum_{m=1}^{M} \tau_m log(p_m(\vec{z}(\vec{\omega}))) = -log(p_{m=y}(\vec{z}(\vec{\omega}))$$

An ideal prediction would result in $L(\omega) = 0$. By shifting the weights according to the gradient $\nabla_\omega L$, we decrease the loss function for each epoch.

$$\omega_{new} = \omega_{old} - \Delta\omega$$

$$L(\omega - \Delta\omega) \leq L(\omega)$$

The gradient is calculated using the chain rule:

$$\nabla_\omega L = \frac{\partial L}{\partial \omega_{n,m}} = \frac{\partial L}{\partial z_m} \frac{\partial z_m}{\partial \omega_{n,m}}$$

where the inner logit derivative is given as

$$\frac{\partial z_m}{\partial \omega_{n,m}} = \begin{cases} x_{i,n} & \text{for weights} \\ 1 & \text{for biases} \end{cases} \tag{4}$$

and the outer derivative as

$$\frac{\partial L}{\partial z_{m=y}} = (p_m - \tau_m) \tag{5}$$

At the update stage, the learning rate γ decides how quickly the model should adapt to new training data. For each training data point at a time, the weights and biases are updated according to:

$$\text{Weight update:} \quad \Delta\omega_{n,m} = \gamma\Big(\overbrace{\underbrace{x_{i,n}}_{\frac{\partial z}{\partial \omega}} \underbrace{(p_m - \tau_m)}_{\frac{\partial L}{\partial z}}}^{\nabla_\omega L = \frac{\partial L}{\partial \omega_{n,m}}} \underbrace{p_m(1-p_m)}_{\text{softmax derivative}} \Big) \qquad \forall n \in \{1, N\} \;\; \forall m \in \{1, M\}$$

$$\text{Bias update:} \quad \Delta\omega_{N+1,m} = \gamma\Big((p_m - \tau_m) p_m (1 - p_m) \Big) \qquad \forall m \in \{1, M\}$$

3.2 Deep Learning

Deep learning is a branch of machine learning which utilizes neural networks. Neural networks are based on simpler machine learning models such as linear regression. Different types of neural networks can be applied to a wide range of problems e.g. image analysis, speech recognition or language processing. The architecture is inspired by how neurons are connected in the human brain and has shown to be able to recognize nontrivial patterns in the input- and output variables.

A neural network consists of one or several layers of nodes where the input to each layer is the output of the layer before. The first layer is called an input layer and is the entry point for the input variables. The input layer is followed by one or several hidden layers where the input variables are transformed and weighted by the network parameters. The final layer is called the output layer. For a classification problem, a neural network with one layer is constructed by using a generalized linear regression model, which is a linear regression model with the parameters ω_i to which a scalar activation function σ is applied. The generalized linear regression model is a non-linear function which predicts the output z from the input $\mathbf{x} = [1\ x_1\ x_2\ \dots\ x_N]^\top$, which for a neural network with one hidden layer it is given by:

$$z = \sigma(\omega_0 + \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_N x_N) \tag{6}$$

The activation function may be any chosen function, but for this project the softmax function was used, see Equation 3. With the help of the activation function the output layer converts the output of the last hidden layer into class predictions. The linear regression model in equation 6 can be generalized to a neural network with several layers. Figure 5 shows the architecture of a basic neural network.

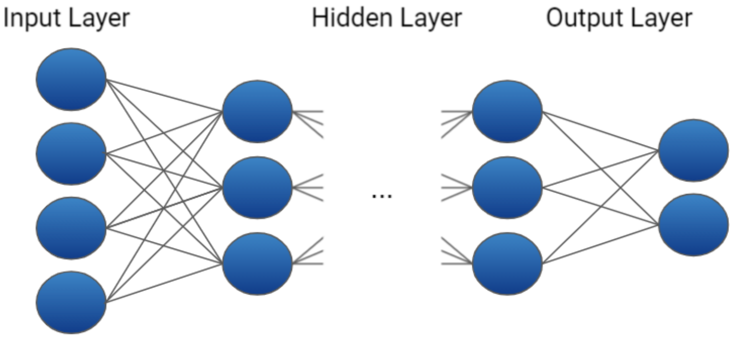


Figure 5: Neural network architecture

Different types of hidden layers serve different purposes, e.g. pooling layers that reduce the size of the data. With the help of the different layers, the neural network can on its own create features and find complex relations between input and output variables in the data. However, a large set of data is usually required for a neural network to be successful [23].

The softmax functions input parameters $z_1, ..., z_M$ are called logits. To learn the classification network the cross entropy loss function is used. With this approach, the vector of predicted probabilities is compared to the one hot encoded output vector. From this comparison, the cross-entropy loss function is minimized and the network's parameters are optimized. The cross-entropy loss function also helps to avoid numerical problems when the probability for a prediction, $p(m|\mathbf{x}_i, \theta)$ is close to zero, by compensating for the effects caused by the softmax function. The cross-entropy loss function is given by the following equation: [22]

$$\hat{\theta} = \arg\min_{\theta} \tfrac{1}{n} \textstyle\sum_{i=1}^{n} L(\mathbf{x}_i, \mathbf{y}_i, \theta) \quad \text{where} \quad L(\mathbf{x}_i, \mathbf{y}_i, \theta) = -\textstyle\sum_{m=1}^{M} y_{im} log\, p(m|\mathbf{x}_i; \theta) \tag{7}$$

3.2.1 Natural Language Processing

Natural Language Processing (NLP) is a branch of machine learning that concerns the interaction between computers and human languages. Some language processing tasks can be easy for computers to perform such as word-for-word translations between languages. However human languages are complex and constantly evolving. It is a difficult task for computers to capture or understand the semantics of human languages. In recent years there has been progress in the field with the introduction of deep learning techniques for NLP [19].

3.2.2 Transformers

A transformer is a deep learning model that uses an encoder-decoder architecture. The model consists of several encoding layers that process and transform the input variables. The encoding layers are followed by decoding layers that inverse transform the encoded input variables to create an output. Transformers use attention which has the advantage that input does not have to be processed in a sequence, unlike other encoder-decoder architectures that utilize recurrent neural networks (neural networks with feedback loops)[25]. For an NLP problem, this means that the transformer model can focus on the context in which the words are used and find the most relevant parts of a sentence.

3.3 BERT

The Bidirectional Encoder Representations from Transformers (BERT) is a pretrained NLP model released by Google [26]. BERT has a transformer-based architecture that allows it to process a free-text input bidirectionally so that it can learn the context of a word based on previous and following words. The specific BERT model used in this project is pretrained on 3.3 billion words from the English Wikipedia and BooksCorpus and has a general understanding of the English language and its structure. However, the model needs to be fine-tuned for each NLP problem it is tasked with by training the model on relevant data.

3.3.1 Tokenization

Before free-text data can be processed by BERT the data needs to be tokenized. Tokenization means that free-text data is segmented into components which the model can process. This is done with the help of BERTs built-in dictionary that contains 30 000 tokens. If a word is not in the dictionary as a single token the word can be separated into several tokens. An example is the word *readable* which is not in the dictionary, it would be tokenized *read*, ##*able* where ## indicated that the token belongs to the first previous token not beginning with ##. There are also special tokens that need to be added before BERT can process a free-text input. Firstly, every input needs to begin with a $[CLS]$ token which indicates to BERT that it is processing a classification problem. Secondly, each free-text input needs to end with a $[SEP]$ token which is used for next sentence prediction problems. Lastly, each free-text input needs to be padded to the same length by $[PAD]$ tokens. The developers of BERT recommend an input length of either 32 or 64 tokens. Inputs longer than the limit will be cut off and ignored in the later process. Every token is then converted to an ID which is unique for every token.

As the final step, an attention mask is created. The attention mask is a vector of ones and zeros that supports BERT in keeping track of which tokens are relevant for further process. Every token except for padding tokens is represented by ones in the attention mask. These processing steps for an example sentance can be observed in Figure 6. As input the BERT model uses the vector of token IDs and the attention mask.

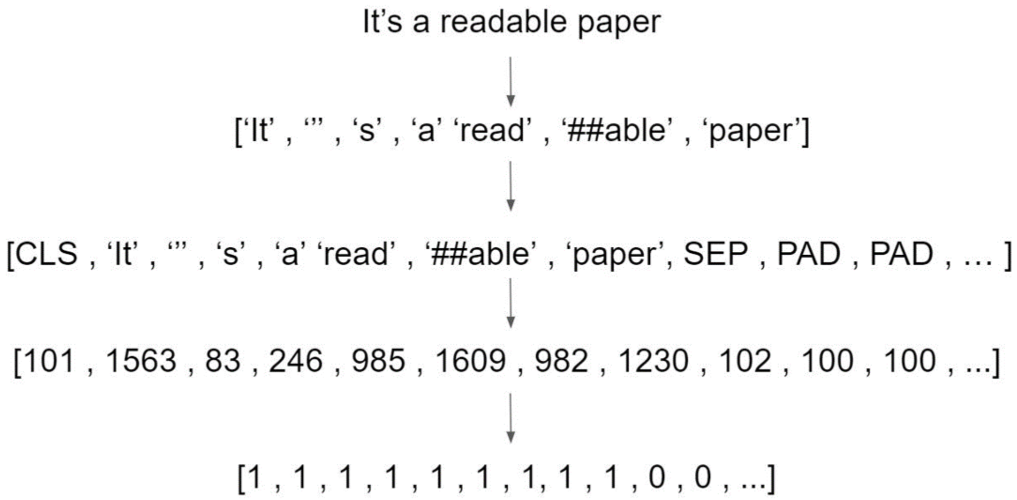


Figure 6: Example of how the BERT model processes a sentence

3.3.2 Classification

When BERT is used for a classification problem a softmax layer is added after the final transformation layer. The input to the softmax layer is a vector of logits $\vec{z} = \{z_1, ..., z_M\}$ that are converted into class probabilities according to equation 3. The $[CLS]$ token values are the only token values that are used as input to the softmax layer this can be seen in Figure 7. The number of logits is equal to the number of classes.

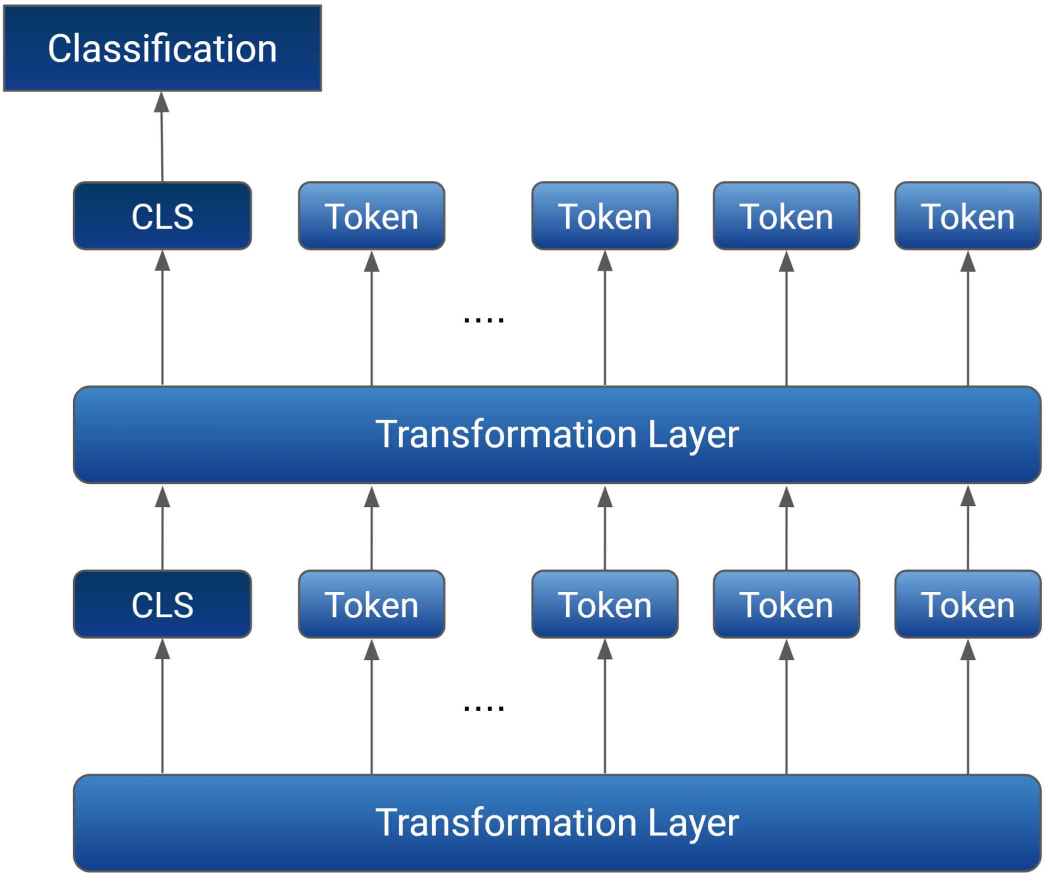


Figure 7: The softmax layer in a BERT model

4 Method

The process as a whole consists of two main areas; free-text processing and creation of SDG basis, which combined add up to the final SDG pipeline. In this project, we have worked with these two main areas as independent projects, since the free-text processing can be viewed as an extension of the creation of SDG basis for the inclusion of free-text data as input. To fully connect the SDG pipeline, some further work needs to be done regarding the WHODrug-coding of ingredients in the reports from which we use free-text verbatims. However, this has not been done in this project.

The end product of the creation of SDG basis is the information that a pharmacist can use to decide which ingredients to include in a QT prolongation SDG, sorted according to level of suspected QT prolonging effect. The input needed is MedDRA-coded ICSRs. The end product of the free-text processing on the other hand is a binary classifier predicting if a free-text verbatim is describing a QT prolonging ADR. In the final pipeline, this is used as a pre-stage extension that allows verbatims describing an ADR as input, even if it is not yet MedDRA coded. If the verbatim is classified as describing a QT prolonging ADR, we wish to code it and include the verbatims' ICSR in the Creation of SDG basis.

