CHALMERS
UNIVERSITY OF TECHNOLOGY

Extraction of Severity Information from Clinical Narratives Using Statistical Natural Language Processing

Master's thesis in Complex Adaptive Systems

REBECKA JACOBSSON

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Master's thesis 2016:EX037

Extraction of Severity Information from Clinical Narratives Using Statistical Natural Language Processing

REBECKA JACOBSSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Signals and Systems
Division of Signal Processing and Biomedical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2016

Extraction of Severity Information from Clinical Narratives Using Statistical Natural Language Processing
Rebecka Jacobsson

Supervisor: Johan Ellenius, Uppsala Monitoring Centre
Examiner: Lennart Svensson, Department of Signals and Systems

Master's Thesis 2016:EX037
Department of Signals and Systems
Division of Signal Processing and Biomedical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Word cloud of severity expressions found in clinical narratives from VigiBase.

Typeset in LaTeX
Gothenburg, Sweden 2016

Extraction of Severity Information from Clinical Narratives Using Statistical Natural Language Processing
Rebecka Jacobsson
Department of Signals and Systems
Chalmers University of Technology

Abstract

Natural language processing methods adapted to the medical domain are quickly becoming important tools in the analysis of large amounts of clinical narratives. One important application is in the field of drug safety, where the investigation of suspected adverse drug reactions is based on millions of individual case safety reports from such incidents, and where much of the relevant information is only available in free-text form. In this thesis, a method for automatically extracting information about the severity of adverse reactions from such reports is developed and tested. A set of 1579 reports, pre-annotated for reactions, are manually annotated for severity information and two support vector machine (SVM) classifiers are then trained and tested on this data. The first SVM performs binary classification of tokens to identify severity descriptors using a large number of token and contextual features, while the second classifies the grade of the severity as mild, moderate or severe and uses as input a bag-of-words of severity descriptors associated with a particular reaction. Running the two SVMs in a pipeline gives an overall precision of 0.837 and a recall of 0.887, resulting in an F1-score of 0.861. This performance is considered good enough for the extracted severity information to be potentially useful in the discovery of new adverse drug reactions.

Keywords: natural language processing, information extraction, relation extraction, pharmacovigilance, machine learning

Acknowledgements

First of all, I would like to thank everyone at Uppsala Monitoring Centre who helped me in my work with this thesis, especially Lovisa Sandberg who spent endless hours helping me deal with tricky annotations and debating the annotation guidelines, Tomas Bergvall who did all the data extraction for me and provided valuable input on many decisions along the way, Sara Vidlin for letting me reuse a lot of her code and explaining to me how it all worked, Kristina Star for patiently taking time to help me grasp the relevance and background of this project in its early stages, Rebecca Chandler and Birgitta Grundmark for providing medical expertise when Lovisa and I got stuck and Henric Taavola for his company as my office mate and for reminding me to take coffee breaks. My supervisor Johan Ellenius always had an open door and willingly discussed with me the problems I ran into. I am especially grateful for all his help in maintaining the scientific stringency of my work.

I would also like to thank Lennart Svensson, who was my examiner at Chalmers University and provided exactly the amount of support that I needed to stay on track. Thank you for pushing me to put down my project in words early on and for answering all my odd questions along the way.

Finally, I would like to extend my deepest gratitude to to my family and friends who have helped me make it through these five years of studies. I particularly want to thank my friends at Chalmers who joined me for the emotional roller coaster of anxiety, excitement, stress and euphoria that is engineering physics. Thank you also to my parents who inspired me to study engineering physics in the first place, to my brother David and to my boyfriend Ludvig for bearing with me throughout all my ups and downs and always providing love and support.

Rebecka Jacobsson, Gothenburg, September 2016

Contents

List of Figures

List of Tables

1

Introduction

Adverse drug reactions (ADRs) are considered a major cause of morbidity and mortality and are often associated with high costs to society [1, 2]. It has been estimated that ADRs cause about 3.6 % of all hospitalizations in Europe, and that 10 % of all patients admitted to European hospitals experience an ADR during their stay [3]. The annual number of deaths in Europe caused by adverse drug reactions has been estimated to lie in the range $42000 - 419000$ per year [3], and the total societal economic burden of ADRs in the European Union has been estimated at €79 billion per year by the European Commission [4]. While pre-market randomized controlled trials do work well for evaluating the efficacy of new drugs, their usefulness in detecting ADRs is limited. This is mainly due to limitations in the size and heterogeneity of the studied populations as well as the relatively short durations of the studies [5]. Detection of adverse drug reactions therefore depends strongly on spontaneous reporting of adverse events discovered after drugs have reached the market.

The arguably biggest ever disaster in the field of drug safety happened in 1961 when it was discovered that the drug Thalidomide, which had been used to treat nausea and morning sickness in pregnant women, had caused at least 10 000 children to be born with missing or underdeveloped limbs. It had taken several years to discover the causal relationship between the drug and the birth defects and physicists and politicians around the world called for improved methodology and increased cooperation between countries to tackle drug safety issues. The field of pharmacovigilance was born as countries struggled to prevent similar disasters from happening again.

Today, most countries in the world have authorities that are responsible for the regulation and surveillance of drugs and medical products before and after they are released on the market. In Sweden, this is the Medical Products Agency (MPA) or Läkemedelsverket in Swedish, while in the United States the responsible agency is the Food and Drug Administration (FDA). These authorities typically collect reports of suspected ADRs, so called Individual Case Safety Reports (ICSRs), from doctors, pharmacists and patients and use these along with other sources of information, such as data from clinical trials and sources in literature, to monitor and act on possible ADRs. The European Union also has the European Medicines Agency, which is currently located in London but is likely to relocate in the aftermath of Brexit. The EMA collects ICSRs from all its member states and leverages this larger data set in the detection of ADRs. But while the EMA pharmacovigilance program thus covers about 740 million people, this is still only about a tenth of the global population. As was concluded after the Thalidomide disaster, sharing information on drug safety

among the countries of the world is necessary in order to quickly and accurately detect new ADRs when they occur.

Pharmacovigilance is performed on a global level by the Uppsala Monitoring Centre (UMC), which is the World Health Organization (WHO) Collaborating Centre for International Drug Monitoring. Since its foundation in 1978, the UMC has received over 13 million ICRSs from more than 130 different countries, all participants of the WHO Pharmacovigilance Program. By joining the program and submitting their own reports, these countries also gain access to VigiBase® , which is the database developed and maintained by the UMC that contains all submitted ICSRs. Among the biggest contributors are the United States along with several European countries, but today many reports are also submitted by developing countries. Efficiently extracting information from this vast set of data is a challenge in itself.

ICSRs typically include both structured and unstructured data. The structured data includes for example standardized codes for all observed symptoms and a list of all the patient's medications and their dosages. This data is used for signal detection, where in a first-pass automated screening reports are selected for further analysis. The unstructured data consists of free-text such as clinical narratives and the reporter's comments, and is today only used in the second step of the signal detection process where human experts analyze the reports in detail and judge their relevance. A long-term goal for UMC is to develop methods for automatically extracting features from this unstructured data that can be used in the first-pass screening. That is also the focus of this thesis, which will focus on developing machine-learning based methods for extracting information about the severity of adverse events in the clinical narratives of ICSRs from VigiBase .

1.1 Background

One of the currently most useful features of the structured data in ICSRs is the reported *seriousness* of the case as a whole. This is essentially a boolean flag that is associated with a report and which can be used for filtering. A reported case is considered serious if it

- results in death
- is life-threatening
- requires inpatient hospitalization or prolongation of existing hospitalization
- results in persistent or significant disability/incapacity (as per reporter's opinion)
- is a congenital anomaly/birth defect

or if it is considered an "other medically important condition", which can for example be convulsions that do not result in hospitalisation or development of drug dependency or drug abuse [6, 7]. The seriousness of a report is often used as a selection criteria in the first-pass screening of a specific drug-ADR combination in

order to aid prioritization. A commonly used approach at UMC is to require that 75 % of the reports in a case series are reported as serious, or else that case series is dismissed as uninteresting.

A related concept that is not captured in structured form is the *severity* of the reported adverse events. Severity and seriousness are not synonyms – while the seriousness of an event describes its outcome, the severity describes its intensity. The severity of an event can for example be described as mild, moderate or severe and this does not necessarily reflect the seriousness of the event, such as in the case of a severe headache which may often be a non-serious event [6]. Severity information may however be a useful feature in the first-pass screening step of signal detection. Today, non-serious and common adverse drug reactions such as headaches and rashes are seldom investigated due to the vast number of reports referring to such reactions – for example, there are currently over 450 000 reports in VigiBase listing headaches as suspected ADRs. This makes it impossible to analyze them all in detail, unless some filtering criteria can be applied.

Developing methods for extracting the severity of adverse drug events from reports in VigiBase would enable analysis of these non-serious but severe adverse drug reactions. This might be particularly useful for reports submitted by patients, since several studies have shown that patients tend to report the severity of ADRs to a greater extent than health care professionals [8, 9, 10]. Another advantage is that severity could then be extracted for specific reactions while the seriousness is only reported for each case as a whole, and most cases report multiple reactions. For example, a report might describe a patient initially experiencing a strong headache and then a long chain of other medical events leading to the patient's death. The case would then be classified as serious because of its fatal outcome, but this would not in any way be related to the severity of the headache. Screening for suspected drug-ADR combinations using severity as a feature for selection might potentially lead to the discovery of previously unknown adverse drug reactions or allow for further characterization of already known adverse reactions. Enabling severity to be used for such screening would require several steps of natural language processing and information extraction. First, medical events and severity modifiers need to be discovered, then the medical events have to be classified as ADRs or non-ADRs and the relations between the severity modifiers and ADRs have to be established. Finally, the grade of severity needs to be mapped to some useful scale that can be used in screening.

1.2 Objective

The objective of this thesis is to enable the use of severity information in the first-pass screening step of the signal detection process at Uppsala Monitoring Centre. This will be done by developing methods to automatically detect words and expressions that describe the severity of ADRs from the free-text narratives of individual case reports. Then methods will be developed for classifying the severity according to a scale of mild/moderate/severe.

1.2.1 Limitations

Discovery of medical events in clinical text is an important task in clinical NLP. Several methods already exist for doing so [11] and therefore, this thesis will not attempt to develop those methods further. Instead, automatic event detection will be performed before any training data is annotated. Furthermore, no advanced methods will be used for determining whether a detected medical event is an ADR nor not. This is in general a quite complex task since a mentioned medical event can also be part of the patient's medical history or even the cause for the starting treatment that later led to an ADR. Since ADR detection is not the focus of this project, a simple method based on looking for matches between reported MedDRA codes and the corresponding codes mapped to events found by the event-detection algorithm will be used. This is explained further in Section 3.3. During the annotation process, severity will then only be marked for events that could be discovered automatically and that were determined to be ADRs. The success rates of the methods developed will thus not depend on the performance of the event detection or ADR identification algorithms.

Furthermore, no lab or test results will be included in the analysis. It would certainly be possible to develop methods for extracting test results and classifying the severity of different diseases associated with them based on the result, but that task is quite different from the general severity extraction problem and will therefore not be included in this thesis. This means that descriptions of severity which require some degree of interpretation or medical reasoning from the reader, such as "fever above 40 degrees", will not be annotated as severity. This is of course a limitation to the usefulness of the developed method, but is considered a necessary limitation for this project. More details on these considerations can be found in Section 3.4 and 4.1.

Finally, only narratives written entirely in English will be considered for this study. The methods developed will not be language-specific but they will be based on an annotated data set which is purely in English. Future work might extend this project by annotating data sets in other languages and applying the algorithms developed here, but this is outside the scope of the current project.

1.2.2 Research Questions

The following research questions will be addressed in the order they are listed below. The main focus of the project will be on answering the first research question.

1. Can natural language processing methods be used to automatically identify indicators of the severity of adverse drug reactions (ADRs) from free text fields in individual case safety reports (ICSRs) from VigiBase ?
2. To what extent can identified indicators of severity be used to automatically classify or grade the severity of ADRs?

1.3 Related Work

The past decade has seen rapid development of natural language processing methods specific to the clinical domain. Much of the work has focused on information extraction from electronic health records and biomedical literature for question answering [12], decision support [13], summarization [14], aiding research [15] and many other tasks [16]. Clinical NLP has also been used in pharmacovigilance [17, 18, 19], but most research has focused on detecting adverse drug reactions from electronic health records and not much work has been focused on spontaneous reports. In fact, there are only two known attempts to apply NLP to the narratives of spontaneous reports. The first is a study performed by Ellenius et al at Uppsala Monitoring Centre which explored named entity recognition of drugs in VigiBase reports while the other is a recent study by Ramesh et al [20], which aimed at named entity recognition (NER) of ADR related information from reports in the American Food and Drug Administration's Adverse Event Reporting System (FAERS). Neither of these studies did however attempt to extract any severity information.

Severity discovery has on the other hand recently been explored in the context of clinical notes. Dligach et al [21] recently developed methods for modifier detection and relation extraction for the UMLS defined DegreeOf relation. Both of these tasks were accomplished by training a support vector machine (SVM) classifier on two annotated corpora that consisted of radiology notes, breast cancer oncology and pathology notes in one case, and intensive care unit notes and discharge summaries in the other. The features used for training of the relation extractor included a large number of lexical, syntactic and semantic features as well as tree kernel features. The features used for modifier detection were not reported, but the publicly released source code suggests that the tokens themselves, their POS-tags and the 2 previous outcomes were used as features. Results are not reported for the modifier detection, but the relation extraction method achieved F1 scores of $0.905 - 0.929$ for the DegreeOf relation.

Severity detection was also part of the "Disease/Disorder Template Filling" task of the ShARe/CLEF eHealth Evaluation Lab 2014. Participants Johri et al [22] based their work on the open source modifier and relation extractors developed by Dligach et al but trained a conditional random fields (CRF) model for annotating severity modifiers. Their model achieved a strict F1 score of 0.828 on the CLEF eHealth corpora, which consisted of discharge summaries, radiology reports, echocardiography reports and electrocardiogram reports. Other participating teams attempted rule-based [23, 24] and deep parsing [25] approaches, but with less successful results.

1.4 Thesis Outline

The next chapter of this thesis will begin with an overview of some of the tasks that clinical natural language processing aims to solve. An introduction to the machine

learning methods used in this thesis will also be given. In the following chapter, the methods used for severity detection and classification are explained in detail. The general frameworks used are also introduced, as well as the methods used for data pre-processing and creation of the human annotated gold standard data set. Next follows a review and analysis of the results from all sub tasks, and in the final section conclusions are drawn from all experiments.

2

Theory

This chapter outlines the theory that the method described in Chapter 3 is based upon. A brief introduction to natural language processing in general and information extraction from clinical text in particular is given first, followed by an description some important machine learning methods used in natural language processing. Finally, some different methods for evaluating the performance of these methods are defined and compared.

2.1 Natural Language Processing

Natural language processing is the field concerned with understanding and generating human language such as written or spoken English, Portuguese or Chinese by using computer languages such as C++ and Java. One can think of it as a collection of tasks aiming to bridge that gap between computer and human communication. A large and important subfield within natural language processing, which will be the focus of this thesis, is information extraction which as its name implies is concerned specifically with extracting and interpreting information that is originally expressed in natural language. Information extraction tasks include both basic tasks such as sentence splitting and tokenization, and more complex problems such as part-of-speech tagging, named entity recognition and relation extraction. These different tasks will be described in the next few sections and then be followed by a brief description of two important frameworks commonly used in natural language processing: UIMA, which is a general purpose architecture for managing unstructured information such as text, and cTAKES, which is a UIMA based system specifically designed for information extraction from clinical text.

