DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2018

Deep Neural Networks for Inverse De-Identification of Medical Case Narratives in Reports of Suspected Adverse Drug Reactions

EVA-LISA MELDAU

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Deep Neural Networks for Inverse De-Identification of Medical Case Narratives in Reports of Suspected Adverse Drug Reactions

EVA-LISA MELDAU

Master in Computer Science
Date: February 24, 2018
Supervisor at KTH: Joel Brynielsson
Supervisor at the Uppsala Monitoring Centre: Niklas Norén
Examiner: Olle Bälter
Swedish title: Djupa neuronnät för omvänd avidentifiering av medicinska fallbeskrivningar i biverkningsrapporter
School of Electrical Engineering and Computer Science

Abstract

Medical research requires detailed and accurate information on individual patients. This is especially so in the context of pharmacovigilance which amongst others seeks to identify previously unknown adverse drug reactions. Here, the clinical stories are often the starting point for assessing whether there is a causal relationship between the drug and the suspected adverse reaction. Reliable automatic de-identification of medical case narratives could allow to share this patient data without compromising the patient's privacy. Current research on de-identification focused on solving the task of labelling the tokens in a narrative with the class of sensitive information they belong to.

In this Master's thesis project, we explore an inverse approach to the task of de-identification. This means that de-identification of medical case narratives is instead understood as identifying tokens which do not need to be removed from the text in order to ensure patient confidentiality. Our results show that this approach can lead to a more reliable method in terms of higher recall. We achieve a recall of sensitive information of 99.1% while the precision is kept above 51% for the 2014-i2b2 benchmark data set. The model was also fine-tuned on case narratives from reports of suspected adverse drug reactions, where a recall of sensitive information of more than 99% was achieved. Although the precision was only at a level of 55%, which is lower than in comparable systems, an expert could still identify information which would be useful for causality assessment in pharmacovigilance in most of the case narratives which were de-identified with our method. In more than 50% of the case narratives no information useful for causality assessment was missing at all.

Sammanfattning

Tillgång till detaljerade kliniska data är en förutsättning för att bedriva medicinsk forskning och i förlängningen hjälpa patienter. Säker avidentifiering av medicinska fallbeskrivningar kan göra det möjligt att dela sådan information utan att äventyra patienters skydd av personliga data. Tidigare forskning inom området har sökt angripa problemet genom att märka ord i en text med vilken typ av känslig information de förmedlar. I detta examensarbete utforskar vi möjligheten att angripa problemet på omvänt vis genom att identifiera de ord som inte behöver avlägsnas för att säkerställa skydd av känslig patientinformation. Våra resultat visar att detta kan avidentifiera en större andel av den känsliga informationen: $99,1\%$ av all känslig information avidentifieras med vår metod, samtidigt som 51% av alla uteslutna ord verkligen förmedlar känslig information, vilket undersökts för 2014-i2b2 jämförelse datamängden. Algoritmen anpassades även till fallbeskrivningar från biverkningsrapporter, och i detta fall avidentifierades $99,1\%$ av all känslig information medan 55% av alla uteslutna ord förmedlar känslig information. Även om denna senare andel är lägre än för jämförbara system så kunde en expert hitta information som är användbar för kausalitetsvärdering i flertalet av de avidentifierade rapporterna; i mer än hälften av de avidentifierade fallbeskrivningarna saknades ingen information med värde för kausalitetsvärdering.

Contents

Chapter 1

Introduction

Medical research requires detailed clinical data. Insight into patient history can be used by researchers or practitioners to investigate treatments, drugs and diseases. Thus, sharing patient data can help researchers to better analyse these. In adverse drug reaction research, where the causality between a suspected adverse reaction and a drug has to be assessed, the case narratives from this patient data can be especially crucial [27]. Case narratives can best help to discover yet unknown adverse drug reactions and to study new explanations for adverse drug reactions [66]. Some information useful for causality assessment is only reported in the case narratives such as the descriptions of the course of the events or adverse drug reaction severity [27].

The Uppsala Monitoring Centre is an independent foundation that is responsible for the scientific and technical operations of the World Health Organisation (WHO) Programme for International Drug Monitoring. Patient safety and the safe and effective use of medicines are the Uppsala Monitoring Centre's goal. Their database VigiBase [36] is one example of a collection of data valuable for medical research. It contains over 15 million individual case safety reports in which suspected adverse drug reactions are reported. These reports contain both information in structured format and in form of free-text case narratives. The reports are provided by national centres, which are recognised by their country's government as national organisations or other entities to participate in the WHO Programme for International Drug Monitoring.

In order for hospitals, practitioners, or national centres for drug monitoring to be able to share electronic records and case narratives in particular, they will have to ensure patient confidentiality. This is the case if information that can be used to identify the patient has been removed from the case narratives. The information that has to be removed is often called protected health information (PHI). Many countries forbid sharing reports unless the PHI is removed. In the United States, PHI is defined in the Health Insurance Portability and Accountability Act (HIPAA) [61]. In this act, 18 types of PHI are listed including names, different identification numbers, contact information and dates [1]. This also applies for the WHO's international drug monitoring programme: in order for national centres, which collect the data on a national level, to be allowed to share the case narratives within the WHO programme, they have to first de-identify the case narratives.

Manual de-identification is not practical for several reasons. First of all, only a restricted number of people have the right to access the data as it contains confidential information. Secondly, when presented with a large amount of data, manual de-identification will be costly both in terms of time and financial costs. Finally, humans are prone to make errors. Ferrández et al. [17], for example, report an agreement between annotators of 83% when considering exact agreement and 91% for inexact agreement during their annotation process. Therefore, the development of automatic de-identification methods is important.

Currently, the standard is to identify the sensitive terms (PHI) in order to remove them. We evaluate the use of an inverse approach in which starting from all words being "removed", "safe" words (non-PHI) are identified and added to the de-identified text. This has previously only been explored to some extent [7, 17]. Our presented method will use deep learning, in which representations in form of neural networks with multiple layers are learnt at the same time as the classifier, as deep learning methods have been showing good results for de-identification in the past [12, 32]. We propose a combination of dictionary look-ups and deep neural networks to identify the safe words, where "safe" means that the word is safe to be left in the de-identified case narrative.

1.1 Purpose and Problem Statement

The goal of this thesis project was to develop an automatic de-identification method that is reliable, in the sense that it has a high recall of PHI, while still preserving useful information for causality assessment. Achieving a high recall means that the method identifies a high percentage of the protected health information which is present in the case narratives. This would make it possible to apply automatic de-identification in practice to de-identify free-text case narratives in order to share them between organisations.

Specifically, in this thesis, we investigate the following question:

- can an automated de-identification method using an inverse approach achieve a recall of protected health information within case narratives of close to 100% while preserving useful information?

Chapter 2

Background

This chapter gives an introduction to de-identification and the field of pharmacovigilance. In Section 2.1, pharmacovigilance is described. This includes an introduction to causality assessment within adverse drug reaction monitoring, as well as the WHO programme for drug monitoring and the global database VigiBase containing reports of suspected adverse drug reactions. Section 2.2 gives a definition of protected health information. Section 2.3 presents the related work on de-identification methods which aim to remove protected health information from free text.

2.1 Pharmacovigilance

The World Health Organization defined *pharmacovigilance* as

> "[t]he science and activities relating to the detection, assessment, understanding and prevention of adverse effects or any other drug-related problem" [46].

Within the WHO International Drug Monitoring Programme, adverse drug reactions are monitored by national centres in the participating countries. In this section, we first summarise the procedure of causality assessment. Furthermore, the WHO Programme is described, as well as the international database in which reports of suspected adverse drug reactions are stored. This database is called VigiBase.

2.1.1 Causality Assessment

In adverse drug reaction research, it is important to assess the likelihood that a drug could be the cause for the suspected adverse reaction. This is referred to as *causality assessment*. Different criteria for causality assessment and *signal detection*, with the aim of detecting potential problems with drugs, have been described in the literature. Some of these criteria are presented in the following.

In [22], Sir Austin Bradford Hill describes which aspects should be considered when a person tries to decide whether an association between two variables is most likely due to causation. His criteria are commonly used in causality assessment for adverse drug reactions. He gives the following nine criteria which we adapted to pharmacovigilance based on the presentation in the video lecture on "The logic of causality" by Savage [55]:

1. Strength of association. How strong is the association? For example, does the adverse drug reaction appear 200 times as often as for people who do not take the drug or only 10 times as often?

2. Consistency. Has the suspected adverse drug reaction to a certain drug been reported by different people, in different places, and/or at different times?

3. Specificity. Is there a specificity of the association between the drug and the adverse reaction in question? For example, are similar associations with the adverse reaction with other drugs used for the same indication reported, in which case it might not be the drug that is the cause? Do we see associations between the same drug with a wide range of adverse reactions that are unlikely to be manifestations of the same underlying medical condition?

4. Temporality. Is the time line possible or plausible? Was the drug taken before the adverse drug reaction appeared? And was there plausible time for the adverse drug reaction to develop, e.g., a tumour does not develop in one day?

5. Dose-response relationship. Does a higher dose lead to a higher effect? Or is this not the case?

6. Plausibility. Is the association between the adverse drug reaction and the drug plausible from a scientific perspective? This can be

helpful but Bradford Hill points out that we cannot demand it as the association could be new to science.

7. Coherence. Does the association not conflict with any generally known facts?

8. Experiment and experimental evidence. Can experiments reproduce the observation? Can a de- and a rechallenge support the causality hypothesis?

9. Analogy. Are similar reactions known to be caused by similar drugs?

Of these criteria, information on (3.), (4.), (5.) and even (8.) could based on our own assessment be (exclusively) given in the case narrative of a report of a suspected adverse reaction.

Edwards et al. [14] try to define the quality of reports for early signal detection. The authors assign different quality labels to the reports based on the information they contain. A report that does not contain all the essential information (identification of the source of the case, identification of the case, description of the reaction, name of the drug, treatment dates, reaction dates) is labelled "unassessable". A report that contains all this essential information is called "feasible". A "substantial" report includes in addition to the essential information all of the following: sex, age, all drugs with doses and dates, indication for treatment/underlying diagnosis, and outcome of the adverse drug reaction. A report that additionally reports a positive rechallenge (reintroduction of the drug which again leads to the same adverse drug reaction) is labelled "presumptive". An "index case" is given if the report is either "presumptive" or if it is not "presumptive" but contains all the information needed for a "substantial" report and in addition does not contain any "confounding variables" (that the underlying disease or another drug that the patient is taking could have caused the adverse drug reaction).

The authors then suggest that a signal could be produced if there are three index cases or cases equivalent to index cases. For this, two substantial cases or four feasible cases are considered equivalent to one index case. The sensitivity or specificity of signal detection can in this method be adjusted by changing the amount of index cases required. It will, however, always be a trade-off between sensitivity and specificity.

From this we can see which information is important for causality assessment, could be given in the case narratives, and is sometimes missing from the structured information. According to Edwards et al.'s definition and our understanding, these are: description of reaction, all drugs with doses and dates which the patient was taking, indication for treatment/underlying diagnosis, and information about a possible rechallenge.

In general, there is always an uncertainty whether the suspected drug actually has caused the adverse reaction or not. According to Meyboom et al. [37] a method for structured causality assessment cannot reduce the uncertainty but it can categorise the uncertainty. The authors further point out that causality assessment only has limited scientific value as the methods have not been (and possibly cannot be) validated.

In a questionnaire for causality assessment the following questions could be asked as presented by Meyboom et al. [37]:

1. When was the drug given? Prior to the event?
2. Is the site of the adverse drug reaction the same as the site of application of the drug?
3. Is the time between onset of the drug and adverse drug reaction reasonable?
4. Did the adverse drug reaction occur immediately after the drug administration?
5. Was there a rechallenge with positive result (adverse drug reaction reoccurred)?
6. Was there a dechallenge with positive result (adverse drug reaction disappeared)?
7. Were other drugs stopped simultaneously?
8. Has the patient previously had the same adverse reaction to the drug?
9. Is the adverse reaction known with this drug?

The answers to some of the questions could be (exclusively) given in the free-text case narrative. The narrative could contain information on the site of the adverse drug reaction, the timeline of the events (time-to-onset, where the onset is the time of introduction of the drug, etc.), (detailed) information on possible de- and rechallenges, information on other drugs and doses, and the patients' medical history.

2.1.2 World Health Organization (WHO) International Drug Monitoring Programme

After the adverse drug reaction disaster caused by the use of thalidomide during pregnancy in the 1960s, several countries started national drug monitoring programmes and subsequently joined forces to start the WHO International Drug Monitoring Programme in 1968 [36]. The joined programme has the aim of detecting emerging risks to patients earlier than is possible based on national data only. Within this programme, the WHO Collaborating Centre in Uppsala, Sweden, the Uppsala Monitoring Centre, is responsible for managing the technical, operational, and scientific aspects. The programme's aim is to detect drug-related problems at an early point by collecting so called individual case safety reports in an international database, called *VigiBase* [36].

The reports of suspected adverse reactions in VigiBase include reports from regulatory and voluntary sources [36]. This way of reporting suspected adverse drug reactions for example by health professionals to national centres is sometimes referred to as "spontaneous reporting". Edwards and Aronson [13] give an overview of the different methods for surveillance of adverse drug reactions. They state that spontaneous reporting is simple but suffers from under-reporting and reporting bias. Furthermore, Pal et al. [47] suggest that spontaneous reporting systems are a reasonable choice of method as they can allow for a large information provision at a low cost.

The reports transmitted by the national centres to VigiBase can be used to identify potential problems with drugs. This process of identifying potential problems is called signal detection. A *signal* is

> "[i]nformation that arises from one or multiple sources (including observations and experiments), which suggest a new potentially causal association, or a new aspect of a known association, between an intervention and an event or set of related events, either adverse or beneficial, that is

> judged to be of sufficient likelihood to justify verificatory action" [67].

A signal does therefore not mean that there is evidence of a causal relationship between the drug and the adverse reaction. Instead, it is a hypothesis and is used to warn at an early point. In order to form a signal, typically, more than one report is needed. In pharmacovigilance, a series of reports is needed in order to draw better conclusions about the causality [37]. Together this series of reports can form a signal.

The data collected in VigiBase is monitored and analysed by researchers at the Uppsala Monitoring Centre as well as users of the database such as national pharmacovigilance centres. Based on such analyses, the Uppsala Monitoring Centre issues signals which indicate potential problems with drugs. The signals are sent to the national pharmacovigilance centres and to the pharmaceutical companies that hold the marketing authorisation for the concerned drugs [36].

2.1.3 VigiBase

VigiBase is the "unique WHO global database of individual case safety reports" [64]. SQL is used to manage the data and VigiBase can be accessed through Internet applications as well as client-server applications and open database connectivity [36]. VigiBase is maintained and developed by the Uppsala Monitoring Centre on behalf of the WHO. The database is used to detect signals. Reports are sent to the database in a standardised format, such as the ICH E2B format [36]. This format includes several free-text fields for case narratives. Older formats, which are still used by some countries, do not include such fields. The Uppsala Monitoring Centre has also developed a web-based case management system called VigiFlow [36].

VigiBase contains over 15 million reports [24]. Reports are collected nationally by the national centres who share them in VigiBase [64]. Thereby, the database is regularly updated and growing. Having a global collection of data can increase the probability of early signal detection. The first reports were collected in 1968. Today, 127 countries are members of the WHO Programme for International Drug Monitoring, representing over 90% of the world's population [64]. The reports collected in VigiBase can come from health professionals, patients and pharmaceutical companies [64]. Access to the database is granted to the member countries of the WHO programme and others with a legitimate interest in accessing this data for pharmacovigilance purposes and with relevant knowledge to interpret the data [64].

A report mostly contains fields that are linked to pre-defined vocabulary but it can further contain free-text fields for patient disease background or a description of the adverse drug reaction [36]. The so called "Narrative Include Clinical" is one of the free text fields of the E2B format and it contains the case summary. The free-text case narratives can be crucial for the causality assessment [27]. Karimi et al. [27] list the severity and site of the adverse drug reaction, interventions, the specifications of the underlying disease, and patient ethnicity as information which could only be found in the VigiBase case narratives. The authors further point out that information on dechallenge, the withdrawal of the drug, or rechallenge, the re-introduction of the drug treatment after withdrawal is sometimes exclusively given in the case narratives. In an evaluation of samples of VigiBase reports which they described as having normal length, 22% of the investigated reports had information crucial to causality assessment included in the case narratives. 26% had provided information in the case narratives that "considerably affected the understanding of the clinical course of the cases". When longer case narratives were considered, 32% of the case narratives included crucial information. Therefore, the authors claim that the case narratives have to be included in interpretation of the suspected adverse drug reaction cases in order to avoid misinterpretations [27].

Incoming reports to VigiBase are checked according to certain quality criteria (see also Section 2.1.1). The entries are linked to the WHO-Drug reference source of Medical Product Information and to medical terminology defined in the WHO Adverse Reaction Terminology or the Medical Dictionary for Regulatory Activities (MedDRA) [36] (for more information on MedDRA, see Section 4.2.2). The case reports stored in VigiBase should in principle be de-identified but contain a reference to the original case report located at a national centre [64]. Nevertheless, there exist reports in VigiBase, which are not completely de-identified.

The advantages of spontaneous reporting are also the advantages of VigiBase: the national centres continuously transmit new data. This way of adverse drug reaction monitoring has low costs and can cover a broad population. Nevertheless, the system suffers from the typical problems of spontaneous reporting, which is missing data and underreporting [36]. In general, the quality of spontaneous case reports can differ as the reports may be incomplete [37].

2.2 Protected Health Information

In the United States' Health Insurance Portability and Accountability Act, sensitive information which can be used to identify a patient's identity is called *Protected Health Information (PHI)* [1]. The term PHI can, however, also be used in a wider sense and include information that is not listed by HIPAA but which could be used to identify a person's identity. Examples of such PHI categories that are not included in HIPAA are for example age under 90 or dates in form of common holidays, which could possibly lead to a re-identification of the person. These are amongst others included in the definition of PHI of the 2014-i2b2 de-identification challenge [60] (see Section 4.1.1 for more information on the data set).

2.2.1 Health Insurance Portability and Accountability Act

The list of PHI, as defined in the Health Insurance Portability and Accountability Act in 45 C.F.R. §164.514(b) [1], is presented in Table 2.1. A summary of the HIPAA Privacy Rule can be found on the website of the U.S. Department of Health and Human Services [61].

2.2.2 European Union

In the European Union (EU), a new regulation for data protection will begin to apply in the end of May 2018. With regards to personal information and handling of anonymous data, the following points from the regulation [50] are interesting:

1. Personal data is defined as "any information relating to an identified or identifiable natural person".

2. An identifiable natural person is defined as a person "who can be identified, directly or indirectly". It is stated that this could

1. Names
2. Geographic subdivisions smaller than a State: street address, city, county, zip code, etc.
3. Dates: all dates, except year, e.g., birth date, admission date, discharge date, etc.; ages over 89 and dates, including year, indicative of such age
4. Telephone numbers
5. Fax numbers
6. Electronic mail addresses
7. Social security numbers
8. Medical record numbers
9. Health plan beneficiary numbers
10. Account numbers
11. Certificate/license numbers
12. Vehicle identifiers and serial numbers, including license plate numbers
13. Device identifiers and serial numbers
14. Web Universal Resource Locators (URLs)
15. Internet Protocol (IP) address numbers
16. Biometric identifiers, including finger and voice prints
17. Full face photographic images and any comparable images
18. Any other unique identifying number, characteristic, or code

Figure 2.1: PHI according to HIPAA.

be done "by reference to an identifier such as a name, an identification number, location data, an online identifier". But the regulation also includes "factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity" of the person as possibilities to identify him or her.

3. The principles for data protection presented in this regulation do "not apply to anonymous information". This is in the following explained to be "information which does not relate to an identified or identifiable natural person". It is also stated that the regulation does not apply to "personal data rendered anonymous in such a manner that the data subject is no longer identifiable". Thus, both data which is intrinsically anonymous or which has been made anonymous can be handled without having to follow the other rules of the regularisation. Here, anonymous means that the person who is the data subject is no longer identifiable.

4. The regulation explains how to “determine whether a natural person is identifiable”. For this, one should consider “all the means” which could be used by “the controller or by another person” in an attempt to “identify the natural person directly or indirectly”. The text also points out that this should be means which are “reasonably likely” to be used. For this, “all objective factors, such as the costs of and the amount of time required for identification” should be considered. In this context, “the available technology at the time of the processing and technological developments” should be taken into account.

From (2), it can be concluded that a de-identification method will need to remove: names, identification number, location data, and online identifiers. The EU regulation does not define which scale of location information has to be removed, for example whether states and countries are considered to be location data which has to be removed. Online identifiers could be interpreted as being email addresses, IP addresses, and websites. It is not clear which information has to be removed with regards to the other factors mentioned in the regulation: physical, physiological, genetic, mental, economic, cultural, and social identity.

2.2.3 Comparison Between Countries

We can see that the difference between the HIPAA and the EU regulation is that HIPAA gives clear guidelines, while the EU regulation does not. The EU regulation instead keeps the definition of PHI more general and moves the responsibility of defining which information is enough to identify a person to the data handler. This way it is guaranteed that a person’s identity is protected in the case where removing the HIPAA PHI is not enough to hide the identity. Imagine for example the case in which only one person with the contained physical, economic, and social characteristics exists in a certain country. It would then not be enough to remove all HIPAA PHI in order to prevent the re-identification of the person. This is due to the fact that HIPAA does not include these four attributes of a person (physical, economic, and social characteristics, and country names). The EU regulation allows the data handler to decide which definition of PHI is reasonable in a certain context.

This should be seen in relation to the concept of “k-anonymity”. In the context of privacy-preserving data mining, “k-anonymity” is a method in which techniques such as generalisation and suppression are used to make the data representation more general such that at least k records share the same attributes [3]. This way, one can prevent the identification of an individual by finding the only person with this combination of attributes since there are always at least k individuals.