The process overview and the sub-modules can be observed in Figure 8. The free-text processing corresponds to sub-modules 1-3 and is described in the sections 4.2-4.4. The creation of an SDG basis corresponds to sub-modules 4-7 and is described in section 4.5-4.12.

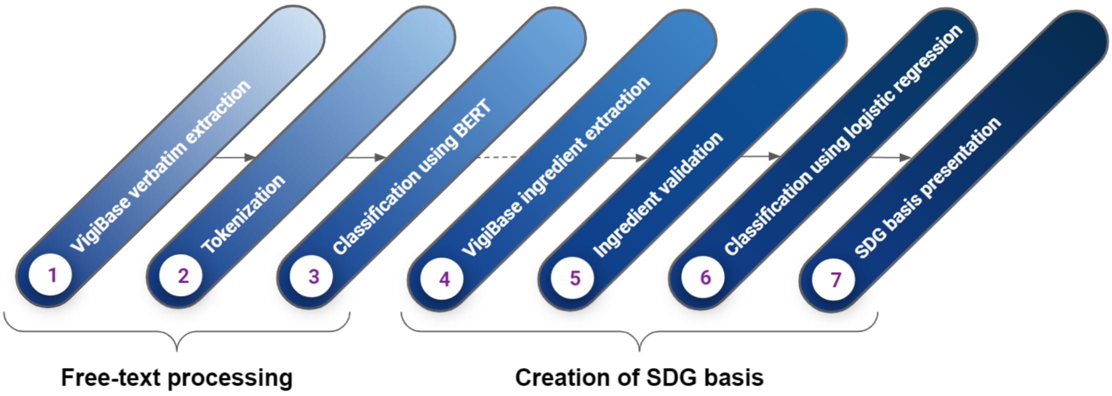

1
VigiBase verbatim extraction
2
Tokenization
3
Classification using BERT
4
VigiBase ingredient extraction
5
Ingredient validation
6
Classification using logistic regression
7
SDG basis presentation
Free-text processing
Creation of SDG basis

Figure 8: Project process scheme

To further explain what is included in the different sub-modules:

1. Extraction of free-text verbatims from VigiBase using language sorting to include only verbatims mainly written in English and a sampling method to reduce the amount of non-QT training data (Section 4.2).
2. Tokenization as verbatim pre-processing (Section 4.3).
3. Binary classification of free-text verbatims using BERT to find out whether or not

a verbatim describes a QT prolonging ADR (Section 4.4).

4. Extraction of active ingredients from VigiBase that indicates a connection to QT prolonging ADRs, together with corresponding relevant information (Section 4.5).
5. Ingredient validation by SPC validation (Set ID extraction using OpenFDA API, SPC scanning and PT coded using a UMC-created pipeline for SPC mining), as well as comparison to CredibleMeds' QT drug list (Section 4.6, 4.9).
6. Ingredient classification trained on manually labeled data using multinomial logistic regression (Section 4.11).
7. Presentation of SDG basis with relevant information and ingredients sorted according to the level of suspected QT prolonging effect (Section 4.12).

4.1 Choice of software

For the VigiBase data extraction and pre-processing, we have used SQL. The Set ID extraction and SPC mining was done using Python 3.8 (the SPC Mining pipeline is Python based). Python was also used for the free-text classification using BERT. For the rest of the implementation we have used C# 9.0 on .NET Core 3.1, where the pre-processed database is connected using Entity Framework. Data has been processed using C# or SQL queries depending on suitability. Plots have been constructed using Microsoft Excel and MathWork's MATLAB.

4.2 VigiBase verbatim extraction

ICSRs can contain a free-text verbatim that describes the reported ADR(s). Pre-coded ICSRs were used as data to train and evaluate the BERT model. The data was extracted from two different databases named `UMCReport_20210103` and `Meddra_20210103`. Both databases are frozen versions of otherwise actively updated databases. The verbatims and the PTs they are coded to, are stored in `UMCReport_20210103` and the latest terminology version of MedDRA is stored in `Meddra_20210103`. In VigiBase there is 10 868 817 verbatim in total, which are reported from different countries and written in different languages. Out of these, 11 216 are coded to PTs in the narrow scope QT Prolongation/TdP SMQ.

In the database `UMCReport_20210103`, each verbatim is associated with a ReactionID which is a unique identifier for each ADR in VigiBase. The reaction-ID was used to match each verbatim to the coded PT term. From the database `Meddra_20210103` the name of each PT was extracted and matched to each PT term. For every verbatim, a label was added. If the verbatim was coded to a PT that is included in the narrow scope QT prolongation/TdP SMQ the label was set to 1 and otherwise it was set to 0, implying no QT-connection.

The extracted data set consisted of the following columns:

- Reaction ID: Unique identifier for each ADR

- Verbatim: Free-text description of an experienced ADR
- PT Code: Identifier for the PT term that the verbatim is coded to
- PT Name: Name of the PT term that the verbatim is coded to
- Label: 1 if the described ADR is coded to a PT within the narrow scope QT prolongation/TdP SMQ, otherwise 0

4.2.1 Language sorting

One of the delimitations in this project is to only include ICSRs written in English, simplifying the NLP method and evaluation. The verbatims extracted from VigiBase are written in several different languages. Non-English verbatims were sorted out from the data set. As a first approach, verbatims from known English-speaking countries were selected. Although this proved efficient, the loss of data was large, heavily affecting the size of training data. Thus our second approach was based on the language sorting process in ”Extracting Adverse Drug Reactions from Product Labels using DeepLearning and Natural Language Processing”[18]. The process utilizes two different strategies to discard non-English verbatims. The first one automatically discards verbatims containing letters not in the Latin alphabet (for example Korean symbols and vowels like ”à,ú,ý”). Secondly, we create a dictionary of all words in the MedDRA LLTs. Each remaining verbatim is then split into separate words which are compared to the dictionary and an English score for the verbatim is calculated. The English score is the percentage of words in the verbatims that are present in the dictionary and is defined as

$$\text{English score} = \frac{\text{Words in dictionary}}{\text{Words in verbatim}}$$

We calculated the English score for each remaining verbatim and discarded all verbatims with a score below 70%. After this language sorting process, 7 790 688 English verbatim remained. 7263 of these are coded to a PT included in the narrow scope QT prolongation/TdP SMQ.

4.2.2 Data sampling

The data is highly unbalanced with 99.9% of the verbatims belonging to the negative class, i.e not included in the narrow scope QT prolongation/TdP SMQ. To cope with this issue and to reduce computation time when training BERT, two approaches to data sampling were investigated. Both approaches utilize the idea of under-sampling where samples from the majority class, in this case, non-QT verbatims, are drawn [21] whereas the minority class is untouched. With the two sampling approaches two different data sets for training was created. The first data set for training was created by using random sampling, where non-QT verbatims are selected at random. The second data set for training was created by using PT distribution sampling where non-QT verbatims are selected while keeping the proportion of non-QT PTs in the data the same. With these two different data sets two different BERT models were trained and their performance compared.

In order to compare both sampling approaches, they need to be evaluated on the same test set. Ideally, the test set should reflect a real world scenario and represent a wide range of different PTs. Therefore the test set was created using PT distribution sampling. The test set was created first, and verbatims not in the test set were then sampled to create the two different training data sets. The test set contained 513547 non-QT verbatims and 2397 QT verbatims. The number of verbatims in each of the training sets are shown in Table 1.

	Nr verbatims for training	
	Non-QT	QT
Random sampling	1042979	4866
PT distribution sampling	1042654	4866

Table 1: Training set distribution for the two sampling approaches

4.3 Tokenization

Before the BERT model can train on the data set each verbatim needs to be processed into a format BERT can handle. Each verbatim needs to be tokenized and padded to the same length. The BERT developers suggest a limit of 32 or 64 tokens in each input [26]. To reduce training time, a limit of 32 tokens was chosen. As a result, information might be lost due to verbatims cut shorter. The $[CLS]$ and $[SEP]$ tokens were also added to the endpoints of each verbatim.

4.4 Classification using BERT

After tokenization, the data for training was split into a training set of 70% and a validation set of 30%. Two different versions of a BERT model were trained were the different training sets based on the different sampling approaches explained in Section 4.2.2. The BERT model works in batches, such that only a subset of the verbatims are processed at the same time. When all batches have been processed, one epoch has passed. The batch size was set to 32 verbatims and the number of epochs to 4. Since the data set is highly unbalanced, we used the F_β-score as the main evaluation metric, focusing on the misclassified rather than the correctly classified verbatims. Based on the results the model does not have a problem with false positives or false negatives. Therefore we chose $\beta = 1$ which will weigh precision and recall equally, such that the F_1-score is used for evaluation.

There are several different types of pre-trained BERT models for different types of problems. For this binary classification problem, the base version ”BERT For Sequence Classification” was selected. The base version has 12 transformation layers and a final softmax layer to compute class probabilities. In order to follow the training progress, the model calculates the cross-entropy loss function between each batch. The loss function was optimized using Adam, an optimization algorithm using SGD for deep learning models [27], and a learning rate $\gamma = 0.00001$.

4.4.1 Predictions

When fully trained, the BERT model will make predictions on the unseen test data set about whether or not the ADR description indicates a QT-connection or not. Since the project goal is to automate the process of creating an SDG, the next step for a QT-predicted verbatim would be to code it to a PT term.

4.5 VigiBase ingredient extraction

For ICSRs pre-coded to MedDRA standard terms, we wished to sort out relevant data to support correlation to QT prolongation. The data was extracted from two different databases named `UMCReport20210103` and `Meddra_20210103`. Both databases are frozen versions of otherwise actively updated databases. `UMCReport20210103` contains ICSRs and `Meddra_20210103` contains the latest version of MedDRA terminology. Data from these databases were extracted, combined and stored on a local database using SQL. To find the relevant reports we set the following requirements on the data:

Date

The extracted reports were submitted on or after January 1st 2018, a restriction set to keep the MedDRA coding version consistent throughout the data.

Exclude foreign reports

To avoid duplicate reports in the data, we chose to only include reports that are written in and submitted from the same country.

Narrow scope TdP/QT prolongation SMQ

A report can include several different ADRs coded to different PTs and LLTs. In these cases, only reactions coded to a PT that is included in the narrow scope TdP/QT prolongation SMQ were selected. To do this, PT terms were selected from the `Meddra_20210103` database with the following requirements:

- Code = 20000001 (SMQ code for QT Prolongation/TdP)
- ScopeID, Term Scope = 2 (narrow scope)
- Term status = A (active SMQ)
- Term level = 4 (hierarchy level: PT)

Drug Characterization ID

We used the Drug Characterization ID to sort out the drugs that are not suspected to have caused the QT prolonging ADR. In our extracted data we set Drug Characterization ID = (1,3) to include only ”suspected” and ”interacting”.

WHODrug

For each ICSR, UMC has validated a trade name for the medicinal product(s) that the patient has taken. Each trade name is linked to an ID in WHODrug. A subset of WHODrug is hosted inside the database `UMCReport20210103` from which we can extract the active ingredients and connected moieties that a specific drug contains. To extract the active ingredient we used the Drug record number and Sequence number 1, see Section 2.5 for definitions. Since there is a many-to-one relationship between the active ingredient and active moiety, we used the active ingredient name to find the active moiety by setting the Sequence number 1 = 1. Since `UMCReport20210103` only hosts a subset of the whole WHODrug dictionary, some active moieties had to be manually searched for in the actual WHODrug dictionary using WHODrug Insight. This will however not be a future issue since the final pipeline will extract data from a different database table with access to all of the WHODrug dictionary.

4.5.1 Sorted data

After the described sorting, 8815 reports remained. Together these reports include 1329 unique ingredients that we wish to validate. The sorted data has the following columns:

- `ReportID` - Unique identifier for a report.
- `UMCValidated_ProductID` - Validated trade name
- `UMCValidated_BaseCompositionName` - Validated active moiety
- `ActiveSubstance` - WHODrug active substance
- `PT_Code` - ADR identifier

We used VigiLyze to evaluate that the amount of extracted data corresponded to the amount when searching accordingly using the VigiLyze search tool.

4.6 Validation against SPC data

To examine if the data extracted from VigiBase is known or suspected to cause QT prolongation, we decided to validate the connection using SPC information from DailyMed. Our approach was to extract a list of Set ID:s, which is a unique identifier for each SPC, for each active ingredient in our sorted VigiBase data. The Set ID:s is then used to access all relevant SPC information and search for indications of QT prolongation reactions.

Set IDs could be obtained by web scraping or by using a suitable Application Programming Interface (API). We found two APIs that handle the SPCs data set, DailyMed RESTful API and OpenFDA Drug API. We opted for OpenFDA since it is more versatile and well-suited for requests based on active ingredient information.

4.6.1 Set ID extraction using OpenFDA

OpenFDA offers several open-source APIs based on the search engine platform Elasticsearch, handling all public FDA data. We have used the weekly updated OpenFDA Drug API that processes Drugs@FDA data which is described as:
”Information about the following FDA-approved products for human use:

- Prescription brand-name drug products, generic drug products, and many therapeutic biological products
- Over-the-counter brand-name and generic drugs" [16]

This corresponds to the SPC data available in DailyMed (although the data is not synchronously updated, hence small differences in the data sets may occur). A request can be sent to an API using an URL with specified search parameters.

Our goal using OpenFDA was to get all SPC Set ID:s for a given list of active ingredients and their corresponding active moieties. The moieties are the base composition for each ingredient, and often they share the same name. For reference, see some examples of active ingredients and corresponding active moieties from our data set in Table 2. As seen, several ingredients can have the same moiety. When they are not the same, the ingredient name is more specific than the moiety.