2.1.1 Sentence Splitting, Tokenization and Part-of-Speech Tagging

Sentence splitting and tokenization are two of the most fundamental tasks in natural language processing. This is because many other problems rely on accurate detection of sentences and tokens. A token is typically a word, but punctuation symbols such as periods, commas and dashes are also tokens as well as newline characters. Sentences are more difficult to define, but they are typically delimited

by punctuation symbols and the difficulty in sentence splitting is determine when a punctuation token does in fact indicate the end of a sentence and when it does not. Both tokenization and sentence splitting can be performed using rule-based methods but today, machine learning methods are more common and there is a plethora of trained models available for all types of domains.

Part-of-speech (POS) tagging is the task of determining the part-of-speech for each token in a text. While POS-tagging may be considered an important application in itself, it is more frequently used as a building block for solving other tasks such as named entity recognition.

2.1.2 Named Entity Recognition

Named entity recognition (NER) is concerned with detecting named entities such as persons, locations, companies or dates from free-text and then classifying them into into their appropriate categories. Some named entities that are specific to the clinical domain are drugs, medical events and treatments. For example, the following sentence contains one medical event and one treatment: ”The myocardial infarction was treated with defibrillation”. Here, the medical event is ”myocardial infarction” and the treatment is ”defibrillation”. These named entities can be identified either using rule-based methods, which can for example make use of dictionaries of medical events (see Section 3.3) or they can be trained on text that has been annotated with the correct answers. Popular methods for NER include support vector machines, conditional random fields and deep neural networks.

Named entity recognition can be modelled in various ways. One method is simply to classify all tokens in a text as belonging either to one of the available named entities or as not being a named entity at all. This means that the named entity recognition problem is approached as a multiclass classification problem, where the number of classes is the number of named entity types plus one for tokens that are not named entities. Using this approach, the tokens in the example above would all be classified independently and the correct output would be

- myocardial → medical event
- infarction → medical event
- defibrillation → treatment

but we would often prefer to know that the tokens ”myocardial” and ”infarction” so actually describe the same medical event and thus be just one named entity. This is typically achieved by expanding the number of classes, and can be done in several ways. The simplest is so called BIO-encoding, where BIO stands for beginning, inside, outside. Each possible named entity type then gets two corresponding classes which each describe the relative position of the token within that named entity. Continuing the example above, we would thus get B-medical event which is assigned to the first token of each medical event, I-medical event which would be assigned to any other other token of a medical event, B-treatment which would be assigned to the first token of a medical event etc. and O which would be assigned to all

tokens not belonging to one of the named entity types. Our example from above thus becomes

- myocardial → B-medical event
- infarction → I-medical event
- defibrillation → treatment

and it is then a simple task to merge subsequent tokens with matching tags, such as ”myocardial” and ”infarcation” into full named entities.

2.1.3 Relation Extraction

A task that is closely related to named entity recognition is relation extraction, which is concerned with mapping relations between named entities. This task is typically modelled as a binary classification problem, where candidate pairs of tokens are classified as either being related or not. How these candidate pairs are defined depends on the task. If for example the goal is to extract relations of the type ”employee of”, it would be appropriate to define the candidate pairs as all possible pairings of a person and a company. This means that from the sentence ”Marie works for Apple and Julia is a Google employee”, we would get four candidate person-company pairs:

- Marie – Apple
- Julia – Google
- Marie – Google
- Julia – Apple

where only the first two are true relations. Dligach et al [21] used a similar model for identifying severity relations in clinical narratives by using all possible pairings of severity entities and medical events as candidate pairs. A prerequisite is then obviously that severity entities must first have been accurately identified. The problem approached in this thesis is thus slightly different and can be seen as a combined NER and relation extraction problem, since the goal is both to detect severity tokens and to relate them to their appropriate reation(s) before finally classifying them.

2.1.4 UIMA and cTAKES

UIMA stands for Unstructured Information Management Architecture and is a system developed by IBM for use in Watson, a question-answering computer developed to compete with humans in the game Jeopardy. The UIMA system is today maintained and developed as open-source by the Apache Software Foundation. It provides a component-based structure for the analysis of unstructured information such as free-text or images. Components can be developed independently and then built into pipelines, which allows for efficient reuse of code.

cTAKES is a system built on top of the UIMA framework, which supplies many useful components specifically developed for information extraction from clinical text [26]. Some of the components included are sentence splitting, tokenization, part-of-speech tagging and dictionary-based detection of medical events (see Section 3.3). While the last is rule-based, the three first use machine learning models that have been trained on clinical data from a large number of sources. This means that they can be expected to work better on clinical text than equivalent models that have been trained on other types of text.

2.2 Machine Learning for Natural Language Processing

Most modern natural language processing is based on machine learning methods which rely on training examples to learn how to predict the output of new examples. The following sections will describe the support vector machines classifier, which is one such algorithm, as well as some practical details of using machine learning in natural language processing.

2.2.1 Support Vector Machines

Support vector machines (SVMs) are supervised machine learning algorithms that can be used for regression or classification. They are particularly efficient on high-dimensional problems and therefore popular in natural language processing and information extraction applications, where the feature space is usually very large. The most basic SVM classifier is the linear hard-margin SVM. It is used to solve classification tasks such as the one shown in figure 2.1a, where the goal is to separate the two classes using a linear boundary. As this figure shows, there are many possible boundaries that achieve perfect separation between the classes and we must choose one. The idea in SVM classification is to identify the plane that achieves the largest separation between the two classes. We call this separation the margin and figure 2.1b shows the boundary that maximizes the margin for the case in figure 2.1a. Mathematically, we can define any linear boundary as a set of points $\mathbf{x}$ that satisfy $\mathbf{w}^T\mathbf{x} + b = 0$ where $\mathbf{w}$ is a vector that is normal to the boundary and b is a bias term. If we denote our points under classification as $\mathbf{x_i}$ with $i = 1, 2, 3, ..., n$ and their classes by $y_i = +1$ for one class and $y_i = -1$ for the other, the maximum margin boundary can be found by solving the optimization problem

$$\min \frac{1}{2}||\mathbf{w}||^2 \quad \text{under the constraint} \quad y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 \tag{2.1}$$

where the minimization part is to maximize the margin and the constraint guarantees that all points are correctly classified by the boundary.

If we instead have a set of points such that the classes are not linearly separable, as in figure 2.1c, there will be no solution to (2.1). We can then decide to accept that some

points will be misclassified, but associate this with a cost that is proportional to the distance between each misclassified point and the boundary. We thus introduce so called slack variables ξ_i which describe this distance for misclassified points $\mathbf{x_i}$ and which are zero for correctly classified points. We then find a separation boundary which satisfies

$$\min \frac{1}{2}||\mathbf{w}||^2 + C\sum \xi_i \quad \text{under the constraints} \quad \begin{cases} y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \tag{2.2}$$

where C is a cost parameter that determines the balance between maximizing the margin and placing as many points as possible on the correct side of the boundary. This is called a soft-margin SVM. Another possibility is to use a nonlinear SVM. This means that we implicitly transform our data points $\mathbf{x_i}$ into some other space where they are linearly separable. In practice, this is achieved by defining a kernel function $k(\mathbf{x_n}, \mathbf{x_m})$ and solving for the Lagrangian dual of (2.2) with that kernel. We will not describe the details here, but only note that the optimization problem becomes computationally heavier than when using a linear kernel.

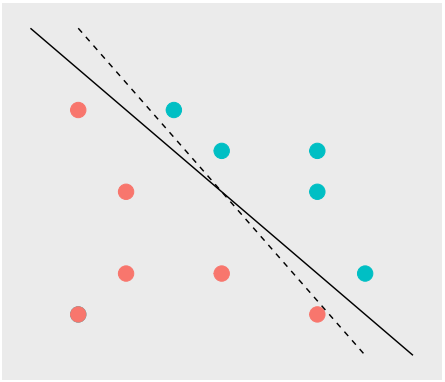
There are many ways of of solving the optimization problem in (2.2). For large, sparse data sets a popular implementation which achieves remarkably fast performance is the LIBLINEAR algorithm developed by Lin et al [27]. It uses a version of coordinate descent developed by Hsieh et al [28] which involves splitting the problem into sub-problems and randomly permuting these in order to achieve fast convergence. Since the algorithm is then stochastic, its output can differ slightly between runs.

In order to use an SVM for multiclass classification, the task is typically transformed into multiple binary classification tasks where the goal of each individual classifier is to discriminate between one class and the rest. This is called *one-vs.-rest* classification and is implemented in LIBLINEAR.

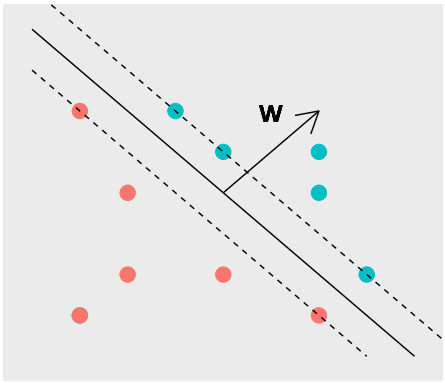
2.2.2 Imbalanced Data

A common problem when training machine learning classifiers for natural language processing tasks is that the classes are often severely imbalanced. When this happens, many classifiers will tend to produce biased results and therefore many methods for dealing with class imbalances have been developed [29]. These methods can be thought of as belonging to two different categories – those that modify the input data and those that modify the classifier itself.

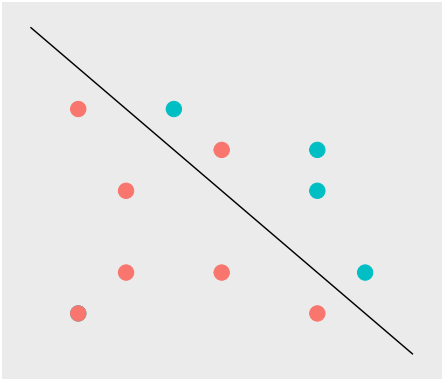
The most common approach in the first category is resampling. This usually means that only a subset of the examples from the majority class are used for training so that the training data becomes balanced. This subset can be selected randomly or in an informed manner and the method is usually referred to as *undersampling*. Another option is *oversampling*, which means that some of the minority class examples are replicated in the training set, thus again achieving balanced training data.



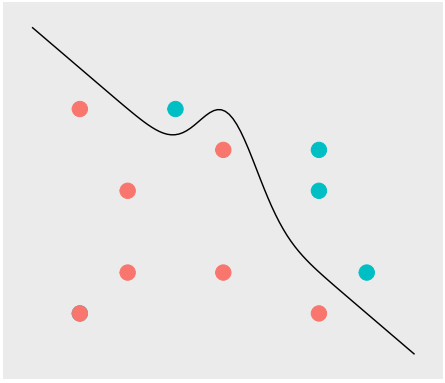
(a) Two possible boundaries.



(b) Maximum margin solution.



(c) Non-separable classes.



(d) Non-linear boundary.

Figure 2.1: Illustrations of support vector machine classification.

Both of these resampling techniques are associated with various problems. In undersampling, the main issue is that a large amount of potentially useful information is simply thrown away. The harm can be limited by performing informed rather than random undersampling but this doesn't remove the problem entirely and performing informed sampling is a complex problem in itself. Oversampling on the other hand leads to the training set containing several identical examples from the minority class, which can cause problems with overfitting.

As an alternative to these resampling methods, the classifier itself can be modified to better handle imbalanced data. The most common way of doing so is called *cost-sensitive learning* and means that the cost associated with misclassification is set differently for the different classes. In a binary SVM classifier, this would mean that instead of solving equation (2.2) we introduce two different cost parameters, C_+ for the positive class and C_- for the negative and then solve the optimization problem

$$\min \frac{1}{2}||\mathbf{w}||^2 + C_+ \sum_{i:y_i=+1} \xi_i + C_- \sum_{i:y_i=-1} \xi_i \quad \text{with} \quad \begin{cases} y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1 - \xi_i \\ \xi_i \geq 0. \end{cases} \tag{2.3}$$

If we let $C_+ = C_-$ we get the ordinary soft-margin SVM in (2.2), but if we instead let $C_+ \neq C_-$ the ratio of the two cost parameters can tuned so that the class imbalance is compensated for. The simplest choice is to let the cost parameters be inversely proportional to the frequencies of the two classes. This leads to a higher cost of misclassification for the minority class and thus allows for better performance on imbalanced data sets.

2.2.3 Feature Scaling

Another general problem when training several different types of machine learning classifiers is how to scale the input features. Many algorithms, including support vector machines, do not work well for "raw" or unscaled features. For example, if the goal is to predict the value of a house using its size in square meters and the number of bedrooms as input features, the first feature will typically be in the range $50 - 200\text{m}^2$ while the second will probably be in the range $1 - 4$. A soft-margin SVM that uses these features unscaled will then be prone to overfitting with respect to the size, since the absolute cost of misclassification is larger for this feature. The typical solution is to scale all the input features to some common range like $[0, 1]$ or in such a way that the mean is $\mu = 0$ and the standard deviation $\sigma = 1$.

However, in many applications involving a very large number of features, including natural language processing, one would want to give different weights to different features depending on their predictive performance. So continuing the house pricing example, the number of tiles in the bathroom is probably not a feature with good predictive power whereas the size of the house is. It is therefore desirable to scale the features differently according to their predictive power. Several methods for doing so

exist, and many of these work by scaling each feature f to some range $[0, x]$, where x is a numeric value determined by the predictive power of that feature. Forman [30] has proposed a method for selecting this upper bound, called Bi-Normal Seperation (BNS) feature scaling, and shows that it is successful for text classification tasks. BNS scaling is most easily described and implemented for binary classification, and as noted in Section 2.2.1 we can always interpret a multiclass classification problem as several binary classification tasks. The method will therefore here be described for binary classification only.

In order to determine the predictive power of each feature, we wish to estimate the conditional probabilities that a training example belongs to the positive and the negative class respectively, given that feature x_i is active. For binary features, x_i could for example indicate the presence of a word w_i. In the binary case, the conditional probabilities are easily estimated by simply counting the fraction of all positive examples with feature x_i active, which we call the true positive rate (TPR), as well as the fraction of all negative training examples with x_i active, denoted as the false positive rate (FPR). These values are then used to calculate the upper bound of the scaling range using

$$\text{BNS} = |F^{-1}(\text{TPR}) - F^{-1}(\text{FPR})| \tag{2.4}$$

as the upper bound. F^{-1} is here the inverse of the cumulative distribution function of the standard normal distribution. As an example, consider the task of classifying email as spam or non-spam. Let w_i be the word "free" and assume that 30 % of the spam emails in the training set contain the word "free" whereas only 1 % of the non-spam emails do. We would then get $BNS = |F^{-1}(0.3) - F^{-1}(0.01)| = 1.8$, and therefore scale the feature x_i to the range $[0, 1.8]$ so that $x_i = 0$ for emails that don't contain the word "free" and $x_i = 1.8$ for emails that do.

BNS scaling can also be generalized to work on continuous features [31]. The idea is then to binarize the continuous features by selecting a threshold v and calculating the TPR and FPR for training examples where $x_i \leq v$. The BNS can then be calculated using equation (2.4), and after doing this for all possible thresholds v of a feature x_i, we select the largest BNS obtained and then scale the feature x_i using this as the upper bound.