When evaluating an automatic de-identification method a definition of PHI according to HIPAA is however useful in order to allow for performance comparisons. Nevertheless, a flexible method which can be adjusted to remove more, less, or different attributes from the text could be desirable for example for the use in EU countries.

Many other countries have passed laws related to the protection of personal data. Here, we present the legislation in three of them: South Africa, Japan, and India.

In the South African Act on Protection of Personal Information from 2013 [59], “to de-identify” is defined as “to delete any information that [...] (a) identifies the data subject; (b) can be used or manipulated by a reasonably foreseeable method to identify the data subject; or (c) can be linked by a reasonably foreseeable method to other information that identifies the data subject”. If the processed data which includes personal information “has been de-identified to the extent that it cannot be re-identConsistency.ified again”, then the Act on Protection of Personal Information does not apply. In the act, “to re-identify” is defined as resurrecting any information that has been de-identified (according to the previous definition of “to de-identify”). Therefore, the new EU legislation is similar to this one. Both do not state what information would be enough to identify a data subject directly, or after manipulation of the information or linkage of the information to other information. The South African legislation does as the EU regulation give the responsibility of defining what is information which can be used to identify a person to the data handler. The act provides a comprehensive definition of the meaning of “personal information”. In the context of medical data, we can note that “information relating to the [. . .] medical history of the person” is considered personal information and that this act does therefore apply.

In Japan, a new act on the protection of personal information is put into full effect on 30th May 2017 [49]. In the act, “personal information” is defined as “information relating to a living individual” which contains “name, date of birth, or other descriptions etc. [. . .] whereby a specific individual can be identified” or which contains “an individual identification code”. If these kind of ID numbers or descriptions have been deleted (or masked) then the personal information is referred to as “anonymously processed information”. After removing the information, it should not be possible “to identify a specific individual”. The act further states that a business operator handling anonymously processed information can share the information. They have however to inform the public about which “categories of information relating to an individual [are] contained in the anonymously processed information” when the anonymously processed information is produced and before it is shared with a third party.

In India, there exist the Information Technology Act from 2000 [63] and additional rules such as the “Information Technology (Reasonable security practices and procedures and sensitive personal data or information) Rules” from 2011 [25]. In the latter, “personal information” is defined as “any information that relates to a natural person, which, either directly or indirectly [. . .] is capable of identifying such [a] person”. With indirectly they mean “in combination with other information available or likely to be available with a body corporate”. These rules further define the concept of “sensitive personal data or information” which could for example be relating to “physical, physiological and mental health condition” or “medical records and history” but also to a lot of other information such as financial information or sexual orientation. The data may be transferred if this is “necessary for the performance of the lawful contract between the body corporate or any person on its behalf and provider of information or where such person has consented to data transfer”. In the law text, there does not seem to be any definition of how to handle “de-identified” data.

2.3 Related Work: De-Identification Systems

Medical reports which have been de-identified can, as previously mentioned, be shared and used for research. Since manual de-identification is costly, there has been much research on the development of automatic de-identification methods. These research projects mostly focus on removing the PHI defined in HIPAA, which is probably due to the amount of medical data that is digitally stored in the U.S. but also because HIPAA explicitly defines which information has to be removed in order for the report to be de-identified. Thus, HIPAA is giving the possibility to use a quantitative measure for the quality of de-identification methods in terms of recall of PHI.

Uzuner et al. [65] defined the goal of de-identification for the i2b2 de-identification challenge as identifying and removing the PHI contained in medical records while the integrity of the data should be preserved as much as possible. The authors further describe the challenges for this task. First of all, the difficulties posed by the ambiguities between PHI and non-PHI such as the name "Parkinson" (PHI) and the "Parkinson's disease" (non-PHI). Secondly, Uzuner et al. point out that it can be difficult to identify PHI which are misspelled or foreign words.

Since the introduction of deep learning methods, in which input features are learned together with the rest of the parameters of a predictive model, a categorisation of de-identification methods into methods using hand-engineered features and those learning the features is a reasonable choice. The methods using hand-engineered features can then further be classified into rule-based methods, also referred to as methods using pattern-matching, and machine-learning-based methods [38, 12].

2.3.1 Systems Using Hand-Engineered Features

Extensive reviews of the different de-identification methods published can be found in [38, 65, 60], containing both rule-based and machine-learning based methods using hand-engineered features.

An example of a purely pattern-matching-based method is presented by Neamatullah et al. [45]. They use a combination of dictionary look-ups, regular expressions and heuristics to identify PHI from free-text medical records. They use four types of dictionaries: a list of known patient names and hospital staff (known PHI), a list with amongst others generic first names, last names, hospital names and locations (potential PHI), a list of PHI indicators such as titles, name indicators ("son", "mother", . . .), location indicators (e.g., "Hospital"), age indicators (e.g., "age"), a list of non-PHI formed of common English words and words from the UMLS nomenclature. Names that are also found in this last list are labelled "ambiguous". The presented algorithm first divides sentences into words. It then identifies PHI by look-ups in dictionaries and regular expressions. PHI including numeric patterns such as street addresses are recognised using regular expressions identifying the numeric pattern while checking for contextual keywords such as “road”. Non-numeric PHI are recognised by lexically matching each word in the text with the dictionaries, i.e., labelling words with the dictionaries they belong to such as known PHI or PHI indicator (e.g., “Mr.” or “Hospital”). Regular expressions are here used to check for combinations of potential PHI and context keywords (e.g., “Mr. Miller” with the context keyword “Mr.” followed by a a known last name which is in the potential PHI dictionary). If an “ambiguous” PHI is found, heuristics are used to decide whether to classify the token as PHI or not. The authors check for example for name patterns such as first name followed by last name. When the algorithm has classified every token, the PHI tokens are removed and replaced by their PHI category, e.g., name or location. In the case of dates as PHI, these are replaced by a date with a patient specific offset. This way time intervals are preserved. The authors developed and evaluated their method using the nursing notes from the MIMIC-II data set [53], which they annotated and enriched by adding more instances of PHI. The method received a recall of close to 100% on names except for initials which it was unable to recognise. Dates were recalled in 94.6% of the cases, locations in 97.3%, phone numbers in 100%, and age over 89 in 75%. The precision was below 90% for all categories except locations.

Yang et al. [68] present a method which combines rule-based techniques with machine learning techniques in form of conditional random fields (see Section 3.6). The authors generate different linguistic features like part-of-speech tags of the token (i.e., is the token a noun, verb, adjective, etc.), the word forms, the position in the sentence, features based on regular expression, as well as task-specific features. For each PHI category, they proceed separately. In order to determine whether a token is of this PHI category, one or several of the following approaches are used: conditional random fields, rules and patterns, or keywords. In a post-processing phase, the method further tries to correct possible errors. This method was the best submission at the 2014-i2b2 de-identification challenge [60], where it achieved a precision of 97.6%, a recall of 93.9%, and an F1 score of 95.7% on the HIPAA-PHI categories during an entitiy-based evaluation.

Sahlström [54] compared the performance of three different de-identification methods on a data set of VigiBase reports and on the 2006 i2b2 de-identification challenge data set. One of the methods he implemented uses regular expressions to de-identify dates and ages and Sahlström evaluated its performance on his VigiBase data set. The method achieved an F1 score of 85.7% on ages and 93.5% on dates. The second method that Sahlström developed was a support vector machine using several different features such as the part-of-speech tag of the token, the result from dictionary look-ups, and the token position in the sentence and the document. This method achieved a recall of 91.5%, a precision of 91.8%, and an F1 score of 91.7% on the 2006-i2b2 data set. His third method was a conditional random field. For this evaluated on the 2006-i2b2 data set, the recall was 87.7%, the precision 95%, and the F1 score 91.2%.

2.3.2 Feature Learning Neural Network Systems

Deep learning has moved into the focus of machine learning following a seminal publication by Krizhevsky et al. [29], which led to advances in the field of Computer Vision. Deep learning is a machine learning method in which the representation, the features of the data, are learned and not hand-engineered [19]. The features are learned at the same time as the desired function itself while using several layers to learn a hierarchy of features. In the field of natural language processing, recurrent neural networks (RNNs) have led to advances in the field of language modelling [40] but have further proven to be applicable to other tasks like chunking, part-of-speech tagging, and named entity recognition [11].

Dernoncourt et al. [12] have successfully applied this kind of neural network to the task of de-identification. The method uses a learned token embedding to represent tokens as vectors, where the token embedding is pre-trained and fine-tuned during the training of the model. Pre-training means that one trains the model, in this case a token embedding, on a different data set and then uses the learned weights or parts of them as initialisation in another, sometimes more complex model. The token-embedding is further enhanced by a vector representation of the characters of a token, which allows for example to handle out-of-vocabulary tokens and misspellings. This character-based token embedding is learned using a type of neural networks called bidirectional Long Short-Term Memory (LSTM) RNNs (see Section 3.4.4 for an introduction to LSTM). In addition to this character-enhanced token embedding layer, the model has two more layers, a label prediction layer and a label sequence optimisation layer. The label prediction layer uses bidirectional LSTMs and a feed-forward layer. Its purpose is to output a sequence of probability vectors to output the probabilities for certain tokens to belong to certain PHI categories. The label sequence optimisation layer is then used to find the optimal sequence of labels for the input (token) sequence based on these probabilities. This layer can take into consideration that two names, first and last name, often follow each other and optimise the label of sequences with this in mind. The model is evaluated using the 2014-i2b2 data set [60] as well as the MIMIC-III data set [26] and compared to a conditional random field method, which it outperforms. This paper brings up ideas for named entity recognition which were presented by Lample et al. [30]. In named entity recognition, the goal is to find and label named entities in natural text such as persons or locations. Lample et al. present a method using a bidirectional LSTM combined with a sequential conditional random field layer, which can be used to tag a sequence. The idea is that the labelling of the sequence is performed jointly while looking at the whole sequence and not only at separate tokens. This is argued by the fact that in named entity recognition tasks the different tokens are dependent, e.g., a last name often follows a first name. Their results have shown that using deep learning with recurrent neural networks can lead to better performance than was previously achieved by the use of other machine learning methods like conditional random fields. The authors propose that this is because their artificial neural network can better use the context information and that it is better capable of dealing with the variations in natural language than a conditional random field.

Dernoncourt et al.'s feature learning approach was improved by Lee et al. [32]. Their method does not only use learned features but also includes hand-engineered features. The authors exploit the fact that one often is given lists of patient and doctor names of a certain hospital when performing the de-identification. Therefore, they use features that they derive from the hospitals' databases. These features are patient's first name, patient's last name, doctor's first name, and doctor's last name. In addition to these dictionary look-ups, they also add other features. These are morphological features (e.g., "first letter capitalised?"), semantic features (e.g., hypernyms), temporal features (e.g., holidays), gazetteers (e.g., honorifics for doctors or first names), and regular expressions (e.g., for email, or age). These features were taken from work by Filannino and Nenadic [18] and from work by Yang and Garibaldi [68], or from online resources. These are binary features which are passed through a feed-forward neural network before they are added to the character-enhanced token embedding from the non-feature-enhanced model [12]. This method is tested on the same subset of the MIMIC-III data set as the previous one. For most categories, except phone and state, the enhanced model outperforms the plain RNN model in terms of precision, recall and F1 score. The additional features not coming from a hospital database did however not generally improve the results and did in some cases even reduce the recall of PHI categories.

Shweta et al. [58] also use RNNs to perform de-identification of clinical records, but they chose two different types of RNNs, the Elman Architecture and the Jordan Architecture. Also with this kind of RNNs, it was possible to outperform conditional random fields. These RNNs use so called context windows to capture short-term dependencies. To form a context window, word embedding vectors are concatenated using the word embeddings from previous and subsequent time steps. This method does however not perform as well as Dernoncourt et al.'s method [12] on the 2014-i2b2 data set.

Li et al. [34] also use bidirectional LSTMs as in [12] but their method also extracts the skeleton of the medical record and uses this as input to a separate RNN. The final classification uses the output of the LSTM for both directions as well as the RNN that processed the skeleton. This method outperforms the method presented in [12] in terms of precision on the 2014-i2b2 data set but it performs worse in terms of recall.

2.3.3 Inverse Approach Systems

The previously presented methods all approach the task of de-identification as a named entity recognition task, that is identifying and removing sensitive words and numbers. Our literature search only identified one method [7] which approaches the inverted task where all words are "removed" and "safe" words are allowed back into the text. This method [7] only retains clinical terms and stopwords while all other words are removed and it is further limited to pathology records.

After parsing for sentences and tokenizing them, the algorithm looks up medical terms in the Unified Medical Language System (UMLS) nomenclature [9] and replaces the found terms by their code in the UMLS. Furthermore, the method takes in stopwords. All other words are replaced by three asterisks. In order to look up which words match the nomenclature, the method takes a phrase and produces all possible ordered concatenations of words of length 1 to n, where n is the length of the phrase. The implementation is nomenclature-independent but Berman, the author, suggests the use of Medical Subject Headings (MESH)[1] and Systematized Nomenclature of Medicine (SNOMED)[2]. The system was implemented in Perl. The author describes the different problems that this algorithm has: it will definitely take out too much information from the text and it might include names that are also common words used in the nomenclature. The system will not take in PHI terms that were misspelled as misspelled terms will just be left out by the system.

Furthermore, [17] includes a component implementing the inverse approach. This system first classifies words as PHI and removes them based on rules and conditional random field predictions. Afterwards, it lets some of these words back in according to a support vector machine classifier[3]. For pre-processing, the authors use sentence segmentation, tokenisation, part-of-speech tagging, phrase chunking, word normalisation, and lexical variant generation. The rule-based method they used are dictionary look-ups in different dictionaries, as well as decisions based on the part-of-speech tag of a token. In order to filter the false-positives, data points which were falsely classified as being positive, i.e., PHI, multi-class support vector machines for certain subsets of the PHI categories are used. The authors, Ferrándes et al., tested the system on their own data set as well as the 2006-i2b2 data set. On the latter they achieved a high recall of more than 90% for all PHI categories except for Street/City. The overall recall was 96.5%.

[1] MESH: `https://www.nlm.nih.gov/mesh/`

[2] SNOMED: `https://www.nlm.nih.gov/healthit/snomedct/`

[3] A support vector machine is a classifier which tries to find a separating hyperplane with maximal margin for the data in a higher dimensional space.

Chapter 3

Theory

This chapter contains the description of the relevant theory underlying our de-identification method. In Section 3.1, a short introduction to artificial neural networks in general is given. In Section 3.2, an introduction to the field of deep learning and neural network training follows. Section 3.3 introduces deep feed-forward neural networks and Section 3.4 introduces the concept of recurrent neural networks, a special type of neural networks commonly used to process sequential data. In Section 3.5, we describe a way of representing natural language input for input to neural networks referred to as word vectors. In Section 3.6, we describe linear-chain conditional random fields, a method for sequence labelling. Section 3.7 contains a description of prevalent evaluation measures.

In machine learning, an algorithm tries to learn a function using a training data set. The algorithm is trained on data in order to approximate the function. This learned approximation can then be used to predict some aspect of a new data point. It is desired that the learning algorithm can *generalise* from the training data to unseen test data [20, Ch. 3]. If the learning algorithm specialises too much to the idiosyncrasy of the training data, one says that the algorithm is *overfitting* to the training data. As the goal of machine learning is to generalise, measures against overfitting are commonly used. In the following, we will focus on a special kind of learning algorithm called artificial neural networks.

3.1 Artificial Neural Networks

Artificial neural networks are a type of machine learning technique that is inspired by the nervous systems [51, Ch. 1]. The simplest artificial neural network consists of one neuron, often called unit. A neuron takes several real values x_i as its inputs. These inputs are multiplied with weights w_i, one weight per input [51, Ch. 1]. The weighted sum of the inputs is called the unit's activation. The unit output is computed as the result of a non-linear activation function which takes the weighted sum as its input [8, Ch. 5]. The non-linear activation function of such an artificial neuron is chosen as a differentiable function which resembles the all-or-nothing activation of a real neuron. Commonly used is the function $tanh$ or the sigmoid function [8, Ch. 5]. Such an artificial neural network neuron is shown in Figure 3.1.

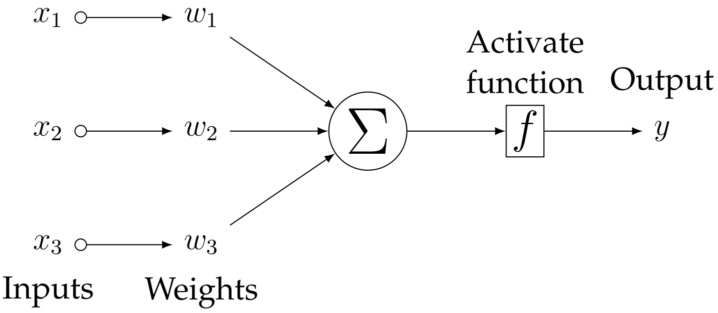

Figure 3.1: A neuron with inputs x_i, weights w_i. It computes the weighted sum of the inputs and outputs y, the result of the activation function f given this sum.

One can use several neurons to form artificial neural networks, where multiple neurons receive the same inputs in parallel or where the output of a neuron is passed as input to neurons in a next layer [8, Ch. 5] (see Figure 3.2).

This kind of neural network is called *feed-forward* neural network because the input is only passed forward through the network due to its lack of loops in its connectivity. For a detailed introduction to artificial neural networks, see [42, 16, 51].

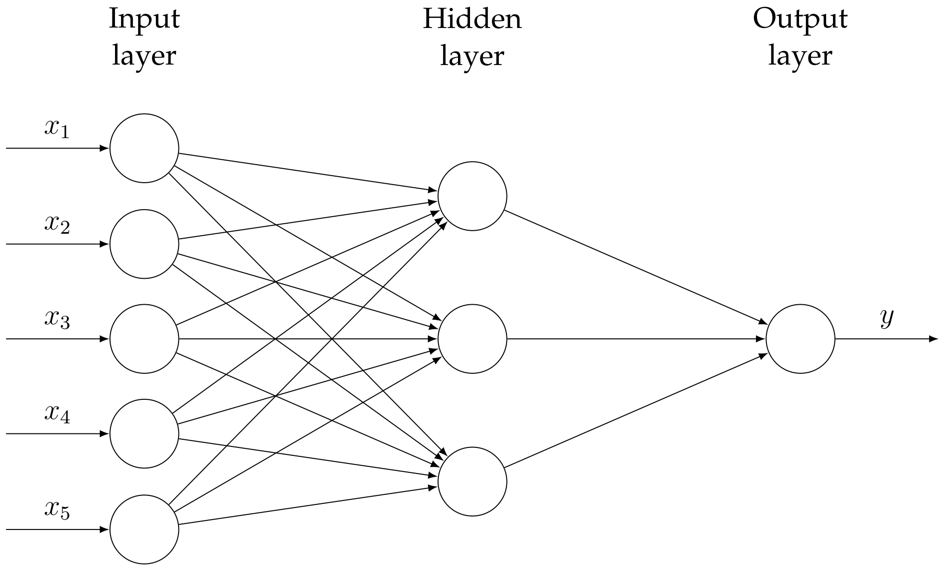

Figure 3.2: An artificial neural network with three layers of neurons, including a so called hidden layer.

3.2 Deep Learning

In *deep learning* methods, features are learned instead of hand-crafted and the networks commonly consist of multiple layers used to learn this data representation [19, Ch. 1]. A review on deep learning was published by LeCun, Bengio, and Hinton [31]. For a detailed introduction to the topic, see [19].

3.2.1 Feature Learning

Conventionally, in order to apply a machine learning technique to a problem, one had to engineer features by hand. It was not common to input raw data into the learning system [31]. This process of feature engineering had to be done carefully and it required domain expertise [31], while for many problems the hard part in solving them is the feature extraction itself. In the case, when given good features, a simple learning system can easily learn to perform the classification or other tasks [19, Ch. 1]. Therefore, "representation learning", the task of automatically discovering the representation of the raw data needed for a classification method [31], became an interesting task for researchers.

A deep learning method learns representations as a hierarchy of features at the same time as it learns to solve the actual problem. A representation learned by a deep learning system consists of multiple levels of representation [31], in which complex concepts are built out of simpler concepts [19, Ch. 1]. This hierarchy of features can be thought of as a deep graph with multiple layers, in which the first layer contains simple concepts while the last layer contains more complex concepts [19, Ch. 1]. Because of this deep hierarchy of features, these methods are referred to as deep learning methods.

3.2.2 Pre-Training and Fine-Tuning

Deep neural networks can be *pre-trained* on a similar task and then *fine-tuned* for the actual task on new data. Pre-training can be useful when data is limited [31]. For visual tasks, Razavian et al. even showed that features learned by a so called convolutional neural network[1] can be used for a variety of tasks different from the task originally used for training [57], even without fine-tuning. Azizpour et al. [4] further researched how to best transfer the learned vision features. They found that when certain "learning factors", like regularisation techniques and network size, are well chosen, it is possible to reach state-of-the-art results on different computer vision tasks by fine-tuning using features pre-trained on the ImageNet dataset [52].

The role of unsupervised pre-training, where the correct answer is not known, has been investigated during several research projects [15, 6]. Erhan et al. [15] propose the hypothesis that the improvements are due to better regularisation, while Bengio et al. [6] suggest that the advantage is due to better optimisation. In both studies (greedy layer-wise) unsupervised pre-training was used. Erhan et al. [15] suggest that this leads to better generalisation because it introduces "a useful prior to the supervised fine-tuning".

Fine-tuning even following supervised pre-training is commonly used, for example by Dernoncourt et al. [12]. They pre-train their token-embedding on a different data set to then fine-tune it during

[1] A convolutional neural network is a type of artificial neural network with a local connectivity pattern, shared weights, which is commonly used in computer vision.

the actual training for the task of de-identification.

3.3 Deep Feed Forward Neural Networks

The simplest example of a deep learning model is the deep feed-forward neural network [19, Ch. 6]. Thus, this is just a classical feed-forward neural network using several layers stacked on each other as previously described (see Section 3.1).