Ingredient	Moiety
Alfuzosin hydrochloride	Alfuzosin
Acetylsalicylate lysine	Acetylsalicylic acid
Acetylsalicylic acid	Acetylsalicylic acid
Hepatitis b vaccine rHBsAg (yeast)	Hepatitis b vaccine
Ferrous sulfate	Iron

Table 2: Data set examples

We created a script that reads a CSV-file with active ingredients and corresponding moieties as input data. For each active ingredient we use the OpenFDA API to request SPC information. In the request, we insert an API key provided by OpenFDA which we need to be able to make more than 1000 requests per day (otherwise no key is needed) and the name of the active ingredient. The request results in a JSON-file containing SPC data where the active ingredient and/or generic name corresponds to the given active ingredient. Since the highest possible number of resulting SPCs per request is 1000, we must make multiple requests in cases we exceed the limit. In the cases where the API request does not result in any matching SPCs for the active ingredient, we search for SPCs related to the active moiety instead.

For all resulting SPC data, we extract all Set ID:s using the `Regular Expressions` package. The active ingredient name, the list of correlated Set ID:s and a DailyMed search link to the first correlated SPC are added to the JSON-file for each iteration. This file is our resulting output data. The extraction process is described in the following algorithm:

Algorithm Set ID Extraction using OpenFDA

```
Read csv-file with active ingredients as input data
Initiate json-file capturing 'Name', 'Set ID:s' and 'Link'
for each active ingredient do
    Request SPC information for active ingredient using OpenFDA
    if no hits on ingredient name then
        Request SPC information for active moiety using OpenFDA
    end if
    Extract all Set ID:s from SPC information
    Append name, Set ID:s and DailyMed-link to json-file
end for
Resulting json-file is output data
```

4.6.2 SPC scanning and coding

For each Set ID (connected to the active ingredient or active moiety) extracted in the previous step, we wish to access the SPC information, more specifically the sections within the SPCs describing ADRs and ingredients. For this purpose, we used a UMC-created pipeline called ”SPCMining”.

The SPCMining Pipeline is based on two separate master thesis projects previously conducted at UMC: ”Extracting Adverse Drug Reactions from Product Labels using Deep Learning and Natural Language Processing” [17] and ”Mapping medical expressions to MedDRA using Natural Language Processing” [18]. As input, the pipeline takes SPCs from DailyMed in XML-file format and converts them into a JSON-file format. Free-text descriptions of ADRs in the SPC are then coded to MedDRA-codes by an NLP model. The NLP model has two evaluation metrics, micro average and macro average F_1-score. Macro average means that the F_1-score is calculated independently for each class while the micro average will weigh the F_1-score with regards to the performance on each class. The macro average F_1-score$= 0.774$ and the micro average F_1-score$= 0.806$. The ADR verbatims are scanned and coded from the following SPC sections:

- Adverse reactions
- Boxed warnings
- Precautions
- Warnings and precautions
- Warnings

The pipeline processes are described in Figure 9.

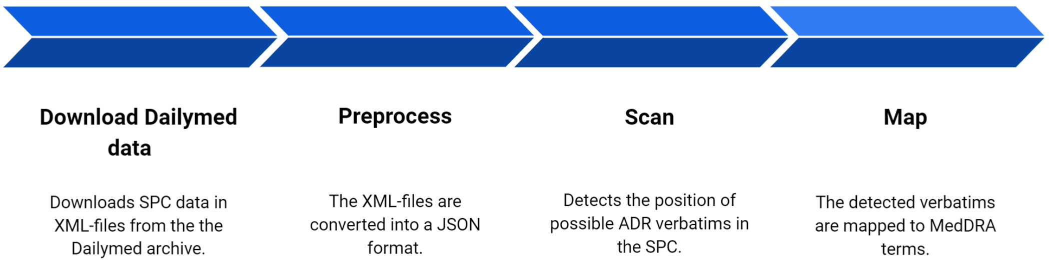

Download Dailymed data
Downloads SPC data in XML-files from the the Dailymed archive.
Preprocess
The XML-files are converted into a JSON format.
Scan
Detects the position of possible ADR verbatims in the SPC.
Map
The detected verbatims are mapped to MedDRA terms.

Figure 9: Flow chart for the SPCMining Pipeline

The number of unique Set ID:s extracted in the previous step was 38 458. We accessed 33 983 of these from a previously mined data set. For the 4475 Set ID:s that was not already mined, we downloaded the raw XML data from DailyMed and ran the pipeline locally on those Set IDs. The pipeline was unable able to mine 373 Set ID:s since the information in those Set ID:s was not complete (the sections that the SPCMining pipeline scans and codes information from were non-existent).

4.6.3 Preferred terms comparison for SPC data

Given our list of active ingredients and corresponding Set ID:s from the previous step, as well as the JSON-files with information of all successfully mined SPCs for these Set ID:s, we now wish to check the SPCs for PTs included in the narrow scope QT Prolongation/TdP SMQ. The process is described in the following algorithm:

Algorithm SPC validation for TdP/QT prolongation SMQ

```
Declare PT-list: PT codes in QT Prolongation/TdP narrow scope SMQ
Read JSON-file with results from 'Set-ID Extraction' as input data, convert to C#
object
for each active ingredient do
    Read list of 'Connected Set ID:s' (for ingredient or moiety)
    for each Set ID do
        Read JSON-file with coded SPC data, convert to C# object
        Check if SPC contains multiple active ingredients
        Find all coded PTs that are included in PT-list
        Check which sections these PTs were found in
    end for
    Present number of successfully mined SPCs
    Present number of successfully mined single-active ingredient SPCs
    Present percentage of SPC:s with hits (total and single-active ingredient only)
end for
Save validation results as output CSV-file
```

We create C# objects for all JSON-files to effectively access only the information needed. For each SPC object, we find all coded PT terms by the route:

`Sections→Mentions→codeds→PtCodes`

Comparing these to our declared list of PT codes, we store information about the number of SPCs with hits (successful comparison). We do not take into account which section the hit is found in (such as ”Warnings” or ”Adverse Drug Reactions”) or which PT we get the hit on (such as ”Long QT Syndrome” or ”Electrocardiogram QT interval abnormal”). Although that information is easily accessed since it might be of interest in future versions of the SDG.

One thing that increases the accuracy of the validation is whether or not the SPCs contain one or multiple active ingredients. If a drug with only one active ingredient is known to cause an ADR (stated in the SPC), we can directly link the reaction to that ingredient. Whereas if the drug has multiple active ingredients, we can not know which one is most likely to be the cause. To label each SPC as multi- or single active ingredient, we examine the route:

`Product→Parts→ActiveIngredients→ActiveMoieties`

In the mined SPC information, all ingredients are listed. Some ingredients are inactive for example water, corn starch or talc. The active ingredients have the corresponding active moieties listed in the mined JSON-file. So to check if an SPC has one single active ingredient, we examine if only one of the listed ingredients has a non-empty list of active moieties.

As output, we extract following data for each ingredient in our input data list:

- Number of successfully mined SPCs
  - How many of these had hits (percentage)
- Number of single-active ingredient SPCs
  - How many of these had hits (percentage)

4.7 Categorization

To get an overview of which ingredients we could validate via SPCs and with what certainty, as well as how they are represented in VigiBase reports, we divided all ingredients into five sub-categories. They are defined as following (the connected reports in Vigibase before and after modification is to be explained in the next section):

1. Validated on a single-active ingredient SPC.
2. Validated on multiple-active ingredient SPCs only.
3. Mined connected SPCs, but not validated.
4. No mined connected SPCs, connected reports in Vigibase after modification.
5. No mined connected SPCs, connected reports in Vigibase only before modification.

4.8 VigiBase occurrences

For the list of ingredients coded to a PT included in the narrow scope QT prolongation/TdP SMQ, we calculate the total number of connected ICSRs in VigiBase for each ingredient (i.e. all reports where the user described an ADR after taking a drug that included that specific ingredient). We also calculate the fraction of these ICSRs that was coded to the narrow scope QT prolongation/TdP SMQ. The resulting percentage describes how many of all reports connected to an ingredient that has a QT-connection:

$$\text{QT occurrences} = \frac{\text{QT coded connected reports}}{\text{All connected reports}}$$

This percentage is not always an accurate measurement of an ingredient's probability to cause QT prolongation/TdP, since we do not know which one of all ingredients connected to the report is causing the ADR. To further improve this parameter, we decided to modify the percentage by adjusting the numerator. Instead of counting all connected reports coded to the narrow scope QT prolongation/TdP SMQ for each ingredient, we temporarily discard reports connected to another ingredient that we have reason to strongly suspect is the cause of the ADR (these reports are still included in the denominator which does not change).

As a threshold to when an ingredient is suspected to be the cause, we choose to include ingredients that we have validated for at least one single-active ingredient SPC, i.e. ingredients in category 1. Thus the ”Modified number of QT coded connected reports” in Equation 8 is all QT coded reports connected to an ingredient, except those where another ingredient included has been validated for a single-active ingredient SPC.

$$\text{Modified QT occurrences} = \frac{\text{Modified number of QT coded connected reports}}{\text{All connected reports}} \tag{8}$$

As an example, ”Aminosalicylic acid” had a total of 28 connected ICSRs. When checking the list of connected ingredients for these 28 reports, they all contained at least one other ingredient that belongs to category 1. Thus they were all discarded and the modified percentage of connected reports is 0. For an example of how Abacavir Sulfate is affected by the parameter modification, see Figure 10.

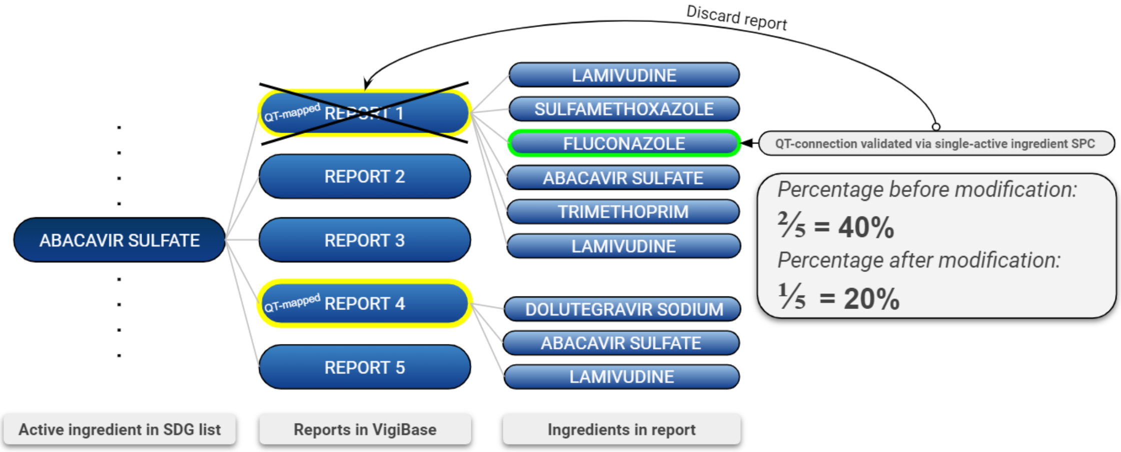


Figure 10: Modification of the VigiBase occurrences percentage for Abacavir Sulfate

After this modification, we were able to separate the sub-category for no mined SPCs into two new sub-categories. Sub-category 4 and 5, where ingredients in category 4 still has VigiBase occurrences after modification whereas ingredients in category 5 do not.

4.9 CredibleMeds comparison

Another source of information that we used to validate our SDG ingredient list is CredibleMeds, see Section 2.9. They provide and manage a list of ingredients with a risk for the user to develop TdP. We contacted CredibleMeds that supplied us with an Excel-file containing this list. The original list consists of 293 ingredients (or a combination of two ingredients) and contains the following information:

- Generic names (ingredient name)
- Drug brand names (partial list)
- Drug Action (e.g. antibiotic, sedative)
- Main therapeutic use (e.g. asthma, cancer)
- Routes administered (e.g. oral, injection)
- Current risk category

We have used the first and last category for evaluation (and later on for ingredient classification improvement). Since we use different ways to rank/categorize ingredients, this list can not be used to directly evaluate our list of ingredients. It does however give us an indication about if the ingredients we have extracted have a known or suspected correlation to QT-prolongation/TdP, and to what risk category it has been assigned by AzCERT. This information is also of value to the pharmacist using the final SDG. The different risk categories used in CredibleMeds are:

- Drugs with known TdP risk

- Drugs with possible TdP risk
- Drugs with conditional TdP risk
- Drugs to be avoided by congenital Long QT

In some cases, the CredibleMeds included some different spelling options and synonyms for ”Generic names”. We altered those to match our ingredient names (if the listed ingredient was in our list). Some alterations can be observed in Table 3. As can be seen, some alterations were easily done by using only one of the suggested spellings, whereas others required using a synonym or non-suggested spelling. It is safe to assume that some ingredients were not matched because of a spelling or synonym we did not know to alter. Some ingredients were listed as a combination in CredibleMeds, where we divided the combination into two separate ingredients to match our format. Thus the modified CredibleMeds list contains 300 ingredients.

Original	After modification
Amphetamine (Amfetamine)	Amfetamine
Eribulin mesylate	Eribulin mesilate
Levalbuterol (Levsalbutamol)	Levosalbutamol
Papaverine HCl	Papaverine hydrochloride
”Fluticasone and Salmeterol”	”Fluticasone” and ”Salmeterol”, respectively

Table 3: Examples of modification of ”Generic Names” in CredibleMeds list

After this modification, we read the data and compares each ”Generic name” to our ingredient names. If it is an exact string match, the program writes that the ingredient has been validated using CredibleMeds onto the local database, as well as to which risk category it belongs. If the ingredient does not match, we repeat the comparison and writing but now using the moiety.

4.10 Manual ingredient labeling

In order to present the SDG ingredients in an order of likeliness to cause QT prolongation, we aimed to train a classification model using logistic regression to categorize ingredients. To get the training data, we received help with the manual classification of a subset of the SDG ingredients.