One potential problem occurs if the TPR or FPR is exactly zero or one, since the inverse of the normal distribution then goes to infinity. Forman [30] handles this problem by limiting TPR and FPR to the range [0.0005, 1-0.0005] and motivates this by his use of a finite size look-up table for the inverse normal distribution, but also suggests that it might be better to substitute TPR and FPR with a fractional count when they are zero.

		True label: +	True label: -
Predicted label	+	True positive (TP)	False positive (FP)
	-	False negative (FN)	True negative(TN)

Table 2.1: Confusion matrix for binary classification.

2.3 Evaluation Metrics

The most commonly used metric for the general classification problem is the accuracy, which measures the fraction of the test cases that are correctly classified. This is however not an appropriate metric to use when the classes are highly imbalanced, since a classifier which always predicts the largest class will achieve a very high accuracy. For example, in the severity detection problem only about 0.3% of the tokens in the test set are actually severity tokens, so a classifier that labels all tokens as non-severity tokens will achieve an accuracy of 99.7%. This does of course not mean that the classifier is performing well, and this motivates the use of other metrics.

For each class we can calculate the number of cases that were correctly assigned to the class (true positives), the number of cases that were wrongly assigned to the class (false positives), the number of cases that were correctly assigned to another class (true negatives) and the number of cases that were wrongly assigned to another class (false negatives). Table 2.1 displays these definitions in the form of a confusion matrix for the case of binary classification, where the task is to assign one of the labels + or - to each entity. The four counts in table 2.1 can be used to calculate a number of different evaluation metrics. The most common in natural language processing tasks are precision, recall and F1-score. These are calculated slightly different for binary and multiclass classification tasks.

2.3.1 Binary Classification

For binary classification, the precision is defined as

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.5}$$

where TP and FP are defined as in table 2.1. It can be thought of as the likelihood that an entity which has been labelled as positive actually does belong to the positive class. The recall, defined as

$$\text{recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{2.6}$$

instead measures the fraction of the entities that belong to the positive class which are correctly labelled. In other words, it measures how likely it is that an entity with a positive label is actually labelled as positive. The goal of a classification task is typically to achieve high precision and high recall simultaneously. Especially in natural language processing, a commonly used metric which weights both of these together is the F1-score, defined as

$$\mathrm{F1} = \frac{2 \cdot \mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}. \tag{2.7}$$

The F1-score gives equal weight to the precision and recall and rewards classifiers which achieve an intermediate precision and recall over those that achieve either a very high precision but low recall, or vice versa. It is typically used as the main evaluation metric in information retrieval and NLP tasks, and will be heavily used in this project.

2.3.2 Multiclass Classification

For multiclass classification problems, the precision, recall and F1-score are typically calculated using some sort of averaging over the classes, but the averaging can be performed in several different ways. The simplest option is to simply calculate the precision and recall for each class separately and then average the results. This is called macro averaging and gives equal weight to all classes irregardless of their frequencies. The definitions of macro precision and macro recall are

$$\mathrm{precision}_M = \frac{1}{k} \sum_{i=1}^{k} \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i} \tag{2.8}$$

$$\mathrm{recall}_M = \frac{1}{k} \sum_{i=1}^{k} \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i} \tag{2.9}$$

where k is the number of classes and M denotes macro. These metrics can then be used to calculate the macro F1 score, defined as

$$\mathrm{F1}_M = \frac{2 \cdot \mathrm{precision}_M \cdot \mathrm{recall}_M}{\mathrm{precision}_M + \mathrm{recall}_M}. \tag{2.10}$$

Alternatively, one can sum the true positives, false positives and false negatives over all classes before using the cumulative values to calculate the overall metric. This is called micro averaging and is defined as

$$\mathrm{precisison}_\mu = \frac{\sum_{i=1}^{k} \mathrm{TP}_i}{\sum_{i=1}^{k} \mathrm{TP}_i + \mathrm{FP}_i} \tag{2.11}$$

$$\mathrm{recall}_\mu = \frac{\sum_{i=1}^{k} \mathrm{TP}_i}{\sum_{i=1}^{k} \mathrm{TP}_i + \mathrm{FN}_i} \tag{2.12}$$

with

$$\mathrm{F1}_\mu = \frac{2 \cdot \mathrm{precision}_\mu \cdot \mathrm{recall}_\mu}{\mathrm{precision}_\mu + \mathrm{recall}_\mu}. \tag{2.13}$$

For multi-class classification tasks where each entity is classified as belonging to exactly one class, we will get

$$\sum_{i=1}^{k} TP_i + FP_i = \sum_{i=1}^{k} TP_i + FN_i = n \tag{2.14}$$

where n is the total number of entities that are being classified. This gives

$$\mathrm{precision}_\mu = \mathrm{recall}_\mu = \mathrm{F1}_\mu = \mathrm{accuracy} \tag{2.15}$$

and consequently the micro-averaged metrics are not very useful for this type of classification task. The severity classification task introduced in Section 3.1 is an example of such a task, and therefore only macro-averaged metrics will be used in the evaluation of that classifier.

2.3.3 Confidence Scoring

The precision, recall and F1 score are all one-dimensional performance indicators and as such say nothing about the variability of the performance or the confidence of the outputted scores. This is especially troublesome when evaluating models trained on very little data, where the performance would be expected to display high variability. One possibility is to estimate confidence intervals by performing bootstrap simulations on the training data, but Goutte and Gaussier [32] point out that these metrics "do not always correspond to sample means or medians" and that "the bootstrap method may fail to give accurate confidence intervals" . They instead propose a probabilistic interpretation of the precision, recall and F1-score which allows for the calculation of confidence intervals for these metrics using their derived probability distributions.

Starting from the initial assumption that the TP, FP, TN and FN counts follow a multinomial distribution with parameters π_{TP}, π_{FP}, π_{TN} and π_{FN}, they show the following properties:

1. The distribution of TP given $M_+ = TP + FP$ is a binomial with parameters M_+ and p.
2. The distribution of TP given $N_+ = TP + FN$ is a binomial with parameters N_+ and r.

where p and r are the precision and recall as defined in (2.5) and (2.6). Using the first of these conclusions, the likelihood function for the precision can be expressed as

$$L(p) = P(D|p) \propto p^{TP}(1-p)^{FP} \tag{2.16}$$

where $D = \{TP, FP, TN, FN\}$ is the data from a run. We wish to find the distribution of p given this data, so we apply Bayes' rule to get

$$P(p|D) \propto P(D|p)P(p) \tag{2.17}$$

where $P(p)$ is the prior distribution of the precision. The beta distribution is a suitable choice since it is a conjugate prior for the binomial distribution. Goutte and Gaussier further argue that there is no reason to favor high or low precision, so they select a symmetric beta distribution such that $p \sim Beta(\lambda, \lambda)$, which gives

$$P(p) = \frac{\Gamma(2\lambda)}{\Gamma(\lambda)^2} p^{\lambda-1}(1-p)^{\lambda-1}. \tag{2.18}$$

Inserting (2.18) and (2.16) into (2.17) results in

$$P(p|D) \propto p^{TP+\lambda-1}(1-p)^{FP+\lambda-1} \tag{2.19}$$

which means that $P(p|D) \sim Beta(TP + \lambda, FP + \lambda)$. Before we can construct a confidence interval for the precision, a value of λ must be selected. Goutte and Gaussier argue that using the so-called Jeffrey's non-informative prior with $\lambda = 0.5$ has advantages over choosing the uniform prior with $\lambda = 1$. Once a λ has been chosen, creating a confidence interval for the precision is simple. All we need to do is find the appropriate quantiles of the beta distribution, which can be done by evaluating the beta inverse cumulative distribution function with these parameters at our desired confidence threshold. This is easily done in most statistical software packages or by using tables. The confidence interval will not be symmetrical since the beta distribution is not. If we instead want a confidence interval for the recall, we simply make use of the second property listed above, and get

$$P(r|D) \sim Beta(TP + \lambda, FN + \lambda). \tag{2.20}$$

Now, moving on the the confidence of the F1-score, Goutte and Gaussier note that for two variables $X \sim \Gamma(\alpha, h)$ and $Y \sim \Gamma(\beta, h)$, which follow gamma distributions with the same shape parameter h, we get

$$\frac{X}{X+Y} \sim Beta(\alpha, \beta). \tag{2.21}$$

This means that we can express the posterior distributions of the precision and recall in terms of gamma distributed random variables, so that

$$p = \frac{X}{X+Y}, \quad r = \frac{X}{X+Z} \quad \text{with} \quad \begin{cases} X \sim \Gamma(TP + \lambda, h) \\ Y \sim \Gamma(FP + \lambda, h) \\ Z \sim \Gamma(FN + \lambda, h). \end{cases} \tag{2.22}$$

The F1-score, defined in (2.7), can then be written as

$$F1 = \frac{U}{U+V} \quad \text{with} \quad \begin{cases} U \sim \Gamma(TP + \lambda, 2h) \\ V \sim \Gamma(FP + FN + 2\lambda, h). \end{cases} \tag{2.23}$$

If we now want to create a confidence interval for the F1-score, finding the appropriate quantiles of this distribution is not as simple as for the beta distribution, since its inverse cumulative distribution is not readily available. Instead, one approach is to simulate $F1$ as defined in (2.23) a large number of times and estimate the desired quantiles from these n samples by simply ordering them and taking the values at position $n \cdot \alpha$ and $n(1-\alpha)$ to achieve a confidence interval with confidence level $1-\alpha$.

3

Method

This chapter will describe how the task of extracting severity information on an event level from ICSRs was addressed. The first section will give an overview of the general approach that was used and then the data preparation is described in detail, including data selection and the use of multiple pre-processing steps. The following section describes how a gold standard annotated data set, which was later used for training machine learning models, was manually created after developing a set of specific annotation guidelines. Next comes a description of the simple rule-based baseline methods used for later comparison against machine learning methods. The following section introduces the methods used for feature extraction, shaping and scaling and after this comes a detailed description of all the machine learning experiments that were performed. The final section then defines the evaluation metrics used to bench-mark both the baseline and the machine learning methods.

3.1 General Approach

A two-step strategy was used to extract severity information from the narratives of ICSRs. The first step was detection of severity expressions from the narratives, i.e. finding words and groups of words that describe the severity of a medical event such as the word ”mild” in the following sentence: ”The child developed a mild rash within 24 hours after vaccination”. This step was performed by training a support vector machine (SVM) classifier on a labelled gold standard data set that had been manually annotated according to a set of guidelines, described in detail in Section 3.4. The severity identification problem was approached as a binary classification task, where each token in every sentence containing a reported ADR was classified as either describing the severity of that ADR or not. The input data to the SVM classifier included features such as the token to be classified itself, its part-of-speech tag, the number of tokens between the token and the ADR in question and many others, all listed in Section 3.6. Features describing the neighboring tokens, such as their POS-tags, were also used. The output from this step was a collection of tokens, or a so-called bag-of-words, describing the severity of each reported ADR in an ICSR.

The second step was then to classify the severity of each ADR on a scale of mild/-moderate/severe, given a bag-of-words assumed to contain all relevant severity descriptors. This was again treated as a classification problem and an SVM was trained

on manually annotated data. The output from this classifier was multi-class rather than binary and the input features much simpler – here, only the bag-of-words itself was fed into the SVM. This step resulted in a severity label for each reported ADR in an ICSR for which any severity descriptor had been found. For reported ADRs without any detected severity descriptors, the severity assigned was ”none”.

Both of the SVM classifiers were first trained separately on a training set of 1103 narratives and then evaluated on a test set of 476 narratives, first separately and then jointly, by feeding the output from the first classifier into the second. The performance was evaluated using the metrics precision, recall and F1 score, all defined in Section 2.3. A number of different parameter settings for both classifiers were tested and for the severity detection classifier a number of other aspects, such as the effect of varying the size of the training set, shaping and scaling the features in different ways and removing one feature at a time, were also explored. All of these experiments are detailed in Section 3.7, and their results in Section 4.2 and 4.3.

3.2 Data Selection

All the data used for this thesis originates from VigiBase, the UMC data base of ICSRs. Reports were selected randomly under some constraints that will be outlined and motivated in detail in this section. First of all, ICSRs exist in various formats. VigiBase contains reports in two different formats: INTDIS and E2B(R2). The older INTDIS format was developed by the WHO but has today mostly been replaced by the newer E2B format, which was developed by the International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use (ICH) and allows for exchange of more detailed information. Most of the reports in VigiBase are in the latter format and since the former only allows for very short narratives, reports submitted in INTDIS format are of marginal interest to this project. The E2B format however, includes several free-text fields of unlimited length. Therefore, only reports in E2B format were used for this study. Additionally, for practical reasons, only reports entered into VigiBase before October 2015 were used.

Furthermore, all reports in VigiBase do not have narratives. It is possible for a reporter to leave some or most of the fields in an ICSR blank, and reports that lack a free-text narrative are naturally not of interest here. UMC has developed a simple filter for finding reports with narratives, which essentially checks that the case narrative section (described below) is non-empty and does not simply contain an expression such as ”None”, ”NA” or ”Not provided”. For this project, only reports that passed the filter were used.

In addition, only narratives in English were of interest for this study. Since there is no field which records of the language of each report, the reports had to be filtered after extraction from VigiBase. This step is described in Section 3.3.

Finally, not all free-text fields of an ICSR are relevant in the search for severity infor-

mation. For example, the literature reference section "should be used for literature article(s) that describe individual case(s)" [7] and thus clearly not for describing the severity of reported ADRs. The case narrative section on the other hand "captures a focused, factual and clear description of the case" so one would expect severity information to be found there. In total, eight of the E2B free-text sections were selected by experienced UMC researchers for inclusion in the study, based on their assessment of which sections were likely to contain useful information. These sections were the following:

- Case narrative including clinical course, therapeutic measures, outcome and additional relevant information
- Reaction/event as reported by primary source
- Text for relevant medical history and concurrent conditions
- Results of tests and procedures relevant to the investigation of the patient
- Additional information on drug
- Reporter's comments
- Sender's comments
- Sender's diagnosis/syndrome and/or reclassification of reaction/event

No restrictions were placed on the reporter qualification field, which captures the role of the reporter of the case, though this was discussed in great detail. Its possible values are

- Physician
- Pharmacist
- Other health professional
- Lawyer
- Consumer or other non health professional

and several of these can be selected. The fractions of reports listing each category in VigiBase are shown in table 3.1. As mentioned in Section 1.1, there is reason to believe that consumers report severity information to a greater extent than health care professionals, and using severity information in signal detection is considered by the UMC to likely be most useful for patient reports. However, limiting the study to only patient reports would make it difficult to draw any general conclusions about the usefulness of severity information in signal detection. This is because patient reports are expected to use vocabulary and expressions for describing the severity of their symptoms which differ significantly from those used by health care professionals, and so a model trained on only patient reports would most likely not perform very well on reports from other reporter qualification groups. The fraction of reports in the selected data set submitted by each reporter type is also shown in table 3.1. As is clear from the table, there are significant differences between the fractions in VigiBase and the selected data set. The reason for this is the previously mentioned selection criteria that selected reports must have a narrative in English,

Reporter qualification	Share of reports in VigiBase	Share of reports in selected data set
Physician	52.0 %	38.6 %
Pharmacist	8.3 %	6.5 %
Other health professional	16.2 %	12.1 %
Lawyer	2.1 %	0.2 %
Consumer or other non health professional	35.0 %	11.3 %
Other	11.8 %	0.0 %
None	18.3 %	39.1 %

Table 3.1: Fraction of reports from each possible reporter qualification type, in VigiBase and in the selected data set. "Other" is not a valid reporter qualification group in the E2B format, and mostly refers to reports in INTDIS format. The category "None" refers to reports for which no reporter qualification has been entered. The fractions sum to more than 100% because multiple reporter qualification groups can be selected for a single report. The differences in the distribution between VigiBase and the selected data set are a consequence of the selection criteria that selected reports must include an English narrative.

which leads to a large fraction of the reports originating from the United States. All U.S. reports lack information about the reporter qualification group, and thus end up in the group "None". The distribution of the reports by country is specified in Section 4.1.