3.3.1 Training

Most of the advances in deep learning have been made using *supervised learning* methods [31]. In supervised learning, the system is presented example inputs together with their correct answer. It can then learn from experience. For this, it will need an *objective (error) function*, which measures how correct (incorrect) the answer was. The system then adjusts its parameters in order to perform better in the future.

The weights are adjusted following the negative gradient of the error [20, Ch. 3]. A step is taken in the direction of the negative gradient of the error function with respect to the weights. This is called *gradient descent* [20, Ch. 3]:

$$\triangledown \boldsymbol{w}(t) = -\eta \frac{\partial O}{\partial \boldsymbol{w}(t)},$$

where t indicates the time step and η is chosen between 0 and 1. η is called the *learning rate* as it defines how much should be learned by defining how big the weight correction should be [20, Ch. 3]. The gradient is then added to the previous weights so that $\boldsymbol{w}(t+1) = \boldsymbol{w}(t) + \triangledown \boldsymbol{w}(t)$. Here, the gradient is computed using the full training set [8, Ch. 5].

In practice, *stochastic gradient descent* is commonly used, in which a randomly picked sample is passed through the network and the derivative of the objective function with respect to the weights is computed based on this sample [8, Ch. 5]. An update is made after seeing each sample, but it can also be made after seeing a batch of samples. Passing all samples through the network in stochastic gradient descent is called a training epoch. This often works well since one can get a good estimate of the gradient based on only a few samples and thus save computational time during gradient computations [44, Ch. 8]. Furthermore, stochastic gradient descent can more easily escape from local minima due to the added "noise" of the stochastic estimate of the gradient [44, Ch. 8].

A method to enable the gradient descent to escape from local minima is to add a *momentum* term to the weight update $\bigtriangledown \boldsymbol{w}(t)$ [20, Ch. 3]:

$$\bigtriangledown \boldsymbol{w}(t) = \alpha \bigtriangledown \boldsymbol{w}(t-1) - \eta \frac{\partial O}{\partial \boldsymbol{w}(t)},$$

where $0 \leq \alpha \leq 1$ defines how much of the old direction for the gradient descent to keep.

In order to train a neural network with multiple layers, a method called *backpropagation* has to be used in order to propagate the error from the output layer backwards to the first layers. Backpropagation applies the chain rule in order to compute derivatives with respect to weights of hidden layers. The following explanation is based on [8, Ch. 5], which we refer to for a detailed explanation and an example of the backpropagation algorithm. In the following explanation we will also use Bishop's terminology and his definition of an example neural network.

Consider a simple feed-forward neural network with two hidden layers (its definition was taken from [8, Ch. 5]):

$$a_j^{(1)} = \sum_i w_{ji}^{(1)} x_i \tag{3.1}$$

$$z_j = h(a_j^{(1)}) \tag{3.2}$$

$$a_k^{(2)} = \sum_j w_{kj}^{(2)} z_j \tag{3.3}$$

$$o_k = h(a_k^{(2)}), \tag{3.4}$$

where

- x_i: the i-th input to the network,
- $a_j^{(1)}$: the weighted sum of inputs at the j-th hidden unit of the first hidden layer,
- $w_{ji}^{(1)}$: the weight of the connection between input i and unit j of the first layer,

- z_j: the activation at the j-th unit of the first hidden layer using activation function $h(\cdot)$,
- $a_k^{(2)}$: the weighted sum of inputs at the k-th unit of the second hidden layer,
- $w_{kj}^{(2)}$: the weight of the connection between unit j of the first layer and unit k of the second layer,
- o_k: the k-th output of the network, the activation at the k-th hidden unit of the second hidden layer using activation function $h(\cdot)$.

If we want to compute the derivative with respect to the second layer weights, $\frac{\partial O}{\partial w_{kj}^{(2)}}$ (where O is the objective function), we can use the chain rule to find:

$$\frac{\partial O}{\partial w_{kj}^{(2)}} = \frac{\partial O}{\partial o_k} \frac{\partial o_k}{\partial w_{kj}^{(2)}}.$$

This is due to the fact that the objective O, which uses the outputs o_k, only depends on $w_{kj}^{(2)}$ through o_k. $\frac{\partial O}{\partial o_k}$ can easily be computed using the chosen definition for the objective function.

To compute $\frac{\partial o_k}{\partial w_{kj}^{(2)}}$, we note Equation (3.3) and (3.4) where o_k depends on $a_k^{(2)}$ which in turns directly depends on $w_{kj}^{(2)}$. Using the chain rule another time, we get:

$$\frac{\partial o_k}{\partial w_{kj}^{(2)}} = h'(a_k^{(2)}) \frac{\partial a_k^{(2)}}{\partial w_{kj}^{(2)}} = h'(a_k^{(2)}) z_j.$$

Thus, we find:

$$\frac{\partial O}{\partial w_{kj}^{(2)}} = \frac{\partial O}{\partial o_k} h'(a_k^{(2)}) z_j = \delta_k z_j, \tag{3.5}$$

where we defined:

$$\delta_k = \frac{\partial O}{\partial o_k} h'(a_k^{(2)}). \tag{3.6}$$

For the derivative of the objective function with respect to the weights of the first layer, we use:

$$\frac{\partial O}{\partial w_{ji}^{(1)}} = \sum_k \frac{\partial O}{\partial o_k} \frac{\partial o_k}{\partial w_{ji}^{(1)}},$$

where we use the sum because all o_k depend on $w_{ji}^{(1)}$. For this, we need to compute $\frac{\partial o_k}{\partial w_{ji}^{(1)}}$:

$$\begin{aligned}\frac{\partial o_k}{\partial w_{ji}^{(1)}} &= \frac{\partial o_k}{\partial a_k^{(2)}} \frac{\partial a_k^{(2)}}{\partial w_{ji}^{(1)}} \\ &= h'(a_k^{(2)}) \frac{\partial a_k^{(2)}}{\partial w_{ji}^{(1)}} \\ &= h'(a_k^{(2)}) \frac{\partial a_k^{(2)}}{\partial z_j} \frac{\partial z_j}{\partial w_{ji}^{(1)}} \\ &= h'(a_k^{(2)}) w_{kj}^{(2)} h'(a_j^{(1)}) x_i,\end{aligned}$$

where we used (3.1)–(3.4), the chain rule, and the fact that only z_l with $l = j$ depends on $w_{ji}^{(1)}$.

Thus, we find:

$$\begin{aligned}\frac{\partial O}{\partial w_{ji}^{(1)}} &= \sum_k \frac{\partial O}{\partial o_k} h'(a_k^{(2)}) w_{kj}^{(2)} h'(a_j^{(1)}) x_i \\ &= \sum_k \delta_k w_{kj}^{(2)} h'(a_j^{(1)}) x_i. \qquad (3.7)\end{aligned}$$

We define

$$\delta_j = \sum_k \delta_k w_{kj}^{(2)} h'(a_j^{(1)}). \qquad (3.8)$$

Thus, from (3.5), (3.6), (3.7), (3.8) we get:

$$\begin{aligned}\frac{\partial O}{\partial w_{kj}^{(2)}} &= \delta_k z_j, \\ \frac{\partial O}{\partial w_{ji}^{(1)}} &= \delta_j x_i,\end{aligned}$$

where

$$\begin{aligned}\delta_k &= \frac{\partial O}{\partial o_k} h'(a_k^{(2)}), \\ \delta_j &= \sum_k \delta_k w_{kj}^{(2)} h'(a_j^{(1)}).\end{aligned}$$

In addition to this general learning algorithm, different strategies are used in order to improve the learning process in practice.

One way to prevent overfitting in neural networks is to use *early stopping* [20, Ch. 3]. In early stopping, one uses a validation set, a part of the training set which was set aside, in order to monitor the error during training. Usually, it can be observed that the training error goes down, while the validation error will decrease at first until a certain point after which it starts to increase again [20, Ch. 3]. This is the moment in which the learning system starts to overfit and in which training should be stopped.

Another regularisation method is called *weight decay*. Weight decay means that the weights are kept small by adding an additional regularisation term to the objective function which penalises large weights [8, Ch. 5]. In weight decay, this is the $L2$-norm of the weights, which requires that the total sum of squares of the weights is kept below a certain threshold.

A method typically used in convolutional neural networks is called dropout. Dropout is described by Goodfellow et al. [19, Ch. 7] as an ensemble method similar to bagging where an ensemble of large neural networks is trained. When using dropout, each connection between two units in the network is not used during training with probability p. This way, the network is forced to make correct predictions without relying on certain inputs to units. The method can even be used for deep recurrent neural networks but only on the non-recurrent connections [69].

When *initialising the weights* of a neural network which is to be trained with gradient descent, small random values should be chosen [20, Ch. 3]. The network can as previously explained also be initialised with pre-trained weights.

3.4 Recurrent Neural Networks

Recurrent neural networks (RNNs) are a type of neural networks that allow the network graph to contain cycles [20, Ch. 3]. This means for example that a unit's value can influence the unit's value at the next time step [19, Ch. 10]. RNNs are mostly used to process sequential input data [19, Ch. 10].

Typically, RNNs are the type of models that should be chosen if the distribution over the target variables $y^{(t)}$ depends on values of the network in the distant past [19, Ch. 10]. The network can also use contextual information from the future if a so called bidirectional architecture is used. The assumption which this model makes is that parameters can be shared so that the same parameters can be used at different time steps [19, Ch. 10].

In the following, the RNN model is described using the layout and terminology of Goodfellow et al. [19, Ch. 10].

3.4.1 Structure

Recurrent neural networks have two forms of visualisation, a circuit graph and an unfolded computational graph. Circuit graphs are crisp, compact visualisations of the network, while unfolded graphs illustrate the flow of information in time. For example [19, Ch. 10]:

$$h^{(t)} = f(h^{(t-1)}, x^{(t)}; \theta)$$

where $h^{(t)}$ is the hidden unit at time t, f is the function that this RNN has learned and which is parameterised by θ. $x^{(t)}$ is the input at time t.

This simple RNN has one hidden unit with a recurrent connection to itself, which can be seen by how $h^{(t-1)}$ is used as an input. The unit further has an incoming connection from the input unit $x^{(t)}$. We can visualise this in form of a circuit graph which is shown in Figure 3.3. When unfolding the graph to see how the recurrent connections create dependencies over time, one gets an unfolded computational graph as shown in Figure 3.4.

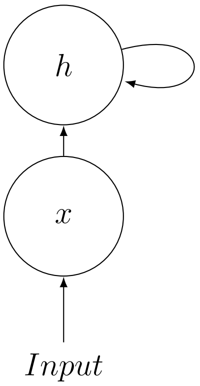

Figure 3.3: The circuit graph of the simple RNN. (Graph as shown in [19, Ch. 10].)

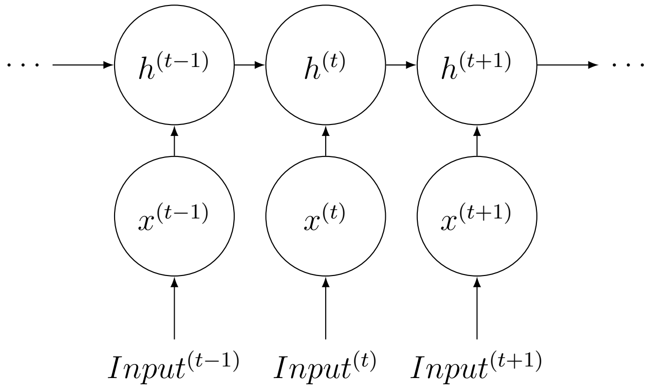

Figure 3.4: The unfolded computational graph of the simple RNN. (Graph as shown in [19, Ch. 10].)

When thinking about the number of parameters of the network, one has to realise that the same function is used at any time step. Thus, the parameters are shared. This appears logical as important information can appear anywhere in the sequence. A lower number of parameters means further that fewer training examples are needed in order for the network to learn. This concept of unfolding is not only good with regards to the number of parameters. It also allows the network to handle inputs with variable input length.

An RNN can also have an output layer. According to Goodfellow et al. [19, Ch. 10], there are three different major design patterns which differ in their recurrent connections as well as in their way of producing outputs:

1. An RNN which has a recurrent connection from the hidden unit to itself and which produces one output per time step.

2. An RNN which has a recurrent connection between the output and the hidden unit and which produces one output per time step.

3. An RNN which has a recurrent connection from the hidden unit to itself and which only produces one output once at the end.

The architecture of the first model is shown in Figures 3.5 and 3.6.

Its forward pass can mathematically be defined as:

$$\boldsymbol{a}^{(t)} = \boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{U}\boldsymbol{x}^{(t)} \tag{3.9}$$

$$\boldsymbol{h}^{(t)} = tanh(\boldsymbol{a}^{(t)}) \tag{3.10}$$

$$\boldsymbol{o}^{(t)} = \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^{(t)} \tag{3.11}$$

$$\hat{\boldsymbol{y}}^{(t)} = softmax(\boldsymbol{o}^{(t)}), \tag{3.12}$$

where the definition is taken from [19, Ch. 10].

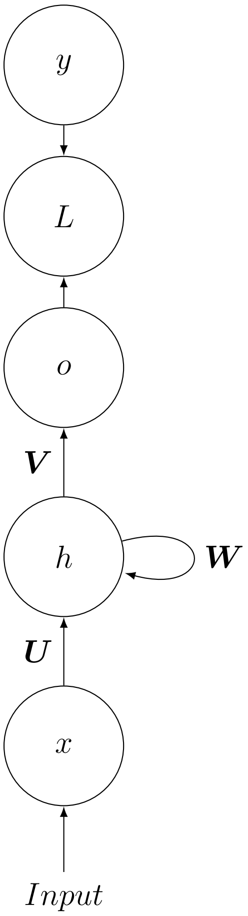

Figure 3.5: The circuit graph of the first RNN architecture (graph as shown in [19, Ch. 10]). x is the input, h the hidden unit, o the network's output, L the loss function (used instead of the objective function in the last example) and y the target value. $\boldsymbol{W}$, $\boldsymbol{V}$, and $\boldsymbol{U}$ are weights.

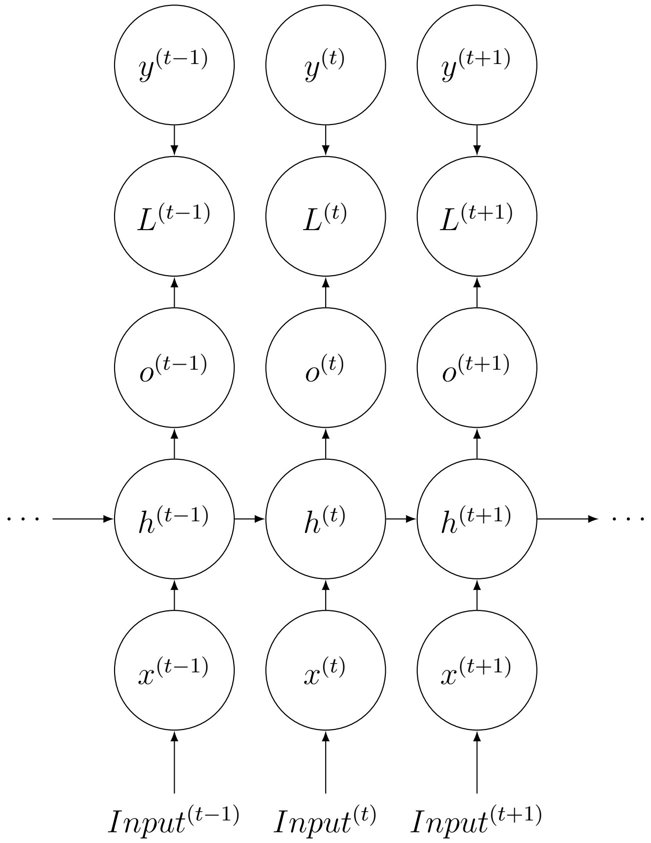

Figure 3.6: The graph of the first RNN architecture, unfolded over time (graph as shown in [19, Ch. 10]). The x are the inputs, h the hidden unit at the different time steps, o the networks outputs, L the loss function (used instead of the objective function in the last example) and y the target values.

Here, $\boldsymbol{W}$ are the parameters of the recurrent connection, $\boldsymbol{U}$ the parameter of the connection between the input and the hidden unit, and $\boldsymbol{b}$ is a bias term. Furthermore, we have $\boldsymbol{V}$, the parameters of the connection between the hidden unit and the output, and $\boldsymbol{c}$, another bias term. In the hidden unit, the RNN can use any kind of activation function. In this case, the hyperbolic tangent is used. The output unit will use a softmax function in order to transform unnormalised log probabilities into normalised probabilities. The softmax function is a type of logistic function which can be used to find the posterior probability of a class given an input in the case of $K > 2$ classes [8, Ch. 4].

The architectures described in this section are shallow ones while one could extend the models to get a deep architecture. A deep architecture can be achieved by using the output of an RNN's hidden layer as input to a new RNN hidden layer, thereby, stacking hidden RNN layers on top of each other [21].

3.4.2 Training

The training method for a recurrent network which has connections between the hidden units is called backpropagation through time. To compute the gradient of the loss function, it is necessary to perform both the forward and backward pass through the unfolded graph. Backpropagation through time cannot be sped up by the use of parallelisation if there are hidden to hidden connections.

We will explain backpropagation through time using the example of the RNN defined in Equations 3.9–3.12.

The loss for this model is defined in terms of the negative log-likelihood:

$$\begin{aligned} L &= \sum_t L^{(t)} \\ &= -\sum_t \sum_k y_k^{(t)} \ln \hat{y}_k^{(t)} + (1 - y_k^{(t)}) \ln(1 - \hat{y}_k^{(t)}) \end{aligned}$$

where $t \in \{1, \ldots, T\}$ is the time step and $k \in \{1, \ldots, K\}$ the class, and $y_k^{(t)}$ the target value at time step t for class k. $y_k^{(t)}$ is 0 or 1. Here, the error is assuming that the class labels are independent given the input vector and that K separate binary classifications are performed using a logistic function to compute the output $\hat{y}_k^{(t)}$ [8, Ch. 5].

We find that

$$\frac{\partial L}{\partial L^{(t)}} = 1 \tag{3.13}$$

and

$$\frac{\partial L^{(t)}}{\partial \hat{y}_i^{(t)}} = \frac{\hat{y}_i^{(t)} - y_i^{(t)}}{\hat{y}_i^{(t)} - (\hat{y}_i^{(t)})^2}. \tag{3.14}$$

One can further find that

$$\frac{\partial \hat{y}_i^{(t)}}{\partial o_i^{(t)}} = \hat{y}_i^{(t)}(1 - \hat{y}_i^{(t)}) = \hat{y}_i^{(t)} - (\hat{y}_i^{(t)})^2 \tag{3.15}$$

from the definition of $\hat{y}^{(t)}$ in (3.12).

We find that

$$
\begin{aligned}
(\nabla_{o^{(t)}} L)_i &= \frac{\partial L}{\partial o_i^{(t)}} \\
&= \frac{\partial L}{\partial L^{(t)}} \frac{\partial L^{(t)}}{\partial o_i^{(t)}} && (3.16) \\
&= \frac{\partial L}{\partial L^{(t)}} \frac{\partial L^{(t)}}{\partial \hat{y}_i^{(t)}} \frac{\partial \hat{y}_i^{(t)}}{\partial o_i^{(t)}} && (3.17) \\
&= 1 \cdot \frac{\hat{y}_i^{(t)} - y_i^{(t)}}{\hat{y}_i^{(t)} - (\hat{y}_i^{(t)})^2} \cdot \left(\hat{y}_i^{(t)} - (\hat{y}_i^{(t)})^2\right) && (3.18) \\
&= \hat{y}_i^{(t)} - y_i^{(t)}, && (3.19)
\end{aligned}
$$

where i is the binary class that we are looking at, t the time step and $\hat{y}_i^{(t)}$ the output of the network for this class and time step, $y_i^{(t)}$ the respective target value. Here, (3.16) and (3.17) follow by the chain rule. (3.18) follows by (3.13), (3.14), and (3.15). (3.19) gives the gradient of the loss function with respect to the output.

When computing the gradient for the loss function with respect to the hidden units $\boldsymbol{h}^{(t)}$, one has to work backwards through the unfolded graph. At time T, the last time step, the gradient is given by

$$
\begin{aligned}
\nabla_{\boldsymbol{h}^{(T)}} L &= \frac{\partial L}{\partial \boldsymbol{o}^{(T)}} \frac{\partial \boldsymbol{o}^{(T)}}{\partial \boldsymbol{h}^{(t)}} \\
&= \mathbf{V}^\top \nabla_{\boldsymbol{o}^{(T)}} L,
\end{aligned}
$$

where we first apply the chain rule, use $\nabla_{\boldsymbol{o}^{(T)}} L$ from above and then compute the derivative of $\boldsymbol{o}^{(T)}$ with respect to $\boldsymbol{h}^{(t)}$ using the definition of $\boldsymbol{o}^{(T)}$ from (3.11).

When we continue the backward pass, we compute

$$
\begin{aligned}
\nabla_{\boldsymbol{h}^{(t)}} L &= \frac{\partial L}{\partial \boldsymbol{h}^{(t+1)}} \frac{\partial \boldsymbol{h}^{(t+1)}}{\partial \boldsymbol{h}^{(t)}} + \frac{\partial L}{\partial \boldsymbol{o}^{(t)}} \frac{\partial \boldsymbol{o}^{(t)}}{\partial \boldsymbol{h}^{(t)}} && (3.20) \\
&= \left(\frac{\partial \boldsymbol{h}^{(t+1)}}{\partial \boldsymbol{h}^{(t)}}\right)^\top (\nabla_{\boldsymbol{h}^{(t+1)}} L) + \left(\frac{\partial \boldsymbol{o}^{(t)}}{\partial \boldsymbol{h}^{(t)}}\right)^\top (\nabla_{\boldsymbol{o}^{(t)}} L) && (3.21) \\
&= \mathbf{W}^\top (\nabla_{\boldsymbol{h}^{(t+1)}} L) \operatorname{diag}\Big(1 - \big(\boldsymbol{h}^{(t+1)}\big)^2\Big) + \mathbf{V}^\top (\nabla_{\boldsymbol{o}^{(t)}} L) && (3.22)
\end{aligned}
$$

for all $t \in \{T-1, \ldots, 1\}$. Here, the chain rule gives (3.20). (3.21) uses the definition of $\nabla_{\boldsymbol{h}^{(t+1)}} L$ and $\nabla_{\boldsymbol{o}^{(t)}} L$. (3.22) follows by the derivative of the $tanh$ function, the chain rule, as well as (3.9) and (3.10) for the derivative of $\boldsymbol{h}^{(t+1)}$ with respect to $\boldsymbol{h}^{(t)}$ and $\boldsymbol{o}^{(t)}$ with respect to $\boldsymbol{h}^{(t)}$. In this formula, $\text{diag}\Big(1 - \big(\boldsymbol{h}^{(t+1)}\big)^2\Big)$ is a diagonal matrix containing elements of $1 - \big(h_i^{(t+1)}\big)^2$.