The manual classification was performed by two pharmacists at UMC. We provided them with a list of ingredients (both got the same list) to categorize independently. That way we could also examine how much manual ranking can vary between professionals. We decided to let the pharmacists use all available information to rank the ingredients as well as possible. For example, they could use SPC information from different countries/centers, VigiBase reports and information on the internet they see as credible. This way we see the effect of using the limited data in our model, such as not being able to access free text information from the SPCs and ICSRs. In our model, we only access the coded data after SPC mining or MedDRA coding, and are also limited to SPCs in FDAs database (such that we can not access SPCs if the active ingredient is not approved at the American market).

The list of ingredients provided contained 110 substances in total. The proportions were 50 substances from sub-category 1, 10 from sub-category 2 and 50 from sub-categories 3, 4 and 5. Other data provided was:

- Instructional guidelines, see Appendix B.
- Country information (from which country the ICSR for each ingredient was received, in decreasing order)
- CredibleMeds QT drug list (the data used for CredibleMeds validation)

Before the pharmacists began their ranking they agreed on a common ranking system. They used DailyMed as their primary source of information. Firstly they would search for the active ingredient and secondly the active moiety. However one of the pharmacists decided to also search for the active ingredient in WHODrug to find medicinal products (that contain the active ingredient) that could be searched for in DailyMed. If the active ingredient could not be validated to have a QT prolonging ADR in DailyMed they would move on to search for SPCs from other countries, research papers or other sources they saw as credible. Each active ingredient was ranked into one of three categories which were:

- Class 1, Strong indication of connection to QT prolongation/TdP based on SPC or other credible source
- Class 2, Weaker indication of connection to QT prolongation/TdP based on SPC or other credible source
- Class 3, No alleged connection to QT prolongation/TdP based on SPC or other credible source

One of the pharmacists had time to rank 100 active ingredients, whereas the other had time to rank all 110 active ingredients. For the 100 ingredients that they both ranked, their labels varied for 38, information used for calculating the Inter-Annotator Agreement (IAA). We made two versions of the training data, one where we used an average in the case where their labels disagreed and one where we used the label indicating a stronger QT-connection. We decided to proceed with the latter version since the pharmacist who labeled the stronger connection has provided reliable sources to motivate the decision (which the other one may have missed). Also, we would rather falsely include ingredients that should not be in the final SDG, than exclude an ingredient with a QT-connection.

4.11 Ingredient classification using logistic regression

To train a model on the manually labeled data to predict the rest of the unlabeled ingredients, we implemented a classifier using multinomial logistic regression (see Section 3.1 for theory). We constructed the classifier to work for any number of features (and classes, but we consistently used three classes since the available training data is labeled to three classes).

4.11.1 Input Data

The labeled input data (110 data points) were kept in a separate database table from the unlabeled. The classification process begins with reading from the labeled data table as input data. As y-data, the three-class labels are read and transformed to target vectors as:
$y = 1 \rightarrow \vec{\tau} = [1,\ 0,\ 0]$
$y = 2 \rightarrow \vec{\tau} = [0,\ 1,\ 0]$
$y = 3 \rightarrow \vec{\tau} = [0,\ 0,\ 1]$

As x-data we choose suitable (best describing QT-connection) parameters as features. A feature vector with values for each ingredient is given as $\vec{a}_n = (a_{n,1}, ..., a_{n,i}, ..., a_{n,I})$ where I is the number of ingredients, $i \in \{1, I\}$, and N is the number of features, $n \in \{1, N\}$. The x-data given by $\vec{x}_i = (a_{1,i}, ..., a_{N,i})$ for each ingredient is the feature values for that ingredient. We began with a basic 2-feature model ($N = 2$), using ”Validated single-active ingredient SPC percentage” and ”Modified QT-coded reports in VigiBase percentage”. When reading the data, we want to normalize each feature value such that it is all presented on a scale from 0 to 1. The normalization for each feature value $a_{n,i}$ is done according to the equation

$$a_{n,i(norm)} = \frac{a_{n,i}}{\vec{a}_{n(max)}},$$

where $a_{n,i(norm)}$ is the normalized feature value and $\vec{a}_{n(max)}$ is the maximum feature value in the feature vector $\vec{a}_n$.

In a second version of the classifier, we included ”CredibleMeds risk group” as a third feature, in order to use the research done by AzCERT in our predictions as well. Since this feature is described alphabetically, we need to convert it to numeric values within the same scale as our normalized numeric features. In consultation with a pharmacist, we set the following data conversion for the CredibleMeds risk groups:

- Not present in CredibleMeds $\rightarrow a_{n,i} = 0$
- Drugs to be avoided by congenital Long QT $\rightarrow a_{n,i} = 0$
- Drugs with conditional TdP risk $\rightarrow a_{n,i} = 0.8$
- Drugs with possible TdP risk $\rightarrow a_{n,i} = 0.9$
- Drugs with known TdP risk $\rightarrow a_{n,i} = 1$

The reason that ”Drugs to be avoided by congenital Long QT” was set to be ignored is that it does not have any proven QT prolonging effects, but rather adrenaline-like effects. To present an example of an input data point for the 3-feature version, consider the ingredient ”Atomoxetine”. The data extracted for ”Atomoxetine” from VigiBase, CredibleMeds and SPC validation results in:

- Validated single-active ingredient SPC percentage: 95.65% $\rightarrow x_1 = 0.9565$
- Modified QT-coded reports in VigiBase percentage: 2.52% $\rightarrow x_2 = 0.0252$
- CredibleMeds risk group: Drugs with possible TdP risk $\rightarrow x_3 = 0.9000$
- Manual label: $y = 1$ $\rightarrow \vec{\tau} = [1,\ 0,\ 0]$

Thus the labeled input data point for ”Atomoxetine” would be:
$(\vec{x};\vec{\tau}) = \Big([0.9565, 0.0252, 0.9000];\ [1, 0, 0]\Big)$

4.11.2 Data division and cross-validation

The labeled input data is divided into training- and validation data, where we have set aside 20% for validation and 80% for training. The training data is used to update the weights and biases to improve the prediction (minimize the mean squared error (MSE)), whereas the validation set is used to evaluate the trained algorithm on unseen data by measuring the performance (MSE and accuracy).

Since the manually labeled data is very limited, we have used 5-fold cross-validation (see Section 5.2.3 for theory) to train the final model, on the whole, labeled data set while still estimate the performance. To understand the procedure, see Figure 11. The whole set of 110 labeled data points are divided into 5 subsets of 22 points each. For 5 iterations (folds), each subset is held out as a validation set and the model is trained on the remaining 80%. The MSE $\bar{\varepsilon}$ and accuracy α are calculated for each iteration and the average over the 5 folds, ε_{cv} and α_{cv}, is used as the final performance estimation. After this 5-fold cross-validation, the final model is trained on all of the labeled data.

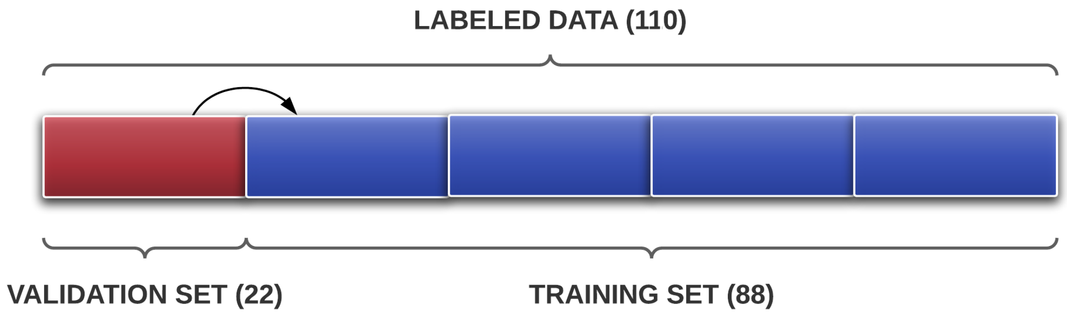


Figure 11: 5-fold cross-validation for the SDG classification, first fold

4.11.3 Epochs and learning rate

An epoch is a term describing one iteration over all data points in a set, so the number of epochs describes how many times the algorithm will train the model on all of the training data set. We have kept a consistent number of 1000 epochs while tuning the learning rate

γ. The learning rate affects how strongly the weights will be updated when presented new data. To decide a suitable learning rate, we plotted ε_{cv} over varying γ and chose the minimum. This tuning was performed for the 2-feature as well as the 3-feature model.

4.11.4 Training

The training consists of updating a weight/bias-matrix using SGD, see Section 3.1.3 for theory. The matrix is initiated with size $(N+1)$xM, where N is the number of features and M number of classes and is initially filled with small randomly assigned values. Since the 3-feature model resulted in a higher accuracy and lower error than the 2-feature model, we decided to proceed with that version.

Let us return to the example of ”Atomoxetine”. After 1000 epochs, the weight/bias-matrix is used to calculate the logits z using Equation 2 for each class. Running the logits through the Softmax function 3 results in following pseudo-probabilities $p_m(\vec{z})$:

- Class 1: $p_1 = 0.8937$
- Class 2: $p_2 = 0.0958$
- Class 3: $p_3 = 0.0105$

In this case, the model prediction is strongly towards class 1. Since that is also the input label y, this would be a correct prediction.

4.11.5 Prediction of unlabeled data

The final 3-feature trained model is used to predict and estimate the QT-connection for all unlabeled ingredients, writing the predicted class onto the data table.

4.12 Final SDG basis

Since we have little to no reason to suspect a QT-correlation for ingredients in category 5 (no mined SPCs and no VigiBase occurrences after modification) labeled to class 3 (no indication of QT connection), we decided to discard these substances from the SDG ingredient list, resulting in a reduction from 1329 to 1086 suspected ingredients.

The final SDG basis consists of these 1086 ingredients with the following information for each:

- Ingredient name
- Moiety name
- SPC validation search subject (ingredient or moiety)
- CredibleMeds comparison search subject (ingredient or moiety)
- Number of reports in VigiBase
- VigiBase occurrences percentage

- VigiBase occurrences percentage after parameter modification
- Number of mined connected SPCs
- Percentage of validated SPCs
- Number of single-active ingredient SPCs
- Percentage of single-active ingredient validated SPCs
- Flag if only multi-active ingredient SPCs are mined
- CredibleMeds risk group (if occurring)
- DailyMed link to the first connected SPC

We have used the manually labeled data to train the broad classes for ingredient sorting. To refine the sorting, we also included additional factors with an impact on the predicted QT-connection. The final order of relevance for the SDG ingredient presentation is sorting according to:

1. Ingredient classifier prediction, ascending order
2. Percentage of single-active ingredient validated SPCs, descending order
3. SPC validation search subject, ingredient before moiety
4. Percentage of all validated SPCs, descending order
5. Modified VigiBase occurrences percentage, descending order
6. Original VigiBase occurrences percentage, descending order
7. Number of mined connected single-active ingredient SPCs, descending order
8. Number of mined connected SPCs, descending order
9. Number of reports in VigiBase, descending order

The final sorted list of ingredients was exported as an Excel-file for pharmacists to use as a basis for creating an SDG. Since we have kept a broad inclusion, the pharmacist can decide upon thresholds for different parameters to narrow down the number of ingredients if wanted.

5 Performance metrics

When the performance of a supervised machine learning model is evaluated, the predicted class labels are compared to the annotated labels. In this section, we explain the metrics used for performance evaluation for the different classification models, as well as an agreement measurement used for manual classification analysis.

5.1 Free-text classification evaluation

The free-text classification of verbatims is a binary classification problem and the metrics described in this section were used to evaluate the performance. For the binary classification problem, there are four prediction outcomes:

- True positives (TP): The model and the annotated label agree that a data point belongs to the positive class.
- True negatives (TN): The model and the annotated label agree that a data point belongs to the negative class.
- False positives (FP): The model predicts that a data point belongs to the positive class, however the data point belongs to the negative class
- False negative (FN): The model predicts that a data point belongs to the negative class, however the data point belongs to the positive class

From these outcomes different evaluation metrics can be constructed. The choice of evaluation metric depends on the type of problem and the class distribution of the data set.

5.1.1 Confusion matrix

Confusion matrices are used for visualizing the model predictions. A perfect classifier would have a confusion matrix where every non-diagonal element is 0.

		Predicted	
		0	1
Actual	0	TN	FP
	1	FN	TP

Table 4: Confusion matrix

5.1.2 Precision and Recall

In order to take false positives and false negatives into account when evaluating a model, recall and precision can be used. Precision measures the fraction of correctly predicted data points that are predicted as positive, i.e. taking the false positive data points into account. Precision is defined as

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

Recall on the other hand takes the false negatives into account and measures the fraction of correctly predicted data points that belong to the positive class.

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

The maximum value for precision and recall is 1, which implies a perfect classifier. A value closer to 0 indicates that the model has problems with false positives or false negatives respectively.

5.1.3 F$_\beta$-Score

F$_\beta$-Score is a measure that combines precision and recall into a single measure and is given as

$$F_\beta = (1+\beta^2)\frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall} \tag{11}$$

The parameter β weighs precision against recall. If $\beta > 1$, recall is weighed higher than the precision. Similarly, $\beta < 1$ weighs precision higher than recall.

5.2 Ingredient classification evaluation

For evaluation of the multinomial logistic regression classifier, we have used MSE and accuracy as performance metrics. The final performance evaluation is done using k-fold cross-validation, where the metrics are an average over the k number of folds.