3.3 Pre-Processing

As mentioned in Section 2.1, most tasks in natural language processing depend heavily on multiple pre-processing steps where the text under analysis is first split into smaller entities such as sections, sentences and tokens and then automatically annotated with for example part-of-speech tags or named entities such as person names and medical events. All the pre-processing in this thesis was performed using existing methods, mostly from the cTAKES framework described in Section 2.1.4. Some of these methods had also previously been adapted by researchers at UMC for another project and their implementations were also used to a large extent.

The very first step of pre-processing was to filter out reports in other languages than English. This was done in the programming language Python, using the language detection package *langdetect*, which was developed in Java by Shuyo [33] and ported to Python by Danilek [34]. The language detection algorithm was run separately on each section longer than 60 characters of every report. All reports with any section not in English were filtered out. Reports with no section longer than 60 characters were also filtered out, since these were too short for the language to be accurately determined. The remaining narratives were then again saved to a delimited text file.

The language filtered text file was then read into the UIMA framework using Java. Each section of every report was then run through a pre-processing pipeline consisting of sentence-splitting, tokenization, POS-tagging and ADE detection. The first three steps simply used cTAKES built-in methods [26] while the last step used the cTAKES fast dictionary look-up algorithm for event detection combined with a UMC developed method for event disambiguation and some additional logic for determining which events were ADEs.

The fast dictionary look-up was used in ”overlap” configuration which means that it searches for matches between sequences of tokens in a text and a supplied dictionary while allowing other tokens in between the tokens of each dictionary term. An example is the expression ”blood pressure was very low” where a standard dictionary look-up algorithm would not find a match for the dictionary term ”blood pressure low” while the overlap configuration does accept the words ”was” and ”very” in the middle of the hit term. The number of tokens allowed in between tokens of a dictionary hit term is configurable and was set to the default value of 2 for consecutive skips and the default 4 for the total number of skipped tokens allowed within the span of a dictionary hit term.

The dictionary used was a subset of the Medical Dictionary for Regulatory Activities (MedDRA) which is a standardized terminology developed by the ICH ”to facilitate sharing of regulatory information internationally for medical products used by humans” [35]. MedDRA contains a hierarchical structure of medical terms, starting at the bottom with ”Lowest Level Term” (LLT) where each term is associated with a ”Preferred Term” (PT). PTs are in turn grouped into ”High Level Terms” (HLT) that are mapped to ”High Level Group Terms” (HLGT) and finally these are divided into ”System Organ Classes” (SOC). The previously mentioned expression "blood pressure low" is an example of a lowest level term. It sorts under the preferred term "Hypotension" which in turn sorts under the high level term "Vascular hypotension disorders" which sorts under the system organ class "Vascular disorders". For this project, only a subset of the dictionary was used. Specifically, only terms contained in 24 of the 27 SOCs were used – the excluded SOCs were ”Surgical and medical prodecures”, ”Social circumstances” and ”Product issues”. These categories were left out in order to avoid annotating medical events for which no severity could reasonably be expected to be found. For example, the MedDRA guidelines state the following [36]:

> Essentially, SOC Social circumstances contains information about the person, not the adverse event. As an example, terms such as PT Drug abuser and PT Death of relative are found in this SOC, whereas their respective disorder terms such as LLT Drug addiction and PT Death are found in SOC Psychiatric disorders and SOC General disorders and administration site conditions, respectively.

It therefore seems reasonable to expect that terms contained by the SOC Social circumstances will themselves never be associated with a severity, and this argument also applies to the other excluded SOCs.

Running the fast dictionary look-up in overlap mode results in a large number of

medical event annotations. Many of these are redundant and need to be removed. An example is the expression ”back pain” for which two medical events will be annotated: ”pain” and ”back pain”. In general, one prefers to keep the more specific term so in this case the medical event ”pain” should be removed. Researchers at UMC have developed their own algorithm for removal of redundant medical events produced by the cTAKES fast dictionary look-up algorithm. The general idea is to keep the terms that contain the longest dictionary hits, rather than the longest annotations. The algorithm works by comparing annotations pairwise. For each pair, their spans are compared and if one annotation is entirely enclosed by the other, the outer annotation is kept. If the spans are exactly the same, the number of reports in VigiBase listing that reaction are compared, and the more commonly occurring reaction is selected. In the unlikely case that the number of reports in VigiBase listing the two events are exactly the same, the term with the largest number of associated HLTs is selected.

If the spans are not identical but overlapping, the algorithm will first try to select the term with the largest number of hit words, defined as the words occurring in the detected dictionary term. For example, the expression ”attempted to commit suicide” results in a dictionary hit on the term ”attempted suicide”, so the number of hits words is 2. If the number of hit words is the same for both annotations, the fraction of all the words in the two annotations that are hit words are compared and the annotation with the larger fraction is selected. Continuing the previous example, the total number of words would there be 4 so the fraction of words that are hit words would be $2/4 = 0.5$. If this again results in a tie between the annotations, the number of reports in VigiBase listing each reaction are compared just like for annotations with identical spans, and as a last resort the number of HLTs associated with each term is used to select one annotation.

The next step after removing redundant events is to determine which events are ADRs. Rather than develop complicated algorithms for determining whether a medical event is likely to be an ADR or not, a simple rule-based approach was taken, making use of the E2B(R2) ”Reaction/event MedDRA term (Preferred Term)” field of the ICSRs. This field is used to supply the MedDRA Preferred Term of each observed reaction that is suspected to be an ADR. The contents of this field were therefore compared against the medical events found in the previous step and the subset of medical events that were also found in the list of reported terms were annotated as reported reactions. Rather than comparing strings, matching was performed by first mapping each medical event and reported Preferred Term to its corresponding PT and HLT codes. These codes are defined in the MedDRA framework and provide one-to-one mappings in the case of PT codes, while one PT code might be associated with several HLT codes. The codes found in the narrative were then compared to the reported codes and a match was registered whenever there was a match either on the PT code or an HLT code. For example, a narrative might contain the word ”rash” and have a reported PT that is ”injection site rash”. These terms have different PT names and codes but both sort under HLT Rashes, eruptions and exanthems (PT Injection site rash *also* sorts under HLT Injection site reactions). ”Rash” would therefore be considered a reported reaction. The reported

reactions found in this way were then used as a proxy for ADRs. One obvious problem with using this method is that many medical events that are ADRs are missed simply because they use a different wording than the reported PTs and the MedDRA hierarchy does not manage to compensate for this. An example is the word ”pain” which sorts under HLT Pain and Discomfort while for example the PT ”injection site pain” sorts only under HLT Injection site reactions. The medical event ”pain” will therefore not be considered a reported reaction in this case, even though it is clearly an ADR. These issues are discussed further in Section 4.1.1.

3.4 Human Annotations

To train learning algorithms for severity detection and classification, labelled training data is needed. A gold standard annotated data set was therefore created from 1579 ICSRs which had undergone the pre-processing steps described in Section 3.3. All of the reports were annotated by the author, following a set of guidelines which were developed specifically for this purpose. Difficult cases were discussed with a UMC research pharmacist, who was also one of the main contributors to the development of the guidelines. In addition, a subset of 100 reports which had not previously been discussed, were also fully annotated by the same UMC pharmacist. These double annotations were performed after all the main annotations had been done and the annotations on this subset were used to calculate inter-annotator agreement F1 scores (defined in Section 2.3) for both the detection and classification problems. The scores were used to evaluate the annotation guidelines since they to some extent measure how clear the guidelines are, and inter-annotator F1 scores are also typically interpreted as an upper bound on the best achievable performance of any algorithm aiming to solve the task in question.

The annotation guidelines were developed based on two other available guidelines for annotating severity in clinical narratives. These other guidelines were however not very detailed or specific and some of their definitions of severity did not agree with the one used in this project. Many additions and modifications were therefore made, resulting in an entirely new set of guidelines, found in appendix A.

Some additional language filtering was also performed as a part of the annotation process. Because several reports were noted to have passed the language filter despite being written partially or entirely in other languages than English, a manual filtering step was also performed. In practise, this meant that during the annotation process, when a non-English report was found it was manually marked as non-English and then excluded from all further analysis.

3.5 Baseline Implementations

To evaluate the performance of the machine learning severity detection and classification algorithms, a number of simple, rule-based baseline methods were also

created so that they could be used for comparison.

3.5.1 Severity Detection Baselines

For the severity detection task, three baselines were created. All made use of a vocabulary of previously seen severity tokens, created from a training data set. For the first baseline, a token in the test set was then classified as being a severity descriptor for a given reported reaction if that token existed in the severity vocabulary and within the same sentence as the reported reaction. The second baseline also required that the number of tokens between the token in question and the reported reaction was not larger than 3. The third and final baseline only classified tokens as severity descriptors if they were positioned immediately before reaction in question and existed in the vocabulary of previously seen severity tokens. The results from running these baselines on the annotated data set are shown and discussed in Section 4.2.1.

3.5.2 Severity Classification Baseline

Only one baseline was constructed for the severity classification task. This baseline made use of three tables, each counting the number of times a token was assigned each grade of severity in the annotated training set. For example, the word ”some” was classified twice in the training set as being of grade ”mild” and once as being of grade ”severe”. When classifying a bag-of-words in the test set, the most frequent grade of each token in that bag-of-words was queried from these tables and the overall grade of the bag-of-words was taken to be the highest of these grades. For bags-of-words containing only previously unseen tokens, the grade was predicted to be ”severe” since this was by far the most frequently occurring grade in the training set. The performance of this baseline is analyzed in Section 4.3.1.

3.6 Feature Extraction

Rather than simply feeding each token that is to be classified into an SVM classifier, it is common in natural language processing to derive a large number of features related to the token in question that are used as input to the classifier. Those features are then typically scaled and since an SVM cannot handle categorical features natively, those must be encoded. The following sections describe the details of all of those steps, for both the detection and the classification problems.

3.6.1 Feature Selection

A total of 33 features of 19 different types were used for the severity detection problem. These features were extracted for all candidate pairs consisting of one token and one reaction both in the same sentence. The 19 different types are shown

in table 3.2, where they are divided into three categories. The first category consists of features relating to a specific token and these were extracted not only for the token under classification but also for the preceding and the following token, giving a total of 21 token features. The second category relates to the reaction for which we are aiming to detect the severity and the third category contains features that describe the context of the token-reaction pair under classification.

For the simpler severity classification problem, only one input feature was used. This feature was a bag-of-words containing all the severity tokens relating to all occurrences in an ICSR of a specific reported reaction.

3.6.2 Feature Encoding

A support vector machine classifier can only handle numeric features. All binary and categorical features must therefore be encoded before being fed into the classifier. Binary features are easily encoded simply as 0 for "false" and 1 for "true". Categorical features are encoded using "one-hot" encoding, which means that a feature with n possible values is represented using n binary features, each of which indicates the presence or absence of a specific value. If we for example have a feature that is a POS-tag with the possible values {noun, verb, adjective}, we can encode a noun as $[1, 0, 0]$, a verb as $[0, 1, 0]$ and an adjective as $[0, 0, 1]$. In general, only $n - 1$ binary features are required since a vector consisting of only zeros can be used to represent the last feature, but for simplicity n features will always be used here.

One issue arising from the use of one-hot encoding is that the total number of features after encoding depends on the number of levels in the categorical variables. This means that when encoding the test set, only levels available in the training set can be used or else the number of encoded features would change, and a model trained on the training set would not be applicable to the test set. An extra level, "unseen", was therefore added to all categorical variables in the training set, and a vocabulary of all the levels of each categorical feature in the training set was saved. All the values of the categorical features in the test set were then checked against this vocabulary and levels only present in the test set were set to "unseen". Continuing the example above, the POS-tag vocabulary created from the training set would be {noun, verb, adjective, unseen} and if the POS-tag "adverb" is encountered in the test set, it would be encoded as "unseen" which has the representation $[0, 0, 0, 1]$.

3.6.3 Feature Scaling

All features used in both classifiers were first scaled to the range [0, 1]. For the severity detection classifier, all binary and continuous features were then scaled again using the BNS feature scaling developed by Forman [30], described in Section 2.2.3. This second scaling was not used on any of the categorical features in the main severity detection model, since this was not found to improve the overall performance. A comparison of the results obtained when using no feature scaling, BNS scaling of all feature types and BNS scaling of only binary and continuous features

Feature category	Feature name	Data type	Description
Token features	Token	Categorical	The token itself, in lower case.
	POS	Categorical	The token's part-of-speech tag, as detected by cTAKES.
	IsInteger	Boolean	Whether the token is an integer.
	IsRoman	Boolean	Whether the token is a roman number in the range 1-5.
	AllUpperCase	Boolean	Whether the token is entirely in upper case.
	IsCapitalized	Boolean	Whether the first letter is upper case and the remaining are not.
	nChars	Numeric	The length of the token.
Reaction features	Reaction	Categorical	The detected reaction itself.
	HitTerm	Categorical	The dictionary term found to match the detected reaction.
	ReportCount	Numeric	The number of reports in VigiBase with this reaction reported.
	IsContiguous	Boolean	Whether the reaction annotation contains tokens that are not part of the dictionary hit term.
Contextual features	nTokensBetween	Numeric	The number of tokens between the token and the reaction.
	IsEnclosed	Boolean	Whether the token is located within the span of the reaction.
	IsBefore	Boolean	Whether the token is located before the span of the reaction.
	IsAfter	Boolean	Whether the token is located after the span of the reaction.
	PartOfEvent	Boolean	Whether the token is contained by a medical event.
	EventBetween	Boolean	Whether there are any medical events located between the token and the reaction.
	InList	Boolean	Whether the token is in a list.
	IsParentheses	Boolean	Whether the token is enclosed by parentheses.

Table 3.2: The 15 different features types used for severity detection. These features were extracted for each candidate token-reaction pair and used to classify whether of not a severity relations exists between the token and the reaction.

is presented in Section 4.2.6. The TPR and FPR were limited to the range $[10^{-6}, 1 - 10^{-6}]$. The cutoff value was chosen based on the size of the data set ($\sim 10^5$ tokens) so that the minimum value would correspond to less than the contribution from one token ($\sim 10^{-5}$).

In practise, the BNS scaling was performed by first scaling all features in the training set to the range [0, 1]. The same scaling was applied to the test set, so that continuous features in the test set with values larger than the maximum or smaller than the minimum value in the training set would be scaled to values outside this range. The BNS was then calculated independently for each feature using data from the training set and applying the threshold method outlined in Section 2.2.3 on continuous features. The features in both the training and test sets were then simply multiplied by the BNS found for that feature.