These gradients can now be used to compute the gradients with respect to the parameters. With respect to $\boldsymbol{c}$ and $\boldsymbol{b}$ these are:

$$\begin{aligned}
\nabla_{\boldsymbol{c}} L &= \sum_t \frac{\partial L}{\partial \boldsymbol{o}^{(t)}} \frac{\partial \boldsymbol{o}^{(t)}}{\partial \boldsymbol{c}} \\
&= \sum_t \Big(\frac{\partial \boldsymbol{o}^{(t)}}{\partial \boldsymbol{c}}\Big)^\top \nabla_{\boldsymbol{o}^{(t)}} L \\
&= \sum_t \nabla_{\boldsymbol{o}^{(t)}} L, \\
\nabla_{\boldsymbol{b}} L &= \sum_t \frac{\partial L}{\partial \boldsymbol{h}^{(t)}} \frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{b}^{(t)}} \\
&= \sum_t \Big(\frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{b}^{(t)}}\Big)^\top \nabla_{\boldsymbol{h}^{(t)}} L \\
&= \sum_t \Big(\frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{a}^{(t)}} \frac{\partial \boldsymbol{a}^{(t)}}{\partial \boldsymbol{b}}\Big)^\top \nabla_{\boldsymbol{h}^{(t)}} L \\
&= \sum_t \text{diag}\Big(1 - \big(\boldsymbol{h}^{(t)}\big)^2\Big) \nabla_{\boldsymbol{h}^{(t)}} L,
\end{aligned}$$

and with respect to $\boldsymbol{V}$, $\boldsymbol{W}$, and $\boldsymbol{U}$:

$$\begin{aligned}
\nabla_{\boldsymbol{V}} L &= \sum_t \sum_i \frac{\partial L}{\partial o_i^{(t)}} \frac{\partial o_i^{(t)}}{\partial \boldsymbol{V}} \\
&= \sum_t \sum_i \Big(\frac{\partial L}{\partial o_i^{(t)}} \Big) \nabla_{\boldsymbol{V}}\, o_i^{(t)} \\
&= \sum_t \Big(\nabla_{\boldsymbol{o}^{(t)}} L \Big) \boldsymbol{h}^{(t)^\top}, \\
\nabla_{\boldsymbol{W}} L &= \sum_t \sum_i \frac{\partial L}{\partial h_i^{(t)}} \frac{\partial h_i^{(t)}}{\partial \boldsymbol{W}^{(t)}} \\
&= \sum_t \sum_i \Big(\frac{\partial L}{\partial h_i^{(t)}} \Big) \nabla_{\boldsymbol{W}^{(t)}}\, h_i^{(t)} \\
&= \sum_t \operatorname{diag}\Big(1 - \big(\boldsymbol{h}^{(t)}\big)^2 \Big) \big(\nabla_{\boldsymbol{h}^{(t)}} L \big) \boldsymbol{h}^{(t-1)^\top}, \\
\nabla_{\boldsymbol{U}} L &= \sum_t \sum_i \frac{\partial L}{\partial h_i^{(t)}} \frac{\partial h_i^{(t)}}{\partial \boldsymbol{U}^{(t)}} \\
&= \sum_t \sum_i \Big(\frac{\partial L}{\partial h_i^{(t)}} \Big) \nabla_{\boldsymbol{U}^{(t)}}\, h_i^{(t)} \\
&= \sum_t \operatorname{diag}\Big(1 - \big(\boldsymbol{h}^{(t)}\big)^2 \Big) \big(\nabla_{\boldsymbol{h}^{(t)}} L \big) \boldsymbol{x}^{(t)^\top}.
\end{aligned}$$

Note how in the case of an unfolded graph the weights such as $\boldsymbol{W}$ are the same at every time step. Therefore, dummy copies of $\boldsymbol{W}$, $\boldsymbol{W}^{(t)}$ are used during backpropagation.

These gradients of the unfolded graph can then be used during the backpropagation algorithm as described in Section 3.3.1 to get the backpropagation through time algorithm.

3.4.3 Bidirectional Recurrent Neural Network

Some task which take sequences as input might benefit from not only taking the past but also the future into consideration. This is for example reasonable to do in a task where the output only has to be made after seeing the full sequence. In order to incorporate both the past and the future, *bidirectional RNNs* can be used. This type of RNN combines two RNNs, one which gets the sequence and another one which gets the sequence in inverted order as input [19, Ch. 10]. The output $o^{(t)}$

will then depend both on the hidden layer of the forward and of the backward RNN [20, Ch. 3].

3.4.4 Long Short-Term Memory

An RNN without any additional architectural changes can only access a limited context [20, Ch. 4]. It has problems to learn long-term dependencies [19, Ch. 10]. This is due to the fact that the effect that the input has on the hidden layer will decay or blow up exponentially while it is passed backwards through the network [20, Ch. 4]. This problem is called the "vanishing (or exploding) gradient problem" [19, Ch. 10].

The most effective solution to the vanishing gradient problem is a special type of RNN, the *Long Short-Term Memory (LSTM)* [20, Ch. 4]. The Long Short-Term Memory model is a type of RNN which uses gates to produce paths on which gradients can flow for longer duration in order to overcome the vanishing gradient problem [19, Ch. 10]. This type of network was first introduced by Hochreiter et al. [23].

The following description of the LSTM is based on [19, Ch. 10] and [20, Ch. 4]. The LSTM model is constructed like a normal RNN, but instead of a normal hidden unit with a recurrent connection, it uses LSTM or memory cells. These LSTM cells have the same in- and outputs as a normal RNN unit. Internally, they additionally have a "self-loop" and they use gates in order to control the flow of information. A simple LSTM as it is commonly used has the following parts:

Internal state unit, $s_i^{(t)}$: The internal state unit is the LSTM cell's internal state. It takes the old internal state, $s_i^{(t-1)}$, the external input, $x_j^{(t)}$, and the external recurrent connection, $h_j^{(t-1)}$, as input. All inputs are gated, by the forget gate for the internal recurrent connection and by the external input gate for the external input and the external recurrent connection. The activation function used by the internal state unit in this case is a sigmoid function,

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\Big(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\Big),$$

where $\boldsymbol{b}, \boldsymbol{U}, \boldsymbol{W}$ are the biases, input weights, and recurrent weights for this internal state unit [19, Ch. 10].

Forget gate unit, $f_i^{(t)}$: The forget gate is used to control when and to what extent to remember the previous state. The decision is based on the external inputs, $x_j^{(t)}$, and the external recurrent connection input, $h_j^{(t-1)}$. The activation function of this unit is a sigmoid function,

$$f_i^{(t)} = \sigma\Big(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\Big),$$

where $\boldsymbol{b}^f, \boldsymbol{U}^f, \boldsymbol{W}^f$ are the biases, input weights, and recurrent weights parameters for this forget gate [19, Ch. 10].

External input gate unit, $g_i^{(t)}$: The external input gate unit is gating the external input, which is $x_j^{(t)}$ and $h_j^{(t-1)}$, based on these same inputs. It is a sigmoid unit,

$$g_i^{(t)} = \sigma\Big(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\Big),$$

where $\boldsymbol{b}^g, \boldsymbol{U}^g, \boldsymbol{W}^g$ are the biases, input weights, and recurrent weights for this external input gate unit [19, Ch. 10].

Output, $h_i^{(t)}$: $h_i^{(t)}$ is the output of the LSTM cell. It is based on the internal state, $s_i^{(t)}$, and gated by the output gate [19, Ch. 10]:

$$h_i^{(t)} = tanh\big(s_i^{(t)}\big)q_i^{(t)}.$$

This unit can have any activation function. In this case the $tanh$ function is used.

Output gate unit, $q_i^{(t)}$: The output gate unit is used for gating the output of the LSTM cell, $h_i^{(t)}$. It also uses a sigmoid activation function.

$$q_i^{(t)} = \sigma\Big(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\Big),$$

where $\boldsymbol{b}^o, \boldsymbol{U}^o, \boldsymbol{W}^o$ are the biases, input weights, and recurrent weights for this output gate unit [19, Ch. 10].

There are different variations of this architecture. For example, one can have $s_i^{(t)}$ as an additional input to the gates. The LSTM is able to learn long-term dependencies, which a simple RNN can only learn with difficulties. LSTMs have been successfully applied to different real-world applications such as speech recognition [21].

3.5 Word Vectors

One way of presenting text as input to a machine learning system is to represent each word by a one-hot vector. This representation is sparse and will suffer from the curse of dimensionality [5]. The vector space is large and the objects are far from each other [5]. Syntactically or semantically close words will also not be located close to each other as there is no notion of similarity between words [39].

Bengio et al. [5] introduced a different, neural-network-based language model in which *word vector* representations and a statistical language model are learned jointly. They learn a word embedding which represents semantically and syntactically similar words close to each other.

The concept of using feed-forward neural networks to learn word vectors was further continued amongst others by Mikolov et al. [39]. The authors present two different model architectures for word embeddings, the continuous skip-gram model and the continuous bag-of-words model. The idea is to use a simple feed-forward neural network with one hidden layer and a softmax function to learn the representations as the weights connecting the input and the hidden layer. In the skip-gram model, for example, the model is trained by learning to predict the context of a word. Thus, if presented a word from a sentence like "The lake is dark blue" and a window size of 2, we find the following (word, context)-pairs: ("The", ["lake", "is"]), ("lake", ["The", "is", "dark"]), ("is", ["The", "lake", "dark", "blue"]), ("dark", ["lake", "is", "blue"]), ("blue", ["is", "dark"]). Note how the context is capturing both the words in the past and in the present. During training of the system, no difference between closer or distant and past or future words is made. The system is presented with training pairs like ("The", "lake") and ("The", "is"). From these it should learn to predict the context word. The learned weights can then be used as the vector representation of the words. While learning this task, the network proved to learn good vector representations for words in the hidden layer. For the continuous bag-of-words model, the inverted task is used for training: to predict the word itself from the context. In [39], this method is improved by Mikolov et al. in terms of speed and quality of the learned representations. The model introduced by Mikolov et al. is called word2vec[2].

The representations learned by this and other models can capture more than simple syntactic regularities. Mikolov et al. [39] showed for example that simple algebraic operations can be used on the vector representation to find the answer to vector(“King”) - vector(“Man”) + vector(“Woman”) = vector(“Queen”).

Another model introduced by Pennington et al. is called GloVe [48]. While word2vec is a prediction-based model, GloVe is a count-based model [48]. Pennington et al. describe GloVe as a “global log-bilinear regression model”. The model captures the statistics of the corpus using word-word co-occurrence counts, or more precisely the ratios of co-occurrence probabilities [48]. The authors present their results which show that they outperform the word2vec model with regards to speed and performance. In the GloVe model, X are the word-word co-occurence counts with $X_{i,j}$ being the number of times word j appears in context of word i. The probability for a word j to occur in the context of word i is computed as $P_{ij} = P(j|i) = X_{ij}/X_i$, where $X_i = \sum_k X_{ik}$. This co-occurrence probability is, according to their observations, not as good for representing the characteristics of the corpus as the ratio between these probabilities. As an example, one can look at the words $i = water$ and $j = pressure$. We can now compute the ratio between the co-occurrence probabilities P_{ik}/P_{jk} with respect to a third word k. This word could be $k_1 = peer$, $k_2 = swimming$, $k_3 = cooker$, and $k_4 = screen$. While “peer” appears in the context of “pressure” but not in the context of “water”, “swimming” occurs in the context of “water” but not of “pressure”, the word “cooker” appears in both contexts while “screen” appears in none of them. Thus, the co-occurrence probability ratio should be low for the word “peer”, high for the word “swimming”, and close to one for the words which relate to both or to none of the words. Thus, the ratios are better than raw probabilities in order to distinguish the words relevant for word representation learning, the ones which are only similar to one of the words, from the words irrelevant for word representation learning. The ratio helps further to better distinguish between the relevant words. Therefore, the authors used these ratios of co-occurence probabilities rather than the raw probabilities as their starting point for word vector learning. Their model is $F(w_i, w_j, \bar{w}_k) = \frac{P_{ik}}{P_{jk}}$, where $w \in \mathbb{R}^d$ are word vectors and

[2] See also `https://code.google.com/archive/p/word2vec/`.

$\bar{w} \in \mathbb{R}^d$ are separate context word vectors. The function F is restricted by them in order to preserve the qualities which a word vector representation should have, such as linearity. Their model learns two sets of word vectors, w and $\bar{w}$, which they claim perform equivalently.

3.6 Linear-Chain Conditional Random Field

As previously mentioned, linear-chain conditional random fields can be used to label a sequence, where the label y is chosen for an input x according to the probability function $p(y|x)$. The function uses features which can depend on the label y, the label of the previous time step, and the input x. The fact that a label is only dependent on the previous but not on the other time steps is called *Markov property*. The linear-chain conditional random field is defined in Equation (3.23) [62]:

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} exp\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t)\}, \tag{3.23}$$

where T is the number of time steps (the sequence length), $Z(x)$ is a normalising factor, $f_k(y, y', x_t)$ is a feature function, K is the number of feature functions, and θ_k are parameters.

3.7 Evaluation Measures

In order to evaluate a machine learning system, commonly used measures can be applied. Systems are often evaluated based on recall, precision, and F1 score. To calculate these measures, one needs to first compute the number of *true positives (TP)*, *true negatives (TN)*, *false positives (FP)*, and *false negatives (FN)*, which form the confusion matrix shown in Table 3.7 [44, Ch. 5].

The *recall* is calculated as:

$$\frac{TP}{TP + FN}.$$

The formula for the *precision* is:

$$\frac{TP}{TP + FP}.$$

		Predicted class	
		+	-
Actual class	+	TP	FN
	-	FP	TN

Figure 3.7: Confusion matrix: a TP is a correctly classified positive sample, a TN is a correctly classified negative sample, a FN is a positive sample which is falsely classified as negative, a FP is a negative sample which is falsely classified as positive.

The *F1 score* is calculated using:

$$\frac{2 \times recall \times precision}{recall + precision}.$$

Chapter 4

Methodology

In this chapter, we present our de-identification method and the evaluation method. Our method uses a combination of both simple and advanced methods in order to solve the inverse de-identification task. In the inverse task, safe tokens are identified to be left in the text. In Section 4.1, the data sets which were used during training and testing of the method are presented. In Section 4.2, we describe two dictionaries used by our method to identify medical terms which are considered non-PHI, thus safe to let back in. In the following, we will refer to non-PHI as being "safe" in the sense that it is safe to let the token back into the text as it cannot be used to identify the data subject. Section 4.3 gives a detailed description of our de-identification method which is a hybrid method combining a rule-based approach with a feature-learning approach. The last section, Section 4.4, contains a presentation of our chosen evaluation method.

4.1 Data Sets

In order to train and test a de-identification model, annotated data is needed. During this project, we worked with two different data sets. The deep neural network was trained and evaluated on data from the 2014-i2b2 de-identification challenge. The second data set, a set of previously annotated case narratives from VigiBase, was used to fine-tune this network. We chose to fine-tune on the VigiBase data set instead of using it as the primary basis for our training due to its limited number of training examples. Furthermore, the similarities between the 2014-i2b2 data set and the VigiBase data set in terms of structure and annotations made this appear to be a reasonable choice.

Both data sets have a separate test set which was used during the evaluation of the method. The VigiBase data set is also used to evaluate the ability of the i2b2-trained network to generalise to VigiBase case narratives without a fine-tuning step.

4.1.1 2014-i2b2 Data Set

This data set, which we will in the following refer to as the 2014-i2b2 data set, was created for "the de-identification track of the 2014 i2b2/UTHealth shared task" [60]. It contains 1,304 medical records of 296 patients. There exist two to five records per patient. Therefore, the authors also refer to the records as "longitudinal medical records". In this project, as described by Stubbs and Uzuner [60], the data was annotated following the HIPAA standard while also introducing additional categories for PHI. The data came from the non-profit organisation Partners HealthCare[1] and PHI was annotated by two annotators. The annotations were manually checked and the PHI afterwards automatically replaced with surrogates. The method to generate surrogates was developed by the authors. Additional categories used, which are not part of HIPAA, were names of hospitals, doctors and nurses, as well as patient's professions and ages below 90 in addition to the ages above 89 included in HIPAA. The annotations were further chosen to be more fine-grained leading to sub-categories such as "Location: Hospital" and "Location: City". The authors decided to label everything that remained with category "Location: Other". This category along with the categories "Location: Room" and "Location: Department" were not used in the final data set for the de-identification challenge. The number of instances of the different PHI categories in the 2014-i2b2 data set can be found in Table 4.1.

The data set has been split by the challenge organisers into a training and a test set as shown in Table 4.1. Of the total number of 790 training documents, 269 were set aside for validation purposes. There are 514 documents in the test set.

[1] Partners HealthCare: http://www.partners.org/

PHI Category	Training	Test	Total
NAME: PATIENT	1,316	879	2,195
NAME: DOCTOR	2,885	1,912	4,797
NAME: USERNAME	264	92	356
PROFESSION	234	179	413
LOCATION: HOSPITAL	1,437	875	2,312
LOCATION: ORGANIZATION	124	82	206
LOCATION: STREET	216	136	352
LOCATION: CITY	394	260	654
LOCATION: STATE	314	190	504
LOCATION: COUNTRY	66	117	183
LOCATION: ZIP CODE	212	140	352
LOCATION: OTHER	4	13	17
AGE	1,233	764	1,997
DATE	7,507	4,980	12,487
CONTACT: PHONE	309	215	524
CONTACT: FAX	8	2	10
CONTACT: EMAIL	4	1	5
CONTACT: URL	2	0	2
CONTACT: IPADDRESS	0	0	0
ID: SSN	0	0	0
ID: MEDICAL RECORD	611	422	1,033
ID: HEALTH PLAN	1	0	1
ID: ACCOUNT	0	0	0
ID: LICENSE	0	0	0
ID: VEHICLE	0	0	0
ID: DEVICE	7	8	15
ID: BIO ID	1	0	1
ID: ID NUMBER	261	195	456
Total	**17,410**	**11,462**	**28,872**

Table 4.1: Number of PHI instances per category in the 2014-i2b2 data set.

4.1.2 VigiBase Data Set

During the Master's thesis project by Sahlström [54], 400 case narratives from the VigiBase database were annotated. The author annotated the samples using the following PHI categories: Date, Age, Location, Organisation, and Person. In order to use the annotated data during this project, it was transformed to the same format used during the i2b2 challenge. For evaluation purposes, all locations were mapped to one location category ("City") and all person annotations were mapped to one name category ("Patient").

The distribution of the PHI examples in the training and test set is presented in Table 4.2. Note the low number of training examples for the categories "Location", "Organisation", and "Person" and note also that the category "Person" is not represented in the test set.

PHI Category	Training	Test	Total
Date	553	185	738
Age	109	30	139
Location	25	9	34
Organisation	5	2	7
Person	4	0	4
Total	**696**	**226**	**922**

Table 4.2: Number of PHI instances per category in the VigiBase data set.

The VigiBase data set contains 300 case narratives for training and 100 case narratives for testing.

4.2 Dictionaries

The rule-based approach to de-identification uses dictionaries to identify safe words which can be let back into a completely masked text. These can be medical terms or common words. These words can however be ambiguous, i.e., some of them could also appear as PHI such as names. Therefore, the rule-based de-identifier will also need some dictionaries to identified known PHI terms or words known to possibly be PHI. Two of the dictionaries used by the rule-based approach, the ones to identify medical terms and drug names, are presented in this chapter. The other dictionaries are introduced in Section 4.3.3.

4.2.1 WHODrug

WHODrug is a drug reference dictionary containing trade names, active ingredients, pharmaceutical formulation, strength and classifications of drugs [36]. It was created within the WHO Programme for International Drug Monitoring to structure the VigiBase data for more efficient analysis and is maintained by the Uppsala Monitoring Centre [36]. It is used to structure and code drug data in reports of suspected adverse drug reactions and in clinical trials for better analysis. The database is filled with medical products appearing in reports of suspected adverse drug reactions, as well as drugs newly registered by the U.S. Food and Drug Administration or European Medicines Agency and based on information from a cooperation with the company IMS Health.

4.2.2 Medical Dictionary for Regulatory Activities

The Medical Dictionary for Regulatory Activities (MedDRA) is a hierarchical terminology by the International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use for classification of medical terms in adverse drug reaction reports [43]. It is used by both regulatory authorities as well as by the pharmaceutical industry [43] and many others including the Uppsala Monitoring Centre, national centres, or academic researchers.

4.3 De-Identification Methods

Our de-identification method is an inverse and hybrid method. The method starts from a text which is fully masked and it then tries to identify safe words to be let back into the text. This does not mean that the method as developed here cannot be extended to be used to identify tokens which possibly are dates, city names, or ages but this is currently not its aim. The method aims to be sure about a token being “safe” before it lets it back into the text. The approach is hybrid because the method solves this task using two rather different approaches. One is a rule-based approach using dictionary look-ups, while the other approach uses deep learning.