5.2.1 Accuracy

The model accuracy is a measurement describing the fraction of correct predictions relative to the labels. The accuracy α is given as

$$\alpha = \frac{\text{Number of correctly labeled data points}}{\text{Total number of data points}} \tag{12}$$

5.2.2 Mean squared error

The MSE measures the average of the squared difference between the targets and the probabilities. For each estimated data point, the squared error ε is given as:

$$\varepsilon = \sum_{m=1}^{M} (\tau_m - p_m)^2$$

Thus the MSE is calculated as:

$$\overline{\varepsilon} = \frac{\sum_{n=1}^{N} \varepsilon_n}{N} \tag{13}$$

5.2.3 k-fold cross-validation

While we train the model on a part of the labeled data known as training data, we also set aside a part for validation. The purpose of this data is to evaluate the model performance on data unseen by the algorithm. A higher fraction of training data results in a better trained algorithm, whereas a higher fraction of validation data gives lower variance in the estimated error and accuracy. This trade-off is especially important when the labeled data is limited. A way to evaluate the model without having to set aside validation data, allowing training on the whole labeled data set, is using k-fold cross-validation:

Algorithm k-fold cross-validation

```
1: Split the labeled data into $k$ batches of validation data
2: for each validation batch do
3:     Train the model on the other $k-1$ batches of data
4:     Evaluate the model on the validation batch, store error and accuracy
5: end for
6: Estimate the model performance on unseen data by calculating the k-fold cross-
   validation error and accuracy
7: Train the final model on all labeled data
```

The k-fold cross-validation error and accuracy are given by taking the average over all k folds for Equations 12 and 13, resulting in the cross-validation metrics:

$$\alpha_{cv} = \frac{\sum_{l=1}^{k} \alpha_l}{k}, \quad \varepsilon_{cv} = \frac{\sum_{l=1}^{k} \overline{\varepsilon}_l}{k}$$

5.3 Cohen kappa

When analyzing the level of agreement between two annotators, a useful measurement is the IAA. We have used this measurement for the level of agreement between two pharmacists who were tasked to individually label a list of ingredients.

There are different varieties of IAAs depending on the number of annotators. Since we have a pair of annotators (the two pharmacists), we used the Cohen kappa metric given as:

$$\kappa = \frac{P_0 - P_e}{1 - P_e} \tag{14}$$

P_0 is the relative measure of agreement, i.e. the percentage of all labels that the pair agreed upon. P_e is the hypothetical probability of chance agreement, i.e. the expected agreement if the annotators would label completely at random. This estimation is obtained using a per-annotator empirical prior over the class labels[28].

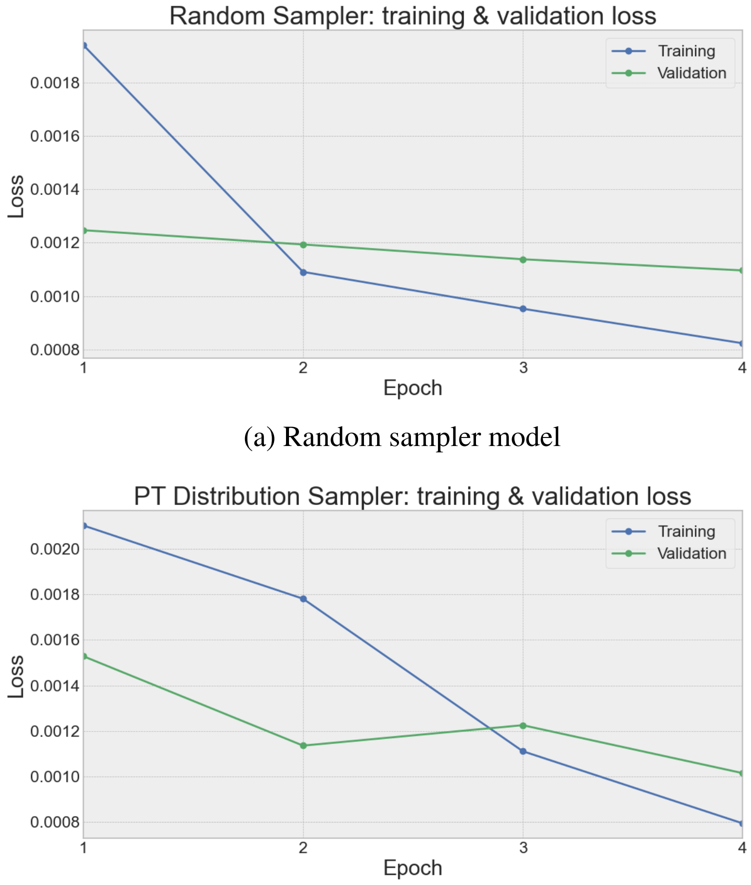
6 Results

6.1 Free-text processing

After verbatim extraction and language sorting, 7 790 688 English verbatims remained, corresponding to about 70% of the original data. 7263 of these are coded to QT prolonging PTs. For the binary classification of free-text verbatims, the following sub-sections will go through the obtained performance measures and results.

6.1.1 Training and validation loss

Between each epoch, the cross-entropy loss function was calculated on the training and validation set. During training and validation, the loss function is minimized using Adam optimizer. The training loss is slightly higher than the validation loss for the first epochs. The training loss decays and becomes lower than the validation loss for the final epochs. The training and validation loss for both sampling approaches can be seen in Figure 12.



(a) Random sampler model

(b) PT Distribution sampler model

Figure 12: Training and validation for the different BERT models

6.1.2 Prediction on test set

The two different BERT models were trained on data created using the two different sampling approaches. The models were tested on the same test set containing 2397 verbatims coded to a PT included in the narrow scope TdP/QT prolongation SMQ and 513547 verbatims coded to other PTs. In table 5 the confusion matrices are shown for both models

		Predicted	
		0	1
Actual	0	513510	37
	1	22	2375

(a) Random sampling

		Predicted	
		0	1
Actual	0	513497	50
	1	22	2375

(b) PT distribution sampling

Table 5: Confusion matrix for Random sampling model and PT distribution sampling model

From the confusion matrices precision, recall and F_1-score can be calculated. For the model that was trained on random sampling data set the $F_1 = 0.9877$ and for the model train on PT distribution sampling data set $F_1 = 0.9850$. Table 6 shows precision, recall and F_1-score.

	Random Sampling	PT Distribution Sampling
Precision	0.9847	0.9794
Recall	0.9908	0.9908
F_1-score	0.9877	0.9850

Table 6: Performance measures

6.2 Set ID Extraction and SPC Mining

After the VigiBase data pre-processing we had a data set of 1329 active ingredients, each corresponding to one active moiety, for which we searched Set ID:s to all connected SPCs. Using OpenFDA, we extracted Set ID:s connected to the ingredient for 944 ingredients and Set ID:s connected to the moiety for 88 ingredients, see Table 7. The remaining number of ingredients with no connected Set ID:s is 297 (22.3%). The number of connected Set ID:s for each ingredient varies widely, up to an order of thousands.

API search subject	Amount with connected Set-ID:s	Percentage of all ingredients
Active ingredient	944	71,03 %
Active moiety	88	6,62 %

Table 7: Results from Set ID Extraction

The total number of Set ID:s extracted was 72 617, with 38 425 unique Set ID:s resulting in 38 052 unique SPCs successfully mined (since 373 Set ID:s were incomplete and therefore unsuccessfully mined). To analyze how many of the duplicates are due to moiety-connected Set IDs (which often share the same name as an active ingredient), we analyze the successfully mined SPCs.

6.3 SPC Validation

Looking at the total set of successfully mined SPCs, we wished to analyze the presence of duplicates, i.e. our mined SPCs that are connected to multiple substances, see Table 8. There are several explanations to the presence of duplicates:

- **Multi-active ingredient drugs**. For example, the SPC for the drug ”Dolishale - levonorgestrel and ethinyl estradiol tablet” belongs to substances ”Levonorgestrel” and ”Estradiol” (both ingredient-connected).
- **Moiety-based identical searches**. For example, the SPC for the drug ”Levofloxacin tablet” belongs to the ingredient ”Levofloxacin” but also ”Levofloxacin hemihydrate” and ”Levofloxacin Mesylate” since we did not find any SPCs directly connected to those ingredients so that the connected lists are connected to the moiety ”Levofloxacin”. Hence the three connected lists of Set-ID:s are identical.

Looking at the 3341 unique moiety-connected SPCs, only 74 was not in an ingredient-connected search as well. Although the proportion of moiety-based searches are small, so a lot of duplicate SPCs is due to multi-ingredient drugs.

API search subject	Total nr of checked SPC:s	Unique SPC:s within search subject
Active ingredient	67624	37978
Active moiety	4475	3341
Both (All SPC:s)	72099	38052

Table 8: Number of SPCs, unique and in total, for each search subject

6.3.1 Multiple active ingredients

When examining how many of all mined SPCs that contains multiple active ingredients, the results are presented in Table 9.

Total nr of SPCs	Nr of multi-ingredient SPCs	Nr of single-ingredient SPCs
72099	30006	42093

Table 9: Number of single- and multiple ingredients SPCs

Out of the 1329 active ingredients, 70 had only multi-ingredient SPCs connected. These are flagged with a warning since the results are more unreliable than for single-ingredient validation.

6.4 Categorization

After categorizing the ingredients based on SPC validation and VigiBase occurrences, see category definitions in Section 4.7, the ingredients were divided into the five categories. The category proportions can be seen in Table 10.

Category	Nr of ingredients in category	Percentage of all ingredients
Category 1	302	22.72 %
Category 2	50	3.76 %
Category 3	503	37.85 %
Category 4	209	15.73 %
Category 5	265	19.94 %

Table 10: Division to categories based on SPC validation and VigiBase occurrences

6.5 CredibleMeds comparison

After comparing if our listed ingredients were also presented in CredibleMeds' list of ingredients with a risk of QT prolongation/TdP, the resulting matches can be seen in Table 11.

Ingredient match	Moiety match	Nr of unmatched ingredients in CredibleMeds' list
208	158	71

Table 11: Comparison to CredibleMeds' list of ingredients with QT-correlation

Out of the 300 ingredients, all but 71 were listed in our data. We believe that this number could be decreased if a pharmacist would go through all data and check for alternate synonyms/spellings. Another reason that some of these ingredients are unrepresented in our data is that no reports connected to those ingredients have been collected to VigiBase after 2018 (thus the ingredient was never represented in our original VigiBase data set).

6.6 Manually labeled data

For the 100 active ingredients that both pharmacists had ranked, they disagreed on the ranking for 38 active ingredients. Out of these, they widely disagreed on 8 active ingredients, meaning that one ranked the active ingredient to class 1 and the other to class 3. The manually labeled ingredients can be observed as data points plotted for VigiBase occurrences and single-active ingredient SPC validation in Figure 13, where the marker color represents the different labels or if the ingredient label was disagreed upon by the two pharmacists.

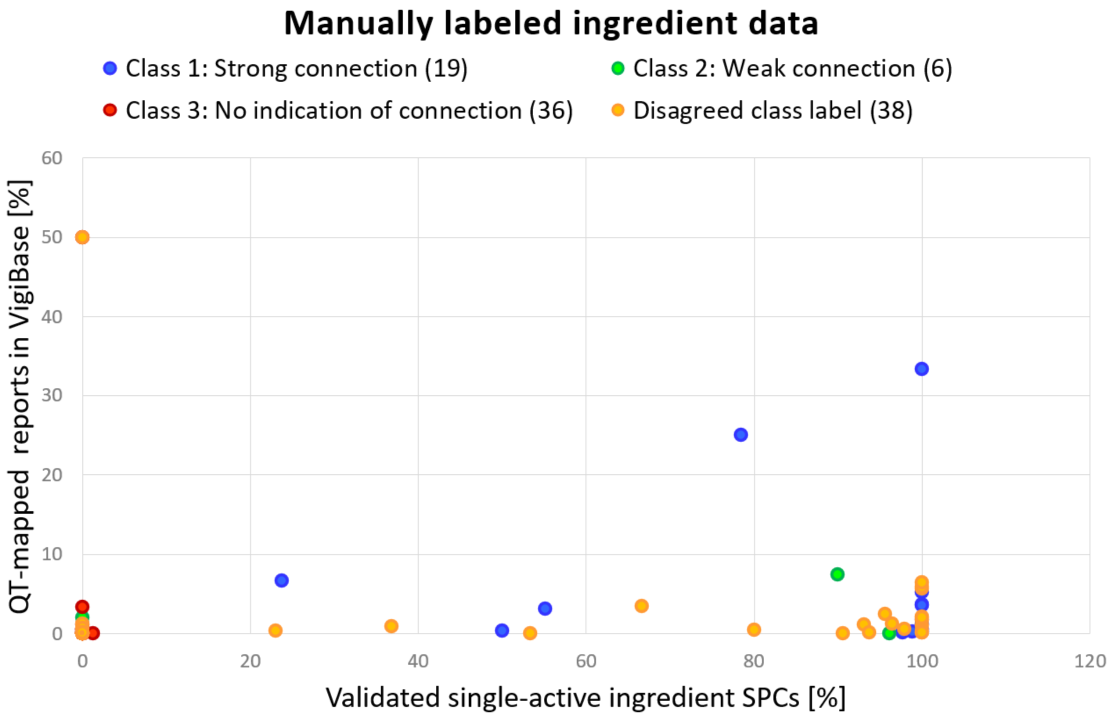


Figure 13: Manually labeled data plotted for VigiBase occurrences and single-active ingredient SPC validation

To further analyze the level of agreement, we calculated the IAA. This measurement tells us more about the actual level of agreement than percentages only, since it also includes the possibility of the agreement occurring by chance. Since we have a pair of annotators (the two pharmacists), we use the Cohen kappa metric, Equation 14, resulting in $\kappa = 0.396$. The measurement expresses the level of agreement on a scale from -1 (zero agreement) to 1 (absolute agreement). $\kappa = 0$ represents the level of agreement expected if the labeling was done at complete random. To interpret the result, the agreement is above 0, which is to be expected since the labeling was not done at random. Although it is relatively far from absolute agreement, indicating that the level of disagreement is high and the labeled data variance is large.

Of the 110 active ingredients, we could not find any SPCs using OpenFDA and the SPC Mining pipeline for 14 of them. The pharmacist that directly searched for active ingredients or moiety could not find these SPCs either, however, the pharmacist who also used WHODrug could find SPCs for 7 out of the 14. This implies that there is a loss of information when not doing manual searches varying from the main method.