The single bag-of-words feature used for severity classification was not BNS scaled. This was partially because in order to perform BNS scaling on a multiclass classification task, the scaling would need to be redone for each binary classification task. Because the transformation from one multiclass to several binary classification problems was handled internally by the SVM package LIBLINEAR, this would have been troublesome to implement. Furthermore, BNS scaling was not assessed as likely to improve the performance of this classifier since BNS scaling of the categorical features in the severity detection classification task had failed to improve the performance of that classifier.

3.7 Training and Tuning of Models

The 1579 annotated reports were split into a training set, consisting of 1103 reports, and a test set consisting of 476 reports. Features were extracted as outlined in Section 3.6 for all reports that passed the manual language filtering step described in Section 3.4. Two separate support vector machine classifiers, one for severity detection and another for severity classification, were then developed and tuned using the training set before being evaluated on the test set. 5-fold cross-validation was used on the training set during the development phase of both algorithms, to ensure that characteristics of the test set were not allowed to influence the design of the classifiers.

3.7.1 Severity Detection

The input data to the severity detection SVM was the features defined in table 3.2, extracted for each candidate token-reaction pair in every ICSR. Since these pairs are defined as combinations of tokens and reported reactions both within the same sentence, not all severity relations were included in the input data. This means that even if perfect performance could be achieved for the severity detection algorithm, severity relations spanning over more than once sentence would not be identified. This limitation was due to computational capacity constraints during the training of

the SVM, and the expectation was that most severity relations should occur within one sentence.

The split into five folds for cross-validation was performed by first randomly shuffling all reports in the training set, and then iterating over them and assigning all the tokens of a report to one fold chosen so that the number of severity tokens and the total number of tokens were roughly equal across all folds. Since both the lengths of the reports and the number of severities per report varied significantly, this led to some folds containing a larger number of reports and/or tokens than others. The reason for assigning entire reports, rather than just individual tokens, to each fold was that reports can sometimes contain the same expression or even paragraph several times, and including different occurrences of the same expression in different folds could therefore lead to an overestimated performance. The folds that were created from the severity detection input data were later reused for cross-validation of the severity classification algorithm.

A linear kernel SVM was used for the severity detection task, using the LIBLINEAR algorithm developed by Lin et al [27] and implemented in R by Helleputte [37]. Due to the large imbalance between the classes, class weighting was performed as described in Section 2.2.2. The cost parameter C was tuned using five-fold cross-validation on the training set and assessing the performance of the classifier within a large range of different values of C. The results from these runs are discussed in Section 4.2.2. The overall performance obtained when using the best value of the cost parameter is reported in Section 4.2.3, where the errors made by this classifier are also analyzed in detail. The effect of varying the size of the training set was then studied, by training classifiers on subsets of the training data and evaluating them on the full test set. The subsets were created by splitting the training data into 10 folds just like for the cross-validation described above, in order to achieve balanced subsets with respect to both the number of tokens and the number of severity tokens. Again, this method of splitting the data also avoids different instances of the the same expression ending up in different subsets. A severity detection SVM was then trained first on one fold, then one two folds and so on until all ten folds were being used. This was repeated five times, generating new folds for each repetition. The results are presented in Section 4.2.4. Furthermore, the contribution from different types of features were investigated by removing one set of features at a time and evaluating the change in performance. The results from these experiments are discussed in Section 4.2.5. Finally, the effects of using the BNS scaling described in Section 3.6.3 were evaluated by training models with and without BNS scaling for different features types. Those results are presented in Section 4.2.6.

3.7.2 Severity Classification

The input data to the SVM for severity classification was a bag-of-words aggregated from all severity tokens associated with a certain reaction in an ICSR. The severities of multiple instances of the same reaction within a report were thus combined. During the manual annotation, all severities has been classified individually so in order to determine the overall severity of the aggregated bag-of-words, the maximum

severity was used meaning that if any of the associated severities was ”severe”, the overall severity was set to ”severe” etc.

This SVM was also trained with a linear kernel and using the LIBLNEAR package. No class weighting was used here, since the classes were not heavily imbalanced. The cost parameter was tuned using cross-validation on the folds created for the severity detection task, and the results from this are shown in Section 4.3.2. Section 4.3.3 discusses the performance of the classifier obtained using the best value, and provides an analysis of its most common errors. The effect of varying the size of the training set was also investigated, and the results are presented in Section 4.3.4.

3.7.3 Combined Severity Detection and Classification

After the SVMs for severity detection and classification had been developed and tuned individually, they were trained on the full training set and then combined into a pipeline where the output from the severity detection classifier was fed into the severity classification classifier. This means that all tokens which were predicted by the first classifier to be severity tokens were aggregated into bags-of-words for each reaction in every ICSR and thus some tokens in these bags-of-words might not actually be severity descriptors. In other words, some propagation of errors should be expected. This pipeline was run on the full test set. The overall performance was then evaluated by comparing the maximum annotated severity grade of each reaction in every ICSR of the test set to that produced by the pipeline. Since only the ”maximum” severity associated with a reaction determines its correct severity grade, this means that some errors in the input data might be cancelled out in the severity classification step. The results form running the full pipeline are presented and discussed in Section 4.4.

4

Results and Analysis

The previous chapter described the methods developed for extraction of severity information from clinical narratives, including the annotation of 1579 reports as well as the training of two support vector machine classifiers, one for detecting severity tokens and the other for classifying collections of such tokens as mild, moderate or severe. This chapter will focus on the results obtained, both from the annotation process itself and when using the annotated data to evaluate the two classifiers.

4.1 Human Annotations

The annotated set of 1579 reports can be used to gain valuable insights on the prevalence and distribution of severity information in reports from VigiBase . Apart from a small pre-study before the start of this project, there is no previous knowledge on the frequency of severity information in ICSRs or how this information correlates with other factors such as the reporter type, country of origin or seriousness.

Overall, 273 of the 1579 reports were found to contain severity information for at least one reaction. Severity information was found for a total of 368 unique reactions and the total number of severity tokens was 931. This implies that roughly one in five ICSRs can be expected to contain severity information, but since severity was only annotated for correctly identified reactions, this should be seen as a lower bound. Out of the 931 severity tokens, 823 (88 %) were found within the same sentence as their associated reaction.

Figure 4.1 shows the distribution of the reports in the annotated data set by country, but only countries with at least 20 reports in the data set are shown separately. The number of reports from each country that contain severity information for at least one reaction is also shown. More than two thirds, 37 % of the reports originate from the United States. This is hardly surprising since the reports are required to have a narrative in English, and the U.S. is the largest English-speaking country in the world. The second and third largest contributors to the data set are Germany (20 %) and India (9 %). The countries with the largest fractions of reports containing severity information are Greece (39 %), Germany (29 %) and Belgium (29 %), whereas the average fraction of reports containing severity is 17 %.

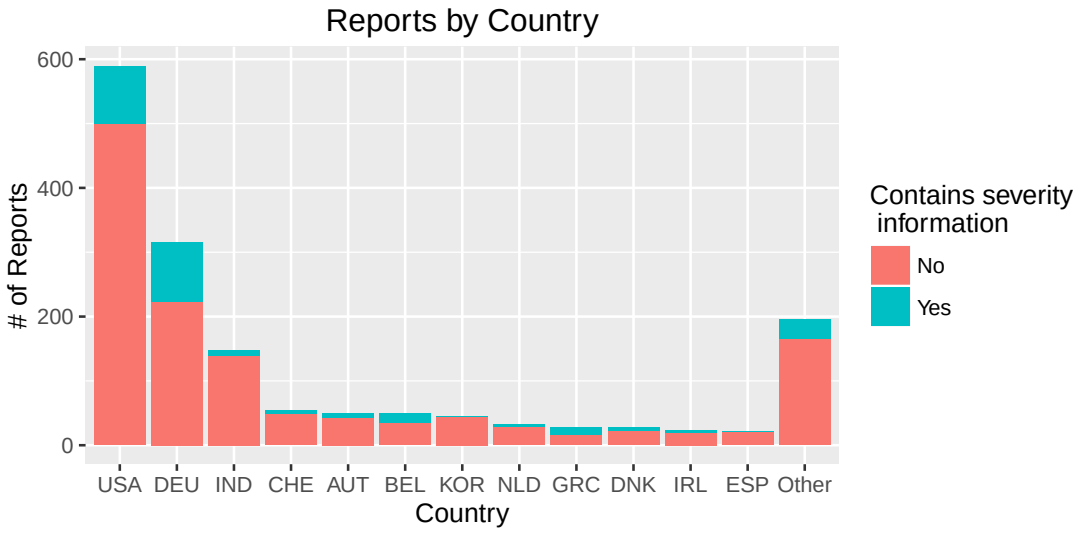


Figure 4.1: Distribution of the annotated reports by country. Countries with less than 20 reports in the data set are grouped together in the ”other” column.

4.1.1 Observations

Some notable observations were made during the annotation process, mostly regarding the performance of the event detection algorithm and the identification of reactions but also on the nature of how severity is reported in ICSRs. These observations will be discussed here.

First of all, as already noted in Section 3.3, the dictionary look-up algorithm used for event detection was far from perfect. Many events were missed entirely and some words were incorrectly identified as events, but unfortunately it was not possible to measure the performance qualitatively. This is because there does not exist a data set of ICSRs that have been annotated for medical events. In general, the look-up algorithm missed many events in plural form, meaning that it would identify for example the word ”rash” as a medical event, but not ”rashes”. The performance could perhaps have been improved by either using an additional or larger dictionary of medical events, or by attempting to incorporate terms in plural form into the existing MedDRA dictionary for example by adding an extra ”-s” or ”-es” to all terms in the dictionary.

Another common problem was that the spans of events were too long, so that the meaning of the event was distorted. An example is the following sentence: ”Medical history included glaucoma since 2009, two unspecified eye operations and atrial fibrillation.”, where two medical events are detected: glaucoma unspecified and atrial fibrillation. The second of these is correct, but the first is not since the word ”unspecified” does in fact not refer to the glaucoma but to the eye operations. This type of error did occur quite frequently, but it was less common for events that were also identified as reactions.

A third common error was for the algorithm to detect only lab values by themselves, rather than together with their evaluations. An example is the expression

”mild elevation of liver function tests” for which the event ”liver function tests” is detected, but this misses the very important information that the test results were elevated. These errors were so frequent that the annotation guidelines were updated to include an exception to the rule that medical events must be correctly detected if a severity is to be annotated. Since a test result itself would rarely be reported as a suspected ADR if it was not abnormal in some way, severities were annotated even to reactions detected without an evaluation such as ”increased”, ”decreased” or ”abnormal”.

The identification of medical events as reactions or ADRs was also non-trivial. As mentioned in Section 3.3, many of the identified medical events did not result in a match with a reported term even though medically they did describe the same event. For example, an expression like ”swelling of upper arm” which gives a dictionary hit on ”swelling arm” does not have any HLT in common with the reported reaction ”injection site swelling”. The conclusion that can be drawn is that the MedDRA hierarchy does not fully capture the relations between different medical terms in a way that would be most useful here. A possibility would be to instead use another hierarchically structured dictionary, but since MedDRA is the industry standard and many reporters intentionally try to use MedDRA terms when writing narratives, it is unlikely that this would improve the overall performance.

It was also noted that the number of reported reactions listed varied significantly between reports. It appears that some reporters tend to list only the final diagnosis of the case as a reaction, while others list all the observed symptoms. This does of course strongly affect the number of medical events in each reports that are marked as reactions. Many reports that listed only one or a few differential diagnoses as reactions were noted to contain detailed descriptions of each symptom, including their severity, but since these were not picked up as reactions their severities could not be annotated. This is of course unfortunate since ideally, for signal detection purposes, we would like to know the severities of all the symptoms as well and not just the diagnosis. Improving the identification of reactions is however outside the scope of this thesis.

The language filter that was applied to all reports in the pre-processing step was noted to work well for reports written entirely in one language, but not as well for reports using a combination of English and another language. Many countries, but particularly the Netherlands and Spain, provide narratives in their own official language, but also provide a brief summary of the case in English. These reports had the be manually labelled as non-English during the annotation process. While this is fine when developing and testing the severity extraction algorithms on a labelled data set, it is likely to affect the performance of any trained algorithm that is later run on unseen data from VigiBase . Developing a better language filter would therefore be desirable.

4.1.2 Inter-Annotator Agreement

Out of the 100 reports annotated by two annotators (A1 and A2), 26 were found to contain severity information, according to both the annotators. In addition, annotator A1 found found severity information in additional 2 reports where annotator A2 found none, and annotator A2 found severity information in 1 report where annotator A1 found none. The total number of severity relations annotated by both annotators was 64. The annotators also recorded an additional 8 severity relations each, which the other had not marked. These numbers give an inter-annotator agreement F1 score for severity detection of F1 = 0.889 which is considered satisfactory since it is better than the results reported for severity detection by Dligach et. al [21], who achieved an inter-annotator agreement score of F1 = 0.871 on the SHARP corpus and F1 = 0.664 on the ShARe corpus. Furthermore, those numbers refer only to the annotation of relations between pre-annotated severity modifiers and medical events, whereas the annotations performed here also required identification of severity modifiers which should reasonably be considered a more difficult task.

All severity relations were also annotated with a severity grade by both annotators. Among the 64 severity relations agreed upon by both annotators, *all* were classified identically. In other words, the inter-annotator F1 score for severity classification was F1 = 1. This is evidently a very satisfactory score.

Overall, the remarkably high inter-annotator scores imply that the annotation guidelines work well and are specific enough that severity can be annotated in a consistent manner. Of note is also that the severity detection problem is clearly the more difficult of the two and it is therefore reasonable to expect weaker results from the detection task than from the classification task, also for a trained SVM classifier.

4.2 Severity Detection

The annotated data set described in the previous section was used to train and evaluate an SVM for the detection of severity tokens, as described in Section 3.7.1. It was also used to evaluate several rule-based baselines and the results from both of these tasks are presented and analyzed in the following sections.

4.2.1 Baseline Performance

The results from running the three rule-based baselines on the annotated severity detection data set are shown in table 4.1. Among these baselines, there is a very clear trade-off between precision and recall where for example the first baseline achieves a very impressive recall of 0.919 but a precision of only 0.029. This means that less than 3% of the tokens identified by this baseline to be severity descriptors actually are, and such poor performance is of course unacceptable if the classifier is to be useful in signal detection. The third baseline does manage to achieve a significantly

Baseline	Precision	Recall	F1	F1 95% C.I.
1	0.029	0.919	0.056	[0.050, 0.064]
2	0.132	0.735	0.224	[0.197, 0.252]
3	0.187	0.406	0.256	[0.216, 0.299]

Table 4.1: Performance of the three severity detection baselines on the test set, with all vocabularies created from the training set. The precision, recall, F1-score and a 95% confidence interval are reported for each baseline.

Figure 4.2: Performance of severity detection on training set as a function of the cost parameter C, using 5-fold cross validation and averaging the results over all folds and classes. The error bars show ± one standard deviation for the performance between folds.

higher precision of 0.187, but this is still a very poor performance and it comes at the cost of the recall dropping to 0.406 which is less than half of its previous value.

It is of course possible that better baselines than these could be constructed, but it seems very unlikely that there is any simple rule-based approach which will work with acceptable performance. The conclusion is therefore that more sophisticated methods are indeed needed, and this justifies the development of the SVM classifier for severity detection, for which the performance is reported in the following sections.

4.2.2 Tuning of Cost Parameter

Figure 4.2 shows the performance of the severity detection SVM when training and evaluating the severity classification SVM on the training set, using 5-fold cross-validation for different values of the cost parameter C. The F1-score can be seen to peak at $C = 1$, where F1 $= 0.7380$. This value of C was therefore used in all subsequent runs.