In our method, the rule-based de-identifier is designed to be very conservative and thereby producing a high recall of PHI but also many tokens which are falsely classified as PHI. The deep de-identifier gets the role of correcting the errors that the rule-based de-identifier makes, i.e., correcting the PHI tokens which are not classified as such and the more numerous tokens which are safe but are classified as PHI.

4.3.1 Overview

The general process of de-identification is presented in form of a flow chart in Figure 4.1. Before an input text is passed through the de-identification part of the method, it is first pre-processed. The text is tokenised and one can think of each token as being classified as “PHI” (and thus masked) until the method decides otherwise. The de-identification part consists of two de-identification algorithms, the rule-based and the deep learning-based de-identifier. Based on the outputs from the two independent de-identification methods, each token is classified as safe or not. Afterwards, this classification result is used during a post-processing step.

During the post-processing phase, two tasks need to be fulfilled. First of all, the recognised unsafe tokens (PHI-tokens) are written to an XML file as annotations in the same format as for the i2b2-challenge. These are all tokens from the original token list which were not recognised as safe. This output can be used to evaluate the method by using the official i2b2 evaluation function. Secondly, the original text and the recognised unsafe tokens are used to compute a de-identified text. This new text is written to a text file. In this text, the unsafe tokens are replaced by “PHI”, thus forming a de-identified version of the text. The other phases of this process are described in the following subsections.

In an inverse approach, the main idea is to only let tokens in which are recognised to be safe. The tokens that are not let back in are thereby classified as PHI because they are not recognised by the method as being safe. This concept of recognising PHI will make it difficult to give names to each of the recognised PHI tokens. Usually, in de-

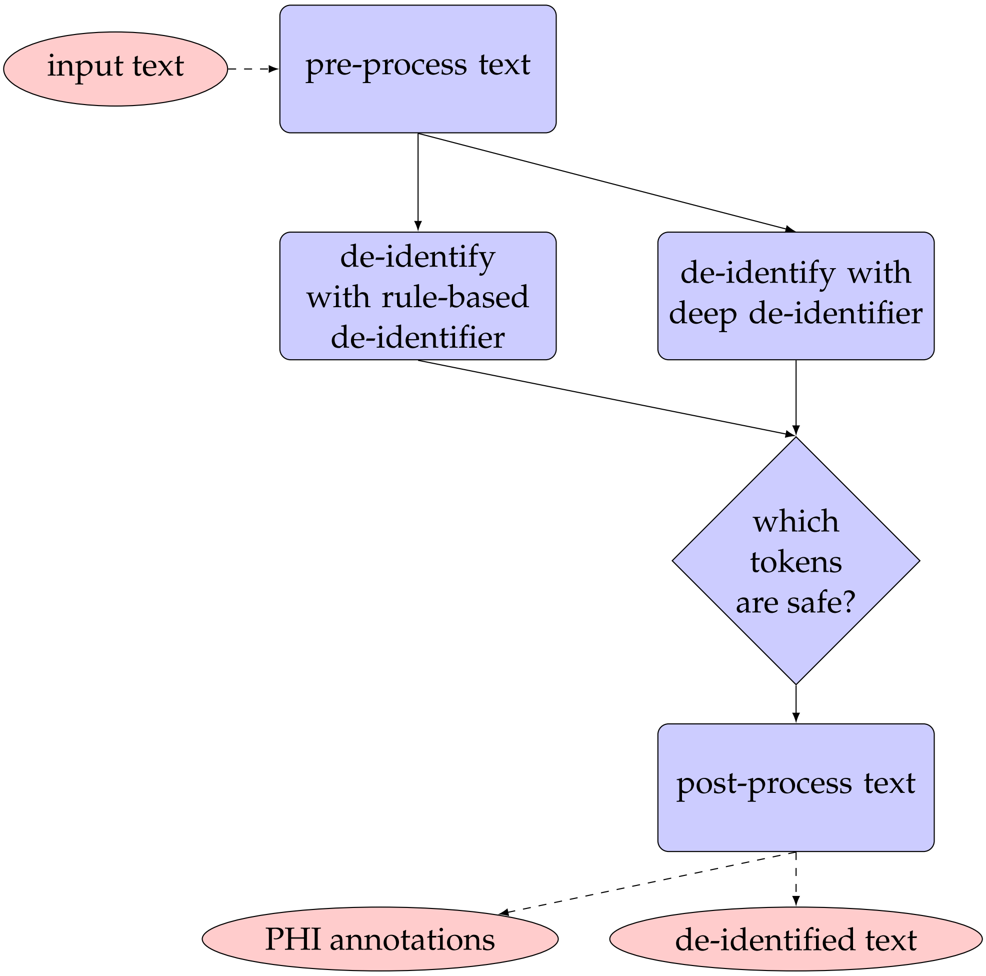

Figure 4.1: The general de-identification process.

identification, the method labels the recognised PHI with a category such as “Person” or “City”. Our method on the other hand does not attempt to label the PHI tokens with categories and does not either try to combine the tokens to form a multi-token PHI instance for the given reason. The output of our method are simply tokens which are safe and tokens which are not safe.

The 2014-i2b2 data set uses, as previously mentioned, both HIPAA-categories and additional PHI categories. The rule-based de-identifier was developed in such a way that it aims to remove all HIPAA-categories, while the deep de-identifier uses all PHI categories and might therefore lead to the removal of non-HIPAA PHI tokens. We assume that this additional information can on the one hand only be helpful for the network to better learn to understand the sentence structure. On the other hand, we designed the method to be especially safe which is increased by taking more possibly PHI out than required by HIPAA.

4.3.2 Annotating and Pre-Processing

The data from both data sets is stored in XML files. It is given in the same format as used in the 2014-i2b2 challenge. In this format, the XML root element is called "deIdi2b2" and it has a child element with the case narrative, the "TEXT", and a child element containing the annotations, the "TAGS". Per annotation such as a name, there is an element of the PHI category as its type and an identification number, a start position, an end position, the text of the annotation, as well as the detailed type such as "City" for the category "Location". We transformed the VigiBase data to the same format and annotation scheme as in the i2b2 challenge. For evaluation of our method, the official Python evaluation script for the 2014-i2b2 challenge can then be used (see Section 4.4.1 for more information on the evaluation method and the evaluation script). The classes from this evaluation module further provide the possibility to load training and test data as well as to write new annotations to XML files.

During the pre-processing phase, the input texts have to be tokenised. This is necessary for the method to be able to process them. In the 2014-i2b2 challenge, one part of the evaluation is made using the tokenised annotations. We use the same tokeniser as used in the challenge's evaluation script. It is based on a regular expression which recognises contiguous sequences of letters and numbers as one token while splitting them at characters other than letters or numbers. This means for example that it splits after symbols and white spaces. For both the rule-based and the deep de-identifier, all tokens are transformed to lower case.

4.3.3 Rule-Based Approach Using Dictionary Lookups

The rule-based part of the method considers each token independently. It checks if the token is included in different dictionaries. It uses the dictionaries to decide whether a word is safe by checking for common English words and for medical terms. To prevent ambiguous words (common word or medical term at the same time as PHI) from being let in, the method also has to use dictionaries with known PHI-terms. The rule-based de-identifier further computes a set of features which can be used later on during the decision phase of the method.

The method will only consider tokens as safe which were found in one of the safe word dictionaries but not in one of the PHI dictionaries. Thus, the results from look-ups in the unsafe term dictionaries will always overwrite the output from the safe term dictionaries in order to remove as many unsafe terms as possible. The rule-based de-identifier also outputs a list of features per token. These are: "is-possibly-city", "has-capital-beginning", "is-sentence-beginning", "is-possibly-name", "is-common word", "is-digit", "contains-digit", "is-medical-term", "is-weekday", "is-month", "is-top-level-domain", "is-street", "is-holiday", "is-single letter", and "is-written out number", which were mostly taken from the previously described dictionaries.

These are the dictionaries with safe terms:

1. Stopwords
2. Common English words
3. WHODrug
4. MedDRA

The list of stopwords is obtained from the Python package "Natural Language Toolkit" (NLTK)[2]. We further check whether a word is a common English word. For this, first of all, the lexical reference system *WordNet* developed by Princeton University's Cognitive Science Laboratory [41] is used. It provides synonym sets for words. If a token has a synonym in WordNet we consider it to be a common English word. Tokens are also checked in the NLTK word list, which includes words retrieved from Wikipedia. Words which are contained in the two presented dictionaries, WHODrug and MedDRA, are also considered to be safe.

These are the dictionaries with unsafe terms:

1. Locations
2. Names
3. Weekdays
4. Months
5. Digits

[2] Natural Language Toolkit: `http://www.nltk.org/`

6. Streets
7. Top-level domains

For the location dictionaries, we obtained a list from NLTK containing U.S. cities and a list of all cities in the world with more than 15,000 inhabitants. The latter was retrieved from the GeoNames Gazetteer[3]. GeoNames provides lists of various geographic data licensed under a Creative Commons Attribution 3.0 License. Note that city names of cities with fewer than 15,000 inhabitants are unlikely to appear in the English dictionary and do therefore not need to be included. Our list of names is a combination of first names obtained from NLTK (corpus by Mark Kantrowitz, Copyright (C) 1991) and last names from the U.S. Census of 2000[4]. From the census we retrieve surnames which occurred 100 or more times. A list of weekdays is used by the method which we assembled ourselves and which also contains abbreviations of the days of the week. The method is further using a list of months, which was also assembled by us and which includes abbreviations. Digits are also considered to be unsafe by the rule-based de-identifier. We check whether a token is a number by using the Python function `isDigit()` and whether a token contains a digit using a Python function which searches for the appearance of a digit. The street name list contains the words "Drive", "Street", and "Avenue" as they commonly occur in street names and are common English words. The top-level domain list contains the common top-level domains which could appear in website names and email addresses, such as "org" or "com". For top-level domains, the rule-based de-identifier checks whether the token which could be a top-level domain is preceded by a dot.

For all safe-word dictionaries with multi-word terms, it applies that we do not want to map the words from the text to the dictionary. This would be useful if we wanted to find where a certain multi-word term occurs. We on the other hand only want to check if a single word from the text occurs in a term in the dictionary. Therefore, we do not need to apply advanced mapping techniques. Instead, we can tokenise the dictionary entries and use the whole dictionary as a bag of words. This simplifies the otherwise complicated way of checking for different orders of the words, left out words in the terms, and so on. These dictio-

[3] GeoNames Gazetteer: `www.geonames.org`

[4] Frequently Occurring Surnames from the Census 2000: `https://www.census.gov/topics/population/genealogy/data/2000_surnames.html`

naries can of course contain words which can also be PHI such as the common example of “Mr. Parkinson” and “Parkinson’s disease”. This can however even be the case for single word medical terms. Thus, we will need to check whether a word is also a possible PHI term and then use the output of the deep de-identifier to make a decision. Matching the multi-word term from the text to the dictionary could help to decide whether a term is actually a medical term but during the decision one would still need to consider the context. The mapping would lead to much overhead while the dictionary is only meant to give a good summary of words which could possibly be used as “safe” words. Afterwards, techniques are still needed to check if this is not the case. We do this both by using the rule-based and the deep de-identifier.

City names differ from multi-word dictionaries as MedDRA in the sense that, for city names, the word order cannot be altered. Thus, the mapping of cities to a text is different than for terms from MedDRA: we need to find occurrences of all tokens of the city name at contiguous positions in the text and in the correct order to find an occurrence. All the unsafe-term dictionaries except for these city names contain only terms of the length of one word and are therefore already a bag of words.

4.3.4 Deep Learning Approach Using Long Short-Term Memory

The deep de-identifier, which uses Long Short-Term Memory (LSTM), focuses on improving the precision by correcting the rule-based de-identifier’s numerous safe tokens mistakenly classified as PHI. It also tries to correct the PHI tokens which were missed by the method. It does so by trying to predict per token with which probability it is safe. It is actually trained to solve the task of labelling all tokens in the sequence with a class, a specific PHI-category or the class “safe”. We check the probabilities for the “safe” class in order to decide if a token might be safe to let back into the text. This is done during the combination step described in Section 4.3.5. There, only tokens are considered to be safe if the deep de-identifier says so with a high probability. How the neural network model is designed and trained to solve this sequence to sequence labelling task is described in the following.

Architecture

The architecture of the deep neural network was chosen similar to the one presented by Dernoncourt et al. [12]. We use RNNs since we are dealing with input data in the form of sequences. LSTM as the type of RNN is chosen because of its ability to overcome the vanishing gradient problem of standard RNNs (see Section 3.4.4). We can apply the LSTM on the sequence both forwards and backwards since we have access to the whole text and thereby also to the future. Thus, a bidirectional LSTM is used. The architecture is shown in Figure 4.2 and described in the following.

Input: Dernoncourt et al.'s architecture (see also Section 2.3.2) as well as ours use both a token-level and a character-level input, which are both transformed by embedding layers to word vectors. A sentence "The patient has..." will look like ["the", "patient", "has"] for the token-level input and [["t", "h", "e"], ["p", "a", "t", "i", "e", "n", "t"], ["h","a", "s"]] for the character-level input. Thus, the token-level input is a sequence of tokens, $x_1, \dots, x_n$, while the character-level input is a sequence of sequences of characters, $[[x_{1,1}, \dots, x_{1,l(1)}]$, ..., $[x_{n,1}, \dots, x_{n,l(n)}]]$, where a token is formed by the character sequence. Note also how all tokens are transformed to lower case during the pre-processing. Here, n is the length of the token sequence and $l(i)$ the length of the i-th token's character sequence.

Token-Based Embedding: Each token from the token sequence is passed through a token embedding function, E_T. The result is a sequence of token-based embeddings, $e_{t_1}, \dots, e_{t_n}$. This embedding function can be pre-trained for example using GloVe or word2vec (see Section 3.5).

Character-Based Embedding: Each sequence of characters is passed through a character embedding layer. This is shown in Figure 4.3. Its purpose is to produce a second embedding for the token based on its sequence of characters. It uses an embedding layer, which is a mapping from a character to a vector, E_C. Note how the same embedding is applied to each character regardless of its position in the sequence. The sequence of vector embeddings for the sequence of characters is then passed through a bidirectional LSTM. It produces one output vector.

The LSTM activation are thus combined to one output for the whole sequence instead of one per character. This way, the method produces an embedding per token which is based on each of its characters combined. For token x_i, this embedding is called e_{c_i} in Figure 4.2.

Bidirectional Long Short-Term Memory: The sequence of the combined embeddings, the concatenations of the character-based and the token-based embeddings, are passed through a bidirectional LSTM which produces an output per token in the sequence, $(a_{1,\rightarrow}, a_{1,\leftarrow}), \ldots,$ $(a_{n,\rightarrow}, a_{n,\leftarrow})$. Each of these outputs is the concatenation of the forward LSTM output and the backward LSTM output for the respective token.

Softmax Layer: This output is in turn transformed by a softmax function into probabilities for the token being of a certain class. Thus, the output of the whole network is a sequence of vectors whose entries are probabilities for the token to belong to one of the output classes.

Variations: The architecture was slightly adapted during some runs in order to try different variations and to evaluate their impact on the performance.

First of all, the character-level input was enhanced by concatenating up to three of the possibly following characters to the character sequence, i.e., characters which follow but which are positioned before the beginning of the next token. This could be white spaces, newlines, or symbols. The idea is that these might help the network to understand the sentence structure better. Especially, it might help in cases where there are dates written in the format "DD/MM/YY", which is quite common. Adding this information to the input gives the network the chance to learn features which use the presence of the characters. We will refer to these as "post space"-characters since they follow the token in the space just behind this token and before the next token.

In a second variation, the architecture was adapted to include a third input. This input is a feature-level input. Thus, hand-engineered features which are expected to be useful for the network are fed into it. These features are concatenated to the combined embeddings. We chose to add information on whether the token is all lower case, all upper case, or starting with a capital letter in order to make up for the information loss which happens during pre-processing. During pre-processing, all tokens are transformed to lower case. We also give the

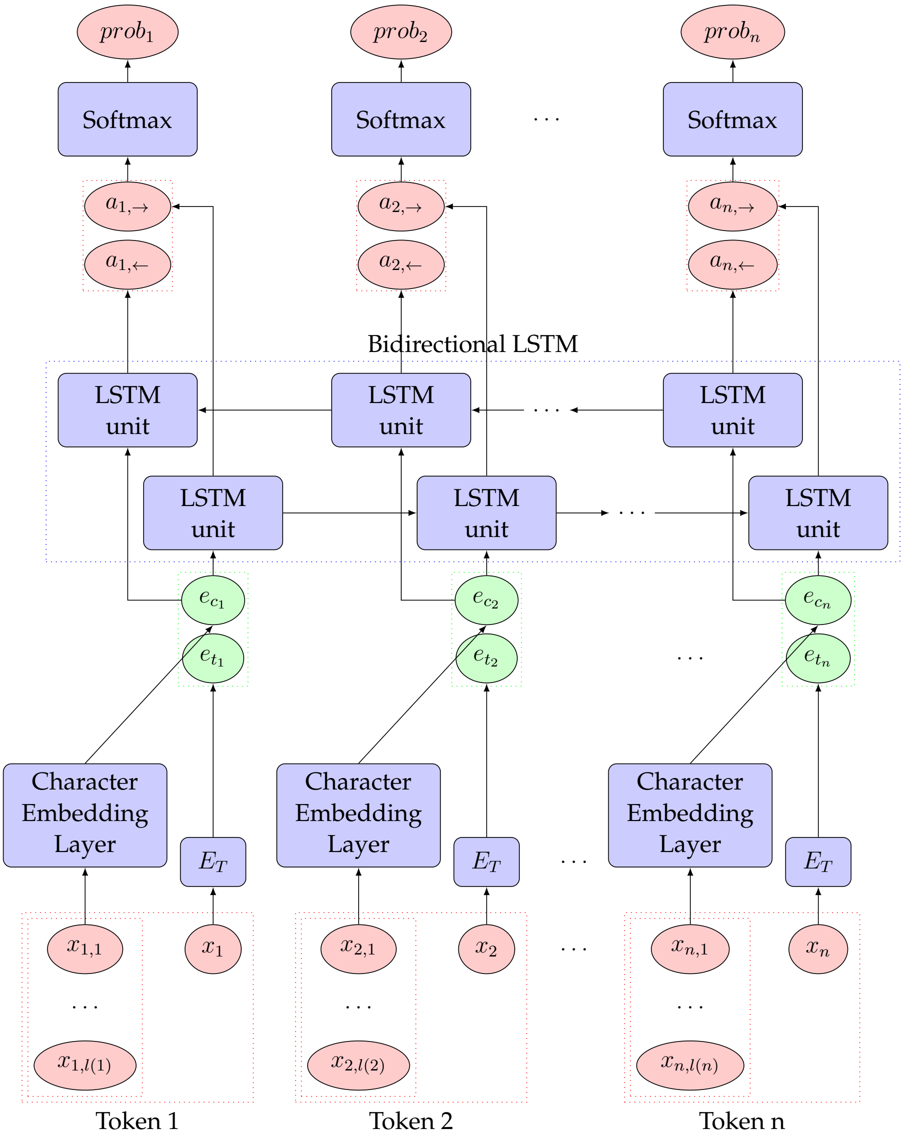

Figure 4.2: Artificial neural network architecture. The character-level input is passed through the character embedding layer from Figure 4.3. The token-level input is passed through the embedding function, E_T. The resulting embeddings are concatenated and passed through a bidirectional LSTM. Its output is transformed into probabilities, $prob_1, \ldots, prob_n$, using a softmax layer.

network additional input indicating the presence of salutation words such as “Mr”, “Mrs”, or “Ms”. The intention is that this might help

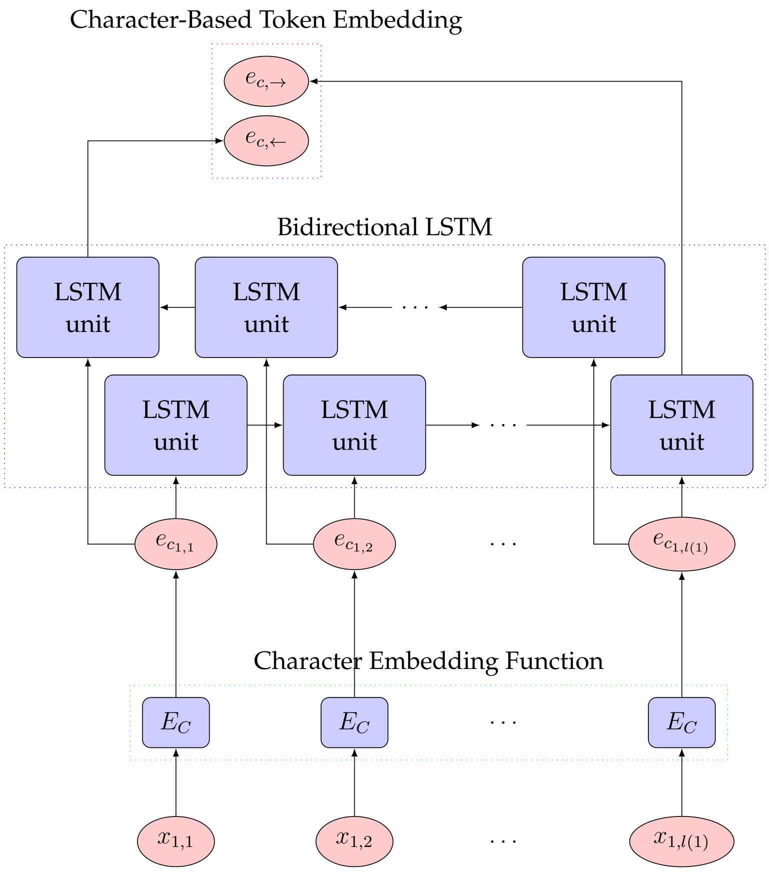

Figure 4.3: Character Embedding Layer. This is the artificial neural network architecture for the character-based embedding of token $x_1 = (x_{1,1}, x_{1,2}, \ldots, x_{1,l(1)})$, with $l(1)$ being the length of x_1. E_C is the character embedding, a mapping from a character to a vector. The same function E_C is applied to all characters regardless of their position in the sequence. The two LSTMs of the bidirectional LSTM are unfolded. The unfolded forward LSTM consists of the units from the lower line and the backward LSTM of the LSTM units of the upper line. Both LSTMs output one vector per character sequence which are concatenated to form one character-based embedding.

the network recognise names more easily. We added this when we noticed that some of the trained networks did not recognise patient names despite the presence of these strong indicator words.