6.7 Classification using logistic regression

6.7.1 Features

For each ingredient, we used data from VigiBase and the results of the SPC validation and CredibleMeds comparison to derive parameters that were used as features for the ingredient classification:

Vigibase Occurrences

From VigiBase we derived three parameters related to the number of reports from 1st of January and on-wards each ingredient occurs in. For each ingredient we calculated the total number of reports the ingredient occurs in. We also calculated how many of these reports had PTs included in the narrow scope QT Prolongation/TdP SMQ. From these parameters, a percentage of QT-related reports was calculated for each ingredient, which we modified for improved precision.

SPC Validation

For each ingredient, four parameters were extracted from the results of the SPC validation. The total number of SPCs and the number of single-ingredient SPCs, as well as the percentages of all SPCs and single-ingredient SPCs that the SPCMining Pipeline coded to PTs included in the narrow scope QT prolongation/TdP SMQ.

CredibleMeds Risk Group

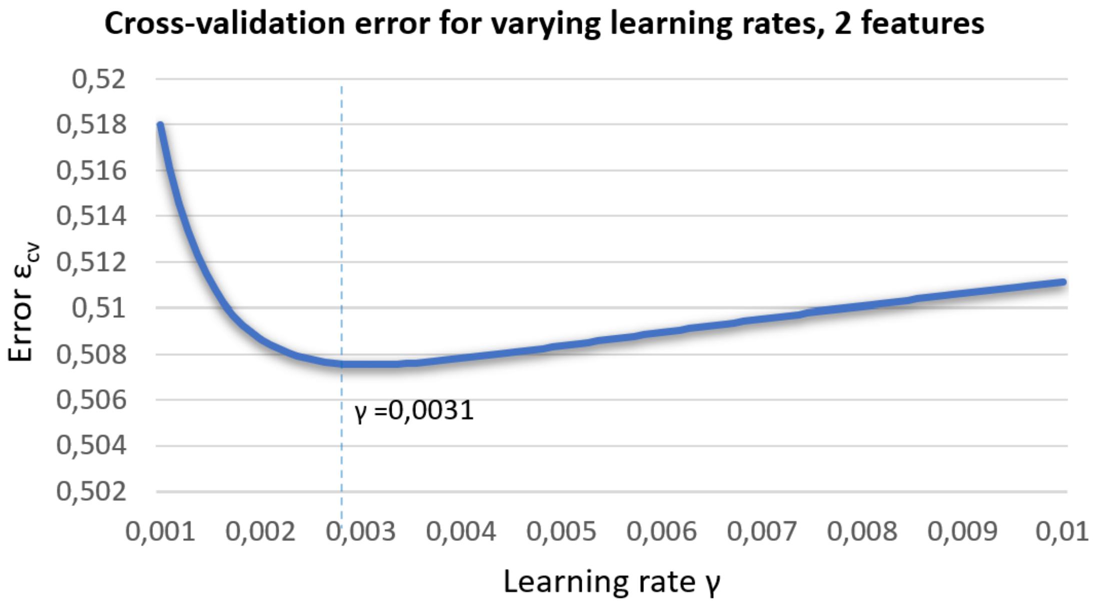
After ingredient comparison to CredibleMeds' list, we use the assigned risk group as a parameter for how strong the ingredient's correlation to QT prolongation/TdP is estimated to be.

The following parameters were used as a basis for classification

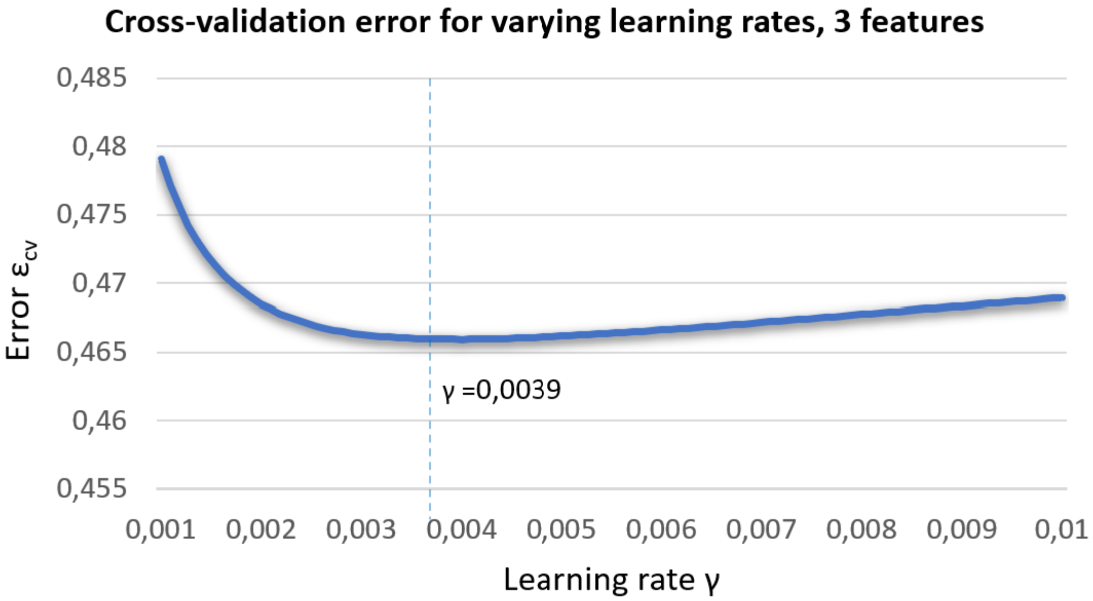
- `QTPercentage_InReports_Modified` - Percentage of QT connected reports where ingredient occurs, after modification
- `Validated_SinglePercentage` - Percentage of single ingredient SPCs coded to a PT included in the narrow scope QT Prolongation/TdP SMQ
- `CredibleMeds_RiskGroup` - States if an ingredient is present in the CredibleMeds list and if so, to which risk group

6.7.2 Choice of learning rates

For a set number of 1000 training epochs, we wanted to tune a learning rate to best train the classifier, i.e. minimize the cross-validation error ε_{cv}. By plotting ε_{cv} over varying γ we identified the minimum, see Figure 14, thus we used $\gamma = 0.0031$ for the two-feature classifier and $\gamma = 0.0039$ for the three-feature one.



(a) Classifier using two features, 1000 epochs



(b) Classifier using three features, 1000 epochs

Figure 14: Choice of learning rates

6.7.3 Performance

Training the classifier for 1000 epochs with said learning rates resulted in the following performance metrics results:

	Two-feature model	Three-feature model
γ	0.0031	0.0039
α_{train}	0.6545	0.6909
ε_{train}	0.4828	0.4380
α_{cv}	0.6545	0.6818
ε_{cv}	0.5075	0.4660

Table 12: Ingredient classifier performance metrics

As often is the case using this kind of learning algorithms, the performance is somewhat higher for the training data than for the validation data. This is simply explained by the fact that the model was trained and adapted to the training data, whereas the validation data is unknown to the model.

6.7.4 Classification of unlabeled data

When using the 3 feature-model trained on all labeled data to make predictions on all ingredients (labeled and unlabeled), the result can be seen in Figure 15, where each dot represents an ingredient and each axis a model feature. Out of the 1329 ingredients, 390 were labeled as class 1 (strong connection), 2 as class 2 (weak connection) and 937 as class 3 (no indication of connection). The colors represent the different labels. From the 937 ingredients labeled as class 3 by the model, 243 were discarded since they also had no validated connected SPCs and no QT-related VigiBase occurrences after modification. The resulting number of ingredients for the SDG basis is 1086.

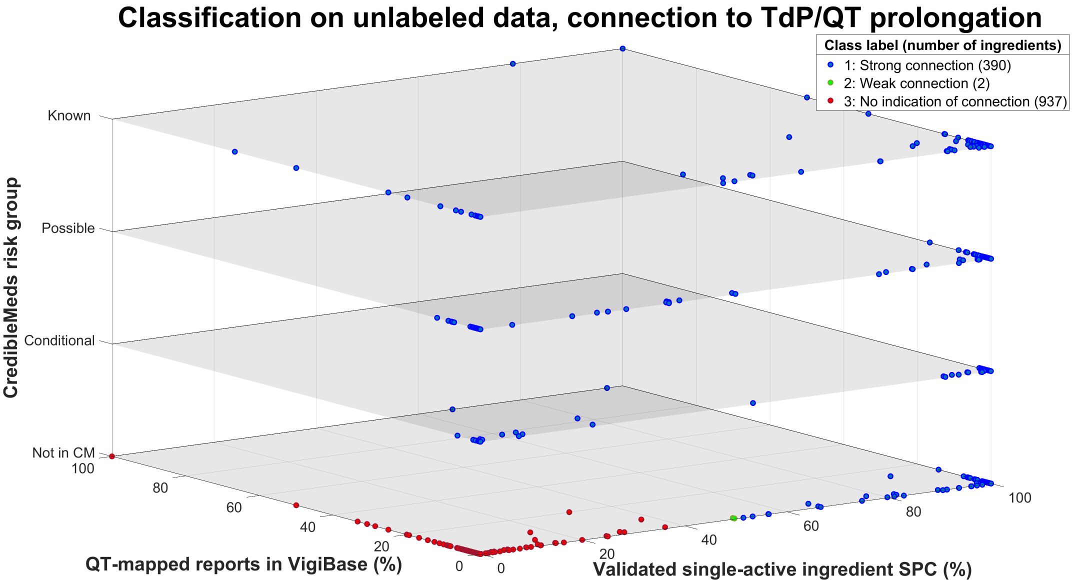


Figure 15: Classification of SDG ingredients using logistic regression

To see how the model predictions relate to the manually labeled data, it can be compared to Figure 16 showing the labeled ingredients in a similar plot with the features as axis. Here, the colors corresponds to the manual labels instead of model predictions. These manually labeled data points are also included in Figure 15, colored after model predictions instead.

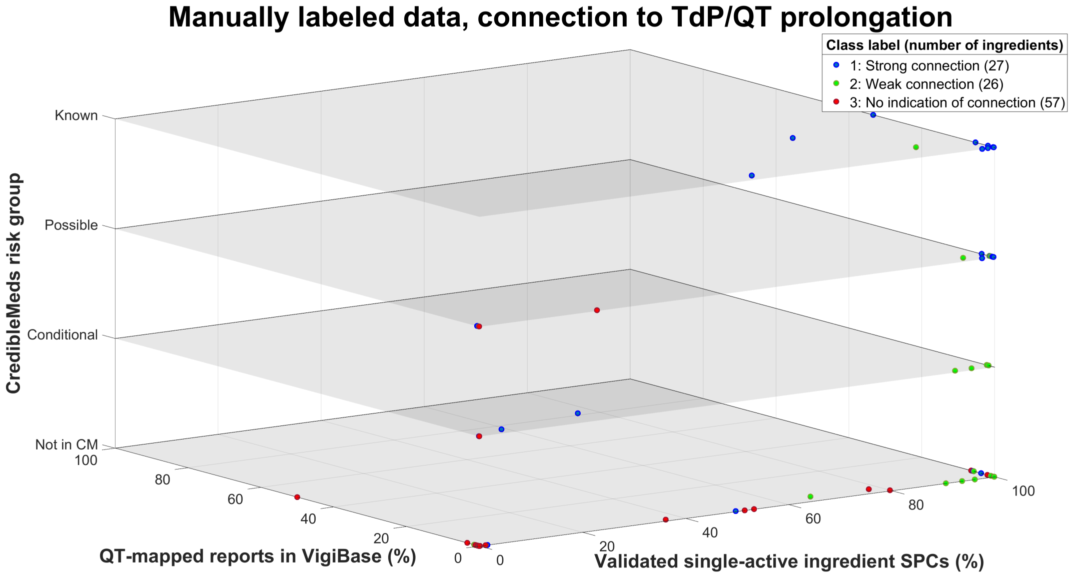


Figure 16: Manually labeled data (if disagreement, the label indicating a stronger QT-connection was chosen)

Since we chose to train the model heavily for the CredibleMeds risk groups (by giving the different risk groups values closer to 1 at the normalization stage), all ingredients included in one of the risk groups we counted as an indication of QT prolonging effects are labeled to class 1. All ingredients are separated by hyperplanes into three prediction zones. The prediction zone for class 2 is very limited, resulting in a small fraction of ingredient predictions. For the ingredients not in the CredibleMeds list, all with a low single-active ingredient SPC validation percentage are predicted as class 3.

7 Discussion

In this section, we discuss the methods used and the results obtained, as well as noted sources of error. We suggest possible improvements, extensions and areas for further investigation.

7.1 Choice of terminology and dictionary

The main reason that we chose MedDRA as medical terminology and WHODrug as drug dictionary is that it is globally recognized standard, hence the VigiBase ICSRs are coded using MedDRA terminology, and the drugs and ingredients described in the reports are referenced using WHODrug. We are content with the usage and flexibility since they are both used internationally and developed to support pharmacovigilance projects. The MedDRA hierarchy allows us to choose the level of specificity, and the SMQ groupings were of utter value for this project. It also makes it easy to include other ADRs by simply choosing a different SMQ as the target.

7.2 Choice of PT terms

We chose to only include PTs in the narrow scope TdP/QT prolongation SMQ. The limit was set after discussions with pharmacists at UMC. Changing to the broad scope would have included PTs that are not exclusively linked to QT prolonging ADRs. An example from the broad scope is the PT ”Cardiac arrest” which a prolonged QT might lead to, however is not the exclusive cause of. A selection of PTs from the broad scope could be included in the project to slightly broaden the scope if wanted.

7.3 Free-text processing

The goal for the free-text processing was to construct a binary classifier that predicts if a free-text verbatim describes an ADR included in the narrow scope TdP/QT prolongation SMQ. According to the results in Section 6.1.2, the classifier can perform this task with only a few errors.

7.3.1 Precoded verbatims

The verbatims used to train the BERT model were gathered from pre-coded ICSRs. As a basis for the coding, a pharmacist has looked at the report as a whole, so the coding may have been done using more information than the free-text verbatim alone. In some cases, the ADR described in the verbatim might not correlate with the coded PT. This makes it harder for the model to make predictions on the verbatim.