		True label	
		Severity	Not severity
Predicted label	Severity	179	39
	Not severity	55	87786

Table 4.2: Confusion matrix for the severity detection classifier, trained on the full training set and evaluated on the test set.

4.2.3 Overall Performance

Table 4.2 shows the confusion matrix obtained when training the severity detection SVM on the full training set and evaluating on the test set, using $C = 1$. The resulting precision was 0.821 and the recall was 0.765, which gives an F1-score of $F1 = 0.792$. The 95% confidence intervals for these estimates, calculated using the probabilistic interpretation developed by Goutte and Gaussier which was outlined in Section 2.3.3, were $[0.766, 0.868]$ for the precision, $[0.708, 0.816]$ for the recall and $[0.748, 0.830]$ for the F1-score. These intervals are clearly quite large, which is unsurprising considering the relatively small size of the data set. Of note in table 4.2 is that there are more false negatives, i.e. tokens that are severity descriptors but not classified as such, than there are false positives, i.e. tokens that are wrongly classified as severity tokens.

Many of the false positives are tokens that do typically describe the severity of reactions, but which are classified as being related to the wrong reaction. An example is the following expression: ”Also had a rash on legs and slight rash on arms”. Here, the word ”slight” is a severity modifier of the second mention of ”rash”, but not the first. The trained SVM identifies this relation correctly, but it also wrongly classifies ”slight” as related to the first occurrence of ”rash”. We thus get one true positive and one false positive from this sentence. In cases like this one, it appears that it is the relation extraction task that is difficult, rather then the NER task of identifying possible severity tokens. Similar issues arise for lists of reactions with associated severities. An example is the following excerpt from a report:

> ... there is a reasonable possibility that the events pancytopenia, neutropenic sepsis (fatal), endocarditis (fatal), septic emboli (fatal) and hyperosmolar coma are related to the administration of ...

The three occurrences of ”fatal” are correctly mapped to their preceding reactions ”neutropenic sepsis”, ”edocartidis” and ”septic emboli”, but they are also each mapped to the reaction directly following them so that we get a relation between the first occurrence of ”fatal” and ”endocarditis”, another between the second occurrence and ”septic emboli” and a third between the third occurrence and ”hyperosmolar coma”. A relation is also incorrectly identified between the first occurrence of ”fatal” and the reaction ”pancytopenia”, resulting in a total of three true positives and four false positives.

In both of the above examples, an underlying problem is that the SVM classifies tokens completely independently of each other. In the first example, this means that when the token ”slight” is classified as related to the first occurrence of ”rash”, the classifier has no information on whether there is a relation between ”slight” and the second occurrence of ”rash” and vice versa. If the classifier did have information on previous outcomes or if it were able to classify several tokens at once, the result might be different and it is reasonable to assume that the performance might improve. One example of a classification algorithm which does have an inherent ability to classify several instances at once is the linear chain conditional random fields (CRF) classifier, which is used in many NLP tasks. In the context of severity detection, a CRF would take as input a sequence of tokens and their corresponding features and output labels (severity/not severity) for all the tokens at once. This allows for the algorithm to take into account the labels of other tokens rather than predicting the label of a single token independently. As mentioned in Section 1.3, Johri et al [22] have used a CRF classifier for severity relation extraction in clinical notes with some success. In the light of their research as well as the nature of the above mentioned false positives produced by the current SVM classifier, it would be interesting to apply a CRF classifier on the severity detection problem.

Another common cause of false positives is incorrect or missing medical events. For example, in the expression ”mild high frequency hearing loss”, the medical event should ideally be ”high frequency hearing loss” but ”hearing loss” would also be acceptable as a more general description. Unfortunately, none of these events are detected by the dictionary look-up algorithm and therefore no severity relation was annotated. However, the word ”high” was detected as a medical event (with PT euphoric mood) and also identified as a reported reaction, since one of the listed reported reactions was ”anger” and these terms share the same HLT ”Emotional and mood disturbances NEC”, where NEC stands for ”not elsewhere classified”. This causes the severity detection algorithm to incorrectly classify the word ”mild” as a severity of the reaction ”high”. Had the event detection algorithm been able to correctly identify ”high frequency hearing loss” as the reaction in question, this false positive might have been avoided. Based on the existence of several examples like this one, it is believed that the performance of the severity detection algorithm might improve significantly if the performance of the event detection could be improved.

As for the false negatives produced by the severity detection algorithm, many of these can be deduced to originate from severity expressions that have not previously been seen. For example, there are four occurrences of the expression ”profuse sweating” in one report in the test set, but the word ”profuse” does not exist at all in the training set. This is most likely the reason that the four occurrences in the test set are all missed. Further examples of severity tokens in the test set that are missing entirely in the training set are: ”remarkable”, ”incapacitating”, ”massively”, ”lethal” and ”huge”. It is therefore reasonable to assume that a larger training set would lead to better performance, and this conclusion is supported further by the results presented in Section 4.2.4.

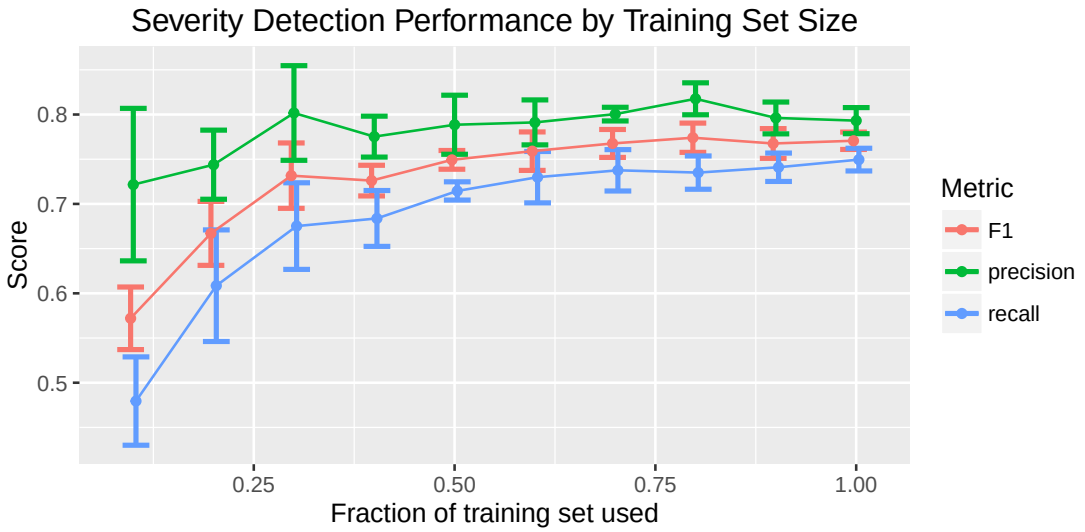


Figure 4.3: Performance of severity detection algorithm as a function of the training set size. Each point represents the average performance obtained by training on five different subsets of the size specified by the x-axis, with the error bars denoting ± one standard deviation for the performance between the five runs.

4.2.4 Dependence on Training Set Size

The performance of the severity detection SVM as a function of the size of the training set is shown in figure 4.3. The plot shows the precision, recall and F1-score achieved using different fractions of the full training set for training, and testing on the full test set. The subsets used for training were generated using the same method as for generating folds for cross-validation, described in Section 3.7.1. This means that the number of tokens in each subset might vary slightly between different repetitions, since the split is performed on a report level. While the error bars in figure 4.2.4 do provide some idea of the stability of the algorithm at different sizes of the training set, they are less accurate for larger training set sizes since when the training set is large, most reports will be selected in all repetitions.

It is clear from figure 4.3 that the performance of the severity detection algorithm does improve as the size of the training set increases, at least up until half of the training set is used for training. After this point, the improvement is less clear. The average performance does continue to improve, but since the error bars are no longer entirely accurate it is difficult to tell if the improvement is significant. The nature of the errors made by the algorithm, which were discussed in Section 4.2.3, does however suggest that using an even larger training set would lead to an improvement in performance.

4.2.5 Feature Ablation

Table 4.3 shows the effect of removing different subsets of the features used for severity detection. The full list of features and their categories can be found in

Features	F1	ΔF1	95% C.I.
All	0.792		[0.748, 0.830]
No token features	0.056	-0.736	[0.046, 0.067]
No reaction features	0.773	-0.019	[0.728, 0.812]
No contextual features	0.533	-0.259	[0.481, 0.582]
No features for surrounding tokens	0.699	-0.093	[0.652, 0.742]

Table 4.3: Effect on the performance of the severity detection algorithm when removing different subsets of features. The F1 score and its 95% confidence interval are shown for each case.

table 3.2. From table 4.3, we note that removing all token features, all contextual features or all features for the tokens surrounding the token under classification does result in a significant decrease in performance, since the confidence intervals for these F1 scores do not overlap with the confidence score achieved when using all features. Removing the reaction features can however not be concluded to significantly reduce the performance, although the obtained F1 score is lower than when all features are used.

That the performance is poor when no token features are used is hardly surprising – more interesting is that removing the reaction features had such a small effect on the overall performance. The fact that they do not does perhaps indicate that other features, such as the distance between the token in question and its associated reaction, are vastly more important and in fact sufficient to predict the status of each token.

4.2.6 Feature Scaling

Table 4.4 shows a comparison of the performance achieved when using BNS feature scaling on just binary and continuous features, on all features and on none of the features. While the differences are small and cannot be concluded to be significant, it appears as if performing BNS scaling on the binary and continuous features does have an positive effect on the classifier performance compared to no scaling at all. While the effect is too small to be very useful, this does at least confirm that BNS scaling does not significantly reduce performance. The reason that it is not more efficient for this classification task is probably that none of the features that were BNS scaled have a very large individual predictive power.

However, when we perform BNS scaling on all features, the performance does appear to decrease, albeit by very little. This is an interesting effect considering that we are then also scaling the one-hot encoded token under classification as well as its preceding and succeeding tokens, which we expect to have a very large predictive power. One possible reason that we do see this decrease in performance is that while the tokens themselves are important to the classifier, it is in conjunction with other features that they achieve significant predictive power. An example is the the word 'severe". While it does obviously often describe the severity of a reaction, it might describe the severity of another reaction than the one currently under classification.

BNS Scaled Features	F1	ΔF1	95% C.I.
Binary & continuous	0.792		[0.748, 0.830]
None	0.776	-0.016	[0.731, 0.815]
All	0.732	-0.060	[0.683, 0.775]

Table 4.4: Comparison of the performance of the severity detection algorithm for three cases: When only binary and continuous valued features are subject to BNS sclaing, when no BNS scaling is performed and when all feature types (listed in table 3.2) are BNS scaled. The F1 score and its 95% confidence interval is displayed for each case.

Consider for instance the sentence ”He experienced a headache, migraine and severe nausea”. Here, the word ”severity” will be part of three candidate token-reaction pairs: {severe, headache}, {severe, migraine} and {severe, nausea}. This means that only in one of three cases is the word ”severe” actually a severity and thus when we see the token ”severe”, we cannot immediately conclude that it is a severity. Instead, we need more information such as the number of tokens between the token and the reaction under classification. The individual predictive power of the token itself is thus small, but its predictive power along with other features is large. This might be true for tokens in general, and could thus explain why BNS scaling the categorical tokens does not improve performance.

4.3 Severity Classification

The performance of the severity classification algorithm alone is reported and analyzed in the following sections. The performance of the rule-based baseline is also presented.

4.3.1 Baseline Performance

Running the baseline for severity classification on the test set resulted in a precision of 0.939 and a recall of 0.957, giving an F1-score of 0.943. This performance is better than expected, and it might even be good enough that a more complex method for severity classification need not be developed. However, it is still possible that using an SVM for severity classification might improve the performance even further.

4.3.2 Tuning of Cost Parameter

Figure 4.4 shows the performance obtained by the severity classification SVM using five-fold cross-validation on the training set, for varying values of the cost parameter C. The best macro F1-score was achieved with $C = 1000$, which gave F1 = 0.921 but it is notable that the variation between folds was relatively large and the performance similar for all values of C in the range $[1, 10^5]$.

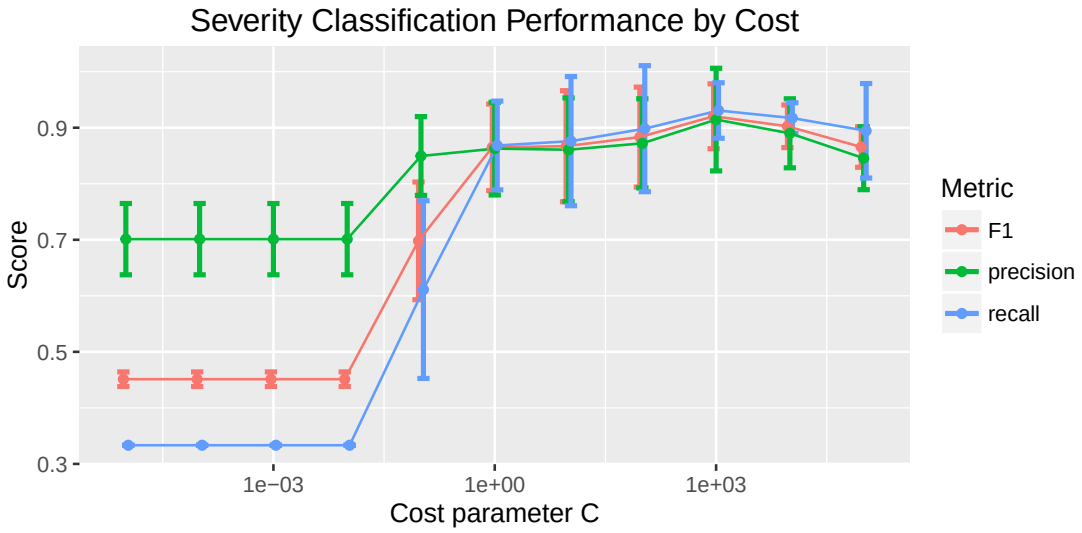


Figure 4.4: Performance of severity classifier on training set as a function of the cost parameter C, using 5-fold cross validation and averaging the results over all folds and classes. The error bars show $\pm$ one standard deviation for the performance between folds.

4.3.3 Overall Performance

Tables 4.5 and 4.6 show the confusion matrix and performance metrics of the severity classification, when trained on the full training set and evaluated on the test set. We can see from the confusion matrix that out of the 121 input bag-of-words, only 3 were misclassified. This gives an accuracy of 0.975 and $F1 = 0.963$, which is considered highly satisfactory.

The three cases that were misclassified are shown in table 4.7. Two of these are bags-of-words that should be graded as ”severe”, but which are predicted by the classifier as ”mild”. Of note is that both of these contain words (”incapacitating” and ”massively”) that were concluded in Section 4.2.3 to not exist in the training set at all. It is therefore no surprise that the severity classification SVM is having difficulties trying to classify these bags-of-words. The correct grade of both ”incapacitating” and ”massively” is ”severe”, but as can be seen from table 4.7 they each occur together with another severity token. Both of these other tokens (”some” and ”slightly”) have the correct grade ”mild”, and thus it appears that the SVM has based its classification decision only on this previously seen token. In general, this seems like an appropriate strategy but it will of course fail in cases like this one, where there is a previously unseen token with a higher-grade severity than the highest known severity.

The cause of the third and final misclassification error is more difficult to analyze, and it is also hard to generalize from since we only have one occurrence. We will therefore abstain from analyzing it in detail here.