Training Details

When training on data for de-identification, it has to be considered that the class distribution is imbalanced. This is due to the fact that the natural text includes more safe words than PHI tokens. Therefore, we weight the samples differently according to the class they belong to. This way, the samples of rare classes contribute more to the loss function. The loss function was chosen as a categorical cross entropy (see the definition of the negative log-likelihood in Section 3.4.2). It is used because it is suitable for the multi-class classification and because it is based on the difference between the target value and the output of the network, thus the margin by which the classification for a data point was wrong. "Adam" was chosen for optimisation. Adam is a stochastic, gradient-based optimisation algorithm which uses "adaptive estimates of lower-order moments" where momentum can be understood as the first order moment of the objective function and where in addition to the first order moment also the second order moment is used [28]. The learning parameters were left at the default values suggested by the deep learning framework: learning rate 0.001, $beta_1 = 0.9$, $beta_2 = 0.999$, $epsilon = 1e - 08$, and no learning rate decay was used. For regularisation purposes, dropout was used after the concatenation of the embeddings. The dropout probability was chosen to be $p = 0.5$.

For the token embedding layer, we chose a pre-trained GloVe layer, trained on Wikipedia text, with 100 as dimension of the embedding. We chose to use GloVe instead of word2vec because Dernoncourt et al. [12] found it to work slightly better in their development of a de-identification method.

The character embedding function is also a layer in the neural network. It is however not pre-trained but trained from scratch at the same time as the rest of the network.

Implementation Details

We implemented our method in Python using version 3.5.3. For the deep learning model, we used the "high-level neural network API" *Keras* [10], version 2.0.2, with TensorFlow [2], version 1.0, as its deep learning backend. Both training and testing of the model was run in GPU mode. For the implementation of the model we used the Keras layers `Dropout`, `Input`, `Embedding`, `TimeDistributed`, `Dense`, `LSTM`, `Bidirectional`, and the function `concatenated`, as well as the `Model` class and the Keras implementation of the `Adam` optimiser. The model is trained using the function `fit` and used using the function `predict` of the `Model`. When training a model using Keras, the target values have to be represented using a 1-hot-encoding.

In order to enable batch learning, the input sequences of the network have to have equal length. Therefore, a maximum length for the token sequences and the character sequences, forming the tokens, was chosen. The sequences have to be padded and possibly split into several parts. The maximum length is limited by the amount of GPU memory available. This is due to the fact that the unfolded model has to fit into the GPU memory. It thereby also depends on the size of the network's layers. The padded values should not influence the network's output. Therefore, they are set to 0 which will lead to them not influencing the LSTM's output. In the loss function these tokens have to be ignored as well. This is reached by weighting the padded tokens with 0 in the loss function.

4.3.5 Combination Strategy

The output of the deep and the rule-based de-identifier have to be combined to make a decision for each token whether it is safe or PHI. The rule-based de-identifier is designed to only let in words for which it is highly confident that they are safe. This makes it a very conservative approach. It is designed to have few missed PHI tokens but thereby also many tokens which were mistakenly classified as PHI resulting in a low precision. The rule-based de-identifier will still lead to some missed PHI as it cannot base its decisions on the context when identifying PHI. This is problematic when a token is not in the list of known PHI but in the common words or the medical term lists. The rule-based de-identifier is designed to not let any token containing numbers in, although these can contain very important information. This rule leads to many tokens falsely recognised as PHI. Both these errors, missed PHI and tokens falsely classified as PHI, might be corrected by combining the rule-based de-identifier's output with the deep de-identifier's output.

One way to combine the two outputs would be to only allow words back in which were classified as safe by both identifiers. This would allow to correct some of the missed PHI of the rule-based de-identifier but not the tokens which were identified as such mistakenly, leading to an extremely low precision. This is impractical and this method is therefore not used.

We use another way of combining the two methods. In this combination method, we use the rule-based de-identifier to decide on how high the threshold of the deep de-identifier should be for it to classify a word as safe. If the rule-based de-identifier says that the word is safe, then a lower threshold can be used in the deep de-identifier. We look at the probability with which the deep de-identifier believes that the output is safe. If this probability is above the threshold, the token is classified as safe. The level of the threshold is chosen according to the output of the rule-based de-identifier. We call this "adaptive-threshold-rule". This combination method requires two thresholds as parameters, a low and a high threshold for when the rule-based de-identifier classifies the token as safe or PHI.

In this "adaptive-threshold-rule", if the deep de-identifier does not say with a high probability that the token is safe it is removed. This is a good idea since the deep de-identifier has a high precision when identifying PHI. If it thinks that something is PHI, i.e., not safe, it most likely is. Thus, it is not likely to cause many tokens wrongly classified as PHI. If the rule-based de-identifier says that a token is PHI it could be one of its many mistakes made classifying a safe token as PHI. Therefore, we want to check if it actually is PHI using the deep de-identifier. The token will in this case be classified as safe if the deep de-identifier classifies it as safe but only if it says so with a very high probability. If the rule-based de-identifier classifies a token as safe it could be one of the rare missed PHI. Therefore, we check with the deep de-identifier if it is safe or not but with a lower threshold to prevent a high number of tokens mistakenly classified as PHI. There is a third case: if the rule-based de-identifier outputs that a token is not safe and that it contains one of the features "is-weekday", "is-month", "is-street", "is-holiday", and "is-written out number", then the threshold for the deep de-identifier is picked as infinity. Thus, these tokens are always removed because they appear to always be PHI.

4.3.6 Model Selection

During this project, we evaluated four different variations of the model which are presented in Table 4.3. In the first variation, a), we use none of the variations mentioned in the architecture description (Section 4.3.4). In the second variation, b), we add the "post space"-characters to the character input as described in Section 4.3.4. In c), we use the additional inputs in form of salutation-features and capitalisation-features as described in Section 4.3.4. Variation d) includes both these concepts.

Model	Post Space	Additional Input
a)	○	○
b)	●	○
c)	○	●
d)	●	●

Table 4.3: The different variations of the method.

In order to select the best model, we train variation a) to d) two times for four epochs. Due to the time constraints of this project, we could not make any experiments with training for more epochs. 10% of the training samples are not used during network training but for validation through accuracy monitoring during training. We evaluate the models' performance after the different numbers of epochs. For this, we run the algorithm on the separate validation data and use a threshold pair with (low $= 0.8$, high $= 0.9$) in the "adaptive-threshold-rule", where low stands for the lower threshold and high for the higher threshold. The best model from the runs is picked, i.e., the model after the epoch with best performance with respect to HIPAA-recall while also taking the general precision into account (see Section 4.4 for more information on the evaluation function).

We rerun these best models for the different variations and the two runs (at best epoch) on the validation set but with a threshold pair of (low $= 0.9$, high $= 0.95$). After comparing the performance on the validation data, we found that this threshold pair generally led to better results on the validation sets (results not shown here). It would have been too computationally expensive to compute the performance on the validation set of this second threshold for all epochs.

We then evaluate the performance of these best models with thresholds (low $= 0.9$, high $= 0.95$) for the four different variations (a–d) on the test set. The results are presented in Chapter 5. Experiments for more different threshold pairs would have been useful but there was not enough time to perform an extensive search.

For the VigiBase model, we took the best model for variation d) to test its performance on the test data both after fine-tuning on VigiBase training data and when solely combined with the "adaptive-threshold-rule". For the "adaptive-threshold-rule" both the threshold pairs (low $= 0.8$, high $= 0.9$) and (low $= 0.9$, high $= 0.95$) were tested. Fine-tuning was performed for 2 epochs and the model was tested after both epochs. We chose variation d) because we did not know at this point which variation would perform best and as we hypothesised that it might be variation d). We also tested the method on the test set when only using its rule-based or deep de-identifier.

4.4 Evaluation

In order to evaluate how the method performs, we compute the recall of PHI on the separate test sets of the i2b2 and the VigiBase data sets. In this way, we can evaluate how safe the method is. For the evaluation of the data preserved, we look at the precision, which however is not expected to be high. For the VigiBase data set, we also let an expert evaluate how much information valuable for causality assessment is preserved. What is considered to be information valuable for causality assessment was presented in Section 2.1.1.

4.4.1 Recall and Precision for Protected Health Information

The method is evaluated based on its recall on tokenised HIPAA annotations from the i2b2 and the VigiBase gold standard. This means that the HIPAA annotations, which can contain more than one token each, are tokenised before they are compared to the method's annotations (which are already single tokens). Since we use the same tokeniser for preprocessing which is also used during the evaluation, the method's annotations will only consist of one token each. The tokenised evaluation is necessary because the method only outputs PHI classified tokens without aggregating them to form a person's name or a city name. This evaluation of tokenised annotations is provided by the evaluation script of the i2b2-challenge. Nevertheless, the script had to be adjusted to properly handle the fact that the method's annotations do not specify any valid PHI category labels for the annotations. During the evaluation, our adjusted algorithm checks for each token of each of the gold standard annotations whether there also is a token annotation made by our method for the token. The gold standard annotation will belong to a certain PHI category while the method's annotation will be a general PHI annotation. Scaiano et al. [56] refer to this as "masking recall". For this, it is important that the token is masked by the method but not that the method knows the token's PHI category. Scaiano et al. point out that it is reasonable to use this type of measure, but using this measure can be problematic if the token frequency per class is not taken into consideration. In our masking recall measure, we can however also output the masking recall for each PHI category (as we know the PHI category of each gold standard annotation) and thereby account for varying PHI category frequencies.

During the evaluation of the method, the recall of PHI is considered as the main measure of performance. The precision, which reflects to what extent masked words are in fact PHI, is, as previously mentioned, not focused on when improving the method at this stage. It is not possible to compute the precision per PHI class since the de-identification method does not assign PHI categories. This means that we cannot compute the number of tokens falsely classified as PHI per category. Since we are removing additional PHI instances which are not included in HIPAA, our method cannot reasonably be evaluated using precision for HIPAA tokens. This is due to the fact that we cannot tell which of the tokens was mistakenly removed due to the attempt to remove HIPAA tokens and which due to the attempt to remove other PHI tokens. We therefore look at the precision for PHI tokens in general when we want to evaluate the method's precision. Since other methods in the literature usually are evaluated based on recall of HIPAA PHI, we evaluated our method based on HIPAA PHI recall.

We changed the implementation of the evaluation script to consider different start and end points of an annotation. Since we are using the same tokeniser as the evaluation script when tokenising our input texts, our correct annotations should in general be the same tokens as the tokens from the tokenised gold standard annotation. There can however be cases in which our annotation token fully covers a gold standard annotation and includes some additional characters in the beginning or end. This occurs when the gold standard annotation starts or ends in the middle of a token which the tokeniser usually keeps together. This means that this probably is a token in which a white space or another separating character is missing. Thus, this token could be seen in the gold standard as two different tokens. It is therefore reasonable to count it once as a correctly recognised PHI token and once as a token which was falsely recognised as being PHI as it represents both in one. It further is not a non-recognised PHI token as its PHI-part was correctly classified as PHI. Therefore, a gold standard HIPAA annotation token is in our version of the evaluation script not counted as missed if the PHI token is fully overlapped by a token of the method and the token is counted as a token which was correctly classified as PHI. This overlapping token is however also counted as a falsely classified PHI token and it therefore lowers the precision of the method. Note that this kind of change to the evaluation function does not make a conservative method like ours look safer than it is. A conservative method should be optimised for high recall. This change only removes the tokens recognised by the original evaluation function as being missed PHI which are actually fully covered by a method's annotation. The recall does therefore not appear better than it is.

During the evaluation of the method's performance on VigiBase data, we also manually check the missed PHI. We assess which leaked PHI token actually could lead to a re-identification and correct the counted number of missed PHI. Here, it is for example important to note that an annotation of a date could include the word "of" in "1st of March". A leaked token "of" does however not allow for re-identification of the data subject.

4.4.2 Retainment of Valuable Information

In order to get an idea about whether or not the method preserves valuable information, we made what can be understood as a type of qualitative precision evaluation which was carried out by an expert.

A pharmacist from the Uppsala Monitoring Centre looked at the resulting de-identified case narratives in order to check their usability for causality assessment. The pharmacist looked at the de-identified version of the 100 VigiBase test samples. The pharmacist read these and checked if there was any useful information preserved and whether there were any parts masked which might be useful during a causality assessment. Afterwards, she checked in the original narrative which words or numbers were removed and whether the information would actually have been necessary for the assessment of causality.

Chapter 5

Results

In this chapter, we present the performance of our method on the test sets while using the evaluation methods as described in Section 4.4. In the first section, Section 5.1, we present the results on the 2014-i2b2 test set. For this, we present the recall and precision for protected health information and the output for some example inputs. In Section 5.2, we present the results in terms of precision and recall on the VigiBase test set. Here, we also present some examples of PHI tokens missed by our method. The section also contains a presentation of the results from the qualitative analysis of the preserved valuable information in the de-identified case narratives.

5.1 2014-i2b2 Data Set

This section includes the results on the i2b2 data set. The performance of the different variations of our method as described in Section 4.3.6 can be found in Section 5.1.1 for the adaptive-threshold combination rule and in Section 5.1.2 for only using the deep de-identifier. In Section 5.1.3, a comparison between the rule-based, deep, and hybrid methods as well as with a method from the related work is made. In Section 5.1.4, we present the results per category and Section 5.1.5 contains a presentation of example outputs.

5.1.1 Evaluation of the Hybrid De-Identifier

For the adaptive-threshold rule, we tested all four variations (a–d) of the deep model from two different runs of training. We selected the higher pair of thresholds as it showed better results on the validation set for all variations. The results are shown in Table 5.1

	Recall – HIPAA	Precision – All
Adaptive-threshold, a-1)	0.985	0.493
Adaptive-threshold, a-2)	**0.994**	0.361
Adaptive-threshold, b-1)	0.991	0.518
Adaptive-threshold, b-2)	0.991	0.494
Adaptive-threshold, c-1)	0.987	0.464
Adaptive-threshold, c-2)	0.981	**0.571**
Adaptive-threshold, d-1)	0.988	0.563
Adaptive-threshold, d-2)	0.990	0.505

Table 5.1: Token-based HIPAA-recall and PHI-precision evaluation on the i2b2 test set for variations a–d) for runs 1 and 2 with adaptive-threshold rules with a high threshold pair (low $= 0.9$, high $= 0.95$). The highest recall and the highest precision are highlighted.

Studying the difference in recall and precision between the different variations a) to d) can give an idea of the impact that the additional inputs, post-space characters and feature inputs, can have on the models' capability to learn the mapping between token and PHI categories (and the safe category). Variation a), the one without any additional inputs, does based on the two runs not appear to be able to reach a recall greater than 99% at the same time as a precision greater than 50%. When the post space character input is added, in variation b), the recall is in both runs above 99% while it seems possible to reach a precision greater than 50% at the same time. If we however add the additional feature input instead of the post space characters, as in variation c), we do not see a clear indication of a result with recall above 99% and precision greater than 50% being possible. Going from only an additional feature input to adding both additional inputs, as in going from variation c) to variation d), seems to allow the recall to be improved, possibly even to a higher value than 99%, while there is no large loss in precision at the same time. In general, for variation d), a result of recall greater than 99% and precision greater than 55% appears possible.

When going from variation b), which uses the post space character input, to variation d), which uses both additional inputs, there does not seem to be a significant difference in performance. In the two runs it looks like adding feature vectors in addition to the post space character input leads to a minor loss in recall but an improvement in precision. Thus, the post space character input appears to improve the performance in general while the results suggest that the additional feature input may improve the precision. Model variation b-1) performs best on the test data with respect to a balance between recall and precision.

This analysis of the repeated runs of training however also shows that there is a substantial variation in performance between runs with the same configuration. The results should therefore be interpreted with some caution.

5.1.2 Evaluation of the Deep De-Identifier

For the deep de-identifier, we used a threshold of 0.8, 0.9, and 0.95 and tested all model variations during the two training runs. The results are presented in Table 5.2.

In this table, we can see that using only the deep de-identifier in a certain variation and run can lead to similarly high recall as in the hybrid method. This is the case when a high threshold such as 0.95 is used. In these cases, the precision is however lower than for the hybrid method. We can also see that using a lower threshold for what is classified as safe can lead to high precision but it is combined with a loss in recall. Such a recall of less than 99%, however, is too low for our purposes. None of these variations of only using the deep de-identifier seems to allow to have both a recall greater than 99% and a precision of 50% and they all perform worse than the corresponding adaptive-threshold rule ones.

5.1.3 Comparisons

In Table 5.3, we compare the result of only using a rule-based de-identifier, with the results of only using a deep de-identifier, as well as with the adaptive-threshold. This table also shows a comparison with other methods.

We can see that the highest recalls were achieved by the rule-based de-identifier, the deep de-identifier when using a high threshold, and

	Recall – HIPAA	Precision – All
Only deep, $th = 0.8$, a-1)	0.946	0.680
Only deep, $th = 0.8$, a-2)	0.964	0.542
Only deep, $th = 0.8$, b-1)	0.971	0.713
Only deep, $th = 0.8$, b-2)	0.969	0.675
Only deep, $th = 0.8$, c-1)	0.951	0.665
Only deep, $th = 0.8$, c-2)	0.935	**0.768**
Only deep, $th = 0.8$, d-1)	0.964	0.740
Only deep, $th = 0.8$, d-2)	0.971	0.697
Only deep, $th = 0.9$, a-1)	0.971	0.571
Only deep, $th = 0.9$, a-2)	0.987	0.423
Only deep, $th = 0.9$, b-1)	0.983	0.568
Only deep, $th = 0.9$, b-2)	0.984	0.600
Only deep, $th = 0.9$, c-1)	0.977	0.550
Only deep, $th = 0.9$, c-2)	0.965	0.664
Only deep, $th = 0.9$, d-1)	0.980	0.643
Only deep, $th = 0.9$, d-2)	0.984	0.586
Only deep, $th = 0.95$, a-1)	0.985	0.470
Only deep, $th = 0.95$, a-2)	**0.994**	0.339
Only deep, $th = 0.95$, b-1)	0.991	0.498
Only deep, $th = 0.95$, b-2)	0.979	0.479
Only deep, $th = 0.95$, c-1)	0.977	0.550
Only deep, $th = 0.95$, c-2)	0.965	0.664
Only deep, $th = 0.95$, d-1)	0.973	0.551
Only deep, $th = 0.95$, d-2)	0.990	0.482

Table 5.2: Token-based HIPAA-recall and PHI-precision evaluation on the i2b2 test set for variations a–d) for runs 1 and 2 only using the deep de-identifier. The highest recall and the highest precision are highlighted.

	Recall – HIPAA	Precision – HIPAA
ANN [12]	0.974	**0.983**
ANN + CRF [12]	0.978	0.979
	Recall – HIPAA	Precision – All
Adaptive-threshold, b-1)	**0.991**	0.518
Only deep, $th = 0.9$, b-2)	0.984	0.600
Only deep, $th = 0.95$, b-1)	**0.991**	0.498
Only rule-based	0.988	0.117

Table 5.3: Token-based HIPAA-recall, HIPAA-precision, and general-PHI-precision evaluation on the i2b2 test set for the best adaptive-threshold rule variation, the best deep de-identifiers, and rule-based de-identifier compared with the artificial neural network model (ANN) and artificial neural network model combined with a conditional random field model (ANN+CRF) of Dernoncourt et al. [12]. The highest recall and the highest precision are highlighted.

by combining the output of the rule-based de-identifier with the deep de-identifier's output. The high recall of the rule-based de-identifier was to be expected as it was designed to be very conservative. Its precision however is with 11.7% too low to be applicable in practice. Many of the adaptive-threshold rule variations (see Table 5.1) reached a higher recall than the rule-based de-identifier while also achieving a higher precision. The adaptive-threshold rule using variation b-1), which includes an additional post-space character input but not an additional feature input, for example, achieved a slightly higher recall while increasing the precision up to 51.8% as compared to the rule-based de-identifier with 11.7%. The deep de-identifier performs better than the rule-based de-identifier both in terms of recall and precision. We have seen that the deep de-identifier when used alone can lead to a recall as high as achieved by the hybrid method. The precision is then however lower than for the hybrid method.

How our results compare to the deep learning approach of Dernoncourt et al. [12] can also be seen in Table 5.3. We compare our results using an adaptive-threshold combination rule with their results for which they were both using an artificial neural network (ANN), which uses an LSTM and a conditional random field layer, and a combination of this artificial neural network with an additional, independent conditional random field model (ANN+CRF). We also added the best only-deep de-identifiers with respect to HIPAA-recall and all-token-precision balance (variation b-2 with $th = 0.9$ and b-1 with $th = 0.95$) and the rule-based de-identifier to the comparison. We find that our hybrid, inverse approach outperforms the state-of-the-art deep learning method with respect to recall but that it has a fairly low precision. Here, we need to note that our method has to be evaluated based on general PHI precision while Dernoncourt et al. evaluated their method based on HIPAA-precision.

5.1.4 Results Per Category

How the adaptive-threshold-rule models perform per category when using the threshold pair (low $= 0.9$, high $= 0.95$) is shown in Table 5.4. Although we found that b-1) is the best model, a-2) has the highest HIPAA-recall which can also be seen in a-2) having achieved the highest recall in more categories than most of the other models but a-2) has a low precision. We expected to see an increase in recall of patient names when adding the additional feature input, i.e., between variation a) to c) and b) to d) but there exist no significant difference. An increase in recall of dates could have been further expected when adding the post space character input from variation a) to b) and from c) to d). This was however not the case. For medical records, an increase in recall can be seen between variations c) and d) but not between a) and b).