Another limitation with the ICSRs is that they do not necessarily have to contain a free-text description of the reported ADRs. This is because there are different reporting practices and legal restrictions in different countries and regions. An example is the reporting practices within EMA where free-text descriptions are written but not always shared with

VigiBase. The missing free-text descriptions limit which ICSRs the classifier can be used on. To be able to use all reports as input (with our without free-text parts), the model could be extended to also take the reported PT into account. Although this might prevent UMCs possibility to validate the reported PT.

7.3.2 Language sorting

In this project, we choose to only include verbatims written in English. This limit was set to ensure that we can understand what the verbatims describe and because no BERT model can process several languages at once. To include non-English verbatims, an NLP model that can process different languages or a translation model needs to be investigated. After discussions with a pharmacist at UMC, a translation model was not further investigated. The reason for this is that translating medicinal terms has proven to be challenging and we wish to avoid compromising the data quality.

The current language sorting approach utilizes a dictionary. Around 30% of all extracted verbatims were discarded when this approach was applied with a 70% threshold for the English-score. The threshold was chosen to allow verbatims containing some non-English words, e.g. Latin words, and after manual review, we are content with the language sorting threshold. If we compare our sorting approach to simply selecting verbatims from English speaking countries, we can include up to 90% more data. Thus the more complex sorting approach is preferable.

7.3.3 Data sampling

The two sampling approaches were implemented to reduce execution time for the training and to make the training data set less unbalanced. Random sampling is easier to implement, however, it might produce a data set that does not reflect the real world scenario. Distribution sampling on the other hand demands a more complex implementation, but the data set reflects the real world better. From the confusion matrices in Table 5 and the F_1-score in Table 6, these differences do not seem to affect the results since the difference in incorrectly classified verbatims is relatively small. Both sampling approaches perform very well for all evaluation metrics.

An alternative to just sampling from the distribution of PTs is to take reporting country or reporter qualification (reported by e.g. consumer, physician, or pharmacist) into account as well. Since the data set is highly unbalanced, some sort of measure should be applied to cope with this problem. The implemented sampling approaches are undersampling the non-QT verbatims. Another approach could be to oversample the QT verbatims by duplication. Because of the simple implementation and superior results, we consider the random sampling approach as preferable.

7.3.4 Misclassification

The majority of verbatims misclassified are the same for both sampling approaches. In Table 13, some examples of verbatims that both model versions (with the two different sampling approaches) misclassified are shown. The first three verbatims in the table are all coded to QT prolonging ADRs but the model predicts them incorrectly. For the following three, the situation is reversed. Regarding probable causes for the misclassification, we can say that the first verbatim is a long text line and therefore cut short by the limited tokenization, such that valuable information about the QT prolonging ADR is lost. The second and the third verbatims suggest that the pharmacist who coded them had access to additional information other than the verbatim. In general, the model seems to react stronger on words it associates strongly with QT prolonging ADRs. This is noticeable with the last three verbatims in the table. The fourth verbatim describes a QT prolonging ADR not included in the narrow scope QT prolongation/TdP SMQ, although correlated. The fifth and sixth verbatims describe two ADRs in each verbatim, one of which is QT prolonging. Both verbatims are coded to PTs suitable for the second ADR described.

Verbatim	Misclassification type
She experienced the first symptoms ... and gallop rhythm	FN
Electric Storm	FN
grade2	FN
Ventricular tachyarrhythmia	FP
QT prolongation and Nausea	FP
Torsades de pointes/cardiac arrest	FP

Table 13: Examples of misclassified verbatims

To deal with these problems the length of the tokenization could be increased, however, it would also increase the execution time. A way to further improve the classification model could be using BioBERT which is a specialized version of BERT trained to process biomedical language [29]. We suggest trying BioBERT to compare the performance to using general BERT.

7.4 VigiBase ingredient extraction

The extracted active ingredients from VigiBase were gathered from ICSRs submitted on or after the 1st of January 2018. The purpose of this date restriction is to keep MedDRA coding conventions consistent. We also want to avoid active ingredients that have become prohibited in the SDG. Changing the date restriction to allow older ICSRs will give more data but might not improve the quality of the data, To include more ICSRs in the data without changing the date restriction the scope of the TdP/QT prolongation SMQ could be changed. However, the broad scope version of the SMQ contains several PTs not exclusively connected to QT-prolonging ADRs. Therefore the correlation would become less precise.

7.5 SPC validation stage

7.5.1 Set ID extraction

The Set ID proved to be a very useful identifier by uniquely describing a specific SPC. After deciding upon several ways to extract Set IDs, we are content with the choice of using the OpenFDA API. It is less demanding than web scraping and more versatile than the DailyMed RESTful API.

7.5.2 Ingredient as a search subject

When we use the OpenFDA API, we insert the ingredient (or moiety if needed) name extracted from VigiBase as a search subject. Using the exact string might sometimes fail since the ingredient name can be spelled differently in FDA:s database. If the name is spelled differently, we will get no search hits and be unable to validate the ingredient. This was also an issue when comparing our ingredient list with the CredibleMeds list of ingredients with QT-correlation, which had to be manually reviewed.

A possible approach to work around this issue would be using a coded identifier for each ingredient instead of the name string, therefor we discussed using Unique Ingredient Identifier (UNII) codes. A UNII code links to a specific ingredient and is a valid searchable field in the OpenFDA API. However, we ended up using the ingredient name to facilitate different searches and using other sources than FDA. Although for improvement we suggest further investigating the use of UNII codes.

7.5.3 SPC source and search subjects

In this project, SPCs are retrieved from DailyMed. This means that only active ingredients that are approved for commercial use in the United States can be validated against an SPC. The main reason for this limitation is that the SPC Mining pipeline that we have used to process the SPCs requires that the SPCs are in a certain format and written in English. Including additional SPC sources would improve the SPC validation stage but would require extending the SPC Mining algorithm.

In the cases where we could not find any SPCs for the searched active ingredient we instead searched for the active moiety. The drawback of this is that the validation becomes less precise. However, if the moiety can be validated to have a QT prolonging ADR then it suggests that the active ingredient also has one, so we believe that including the moiety as well improves the quality of the final product.

7.5.4 SPC Mining

The SPC Mining pipeline, created by UMC was used to scan and code ADRs in SPCs to PT codes. The pipeline uses a NLP model to code the free-text descriptions in the SPCs. The model is not a perfect classifier and will make mistakes in the coding. The performance is still sufficient and provides good results. The pipeline could be extended to also return if the SPC is a single or multi-ingredient SPC (the information is provided but requires data processing).

We have used the SPC Mining pipeline in its existing format, which is trained on the 5000 most frequently occurring PTs. It would be interesting to re-train it to better adapt it for QT prolonging ADRs especially, and investigate if that would improve the performance.

7.6 VigiBase occurrences

The reason for the modification of the VigiBase occurrences parameter is that an active ingredient in sub-category 1 is validated to cause QT prolonging ADR. We, therefore have reason to believe that sub-category 1 ingredients are the cause of QT prolonging ADRs. By doing this modification the VigiBase occurrence will better highlight the probability of an ingredient causing a QT prolonging ADR.

A possible drawback of the modification is that we might lose information about ingredients that are often coreported with sub-category 1 active ingredients. To mitigate this problem, the original parameter can be used alongside the modified one. Since the modification is depending on the SPC validation, improvement of the validation stage would also improve the modified parameter. Using DailyMed for SPC validation, only medicinal products permitted in the United States can be validated. If SPCs from other countries could be validated as well, the categorization of the active ingredients would be more accurate and in turn also the modified VigiBase occurrence parameter.

7.7 CredibleMeds as a feature

We used CredibleMeds as a validation source since it is the most credible source of information we found listing ingredients with a correlation to QT prolongation and TdP. Although it focuses mainly on TdP, a risk of TdP implies a QT prolonging effect as well. The fact that the ingredient classifiers performance was improved by including CredibleMeds comparison as a feature also implies that it is a valuable validation source.

7.8 Ingredient classification

7.8.1 Training data

Given the limited amount of manually labeled training data and the relatively low IAA value, we can conclude that the quality of the training data is a weak link in the classification. The training data could be improved by increasing the number of ingredients to be manually labeled, and/or using more than two pharmacists as annotators. For evaluation purposes, we wanted the labeling to be performed individually. Although for model optimization, it would be more valuable to assign a group of pharmacists that together agrees upon the labels used for training. It would probably be a more accurate representation of how this kind of ingredient sorting would be performed in reality. These improving suggestions require more manpower, thus it is a trade-off between manual work and model optimality.

Because of the training data limitations, a categorization model using semi-supervised learning could be a useful approach. It combines a smaller amount of labeled data with unlabeled data for training. This could return in a better model performance for the limited amount of labeled data.

7.8.2 Logistic regression model

We chose logistic regression since it is useful for multi-class problems and a method that we were able to implement without using external models/tools except for basic packages. Other methods for example support vector machines, deep learning algorithms, or a semi-supervised approach (as previously mentioned) could be used for this type of classification as well. Since it is used to classify a very limited amount of data, we opted for the more basic choice of logistic regression, which still offers a lot of model flexibility.

There are many ways to extend the logistic regression model. We trained it for three features at maximum, but an extension could be using more and/or different features. To avoid overfitting, lasso regression could be used, a regression model using L1 regularization. By adding a regularization penalty term, it supports feature selection.

Some feature data might be misleading because the data is too limited. For example, if an ingredient is only mentioned in one ICSR that is QT-coded, the feature for QT-related VigiBase occurrences would be 100%. If the number of reports was higher, the same information would tell us a lot more since the variance is decreased. To avoid information with high variance, an extension to discard misleading feature data could be setting thresholds for when to include certain information. This could be for example a threshold for the number of reports or SPCs needed to include the related percentages. However, we chose not to implement these extensions since the information is still of value. Instead of discarding information below the threshold, the information could be penalized.

Thresholds could also be used for the model predictions. Our model predicts all input data based on the highest pseudo probability p_m. A limit between the two highest class logits z_m (or probabilities since they correspond) could be set to discover predictions with an uncertainty considered too high. The result could be ignoring or flagging uncertain predictions, demanding a manual check for those cases.

7.8.3 Performance and result

Looking at the classifier results in Table 12, we were able to improve the performance by including CredibleMeds data. This was expected since it is a clear indication of known or suspected correlation. The final cross-validation performance measures for the three-feature model, $\alpha_{cv} = 0.6818$ and $\varepsilon_{cv} = 0.4660$, indicates that the data is not easily separated for the chosen features. This is also observed when plotting the training data for these features in Figure 16. Even though there is a pattern in where the different class data points are more likely to be, the data is very mixed and hard to separate. This is especially true for class 2 (weak connection), in which scattered training data results in a very small prediction zone seen in Figure 15. We believe that the number of unlabeled ingredients assigned to class 2 should be higher to better simulate the manual labeling. With the current training data, the multinomial logistic regression model behaves almost like a binary classifier (predicting only two ingredients to class 2). Because of this result, one could interpret the problem as a binary classification problem instead, giving the manual annotators only two options (QT-prolonging or non-QT-prolonging).

In the figure we can also see that Validated single-active ingredient SPC percentage is the dominating feature when comparing it to QT-coded reports in VigiBase. looking at the outlier with 100% QT-coded reports in VigiBase, it was still predicted as class 3 because it was not validated for any single-ingredient SPC. This shows the importance of the SPC validation stage, why we suggest focusing on that stage if further optimizing the process.

7.9 Final product

When analyzing the final product, it is clear that we have included more ingredients than we believe have an actual connection to QT prolongation. This is due to our strategy of rather including more ingredients than excluding ingredients that might belong in the SDG. The product is therefore adaptable to set new thresholds and/or manually exclude ingredients to narrow down the list of ingredients. Looking at the bottom of the list, we see ingredients assigned to class 3 by the classifier, that we have not been able to validate for SPCs or CredibleMeds comparison. The only connection for these ingredients is a low percentage of QT occurrences in VigiBase after modification. As an example, we find several vaccines here that are usually taken as a set, e.g. diphtheria, tetanus, acellular pertussis, and polio vaccine. These have 1913 reports in VigiBase, from which only 2 have been coded as QT prolonging. A fraction that small is probably due to other reasons than an actual correlation, such as other QT-prolonging drugs being taken by the user or a false interpretation of the reaction. As a first step to narrow down the list further, we suggest setting a threshold for the modified QT VigiBase occurrences acting on the ingredients classified as class 3 (no indication of connection).

7.10 Adapting the process for other ADRs

An interesting aspect of the SDG basis creation is the possibility to adapt the process for other ADRs than QT prolongation. We have tried to keep the pipeline as general as possible, to ease a transition to different ADRs. The VigiBase occurrences parameter would be calculated the same if another MedDRA SMQ were used for the grouping, but could also be adapted for a customized set of PTs. For the SPC validation stage, the only adaptation would be adjusting the list of PTs that we compare with the coded SPCs. The only parameter that we would not be able to use is the CredibleMeds validation. Instead, there might be other validation possibilities for the new ADR.

A challenge for certain ADRs, which we did not experience focusing on QT prolongation, is that some drug reactions can be both wanted and unwanted. Let us take hypotension (low blood pressure) as an example ADR. For a patient suffering from hypertension (high blood pressure), lowering the blood pressure is probably a wanted drug reaction. For other patients, it is an unwanted, and sometimes dangerous, ADR. This difference would be of importance when creating a hypotension SDG.

Another difficulty with groupings for different ADRs is the ability to measure the effect. For QT prolongation, it is directly observable using ECG. Other reactions might be harder to measure and more defendant on the patient's expressed experience, thus harder to quantify.

8 Conclusion

To conclude the project as a whole, we were able to create a basis for a QT prolongation SDG, sorted in order of suspected correlation. We were also able to predict if a free-text verbatim from an ICSR describes a QT prolonging ADR, with satisfactory precision. The final product consists of information useful for a pharmacist to decide if an ingredient should be included in the SDG.