		True label		
		Mild	Moderate	Severe
Predicted label	Mild	33	0	2
	Moderate	0	7	0
	Severe	0	1	78

Table 4.5: Confusion matrix for the severity classification algorithm, evaluated on the test set.

	Class			
	Mild	Moderate	Severe	**Macro Average**
Precision	0.943	1.000	0.987	**0.977**
Recall	1.000	0.875	0.975	**0.950**
F1	-	-	-	**0.963**

Table 4.6: Per-class evaluation metrics for severity classification.

4.3.4 Dependence on Training Set Size

The dependence of the classification performance on the training set size is shown in figure 4.5. These runs were performed in the same way as those in Section 4.2.4 and the results are similar. We see a clear improvement as the size of the training set increases, up until roughly 30% of the training set is used. After this, the average performance continues to increase slowly, but it is difficult to tell if the improvement is significant. The analysis of the misclassification errors in Section 4.3.3 indicate that while there are few errors at all, previously unseen tokens are the cause of two out of three misclassification errors. We can therefore conclude that increasing the size of the training set further would probably improve the performance slightly, but not as much as in the severity detection problem.

4.4 Combined Severity Detection and Classification

The results from combining the severity detection and classification algorithms are shown in tables 4.8 and 4.9. We can see that the overall performance is better than for the severity detection problem alone, for all metrics. This indicates that

Severity Tokens	**Predicted Grade**	**True Grade**
some, incapacitating	mild	severe
massively, slightly	mild	severe
stage II to II, stage II to III	severe	moderate

Table 4.7: Misclassified severities in the test set.

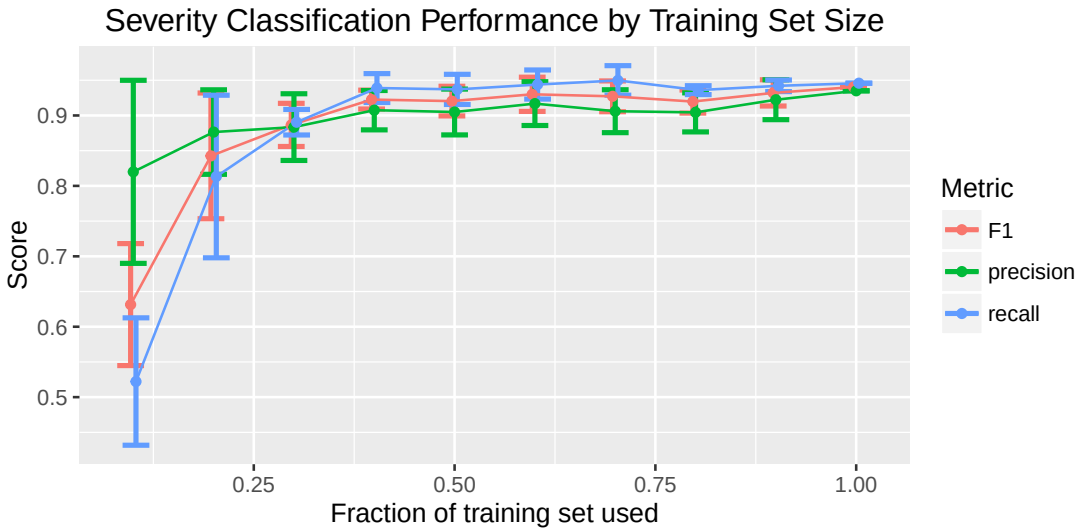


Figure 4.5: Performance of severity classification algorithm as a function of the training set size. Each point represents the average performance obtained by training on five different subsets of the size specified by the x-axis, with the error bars denoting ± one standard deviation for the performance between the five runs.

while the severity detection algorithm does make mistakes by producing both false positives and false negatives, these errors are somewhat compensated for by the classification step, where all severity tokens found to be associated with a particular reaction are merged together into a single bag-of-words which is used to determine the overall grade of the severity. For example, in the expression ”Also had a rash on legs and slight rash on arms”, which was discussed in Section 4.2.3, we noted that the severity detection algorithm incorrectly maps the severity ”slight” to both occurrences of ”rash”. This means the classifier is given the bag-of-words {slight, slight} instead of correct input which would be just {slight}, but this won't affect the output since it will be ”mild” for both cases. In other words, the error in the severity detection step has no effect on the final output of the full pipeline.

Of note is also that the severity classification algorithm only has three possible output values: mild, moderate and severe. It will never output ”none”, and this grade is instead given to all reactions for which no severity tokens were found. This means that all severities missed by the severity detection algorithm will end up with grade ”none” unless the associated reaction has other severity tokens that were picked up in the severity detection step. Conversely, when the severity classification algorithm encounters a bag-of-words containing only previously unseen tokens, it will still be forced to classify the severity as either mild, moderate or severe. In other words, any false positives produced by the severity detection algorithm will always propagate into the severity classification step and lead to false positives there as well, unless there are other severity tokens which are correctly detected and associated with the same reaction.

		True label			
		None	Mild	Moderate	Severe
Predicted label	None	907	7	0	15
	Mild	1	26	0	1
	Moderate	5	0	8	2
	Severe	6	0	0	62

Table 4.8: Confusion matrix for performance of combined severity detection and classification.

	Class				
	None	Mild	Moderate	Severe	**Macro Average**
Precision	0.976	0.929	0.533	0.912	**0.837**
Recall	0.987	0.788	1.000	0.775	**0.887**
F1	-	-	-	-	**0.861**

Table 4.9: Per-class evaluation metrics for combined severity detection and classification.

5

Conclusion

The goal of this thesis was to develop methods for extraction of severity information from the narratives of individual case safety reports describing suspected adverse drug reactions. The two-step approach of first detecting severity tokens using a binary classifier and then determining the overall severity of each reaction based on those tokens was shown to be successful. It is believed that the developed algorithm for severity detection and classification performs well enough to be valuable in the first pass screening step of signal detection at the Uppsala Monitoring Centre and its usefulness for this purpose will be evaluated in the near future.

Using optimal parameter values and the full training set of 1103 reports resulted in an overall F1-sore of $F1 = 0.861$ with a precision of 0.837 and a recall of 0.887 on the test set, consisting of 476 reports. Many of the features used for severity detection proved to contribute significantly to the performance while for severity classification, using only a bag-of-words of severity tokens was enough to achieve near-perfect performance. BNS feature scaling was explored for the severity detection task, but the results do not indicate that this led to any significant performance improvements. The conclusion is therefore that BNS feature scaling is probably best suited for document classification tasks, which it was originally developed for.

Both classifiers outperformed their corresponding rule-based baselines, but while the difference in performance was very large for the severity detection it was not significant for the severity classification. This indicates that using an SVM classifier for severity classification might not be necessary or worthwhile, but on the other hand the cost of training this second classifier is small compared to the first.

The largest sources of misclassification errors were incorrect event detection, sentences containing multiple reactions and severities and the occurrence of tokens in the test set that did not appear in the training set. It is likely that the overall performance could be improved by using a sequence classification algorithm such as a conditional random fields classifier, rather than classifying tokens completely independently as was done here. The use of a non-linear kernel might also be worth exploring, though in other NLP tasks this typically gives a small improvement at most. The performance is also expected to improve with better event detection algorithms. One final possibility for achieving further performance gains is to annotate more data in order to increase the size of the training set. The results in this thesis do however suggest that this might not be worth the effort since the performance gains from using all the available training data compared to using only half were

barely significant for both severity detection and classification.

While there is definitely room for significant improvements to the developed algorithm, perfect performance can never be expected for severity detection. It was clear during the course of this project that even medically trained persons often cannot agree on how to define severity and the reported inter-annotator agreement score of $F1 = 0.899$ provides a rough upper bound on the performance that is achievable using the current annotation guidelines.

To conclude, this thesis has established that it is possible to accurately extract severity information from clinical narratives in individual case safety reports using statistical natural language processing. This opens up for many new possibilities for using natural language processing in the signal detection process at Uppsala Monitoring Centre and other pharmacovigilance centers across the world. Future projects that are closely related to this one might be detecting seriousness information from clinical narratives, developing more sophisticated methods for distinguishing between reported reactions and other medical events in narratives or perhaps expanding the notion of reported reactions to include symptoms detected in the narrative that are related to a reported diagnosis.

Bibliography

[1] Munir Pirmohamed, Sally James, Shaun Meakin, Chris Green, Andrew K Scott, Thomas J Walley, Keith Farrar, B Kevin Park, and Alasdair M Breckenridge. Adverse drug reactions as cause of admission to hospital: prospective analysis of 18 820 patients. *Bmj*, 329(7456):15–19, 2004.

[2] Jonas Lundkvist and Bengt Jönsson. Pharmacoeconomics of adverse drug reactions. *Fundamental & clinical pharmacology*, 18(3):275–280, 2004.

[3] Jacoline C Bouvy, Marie L De Bruin, and Marc A Koopmanschap. Epidemiology of adverse drug reactions in europe: a review of recent observational studies. *Drug safety*, 38(5):437–453, 2015.

[4] Commission of the European Communities. Impact assessment, Accompanying document to the proposal for a regulation of the European Parliament and of the council amending, as regards pharmacovigilance of medicinal products for human use, regulation (EC) No 726/2004 laying down Community procedures for the authorisation and supervision of medicinal products for human and veterinary use and establishing a European Medicines Agency and the proposal for a directive of the European Parliament and of the council amending, as regards pharmacovigilance, Directive 2001/83/EC on the Community code relating to medicinal products for human use, 2008. URL `http://ec.europa.eu/health/files/pharmacos/pharmpack_12_2008/pharmacovigilance-ia-vol1_en.pdf`.

[5] Janet Sultana, Paola Cutroneo, Gianluca Trifirò, et al. Clinical and economic burden of adverse drug reactions. *Journal of Pharmacology and Pharmacotherapeutics*, 4(5):73, 2013.

[6] The International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. ICH harmonized tripartite guideline. Clinical safety data management: definitions and standards for expedited reporting, 1994.

[7] The International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. ICH harmonized tripartite guideline. Maintenance of the ICH guideline on clinical safety data management: data elements for transmission of individual case safety reports E2B (R3), 2005.

[8] Leàn Rolfes, Sarah Wilkes, Florence Hunsel, Eugène Puijenbroek, and Kees Grootheest. Important information regarding reporting of adverse drug reactions: a qualitative study. *International Journal of Pharmacy Practice*, 22(3): 231–233, 2014.

[9] Leàn Rolfes, Florence Hunsel, Sarah Wilkes, Kees Grootheest, and Eugène Puijenbroek. Adverse drug reaction reports of patients and healthcare professionals—differences in reported information. *Pharmacoepidemiology and drug safety*, 24(2):152–158, 2015.

[10] Anthony J Avery, Claire Anderson, CM Bond, Heather Fortnum, Alison Gifford, Philip C Hannaford, L Hazell, Janet Krska, AJ Lee, David J Mclernon, et al. Evaluation of patient reporting of adverse drug reactions to the UK 'yellow card scheme': literature review, descriptive and qualitative analyses, and questionnaire surveys. *Health Technology Assessment*, 15, 2011.

[11] Rave Harpaz, Alison Callahan, Suzanne Tamang, Yen Low, David Odgers, Sam Finlayson, Kenneth Jung, Paea LePendu, and Nigam H Shah. Text mining for adverse drug events: the promise, challenges, and state of the art. *Drug safety*, 37(10):777–790, 2014.

[12] Dina Demner-Fushman and Jimmy Lin. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*, 33(1): 63–103, 2007.

[13] Dina Demner-Fushman, Wendy W Chapman, and Clement J McDonald. What can natural language processing do for clinical decision support? *Journal of biomedical informatics*, 42(5):760–772, 2009.

[14] Dina Demner-Fushman and Jimmy Lin. Answer extraction, semantic clustering, and extractive summarization for clinical question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 841–848. Association for Computational Linguistics, 2006.

[15] Carol Friedman and Noémie Elhadad. Natural language processing in health care and biomedicine. In *Biomedical Informatics*, pages 255–284. Springer, 2014.

[16] Stéphane M Meystre, Guergana K Savova, Karin C Kipper-Schuler, John F Hurdle, et al. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearb Med Inform*, 35:128–44, 2008.

[17] Paea LePendu, Srinivasan V Iyer, Anna Bauer-Mehren, Rave Harpaz, Jonathan M Mortensen, Tanya Podchiyska, Todd A Ferris, and Nigam H Shah. Pharmacovigilance using clinical notes. *Clinical pharmacology & therapeutics*, 93(6):547–555, 2013.

[18] K Haerian, D Varn, S Vaidya, L Ena, HS Chase, and C Friedman. Detection of pharmacovigilance-related adverse events using electronic health records and automated methods. *Clinical Pharmacology & Therapeutics*, 92(2):228–234, 2012.

[19] Carol Friedman. Discovering novel adverse drug events using natural language processing and mining of the electronic health record. In *Artificial Intelligence in Medicine*, pages 1–5. Springer, 2009.

[20] B Polepalli Ramesh, SM Belknap, Z Li, N Frid, DP West, H Yu, et al. Automatically recognizing medication and adverse event information from food and drug administration's adverse event reporting system narratives. *JMIR medical informatics*, 2(1):e10–e10, 2013.

[21] Dmitriy Dligach, Steven Bethard, Lee Becker, Timothy Miller, and Guergana K Savova. Discovering body site and severity modifiers in clinical texts. *Journal of the American Medical Informatics Association*, 21(3):448–454, 2014.

[22] Nishikant Johri, Yoshiki Niwa, and Veera Raghavendra Chikka. Optimizing Apache cTAKES for disease/disorder template filling: Team HITACHI in 2014 ShARe/CLEF eHealth evaluation lab, 2014.

[23] Thierry Hamon, Cyril Grouin, and Pierre Zweigenbaum. Disease and disorder template filling using rule-based and statistical approaches. In *CLEF (Working Notes)*, pages 79–90, 2014.

[24] João Sequeira, Nuno Miranda, Teresa Gonçalves, and Paulo Quaresma. TeamUEvora at clef eHealth 2014 task2a. In *CLEF (Working Notes)*, pages 156–166, 2014.

[25] Tigran Mkrtchyan and Daniel Sonntag. Deep parsing at the CLEF2014 IE task (DFKI-Medical). In *CEUR Workshop Proceedings*, volume 1180, pages 138–146, 2014.

[26] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

[27] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[28] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and Sellamanickam Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *Proceedings of the 25th international conference on Machine learning*, pages 408–415. ACM, 2008.

[29] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[30] George Forman. BNS feature scaling: an improved representation over tf-idf for svm text classification. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 263–270. ACM, 2008.

[31] George Forman, Martin Scholz, and Shyamsundar Rajaram. Feature shaping for linear SVM classifiers. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 299–308. ACM, 2009.

[32] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European Conference on Information Retrieval*, pages 345–359. Springer, 2005.

[33] Nakatani Shuyo. Language detection library for java, 2010. URL `http://code.google.com/p/language-detection/`.

[34] Michal M Danilak. Langdetect, 2014. URL `https://pypi.python.org/pypi/langdetect?`

[35] International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. MedDRA, 2016. URL `http://www.ich.org/products/meddra.html`.

[36] International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. Introductory guide MedDRA version 19.0, 2016. URL `http://www.meddra.org/sites/default/files/guidance/file/intguide_19_0_english.pdf`.