	Adaptive-threshold rule							
	a-1)	a-2)	b-1)	b-2)	c-1)	c-2)	d-1)	d-2)
Date	0.988	**0.997**	0.995	0.994	0.991	0.983	0.992	0.991
Patient	0.946	**0.977**	0.968	0.966	0.964	0.965	0.968	0.975
Med. Rec.	**0.996**	0.990	0.989	**0.996**	0.974	0.981	0.993	0.993
Age	0.982	**0.987**	0.979	0.982	0.975	0.956	0.961	0.985
Street	0.998	0.998	**1.0**	0.998	**1.0**	0.998	**1.0**	**1.0**
Phone	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	0.998	**1.0**	**1.0**
City	0.997	0.997	**1.0**	0.994	**1.0**	0.997	0.997	**1.0**
Org.	0.891	0.905	0.884	0.891	**0.925**	0.884	0.864	0.905
ZIP	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Device ID	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Fax	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Email	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

Table 5.4: Token-based per category recall on the i2b2 test set for variations a–d) for runs 1 and 2 with an adaptive-threshold-rule and the threshold pair (low $= 0.9$, high $= 0.95$).

5.1.5 Example Outputs

In general, one can say that many documents did not have any missed PHI tokens. If they contained leaked PHI, it was often only one token that leaked, e.g., only one part of a date. Thus, the leaked information could often be considered insufficient for re-identification. There were however patient names which leaked. Some of them were common words but even names which are not common words and which were identified by the rule-based de-identifier as possible names were missed. There were also initials of people's names which leaked. With regard to dates, some of the leaked PHI tokens are actually harmless such as "s" and "of" but there were also numbers, seasons, and weekdays which leaked. The method could sometimes not recognise a date which was included in a common date format. The method also had problems recognising a weekday which was combined with another letter because a white-space was missing in between. Many of the missed "Age"-tokens were the token "s" but even numbers followed by clear indicators such as "years" and "y o" were missed.

In Tables 5.5, 5.6, and 5.7, we show three sample reports which we generated ourselves. They were designed to be similar to reports or excerpts from reports from the i2b2 data set. The tokens marked in *blue* are PHI which were correctly recalled and removed. The **red** tokens are those PHI tokens which were missed by the de-identification method. The underlined tokens are safe tokens which were mistaken for being PHI and therefore removed. We can see two missed PHI which were both of the type "Patient". One is an initial, the other one is a first name used in a normal sentence. This last kind of mistake appeared several times on the i2b2 test set. We can see that dates are recalled in different formats and contexts. Some of the mistakenly removed safe tokens, such as "Name" or "M" in "M.D.", are less important for understanding the narrative than others such as doses or the blood pressure. Thus, although the narratives are still readable, they are sometimes missing important information.

Original	De-identified
Record date: *2063*-*05*-*27*	Record date: *PHI*-*PHI*-*PHI*
JOHNS, *EVA* *96735682* *05*/*27*/*2063*	*PHI*, *PHI* *PHI* *PHI*/*PHI*/*PHI*
David Mason, M.D. *320 Loretto Rd*, *Lebanon*, *KY 40033*	*PHI PHI*, M.D. *PHI PHI PHI*, *PHI*, *PHI PHI*
Dear Dr. *Mason*:	Dear Dr. *PHI*:
Your patient, *Eva Johns*, was in the Surgery Clinic today. As you know, she is a *79*-year-old woman with a history of osteoporosis. She underwent a surgery of her left foot on *Nov 20*, *2062*. She has done well since the procedure. She is now comes here complaining of pain in the left foot. She had an x-ray to rule out a fracture of the metatarsal bones. She explains the discomfort as pain and burning in the foot.	Your patient, *PHI PHI*, was in PHI Surgery PHI today. As you know, she is a *PHI*-year-old woman with a history of osteoporosis. She underwent a surgery of her left foot on *PHI PHI*, *PHI*. She has done well since the procedure. She is now comes here complaining of pain in the left foot. She had an x-ray to rule out a fracture of the metatarsal bones. She explains the discomfort as pain and burning in the foot.
I have discussed with **Eva** that her discomfort most likely is due to PN. She also experienced this discomfort on *December* of *2061*.	I have discussed with **Eva** that her discomfort most likely is due to PN. She also experienced this discomfort on *PHI* of *PHI*.
If I can be of further assistance in her care, do not hesitate to contact me.	If I can be of further assistance in her care, do not hesitate to contact me.
Sincerely,	Sincerely,
Robert Short, MD	*PHI PHI*, MD

Table 5.5: Example 1. Made-up example document intended to resemble the real i2b2 data showing both the original and the de-identified narrative. *Blue* tokens are correctly removed PHI tokens, underlined tokens are those which were unnecessarily removed, and the **red** tokens are leaked PHI tokens.

Original	De-identified
Name: *Sierra*, *Arsenio* **P**	PHI: *PHI*, *PHI* **P**
MRN: *35696235*	MRN: *PHI*
Date: *07*-*13*-*49*	Date: *PHI*-*PHI*-*PHI*
Address: *26 C Main Street*, *Cincinnati*, *Ohio*	Address: *PHI PHI PHI PHI*, *PHI*, *PHI*

Table 5.6: Example 2. Made-up example document intended to resemble the real i2b2 data showing both the original and the de-identified narrative. *Blue* tokens are correctly removed PHI tokens, underlined tokens are those which were unnecessarily removed, and the **red** tokens are leaked PHI tokens.

Original	De-identified
Record date: *2092*-*02*-*03*	Record date: *PHI*-*PHI*-*PHI*
Patient Name: *FRIEDMAN*, *JIM*; MRN: *5983265*	Patient PHI: *PHI*, *PHI*; PHI: *PHI*
Dictated at: *02*/*04*/*92* by *LISA B. Li*, M.D.	Dictated at: *PHI*/*PHI*/*PHI* by *PHI PHI*. *PHI*, M.D.
Mr. *Friedman* returns with a history of depression and chronic pain.	Mr. *PHI* returns with a history of depression and chronic pain.
Medications at Transfer: Cipramil 10 mg po qd Actiq, 200 mg p.n. Nicotine patch 14 mcg/d q 24 hrs	Medications at Transfer: Cipramil PHI mg po qd Actiq, 200 mg p.n. Nicotine patch PHI mcg/d q PHI hrs
Allergies: NKDA	Allergies: NKDA
PHYSICAL EXAMINATION: Pulse of 80, blood pressure 156/78, oxygen saturation 96%, and temperature is 97.9.	PHYSICAL EXAMINATION: Pulse of 80, blood pressure 156/PHI, oxygen saturation 96%, and temperature is PHI.PHI.
FARMER, *PAUL* M.D. D: *02*/*03*/*92* T: *02*/*03*/*92*	*PHI*, *PHI* PHI.D. PHI: *PHI*/*PHI*/*PHI* PHI: *PHI*/*PHI*/*PHI*

Table 5.7: Example 3. Made-up example document intended to resemble the real i2b2 data showing both the original and the de-identified narrative. *Blue* tokens are correctly removed PHI tokens, underlined tokens are those which were unnecessarily removed, and the **red** tokens are leaked PHI tokens.

5.2 VigiBase Data Set

This section contains the presentation of the results for the VigiBase data set. In Section 5.2.1, the general results are presented. In Section 5.2.2, we present the results per category. In Section 5.2.3, we give some examples of PHI which was missed by our method. In Section 5.2.4, we present the results from the evaluation of valuable information preserved in the de-identified VigiBase reports.

5.2.1 General Results

During the evaluation on the i2b2 validation set for model selection, we found that d-2) performed best, which is why it was used during the VigiBase evaluations rather than variation b-1) which achieved the best results (highest HIPAA-recall combined with acceptable precision) on the i2b2 test data. The general results of our method on the VigiBase test set are presented in Table 5.8. This table contains the results for model variation d-2) after fine-tuning on the VigiBase training data for 1 or 2 epochs and for model variation d-2)'s performance when applied without fine-tuning.

In all cases, we are using an adaptive-threshold rule with two pairs of thresholds, (low = 0.9, high = 0.95) and (low = 0.8, high = 0.9). The results are compared to the performance of the methods developed during a previous Master's thesis by Sahlström [54]. He used regular expression patterns (RegEx patterns), conditional random fields (CRF), and a support vector machine (SVM) to de-identify VigiBase data. In his project, he made multiple token annotations and evaluated different recalls. He checked whether his annotation overlapped with the gold standard annotation, if it was exactly the same or if it covered the gold standard annotation. For our comparison we chose his "covering-criteria" results. The precision for our method is slightly higher as presented here on all tokens than it is on only HIPAA-tokens since the data set also included some other PHI-categories such as ID numbers which we correctly masked as PHI. Thus, we present the overall PHI-precision instead of the HIPAA-precision, while Sahlström's results are based on only four categories, "Date", "Age", "Location", and "Organisation".

We can see that Sahlström's models were developed to have a high F1 score and therefore both a high recall and a high precision. Thus, his

	Recall	Precision
d-2) fine-tuned, 2 epochs, lower th	0.983	**0.570**
d-2) fine-tuned, 2 epochs, higher th	**0.990**	0.455
d-2) fine-tuned, 1 epoch, lower th	0.979	0.550
d-2) fine-tuned, 1 epoch, higher th	0.981	0.389
d-2) not fine-tuned, lower th	0.901	0.347
d-2) not fine-tuned, higher th	0.929	0.285
RegEx patterns [54]	0.881	**0.930**
CRF [54]	0.783	0.859
SVM [54]	**0.889**	0.905

Table 5.8: Token-based recall and precision on the VigiBase test set for d-2) both fine-tuned and not fine-tuned using an adaptive-threshold rule with two different threshold pairs: (low $= 0.9$, high $= 0.95$) and (low $= 0.8$, high $= 0.9$). This is compared with the results from the previous Master's thesis project by Sahlström [54] which used regular expression patterns (RegEx patterns), a conditional random field model (CRF), and a support vector machine (SVM). The highest recall and precision per project are highlighted.

methods outperform our method by far with respect to precision but with respect to recall our models are all superior. The highest recall, 99.0%, is gained by the variation that was fine-tuned for two epochs and which was using the higher threshold pair (low $= 0.9$, high $= 0.95$).

5.2.2 Results Per Category

In the following, we describe the results of the method per PHI category. The limited size of the VigiBase test set allows us to manually check all PHI tokens missed by our method. As the annotations in the gold standard were made in such a way that ages were annotated while including the words "years" and "old" or similar in the annotation, there are missed PHI tokens which include only the information that the person is a number of "years" old. We further found that some missed "Organisation" PHI tokens included the word "of", which we decided would not increase the probability of re-identification. If we correct the recall per category by removing these tokens from the list of missed PHI, we get the corrected recalls as presented in Table 5.9 and Table 5.10.

	2 epochs				1 epoch			
	low th		high th		low th		high th	
	R	R new	R	R new	R	R new	R	R new
Date	**0.998**	**0.998**	**0.998**	**0.998**	0.996	0.996	**0.998**	**0.998**
Age	0.917	**0.964**	**0.964**	**0.964**	0.905	**0.964**	0.905	**0.964**
Loc.	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Org.	0.75	**1.0**	0.75	**1.0**	0.75	**1.0**	0.75	**1.0**
All	0.983	0.993	0.990	0.993	0.979	0.991	0.981	0.993

Table 5.9: Token-based per category HIPAA-recall on the VigiBase test set for variation d-2) fine-tuned for 1 or 2 epochs and using an adaptive-threshold rule with a lower (low $= 0.8$, high $= 0.9$) and a higher (low $= 0.9$, high $= 0.95$) threshold pair. Here, recall is abbreviated as "R" and "R new" stands for the corrected recall.

We find that the fine-tuned model has very high recall for locations, organisations, and dates, while it performs slightly worse on ages. It seems that learning for more than one epoch does not decrease but might even increase the recall per category. It also increased the overall precision as we can see in Table 5.8. Fine-tuning increases the precision from 34.7% up to 55.0% for use of low threshold pairs and one epoch fine-tuning and even up to 57.0% after two epochs of fine-tuning. From the per category recall tables, we can learn that there are more corrections of missed PHI to make when the model was not fine-tuned on the data set. This means that the model has learned to mask "years" and "old" after fine-tuning but it also recognises more ages even after the recall was corrected. Thus, the model seems to learn something about the special form of VigiBase case narratives which

	low th		high th	
	R	R new	R	R new
Date	0.998	0.998	**1.0**	**1.0**
Age	0.333	0.929	0.512	**0.941**
Loc.	**1.0**	**1.0**	**1.0**	**1.0**
Org.	**1.0**	**1.0**	**1.0**	**1.0**
All	0.901	0.988	0.929	0.991

Table 5.10: Token-based per category HIPAA-recall on the VigiBase test set for variation d-2) not fine-tuned and using an adaptive-threshold rule with a lower, "low th" = (low $= 0.8$, high $= 0.9$), and a higher, "high th" = (low $= 0.9$, high $= 0.95$), threshold pair. Here, recall is abbreviated as "R" and "R new" stands for the corrected recall.

helps to recognise ages.

A comparison between our de-identification method and the ones using regular expressions, conditional random fields, and support vector machines is made for the per-category-recall. This comparison can be found in Table 5.11.

We can see that our model both after fine-tuning or when used "off-the-shelf" outperforms the regular expression, conditional random field, and support vector machine methods with respect to recall. Our models are better especially with respect to locations and organisations which the regular expressions and the conditional random field do not recognise at all. Even the support vector machine is not much better on these categories. It has a recall of only 22.2% on locations and does not recall any organisations. These models do however have a high precision on all categories: the lowest precision on a category is 84% for dates achieved by the conditional random field. Some of the models have a precision of 100% on the "Age" category and the lowest overall precision is 85.9%, while our model has an overall precision of at most 57.0% when using the mentioned combination methods.

	Fine-tuned	Non-fine-tuned	RegEx	CRF	SVM
Date	0.998	**1.0**	0.946	0.822	0.924
Age	**0.964**	0.941	0.800	0.833	0.889
Location	**1.0**	**1.0**	0.000	0.000	0.222
Organisation	**1.0**	**1.0**	0.000	0.000	0.000
All	0.993	0.991	0.881	0.783	0.889

Table 5.11: Token-based per category recall on the VigiBase test set for the best fine-tuned model with an adaptive-threshold-rule with a low threshold pair (low = 0.8, high = 0.9), for the best non-fine-tuned model with a high threshold pair (low = 0.9, high = 0.95), and for the models used by Sahlström [54]. Note that the corrected recall is used for our models. For our model the token-based recall is given while Sahlström's models were evaluated based on the PHI annotation instances.

5.2.3 Examples of Leaked Protected Health Information

Here, we list the leaked PHI tokens from the VigiBase test set after correction when using the fine-tuned variation d-2) after two epochs and with a low threshold pair. The highlighted token is the missed PHI token while the other tokens are the preceding and the subsequent ones:

- Age: "not", "reported", "At", **"4"**, "years", "of", "age"
- Age: "was", "experienced", "at", **"2"**, "years", "of", "age"
- Age: "reported", "to", "be", **"5"**, "years", "of", "age"
- Date: "in", "FU", "FU", **"7"**, "10", "2003", "A"

For patient privacy reasons, the ages were replaced by values in the same age group and the dates were replaced by a different date in the same format and with only a minor change of the year.

The missed "Age"-annotation tokens are easy to detect for a human but the model might not have seen anything similar to "years of age" during training. The date might have gone undetected because of the context "FU FU" which might be very different from the examples seen during training. "FU" stands for "Follow-Up" but this abbreviation does not seem to be commonly used.

The organisation tokens which were missed were twice the token "of" which followed the word "Hospital" and were removed by us.

5.2.4 Valuable Information for Causality Assessment

A pharmacist from the Uppsala Monitoring Centre assessed how much valuable information was left in the de-identified VigiBase case narratives. The case narratives were de-identified using the variation which was fine-tuned for 2 epochs and which was using the lower threshold. She found that 96% of the de-identified reports still contained some type of valuable information. Information included was for example the information on the reaction itself, the seriousness of the adverse reaction, the outcome of the adverse reaction, drug names, information on the medical history, doses, lab results, or information on the temporal relationship of the events. The de-identified narratives did however also miss information such as numbers for lab results, severity scales, or doses, as well as information in form of common words, and even drug names and information on time-to-onset (e.g., "seven days later"). However, 52% of the reports did not miss any valuable information for causality assessment. Drug and brand names, doses, and lab data and other measurements were the types of valuable information which were most often mistakenly removed.

Chapter 6

Discussion

The combination of a deep neural network with a rule-based method using an adaptive-threshold combination method achieved an performance improvement compared to the rule-based de-identifier in terms of recall and especially in terms of precision, as we can tell from the i2b2 test results. The hybrid method is also better than a deep de-identifier with a similarly high threshold (0.95) with respect to precision or with respect to recall when compared to a deep de-identifier with a lower threshold (0.9 or 0.8). This is logical since the rule-based de-identifier is very conservative and will recall many tokens as being possibly PHI so that we can adjust the threshold used on the deep de-identifier to a higher value which will lead to a more secure estimation. However, if we only use a high threshold on the deep de-identifier without using the rule-based de-identifier we see that we recall equally many tokens but that the precision decreases to values lower than for the hybrid method. Using an adaptive-threshold appears logical and gives good recall and precision. The recall is higher than the one for the rule-based de-identifier since the rule-based de-identifier will miss some PHI tokens when the PHI words are also common words or medical terms but not in any of the known PHI dictionaries. In these cases, the deep de-identifier can often recognise PHI by using the context around the token.

For the different variations of the model when using the adaptive-threshold, we could see that adding post-space characters could lead to a sufficiently high recall, larger than 99%, while improving the precision to values above 50% which is a significant improvement from the rule-based de-identifier and its precision of 11.7%. As we could see from the differences in performance for the separate runs of the same variation, the models' performance differs per run. This is because the learned model weights depend on their random initialisation and the random shuffling of training examples during stochastic gradient descent, and the convergence of the model weights to local minima during model training. In general, there is an uncertainty about the results due to the limited amount of training data as well as the fact that we might not have found the optimal weights yet. The convergence time might also differ for different numbers of inputs and for different weight initialisations, which means that we might be training too short for some of the models if we only train for a maximum of four epochs. We did however notice a drop in recall on the validation data after training for more than 3 epochs in most of the cases. One further needs to note that the thresholds were chosen to the arbitrary values of (low $= 0.8$, high $= 0.9$) for the lower pair and (low $= 0.9$, high $= 0.95$) for the higher pair which might work better for some models than for others and that there might even be more complex ways to better combine the two de-identifiers. As we trained two models we could show that there are significant differences between the runs but looking at two runs per variation helps support the observations described in Chapter 5.

It appears logical that adding post-space characters could improve the performance since it allows to recognise sentence structure more easily and since it helps to recognise specific character-symbol formats. It was surprising that it did not significantly improve the recall of dates. In this context, one has to discuss whether adding additional features to a deep network in general will increase its performance or whether it will prevent it from generalising. Using a pure feature learning approach would mean that the deep neural network is trained with pure data to learn any possible feature on its own. Of course, this is not simple in a natural language processing context. A suitable input format has to be chosen. We chose to tokenise the text. We represent the tokens by numbers which are then transformed to embedding vectors within the network. We also represent the tokens by a sequence of characters where each character is represented by a number which is transformed to an embedding vector by the network. This is of course already some kind of feature engineering. In addition to these necessary "features" one might wish to feed the network with more information which we know could be useful when interpreting natural language. One feature which can be added is a binary input indicating whether a token in the original text is lower case, beginning with a capital letter or fully capitalised. We also saw that the neural network might not learn to recognise an instance of a name even though the name-token is preceded by the token "Mr." or "Mrs.". One might believe that giving these additional features as input to the network, the network would improve in performance. We could even extend this to more features, e.g., all features of the rule-based de-identifier. This however raises the question whether giving additional input will actually help the network to learn to predict difficult cases or whether it will prevent the network from generalising to more general features for the tokens. Our network did not improve its performance in terms of recall and only possibly and slightly in terms of precision when the additional feature input was added. The network did not show any signs of increased recall of names but possibly an increase in precision in general. This might mean that the network classifies fewer tokens mistakenly as PHI that it previously mistakenly classified as names. Lee et al. [32] who also used additional feature input reported no improvement through non-database feature-augmentation. They even experienced a decrease in recall on some PHI categories when using hand-engineered features based on for example morphological features or regular expressions, at least when used in combination with features based on a known-PHI database. The authors suggest that the reason for a drop in recall is that features engineered by humans tend to have a higher precision than recall. They further argue that the hand-engineered features might lead to the model losing its ability to recognise tokens which do not conform with the human-engineered features. When adding features to the input, one thus has to evaluate its benefits carefully. If it for example increases the recall of patient names while decreasing the recall of other, less sensitive, categories, this might be desirable. One might want to increase the recall on names even when this means that there is a drop in the recall of organisations. The use of these additional inputs should therefore be evaluated more carefully before a decision can be made regarding whether they are useful or not. For this, one should look at each category and decide what is a good trade-off between per-category recall and overall precision for this category.

Our method improved the recall achieved by the deep learning de-identification method by Dernoncourt et al. [12]. It does however not reach a precision close to the one achieved by Dernoncourt et al. This is probably due to the fact that we do not classify using the class with maximum probability but by checking whether a class is safe using a threshold. This way we can improve the recall at the expense of the precision. The higher precision of Dernoncourt et al. might be achieved due to their use of an additional conditional random field layer on top of the network's softmax layer. Their classification is based on the label returned by the conditional random field layer. One can very well imagine that the conditional random field layer will increase the recall and the precision of PHI for example in the case of dates where often three numbers follow each other. Dernoncourt et al. also mention ZIP codes followed by city names as an example where certain labels are more likely to follow each other. The network could in this case decide that the number is a date if it knew that the previous or following token is also likely to be a date. In their ablation study, the authors showed that they could increase the F1 score of the model from approximately 97% to approximately 98%. Adding a conditional random field layer to our model would be interesting but is problematic since the conditional random field layer would change the output from being a vector of probabilities to a choice of class. Our model does however use the probabilities to make a decision about the safety of a token. The combination method as it currently exists depends on the probability output. One could, during future work, try to explore the possibilities of combining a network with a conditional random field layer with our rule-based de-identifier.