We worked with the free-text processing and the creation of SDG basis separately, but they could be used together in a shared pipeline, including reports that are not coded to MedDRA terms by instead making predictions based on the free-text verbatims. To include the reports that our free-text classifier predicted as describing a QT prolonging ADR in the creation of SDG basis process, there is a need to code the medicinal products mentioned in the report to WHODrug drug codes. There is a UMC automated coding service that could be used for this purpose, creating a finalized pipeline.

Regarding the free-text processing, we are content with the results and would suggest using the implemented language sorting and the random sampling approach. One easily implemented approach to improve the performance would be to increase the number of tokens, avoiding important information being cut off.

Due to the limited and scattered training data and the low IAA score, the ingredient classification has an unsatisfactory performance. The classification works well for a first sorting, but we would not count it as exact or reliable. Since the prediction zone for ”Class 2: Weak connection” is so limited, the classifier almost acts as binary even though it is trained for three classes.

We consider the methods and systems used to be well-performing, and the final result to be a good basis for future work or to directly use as a decision basis. Although as presented in the Discussion section, there are multiple suggested approaches to further optimize and extend the process.

9 References

References

[1] UMC — Who we are. Available at: https://www.who-umc.org/about-us/who-we-are/ (Accessed: 29 January 2021).

[2] UMC — WHODrug Standardised Drug Groupings (SDGs) (September 2020). Available at: https://www.who-umc.org/whodrug/whodrug-portfolio/whodrug-standardised-drug-groupings-sdgs/ (Accessed: 20 January 2021).

[3] UMC — Signal detection (September 2020). Available at: https://www.who-umc.org/research-scientific-development/signal-detection/ (Accessed: 20 January 2021).

[4] EMA — Individual case safety report. Available at: https://www.ema.europa.eu/en/glossary/individual-case-safety-report (Accessed: 20 January 2021).

[5] Derick G, M. et al. (2011), 'Medication-Induced QT-Interval Prolongation and Torsades de Pointes', U.S. Pharmacist, 32(2), HS-2-HS-8.

[6] ICH Official web site — MedDRA. Available at: https://www.ich.org/page/meddra (Accessed: 29 January 2021).

[7] ICH — Introductory Guide MedDRA Version 23.1 (September 2020). Available at: https://admin.new.meddra.org/sites/default/files/guidance/file/intguide_%2023_1_English.pdf (Accessed: 1 February 2021).

[8] Sharma, R et. al. (2013) Everything You Need To Know About Standardised MedDRA Queries. PharmaSUG 2013, Chicago, USA, May 12-15 2013. (Accessed: 5 February 2021).

[9] UMC — WHODrug Global (November 2020). Available at: https://www.who-umc.org/whodrug/whodrug-portfolio/whodrug-global/ (Accessed: 9 February 2021).

[10] DAUE, R. (2017) Pharmacovigilance, Public Health - European Commission. Available at: https://ec.europa.eu/health/human-use/pharmacovigilance_en (Accessed: 20 January 2021).

[11] UMC — Global Pharmacovigilance. Available at: https://www.who-umc.org/global-pharmacovigilance/global-pharmacovigilance/ (Accessed: 20 January 2021).

[12] UMC — Guideline for using VigiBase data in studies. (Mars 2018). Available at: https://www.who-umc.org/media/164772/guidelineusingvigibaseinstudies.pdf (Accessed: 26 January 2021).

[13] UMC — WHO programme members (November 2020). Available at: https://www.who-umc.org/global-pharmacovigilance/who-programme-for-international-drug-monitoring/who-programme-members/?id=

100653&mn1=7347&mn2=7252&mn3=7322&mn4=7442 (Accessed: 26 January 2021).

[14] ICH — MedDRA Distribution File Format Document Version 23.1 (September 2020), Available at: https://admin.new.meddra.org/sites/default/files/guidance/file/dist_file_format_23_1_English_0.pdf (Accessed: 1 February 2021).

[15] UMC — WHODrug Insight (September 2020), Available at: https://www.who-umc.org/whodrug/access-tools/whodrug-insight/ (Accessed: 23 February 2021).

[16] openFDA — Drugs@FDA, Available at: https://open.fda.gov/data/drugsfda/ (Accessed: 23 February 2021).

[17] Bista, S. (2020) 'Extracting Adverse Drug Reactions from Product Labels using Deep Learning and Natural Language Processing', Available at: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-277815 (Accessed: 24 February 2021).

[18] Wallner, V. (2020) 'Mapping medical expressions to MedDRA using Natural Language Processing', Available at: http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-426916 (Accessed: 19 January 2021).

[19] Yang Liu, Meng Zhang, 'Neural Network Methods for Natural Language Processing', Computational Linguistics 2018; 44(1), pp 193-195

[20] Crediblemeds — About Crediblemeds. Available at:
https://www.crediblemeds.org/everyone/about-crediblemeds (Accessed: 7
April 2021).

[21] Boulicaut, J. F. et al. (2004) 'Applying Support Vector Machines to Imbalanced Datasets'. European Conference on Machine Learning (ECML), Pisa, Italy, September 20-24, 2004. Available at: https://link.springer.com/chapter/10.1007/978-3-540-30115-8_7 (Accessed: 21 May 2021)

[22] Tiensuu, J. et al. (2019) 'Detecting exoplanets with machine learning: A comparative study between convolutional neural networks and support vector machines', Available at: http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-385690 (Accessed: 21 May 2021).

[23] Lindholm, A. et al. (2021). 'Machine Learning - A First Course for Engineers and Scientists', Available at: http://smlbook.org/ (Accessed: 21 May 2021).

[24] L.R. Medsker et al. (200) 'Recurrent Neural Networks - Design and Applications', Boca Raton, United States, CRC Press.

[25] Vaswani, A. et al. (2017) 'Attention Is All You Need'. Conference on Neural Information Processing Systems (NIPS), Long Beach, USA, December 4-9, 2017. (Accessed: 22 May 2021)

[26] Devlin, J. et al. (2019) 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', arXiv:1810.04805. Available at:

http://arxiv.org/abs/1810.04805 (Accessed: 22 May 2021).

[27] Kingma, D. P. et al. (2017) ‘Adam: A Method for Stochastic Optimization’. International Conference on Learning Representations (ICLR), San Diego, USA, May 7-9, 2015, Available at: http://arxiv.org/abs/1412.6980 (Accessed: 22 May 2021)

[28] Artstein, R. et al. (2008) ‘Inter-Coder Agreement for Computational Linguistics’, Computational Linguistics, 34(4), pp. 555–596.

[29] Lee, J. et al. (2020) ‘BioBERT: a pre-trained biomedical language representation model for biomedical text mining’, Bioinformatics, 36(4), pp. 1234-1240

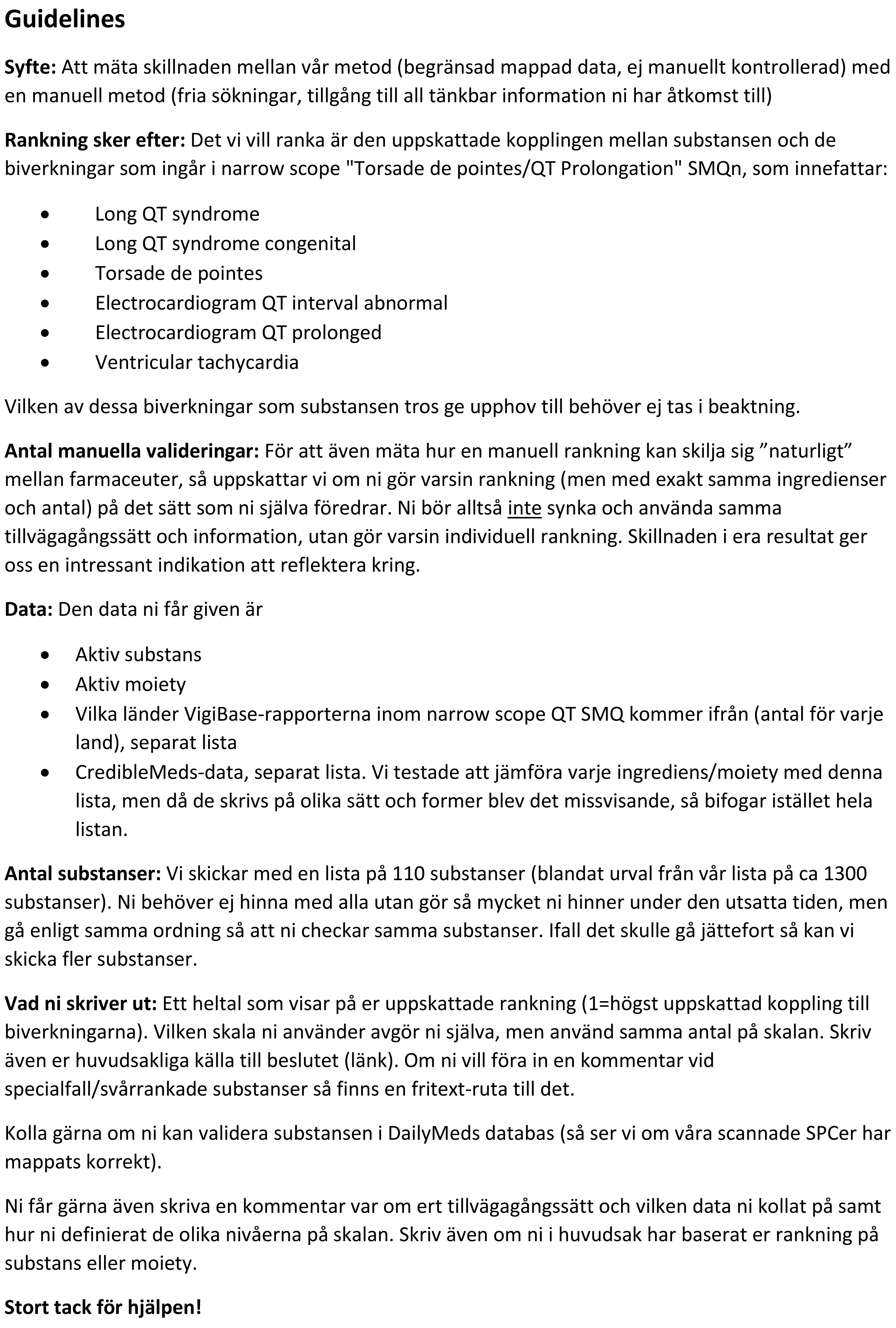
A Division of work

We have collaborated on the project as a whole, but divided the responsibilities for the different sub-modules for an efficient workflow. We have used an agile approach working in sprints of two weeks, which helped us to continuously stay updated on each other's areas of responsibility while preparing sprint reviews.

Jacob was responsible for the extraction of relevant data in VigiBase, regarding ingredients as well as free-text verbatims. Elsa focused on the processing of the ingredient data by extracting Set ID:s, validate for SPCs, review and modify CredibleMeds' QT drug list and compare included ingredients to the VigiBase extracted ingredients. She also analyzed the different validation results.

For the two classification models, Elsa managed the ingredient classifier using multinomial logistic regression and Jacob the free-text classification using BERT, including tokenization, language sorting and sampling approaches. We shared the responsibility for the SPC Mining process, the manual labeling process and analysis, the modification of the VigiBase occurrences parameter and the final content and presentation of the SDG basis.

B Guidelines provided for manual labeling



Guidelines

Syfte: Att mäta skillnaden mellan vår metod (begränsad mappad data, ej manuellt kontrollerad) med en manuell metod (fria sökningar, tillgång till all tänkbar information ni har åtkomst till)

Rankning sker efter: Det vi vill ranka är den uppskattade kopplingen mellan substansen och de biverkningar som ingår i narrow scope "Torsade de pointes/QT Prolongation" SMQn, som innefattar:

- Long QT syndrome
- Long QT syndrome congenital
- Torsade de pointes
- Electrocardiogram QT interval abnormal
- Electrocardiogram QT prolonged
- Ventricular tachycardia

Vilken av dessa biverkningar som substansen tros ge upphov till behöver ej tas i beaktning.

Antal manuella valideringar: För att även mäta hur en manuell rankning kan skilja sig ”naturligt” mellan farmaceuter, så uppskattar vi om ni gör varsin rankning (men med exakt samma ingredienser och antal) på det sätt som ni själva föredrar. Ni bör alltså inte synka och använda samma tillvägagångssätt och information, utan gör varsin individuell rankning. Skillnaden i era resultat ger oss en intressant indikation att reflektera kring.

Data: Den data ni får given är

- Aktiv substans
- Aktiv moiety
- Vilka länder VigiBase-rapporterna inom narrow scope QT SMQ kommer ifrån (antal för varje land), separat lista
- CredibleMeds-data, separat lista. Vi testade att jämföra varje ingrediens/moiety med denna lista, men då de skrivs på olika sätt och former blev det missvisande, så bifogar istället hela listan.

Antal substanser: Vi skickar med en lista på 110 substanser (blandat urval från vår lista på ca 1300 substanser). Ni behöver ej hinna med alla utan gör så mycket ni hinner under den utsatta tiden, men gå enligt samma ordning så att ni checkar samma substanser. Ifall det skulle gå jättefort så kan vi skicka fler substanser.

Vad ni skriver ut: Ett heltal som visar på er uppskattade rankning (1=högst uppskattad koppling till biverkningarna). Vilken skala ni använder avgör ni själva, men använd samma antal på skalan. Skriv även er huvudsakliga källa till beslutet (länk). Om ni vill föra in en kommentar vid specialfall/svårrankade substanser så finns en fritext-ruta till det.

Kolla gärna om ni kan validera substansen i DailyMeds databas (så ser vi om våra scannade SPCer har mappats korrekt).

Ni får gärna även skriva en kommentar var om ert tillvägagångssätt och vilken data ni kollat på samt hur ni definierat de olika nivåerna på skalan. Skriv även om ni i huvudsak har baserat er rankning på substans eller moiety.

Stort tack för hjälpen!