[37] Thibault Helleputte. *LiblineaR: Linear Predictive Models Based on the LIBLINEAR C/C++ Library*, 2015. R package version 1.94-2.

[38] Noémie Elhadad, Guergana Savova, Wendy Chapman, Glenn Zaramba, David Harris, Amy Vogel, Danielle Mowery, and Sumithra Velupillai. ShARe guidelines for the annotation of modifiers for disorders in clinical notes, Dec 2013.

[39] S Belknap, E Freund, N Frid, E Granillo, H Keating, Z Li, R Prasad, B Ramesh, and H Wang, Vand Yu. The annotation guideline manual: Extracting adverse drug event information from clinical narratives in electrical medical records, Oct 2015. URL `http://www.bio-nlp.org/papers/ADE%20Annotation%20Guidelines%202015Oct29v3.pdf`.

[40] National Cancer Institute. Common terminology criteria for adverse events (CTCAE) version 4.0, 2009.

[41] New York Heart Association. Classes of heart failure, 2015. URL `http://www.heart.org/HEARTORG/Conditions/HeartFailure/AboutHeartFailure/Classes-of-Heart-Failure_UCM_306328_Article.jsp#.V7GMwZh97RY`.

A

Annotation Guidelines

These guidelines, which are intended to be used for annotating the severity of adverse drug events in individual case safety reports from the WHO international database of suspected adverse drug reactions VigiBase, are partially based on two other annotation guidelines. The first is the "ShARe Guidelines for the Annotation of Modifiers for Disorders in Clinical Notes" which was developed for and applied to the publicly available ShARe corpus of clinical notes [38]. The second is "The annotation guideline manual: Extracting adverse drug event information from clinical narratives in electrical medical records", developed by the University of Massachusetts Medical School's Biomedical Informatics Natural Language Processing Group, specifically for use on adverse events [39].

A.1 Annotation Types

A.1.1 Medical Event

These annotation guidelines assume that medical events have been pre-annotated and therefore no annotation guidelines for these are provided. Annotated medical events are not marked in this document.

Of note is that the pre-annotation of medical events may be imperfect in such a way that medical events are missed, annotated with the wrong span or annotated for expressions which are not in fact medical events. In cases where this is obvious to the annotator, medical events thst are clearly wrong should be ignored whereas annotations that are only partially correct should be considered correct if the meaning is similar to what it should be. An example is the event "back pain". If only the word "pain" has been pre-annotated as a medical event, it should be considered correct since "back pain" is in fact a type of "pain" and thus the meanings are similar. A counter example is the expression "high fever" annotated with two medical events, "high" and "fever". Fever should then be considered a correct medical event, but not "high" since alone, this refers to a state of elevation and not a fever.

For the case of medical events where the event itself is an elevated or decreased lab value, and only the lab value or test itself has been annotated rather than the full event, it should still be considered correct. An example is "leukocyte count decreased" – if the event found is "leukocyte count" this should be considered cor-

rect.

A.1.2 Adverse Drug Reaction

Adverse drug reactions, which are a subset of the medical events found in an individual case safety report, are also assumed to be pre-annotated. Annotated adverse drug reactions (hereafter: reactions) are underlined in the examples given in this document.

Note that the type of pre-annotation errors described above can propagate to adverse reaction annotations. The above guidelines for determining whether to accept a pre-annotated medical event or not should also be applied to adverse drug reactions.

A.1.3 Severity

The ShARe guidelines define severity as "the relative intensity of a process or the relative intensity or amount of a quality or attribute". This definition will also be used here. The ADE Guidelines state that:

> Severity is often indicated by modifying words such as mild, minimal, markedly, severe, endstage, small, extremely, substantial. Severity terms can also be phrases such as borderline to slightly high, moderate-to-severe.
>
> **Example:** He is *rather* diffusely tender to palpitation.
>
> Here "rather" can be a severity meaning "to some degree".

The severity can be a single word or an entire expression and it is always the longest relevant span that should be annotated. The ADE guidelines state:

> In general avoid redundancy in severity annotation. For example: "some *mild*", annotate only "mild". Sometimes a second word modifies the first and it is necessary to capture both. For example "*very slight*" shows less severity than simply "slight".

Severity must always be associated with a reaction, i.e. the severity annotation must describe the severity of a specific adverse drug reaction. In some cases, the pre-annotation of reactions may not be entirely accurate, and a severity should then only be annotated for reactions that appear to be mostly correct.

Examples: lower back pain,

In this document, annotated severity is marked using italics.

Each severity annotation has an attribute which we call "Grade". The grade of a severity annotation has three possible values: mild, moderate or severe. This attribute is used for classification of severity annotations according to the guidelines in section A.9.

A.1.4 Severity Relation

This annotation type is used to mark a connection between an annotation of the type ”Reaction” and one of the type ”Severity”. A severity relation itself is therefore never marked in the examples below. Each reaction or severity modifier may have multiple severity relations.

Example: The following day the patient developed *severe* arm pain and swelling at the site of injection.

Here, two severity relations should be annotated, one between ”severe” and ”arm pain” and another between ”severe” and ”swelling”. In general, when a severity modifier is directly followed by more than two medical events and it is not clear from the context whether the severity refers to all or only some of them, a relation should only be annotated to the closest event and only if it is a reaction. If there are only two medical events, such as in the example above, the severity should be annotated to all that are reactions.

Example: The patient complained of *severe* nausea, headache, dizziness and tiredness.

In this example only one severity relation, between ”severe” and ”nausea”, should be annotated.

A.2 Explicitness

The ShARe guidelines require severity to be explicit:

> The severity should be explicit. If inference is required to determine the severity of a disorder, then no Severity modifier should be annotated.

Important: Our interpretation of this is that the severity must have been evaluated by the patient, health care professional or reporter and not require the reader to draw any own conclusions. This applies in particular to laboratory and test results that have not been interpreted. Another example is words that are used to describe the location or spread of a reaction, such as ”local”, ”widespread” or ”all over body”. These expressions should not be annotated as severity, since they have not been explicitly interpreted. A local itching could for instance be severe, while a widespread rash might be mild.

A.3 Overlap

According to the ShARe guidelines, the severity can overlap with a medical event.

> The severity can overlap with or be part of the disorder mention. The span of the severity can be outside, overlapping with, or inside the span of the mention disorder.

> **Example:** The patient presented with *severe* pre-eclempsia.

The disorder ”severe pre-eclempsia” has the modifier Severity associated with the span ”severe”.

This practice will be directly adapted. However, the ADE Guidelines point out that this is not always the case.

> Note that some terms you might consider as severity are actually part of the disease name and are not annotated. For example, large is part of the name ”large B cell lymphoma”. /.../ Medical use of the following terms can be used as a disease name or temporal indicator. These words are not severity: acute, chronic, acute-on-chronic, flare. Annotate the example below as follows:

> **Example:** *significant* flares of fibromyalgia

This practice will also be followed. Some examples from ICSRs are:

> **Examples:** shooting pain, projectile vomiting

In both of these cases, no severity should be annotated since ”shooting” and ”projectile” are descriptions of the of the medical events themselves, rather than their severity.

A.4 Laboratory and Test Results

The severity of a laboratory or test result is only to be annotated if the severity of the result has been evaluated and/or interpreted by the patient, health care professional or reporter, see section A.2.

> **Example:** Medical history included a *serious* event of leukocyte count decreased.

In this example, ”leukocyte count decreased” is the medical event and it has the severity modifier ”serious”. The word ”decreased” is not a severity modifier. Also note that according to the example in section A.1.1, if the detected reaction here was simply ”leukocyte count”, it would still have the severity modifier ”serious” and not ”decreased”.

> **Example:** Medical history included high blood pressure.

In this example there is no severity modifier. That is because ”high” is part of the event itself (just like ”decreased” in the previous example) and the severity of the high blood pressure has not been evaluated.

The following two examples are intended to clarify the difference between a value that has been interpreted by the reporter and one that has not.

Example: During the night, the child developed a fever of 40° C.

This first example contains only a value that has not been interpreted or evaluated by the reporter. One might guess that a fever of During the night, the child developed a fever of 40° C is probably quite severe, but that conclusion requires medical inference from the reader and should therefore not be annotated.

Example: During the night, the child developed a *high* fever.

In this second example however, the use of the word ”high” implies that the reporter has evaluated the severity of the fever.

A.5 Numerical Values

The ADE guidelines states that numerical values should only be annotated under certain conditions:

> We only annotate numbers or references to quantity as severity when there is a frame of reference or scale such as %, mm, X out of Y etc. Annotate where you can understand the meaning, otherwise it is diagnosing.
>
> **Example:** fever greater than 100.5
>
> Fever is annotated as an entity but the temperature is not annotated as severity. /.../ Annotate pain in his back twice to create a relation to each – (1) severe and (2) ”10 on a scale of 1-10”. 8 is not annotated since in isolation there is no scale.
>
> **Example:** Remarkable for the aforementioned *severe* pain in his back which she states is, without pain medicine, *10 on a scale of 1-10.* At the moment, it is down to 8.

The ADE guidelines give further examples, but these do not seem to be consistent with the above statements and are therefore not included here. We will generally, as described in section A.2, require an interpretation of any given number to be present for it to be marked as severity. For example, the statement ”innumerable 1mm non blanching papules” does not include any severity information since there is no interpretation of how severe ”innumerable” or ”1mm” is in this case. In other words, the presence of a scale (such as mm) is not a sufficient condition for annotating severity. Furthermore, a frame of reference is not a necessary condition. When numbers refer to a well-established scale they will be annotated even when the frame of reference is not given such as in the examples below.

Examples: *grade 3* loss of consciousness, neuroblastoma *stage 4*

The first example is not consistent with the ADE guidelines, but the latter is since they state that ”disease stages are specific indicators of severity and should be annotated as severity”.

While a description of size such as ”10 x 10 cm rash” does not fulfill the above requirements for being a severity, sizes described as for example ”small” or ”large” should be annotated as severity since their usage implies that the author has assessed the size to be smaller or larger than what would normally be expected.

A.6 Temporality and Disease Dynamics

The ADE guidelines state that disease dynamics should not be annotated.

> We do not annotate words that indicate severity dynamics such as worsening or increasing, decreasing.

This convention will be followed. The ADE guidelines do however also state that expressions such as ”decreased” and ”increased” should be annotated as severity, but we will not annotate such relative expressions as severity. They can however sometimes be part of the annotation of a medical event (see example in section A.4).

Furthermore, we will not annotate any descriptions of the temporality of an event as severity.

Examples: sudden BP elevation, persistent fatigue

A.7 Qualitative and Quantitative Modifiers

Certain words can be used both for indicating quality and quantity. The ADE Guidelines state that:

> ”Some” and ”somewhat” are terms that can be used for descriptors of severity or as quasi severity quantifiers. ”Some pain” means approximately ”mild pain”. Some is vague but it is a description of severity, i.e. not very severe, and can be useful. Don’t annotate when used as a quantifier such as ”some blisters”.

These conventions will be followed.

A.8 Special Terms

The terms ”serious” and ”significant” (and variations of them) are commonly used in ICSRs to indicate that the case fulfills certain conditions, such as for example those for a serious report. In this context, the words ”serious” and ”significant” do not describe the severity of an event. Those same words can however also be used in different contexts to describe the severity of an event, in this latter case they should be annotated as severity.

Example: The patient experienced circulatory collapse which was considered *serious* due to hospitalization. The patient also experienced dizziness which was considered non serious.

In this example the expressions ”serious” and ”non serious” clearly refer to the classification of the medical event as serious/non-serious, based on the ICH guidelines which state for example that an event which leads to hospitalization should be considered serious. Therefore, no severity should be annotated. Other common examples of expressions used to determine whether the case is serious are ”medically significant”, ”signficant” and ”medically important”.

Example: Suddenly patient started suffering from *serious* vision disturbances.

In this example however, the use of the word serious probably has nothing to do with the ICH guidelines or determining the overall seriousness of the case. It should therefore be annotated as severity.

The term ”life-threatening” if often used to justify why a case has been reported as serious. This term does however still describe the severity of the event, irregardless of the context it is used in. The expression ”life-threatening” should therefore always be annotated as long it can be clearly linked to a reported reaction. An example is given below.

Example: The investigator assessed the pulmonary embolism as *life threatening* and related to Bevacizumab and unrelated to Capecitabine.

A.9 Grades of Severity

As previously mentioned, the grade attribute of a severity annotation has three possible values: mild, moderate and severe. This section will describe how the grade of a severity annotation is determined.

First of all, if the author has clearly meant to emphasize the intensity of a medical event, the grade should be set to ”severe”. Examples of this are expressions such as ”severe”, ”strong”, ”marked”, ”very” and ”major”. Conversely, if the author is clearly aiming to diminish the severity by using a modifier, the grade should be ”mild”. Examples include expressions such as ”mild”, ”some”, ”slight” and ”minor”.

A common way to express the severity of an adverse event is to use the Common Terminology Criteria for Adverse Events (CTCAE), developed by the United States National Cancer Institute (NCI) [40]. This terminology is used to map the severity of av event to a numeric scale of 1-5, where 1 is defined as ”mild”, 2 is ”moderate”, 3 is ”severe”, 4 is ”life-threatening” and 5 is ”fatal”. The mild/moderate/severe grades used here thus correspond to CTCAE grades 1-3, while CTCAE grades 4 and 5 are mapped to ”severe”.

Example: The following day the patient developed neutropenic colitis (*CTCAE grade 4*), reduced general condition (*CTCAE grade 3*) and vomiting (*CTCAE grade 1*).

The CTCAE was previously called the NCI Common Toxicity Criteria. Therefore, severities such as ”NCI-CTC grade 2” or ”CTC grade 1” are mapped as if they were CTCAE grades (i.e. as ”moderate” and ”mild” respectively, in this case).

Several other severity scales exist for more specific diseases or disease types. For example, the Canadian Cardiovascular Society has a four step grading system for angina pectoris, where grade 1 and 2 both correspond to ”mild” while grade 3 is ”moderate” and grade 4 is ”severe”. Whenever a medical event is reported with an unspecified grade, and there is a specific scale for that type of medical event available, that scale should be used to define the severity of the event. However, if no such specific scale exists, it should be assumed that the grade refers to a CTCAE grade.

Just like for grades, there are many systems for defining classes of different diseases. An example is the New York Heart Association (NYHA) classification system for heart failure, where a class I or II heart failure is defined as ”mild”, class III is ”moderate” and class IV is ”severe” [41]. As for grades, when such scales are available they should be used.

The severity of a medical event can sometimes also be described by a stage. For cancer, all stages are considered to be severe except for stage 0 and 1 which are considered mild. When there are established scales that map stages to severities, they should be used. Otherwise, stage I or 1 should be graded as ”mild” and everything else as ”moderate”, except for the maximum stage (if it is known) which should be graded as ”severe”. Expressions such as ”end stage” and ”advanced” should be interpreted as referring to the maximum stage and should thus be graded as ”severe”.

Established scales that serve to characterize a condition or disease, but which do not explicitly map values to severities should not be annotated as severity in the first place. An example of this is the Score of Toxic Epidermal Necrosis (SCORTEN) scale which for each score provides an estimate of the mortality rate, but does not map scores to severity.

Many different scales exist for describing the intensity of pain. When there is an established system for mapping between values on a pain scale and severity, that should be used. Otherwise, the pain should be considered ”mild” if its value is within the first 1/3 of the scale, ”moderate” if the value is in the middle third of

the scale and ”severe” if the value is larger than 2/3 of the maximum value on the scale.