Dernoncourt et al. [12] could improve their recall to 97.8% when using an independent conditional random field in combination with their neural network model which also includes a conditional random field layer. This suggests that our method might even improve by combining results not only from the rule-based and the deep de-identfier, but also from an independent conditional random field de-identifier. This use of a separate conditional random field model should be explored during future research.

Our method manages to remove many of the PHI-categories completely from the i2b2 test set. These categories are "Street", "Phone", "City", "ZIP", "Device ID", "Fax", "Email". Many of these follow a certain pattern or will appear in a similar context in all records and are therefore easy to detect. Our method performs worst on the category "Organisation" which is probably because it is less clearly defined and can only be recognised by the appearance of certain words such as “University” or “Association” and by the context. The performance on names is also with 97.7% in the best adaptive-rule variation not enough considering the fact that these tokens might help to directly identify a person. Fortunately, some of the missed “Person”-PHI are only initials which will only slightly increase the probability for re-identification of the person. There are some last names that have leaked, mainly those which are common names but also some known names which are not common words or medical terms. “Patient” appears to be a category which the network struggles the most not to identify as safe with a high probability. This might also be because names appear in many different places in the report and in different contexts. The use of a named entity recognition method which can be trained to recognise which tokens are persons might therefore help improve the method. This could either be added to the rule-based de-identifier, as an input to the deep de-identifier, or the deep de-identifier could be pre-trained for a more general named entity recognition task. It is further surprising that the method does not recognise ages even when they are followed by “years old”. It is probably harder to recognise numbers as PHI since many numbers are also safe. Thus, tokens which themselves look the same, might be PHI as in an age or are safe as in a dose or a pulse. It might be interesting to see whether the performance of the method on ages improves if more training data is used. Dernoncourt et al.’s method performed worst on the “Location” PHI categories. Our hybrid, inverse approach has a recall of 100% for the locations “Street”, “City” and “ZIP”, but only 88.4% on “Organisation” (for variation b-2). We have not evaluated the recall of the non-HIPAA location categories. Our “Location” recall might thus be similarly low as theirs even though it appears to be good when only HIPAA categories are considered.

It is not possible to compare the results from our de-identification with other inverse de-identification methods because neither Berman’s system [7], nor Ferrándes et al.’s system BoB [17] were evaluated on the 2014-i2b2 data set.

On the VigiBase data, our method performs well with respect to recall with values of up to 99.0% and it can even reach a precision of more than 55%. The method benefits significantly from fine-tuning the neural network, especially in terms of precision. The case narratives from VigiBase are different in their structure and language and they are shorter. Therefore, it appears logical that the method can benefit from fine-tuning its weights. Fine-tuning for two epochs does not seem to have led to any overfitting so it might be useful to try to fine-tune the neural network for even more epochs in the future. It is interesting to see how the neural network could adjust to the different annotation scheme of the VigiBase data. The model learned to also mask "years" and "old" in age annotations as these were annotated in the VigiBase gold standard but not in the i2b2 data set. Fine-tuning helped to recognise more "Age" annotations than without fine-tuning. This is not just due to the network learning how to recognise "years" and "old" but it must be learning something more about the structure of the sentences or the way ages are described. The ages which were not recognised by the network were below ten and were followed by "years of age". This could be due to the fact that the neural network might not have seen similar examples during the pre-training on the i2b2 data. The model could maybe learn how to recall these by seeing more examples during another epoch of training or by adding synthetic data of this form to the training set. It is interesting that the performance on locations did not suffer after fine-tuning although all locations were labelled as "City" even if they actually were streets or other locations.

The de-identification methods which were developed by Sahlström did perform significantly better with respect to precision. Our inverse hybrid method achieved a precision of 55% on the VigiBase data while for example the regular expression method by Sahlström had a precision of 93%. This is however also the case because Sahlström's methods do not attempt or succeed to recognise locations and organisations at all or do so only with a very low recall. Thus, our method outperforms these methods with respect to recall on all categories and on some categories with a large difference. Our method struggles mostly with recognising ages which might be due to the previously described difference in formulations.

As the test set does not contain any "Name" samples, it is not reasonable to make any judgement whether the method is leading to a safe de-identification of VigiBase case narratives. Names are in particular sensitive as they can directly lead to the re-identification of the report. In order to develop a good method, training data with more name instances than in the current training set would be required. Furthermore, examples with names would also need to be present in the test set in order to judge the performance of the de-identification method. This does also apply to other PHI categories such as ID numbers and contact information. In general, more VigiBase data with more, different PHI categories is required both to train and to test the method.

The missed organisation annotation tokens, "of" following after "Hospital", could have been detected if a conditional random field in addition to the network's output was used. The model could thus have recognised that the word hospital is PHI and thereby concluded that also the following "of" is PHI. This might also have helped to recognise the missed date annotation token since the model might conclude that the date is missing the number for the date in front of the month and the year.

One problem when transferring the model from i2b2 to VigiBase data might be that, in the i2b2 data set, dates were shifted by several years into the future while VigiBase reports are from the past, thus differing in number by 50 to 100 years. This could hinder the non-fine-tuned model from recognising dates and this could even not be learned fully after fine-tuning for 2 epochs.

Our method was designed to remove more classes than specified by the HIPAA standard by training the deep de-identifier to recognise these while the rule-based de-identifier was designed for HIPAA categories only. We assume that using all classes during training will help the deep network to better understand the text structure and to learn better features. This was however not evaluated during our research. It would be interesting to explore during future research whether the deep de-identifier will perform better if the network is trained on the binary classes "safe" and "PHI" or on hyper-classes such as "Location" and "Person" (instead of "City", "Street", . . . , and "Patient" and "Doctor"). The fact that we developed the network in such a way that it takes away more than required by HIPAA makes it however even safer and allows for an easier application of the method to other countries such as EU member states. The EU regulation states that all information which could lead to the identification of the data subject should be removed. In order to fulfil this requirement, it is likely that many situations will require the method to remove more information than defined in the HIPAA definition. As discussed by the authors of the 2014-i2b2 challenge data set [60], this is especially important in cases where the attacker could use information from several reports of the same patient by combining information from the different reports. These longitudinal reports are however less common in VigiBase.

This research project which developed an inverse, hybrid method was a first evaluation of the possibility to approach the problem of de-identification in this inverse way and by using this kind of hybrid method. Both the rule-based de-identifier and the deep de-identifier as well as the combination methods have room for improvements. The rule-based de-identifier could benefit from filtering the common word list in order to create a common word list with unambiguous words which are not possibly names, dates or other PHI categories, as done by Neamatullah et al. [45]. The rule-based de-identifier could also use a named entity recognition tool, as previously mentioned, to identify persons and locations and other named entities. The dictionaries could in general be improved. For example, the rule-based de-identifier currently only has a list of common names which is based on the last names from the U.S. census. These dictionaries could however easily be extended and improved.

The deep de-identifier could possibly be improved by training on data which is annotated with hyper-categories of PHI or with binary classes (safe or PHI). Adding more training data is also likely to improve the results. This could be more reports of adverse drug reactions from national centres or data produced by the use of data augmentation techniques similar to the ones suggested by Lin et al. [35]. Even the use of a different tokeniser could potentially change the performance of the method. The presented results were not based on an extensive search of optimal optimisers and their hyperparameters due to the limited time available. The limited time was also the reason why we did not explore the effect of training for more epochs. All this could be evaluated during future work.

The combination method of the model could benefit from an extensive search for the best threshold pair. It might also be useful to combine the results of the rule-based and deep de-identifier with the results from the regular expressions from Sahlström's thesis [54]. The impact of combining the deep de-identifier's and the rule-based de-identifier's output using a machine learning technique such as decision trees[1] or support vector machines would also be an interesting

[1] A decision tree is a machine learning model in which a tree of binary questions about the data is built which can be followed down to the leaves per data point in order to find the class to which it belongs in the leaf of the tree.

focus for future work.

Names are an interesting case of PHI since they tend to be repeated several times. It might be useful to develop a method to recognise if a word which possibly is a name is repeated several times and at least once in combination with a salutation so that we can be confident that it is a name.

In de-identification of medical case narratives for adverse drug reaction research, dates as PHI play a special role. This is due to the importance of dates in causality assessment in which the time interval between the occurrence of certain events can be important. For example, it can be useful to know how much time passed between the introduction of a drug (onset) and the reported symptoms in order to draw conclusions about the causality. This time interval is called time-to-onset [33]. Therefore, time information should not be removed like other PHI but should instead be replaced by an onset time and the time-to-onset for following events [45]. This is an important task during the future development of an inverse method which should be applicable in practice. It might even be desirable to try to use the output of the rule-based and deep de-identifier to assign place holders other than PHI to the PHI tokens when the method suspects what kind of category is present. This might already be possible with this method by using the probabilities of the PHI categories as well as the features returned by the rule-based de-identifier. This task was however outside of the scope of this thesis project.

In order to get a good estimate for the level of reliability that the method provides, one should, during future development, evaluate the performance of the method with respect to the risk for re-identification. This could be based on current research such as the work by Scaiano et al. [56].

With recall on both data sets of close to or even above 99%, a very reliable method was constructed. The qualitative analysis by an expert has further shown that the method managed to preserve a lot of valuable information for causality assessment. Nevertheless, one needs to consider that a recall of 99% of the HIPAA PHI tokens does not mean that all direct identifiers (those which could be used to directly identify a person) were removed. Thus, even though a recall close to 100% was reached one has to consider that this does not mean that the probability for re-identification is 0. Furthermore, we need to realise that even though there is valuable information contained, there is still some valuable information which is mistakenly removed. This is probably unavoidable when we aim for an extremely high recall but we are convinced that some improvements with respect to the precision could be made, for example by adding a conditional random field to the method. With respect to the precision, one also has to consider that only around 7% of the tokens in a text are PHI (a result obtained when tokenising with the i2b2 tokeniser and when using a random split of 90% of the training data). This means that even though a precision of only 51% appears low, this does not mean that the total amount of mistakenly removed tokens is very large. Out of all tokens (PHI and safe tokens) only 13.6% of the tokens were safe tokens which were mistakenly removed by variation a-2), which was the one with the lowest precision.

Pre-training the method's neural network on the i2b2 data and fine-tuning it on VigiBase data helps to overcome the problem of limited VigiBase training data being available. With more VigiBase training data, we are likely to improve the results for the already considered categories as well as achieve similarly high recall as on the i2b2 data for categories which are not included in the VigiBase test set so far. Thus, even though the method is not yet ready to be applied in practice, it was shown to be both very reliable and reasonably precise. After some improvements have been made during future research using more data from adverse drug reaction reports, the method could most likely be used in practice. The future work could be carried out in co-operation with national centres in order to test the method in a real life setting.

Chapter 7

Conclusion

From the evaluations of our inverse, hybrid de-identification method on both the 2014-i2b2 data set as well as the VigiBase data set, we can see that it is possible to effectively de-identify case narratives in reports of suspected adverse reactions. Combining both a rule-based approach with a deep-learning approach leads to an improved recall as compared to other methods while still achieving acceptable precision. On the 2014-i2b2 test set, our method achieves a recall of protected health information of 99.1% and a precision of more than 51%. On the VigiBase test set, our method could, after fine-tuning, recall more than 99% of the protected health information. The precision corresponding to 99% recall on this data was 55% but 96% of the narratives contained some kind of valuable information and 52% of the narratives did not miss any valuable information at all.

With more VigiBase data containing annotations for all categories of available protected health information and after some improvements during future work, this method should be reliable and precise enough to be used in practice.

Bibliography

[1] *45 CRF § 164.514: other requirements relating to uses and disclosures of protected health information*. (Last checked: 30/01/2017). URL: `https://www.law.cornell.edu/cfr/text/45/164.514`.

[2] Martín Abadi et al. *TensorFlow: large-scale machine learning on heterogeneous systems*. Software available from `https://www.tensorflow.org`. 2015.

[3] Charu C Aggarwal and Philip S Yu, eds. *Privacy-preserving data mining*. Springer Science+Business Media, LLC, 2008.

[4] Hossein Azizpour et al. "From generic to specific deep representations for visual recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2015, pp. 36–45.

[5] Yoshua Bengio et al. "A neural probabilistic language model". In: *JMLR* 3.Feb (2003), pp. 1137–1155.

[6] Yoshua Bengio et al. "Greedy layer-wise training of deep networks". In: *Advances in Neural Information Processing Systems* 19 (2007), pp. 153–160.

[7] Jules J Berman. "Concept-match medical data scrubbing: how pathology text can be used in research". In: *Arch Pathol Lab Med* 127.6 (2003), pp. 680–686.

[8] Christopher M Bishop. *Pattern recognition and machine learning (information science and statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[9] Olivier Bodenreider. "The unified medical language system (UMLS): integrating biomedical terminology". In: *Nucleic Acids Research* 32.Database issue (2004), pp. D267–D270.

[10] François Chollet et al. *Keras*. Software available from `https://github.com/fchollet/keras`. 2015.

[11] Ronan Collobert et al. “Natural language processing (almost) from scratch”. In: *JMLR* 12.Aug (2011), pp. 2493–2537.

[12] Franck Dernoncourt et al. “De-identification of patient notes with recurrent neural networks”. In: *JAMIA* (2016).

[13] I. Ralph Edwards and Jeffrey K. Aronson. “Adverse drug reactions: definitions, diagnosis, and management”. In: *The Lancet* 356.9237 (2000), pp. 1255–1259.

[14] I Ralph Edwards et al. “Quality criteria for early signals of possible adverse drug reactions”. In: *The Lancet* 336.8708 (1990), pp. 156–158.

[15] Dumitru Erhan et al. “Why does unsupervised pre-training help deep learning?” In: *JMLR* 11.Feb (2010), pp. 625–660.

[16] Laurene Fausett. *Fundamentals of neural networks: architectures, algorithms and applications*. Prentice-Hall, 1994.

[17] Oscar Ferrández et al. “BoB, a best-of-breed automated text de-identification system for VHA clinical documents”. In: *JAMIA* 20.1 (2013), pp. 77–83.

[18] Michele Filannino and Goran Nenadic. “Temporal expression extraction with extensive feature type selection and a posteriori label adjustment”. In: *Data & Knowledge Engineering* 100.Part A (2015), pp. 19–33.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.

[20] Alex Graves. “Supervised sequence labelling with recurrent neural networks”. PhD thesis. Technical University of Munich, 2008.

[21] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. “Speech recognition with deep recurrent neural networks”. In: *Proceedings of the IEEE international conference on acoustics, speech and signal*. IEEE. 2013, pp. 6645–6649.

[22] Austin Bradford Hill. “The environment and disease: association or causation?” In: *Proceedings of the Royal Society of Medicine* 58.5 (1965), pp. 295–300.

[23] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[24] Alexandra Hoegberg. "Down memory lane: four decades of building drug safety". In: *Uppsala Reports* 75 (2017), pp. 9–13.

[25] *Information technology (reasonable security practices and procedures and sensitive personal data or information) rules, 2011*. (Last checked: 01/06/2017). 2011. URL: `http://meity.gov.in/writereaddata/files/GSR313E_10511%281%29_0.pdf`.

[26] Alistair E W Johnson et al. "MIMIC-III, a freely accessible critical care database". In: *Scientific data* 3 (2016).

[27] Ghazaleh Karimi et al. "Clinical stories are necessary for drug safety". In: *Clin Med* 14.3 (2014), pp. 326–327.

[28] Diederik P Kingma and Jimmy Ba. "Adam: a method for stochastic optimization". In: *CoRR* (2014).

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[30] Guillaume Lample et al. "Neural architectures for named entity recognition". In: *CoRR* (2016).

[31] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444.

[32] Ji Young Lee et al. "Feature-augmented neural networks for patient note de-identification". In: *CoRR* (2016).

[33] Fanny Leroy et al. "Estimating time-to-onset of adverse drug reactions from spontaneous reporting databases". In: *BMC Med Res Methodol* 14.1 (2014), p. 17.

[34] Kun Li et al. "Learning to recognize protected health information in electronic health records with recurrent neural network". In: *International conference on computer processing of oriental languages*. Springer International Publishing AG. 2016, pp. 575–582.

[35] Chen Lin et al. "Improving temporal relation extraction with training instance augmentation". In: *ACL: proceedings of the 15th workshop on biomedical natural language processing*. 2016, pp. 108–113.

[36] Marie Lindquist. "Vigibase, the WHO global ICSR database system: basic facts". In: *DIJ* 42.5 (2008), pp. 409–419.

[37] Ronald H B Meyboom et al. "Causal or casual?" In: *Drug Safety* 17.6 (1997), pp. 374–389.

[38] Stephane M Meystre et al. "Automatic de-identification of textual documents in the electronic health record: a review of recent research". In: *BMC Med Res Methodol* 10.1 (2010), p. 70.

[39] Tomas Mikolov et al. "Efficient estimation of word representations in vector space". In: *CoRR* (2013).

[40] Tomas Mikolov et al. "Recurrent neural network based language model". In: *Interspeech 2010*. 2010, pp. 1045–1048.

[41] George A Miller. "WordNet: a lexical database for english". In: *Commun ACM* 38.11 (Nov. 1995), pp. 39–41.

[42] T M Mitchell. *Machine learning*. International ed., [Reprint.] McGraw-Hill series in computer science. New York, NY: McGraw-Hill, 2010.

[43] Patricia Mozzicato. "MedDRA: an overview of the medical dictionary for regulatory activities". In: *Pharm Med* 23.2 (2009), pp. 65–75.

[44] Kevin P Murphy. *Machine learning: a probabilistic perspective*. Cambridge, Massachusetts, London, England: The MIT Press, 2012.

[45] Ishna Neamatullah et al. "Automated de-identification of free-text medical records". In: *BMC Med Inform Decis Mak* 8.1 (2008), p. 32.

[46] World Health Organization et al. *The importance of pharmacovigilance*. 2002.

[47] Shanthi N Pal et al. "WHO strategy for collecting safety data in public health programmes: complementing spontaneous reporting systems". In: *Drug Safety* 36.2 (2013), pp. 75–81.

[48] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: global vectors for word representation". In: *Empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[49] Japan Personal Information Protection Commission. *Amended act on the protection of personal information*. (Last checked: 01/06/2017). 2016. URL: `https://www.ppc.go.jp/files/pdf/Act_on_the_Protection_of_Personal_Infomration.pdf`.

[50] *Regulation (EU) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/EC (general data protection regulation)*. (Last checked: 21/04/2017). URL: `http://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX%3A32016R0679`.

[51] Raúl Rojas. *Neural networks: a systematic introduction*. Springer International Publishing AG, 2013.

[52] O Russakovsky et al. "ImageNet large scale visual recognition challenge". In: *International Journal of Computer Vision (IJCV)* (2015).

[53] M Saeed et al. "MIMIC II: a massive temporal icu patient database to support research in intelligent patient monitoring". In: *Computers in cardiology*. Sept. 2002, pp. 641–644.

[54] Jakob Sahlström. "Automatic de-identification of case narratives from spontaneous reports in vigibase". MA thesis. Uppsala University, 2015.

[55] Ruth Savage. *The logic of causality (part 1)*. (Last checked: 29/06/2017). 2013. URL: `https://media.medfarm.uu.se/play/kanal/142/video/3523`.

[56] Martin Scaiano et al. "A unified framework for evaluating the risk of re-identification of text de-identification tools". In: *J Biomed Inform* 63 (2016), pp. 174–183.

[57] Ali Sharif Razavian et al. "CNN features off-the-shelf: an astounding baseline for recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. June 2014, pp. 806–813.

[58] Asif Ekbal Shweta, Sriparna Saha, and Pushpak Bhattacharyya. "Deep learning architecture for patient data de-identification in clinical records". In: *Proceedings of the clinical natural language processing workshop*. 2016, pp. 32–41.

[59] Republic of South Africa. *Act no.4 of 2013: protection of personal information act, 2013*. (Last checked: 01/06/2017). URL: http://www.justice.gov.za/legislation/acts/2013-004.pdf.

[60] Amber Stubbs and Özlem Uzuner. "Annotating longitudinal clinical narratives for de-identification: the 2014 i2b2/uthealth corpus". In: *J biomed inform* 58, Supplement (2015). Proceedings of the 2014 i2b2/UTHealth Shared-Tasks and Workshop on Challenges in Natural Language Processing for Clinical Data, S20–S29.

[61] *Summary of the HIPAA privacy rule*. (Last checked: 30/01/2017). URL: https://www.hhs.gov/hipaa/for-professionals/privacy/laws-regulations/index.html.

[62] Charles Sutton and Andrew McCallum. "An introduction to conditional random fields". In: *Foundations and Trends® in Machine Learning* 4.4 (2012), pp. 267–373.

[63] *The information technology act, 2000*. (Last checked: 01/06/2017). URL: http://meity.gov.in/writereaddata/files/The%20Information%20Technology%20Act%2C%202000%283%29.pdf.

[64] Uppsala Monitoring Centre (UMC). *Vigibase*. (Last checked: 14/02/2017). URL: https://www.who-umc.org/vigibase/vigibase/.

[65] Özlem Uzuner, Yuan Luo, and Peter Szolovits. "Evaluating the state-of-the-art in automatic de-identification". In: *JAMIA* 14.5 (2007), pp. 550–563.

[66] Jan P Vandenbroucke. "Observational research, randomised trials, and two views of medical science". In: *PLOS Medicine* 5.3 (Mar. 2008), pp. 339–343.

[67] CIOMS Working Group VIII. *Practical aspects of signal detection in pharmacovigilance*. Tech. rep. 2010.

[68] Hui Yang and Jonathan M Garibaldi. "Automatic detection of protected health information from clinic narratives". In: *J Biomed Inform* 58, Supplement (2015). Proceedings of the 2014 i2b2/UTHealth Shared-Tasks and Workshop on Challenges in Natural Language Processing for Clinical Data, S30–S38.

[69] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. "Recurrent neural network regularization". In: *CoRR* (2014).