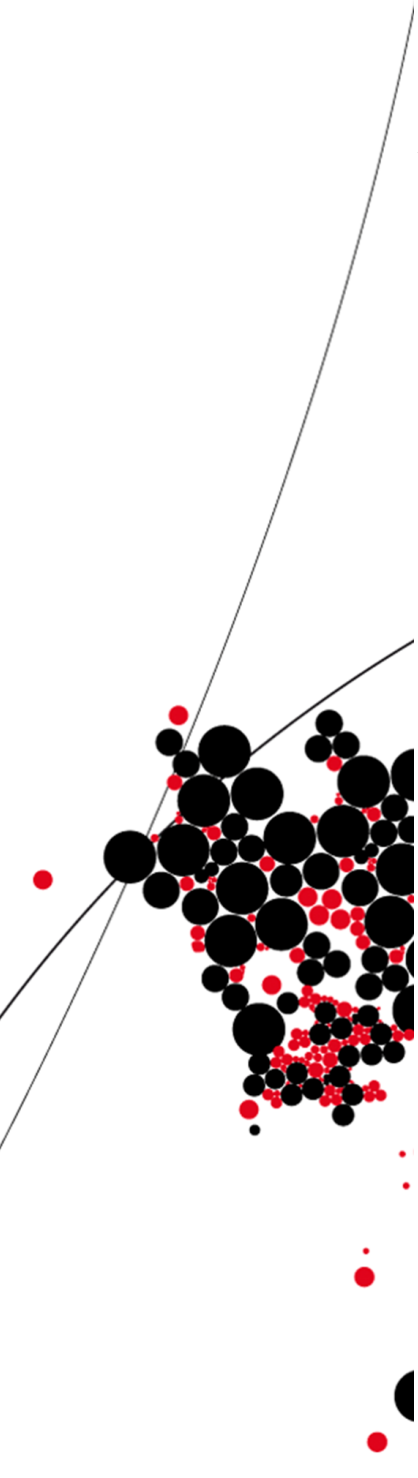
UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

Automatic Registration of Clinical Audits for Head and Neck Oncology at MST

Wybren Kortstra

M.Sc. Thesis
January 2020



Supervisors:
dr. ir. M. van Keulen
prof. dr. ir. A. Rensink
B. Kolenaar MD, DDS

Formal Methods and Tools
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

Clinical audits are used to analyze the quality of health care and improve treatment. Dutch Head and Neck Audit (DHNA) is an audit form that contains around 180 items. The registration form is currently filled manually which is a time consuming process. Most data in Electronic Health Record (EHR) systems is stored as plain text. This research aims to automate value extraction for items in this audit form. We proposed a solution that uses natural language processing to analyze the plain text and extract the There are two types of items that need to be registered: categorical and continuous. For categorical items we proposed classification methods that use medical text documents. We used different types of preprocessing to zoom in on relevant data to improve the classification results. For the continuous items we proposed a technique which adds labels to words in medical text documents. We found that classification without preprocessing scores higher than classification with the preprocessing, but when looking at the features that are most important to this score we found no relevant features. The labeling technique performed very well on the text and extracting the values for the continuous items was very successful as a result of that. Even though the methods for extracting information from the EHR are not perfect they can aid doctors in registering the patient information for DHNA.

Contents

List of acronyms

EHR Electronic Health Record

DHNA Dutch Head and Neck Audit

DICA Dutch Institute for Clinical Auditing

FAIR Findable Accesible Interoperable Reusable

IKNL Integraal Kankercentrum Nederland (Integral Cancer Centre Netherlands)

LR Logistic Regression

NB Naive Bayes

NLP Natural Language Processing

MRDM Medical Research Data Manager

MST Medical Spectrum Twente

NVRO Nederlandse Vereniging voor Radiotherapeutische Oncologie (Dutch Association for Radiotherapeutic Oncology)

RadboudUMC Radboud University Medical Center

RANK Registratie Applicatie Nederlandse Kankerregistratie (Registration Application Dutch Cancer Registration)

ZN Zorgverzekeraars Nederland (Health insurers Netherlands)

Chapter 1

Introduction

Hospitals see hundreds of patients every day and each patient has their own medical history. This history involves previous appointments with doctors, medication, diagnosis, treatments and other health related records. All this information is stored digitally. Digital records can be looked up more easily than paper records, can be shared more easily between physicians than paper records and are less prone to errors due to typing instead of writing with a bad handwriting. A system for keeping digital records is called an Electronic Health Record (EHR). EHR first started in the sixties with the Problem Oriented Medical Record [1] and has since then evolved. The introduction of the internet made it possible for EHR's to share medical records and by 2000 standardization of data formats had made its way. Health Level Seven (HL7), founded in 1987, developed standards for storing and sharing medical data with Clinical Document Architecture Release 1 (CDA R1) as one of the best early examples [2] [3]. Currently EHR systems, which have an adoption rate of over 90%[1], are almost everywhere and both patient and doctor benefit from them. HiX and Epic[2] are currently the largest players in the Netherlands.

Hospitals in the Netherlands have an obligation to report all procedures to the Dutch Institute for Clinical Auditing (DICA). DICA is an organisation that helps to ensure and improve the quality of care and to save costs. They do this by collecting and analysing data of all invasive surgeries and interventions in hospitals. A complete list of all medical specialisms that have to be reported can be found in appendix A. The clinical auditing forms of DICA have to be filled with information about the medical process of the patient and all this information can be found in the hospital's EHR. Audits are used to verify the care a patient has received and improve on the care process. The care process is everything a patient goes through in the hospital. This starts with the take in, then the diagnoses, treatment, recovery and ends with discharge.

Medical Spectrum Twente (MST), a large hospital in the eastern part of the Netherlands, was an early adopter of digitizing patient records and developed one of the first EHR systems. MST originated from the merge of hospitals in Enschede and Oldenzaal and is now the largest hospital in the eastern part of the Netherlands with over 100.000 patients every year[3]. MST has, as a result of the early adoption, different systems for different departments. These systems cannot easily interact with each other and physicians have a hard time finding patient data. For instance, if a patient has been referred by another physician, then information of that earlier visit is difficult to find for the physician the patient is referred to. Therefore, physicians have to send an email with the history of the patient. This makes it very difficult to find the required information for DICA.

The process of registering the patient data in the audit is a time and money consuming process. Currently filling out these audit forms is a manual process: someone has to open the EHR, look up the patient, find the relevant procedure (patients could have been in the hospital for other procedures) and extract the data. For the department of dental surgery DICA has a clinical audit form that is called Dutch Head and Neck Audit (DHNA). This audit focuses on head and neck oncology, tumors in the head and neck area. DHNA contains 180 fields that can be filled out. This audit is registered by an external party called Integraal Kankercentrum Nederland (Integral Cancer Centre

[1] https://ehrintelligence.com/news/outpatient-ehr-adoption-reaches-92-nears-market-saturation
[2] https://www.zorgvisie.nl/content/uploads/sites/2/2018/04/Epd-overzicht2018.pdf
[3] https://www.mst.nl/storage_static/2017/02/mst-1-1.pdf

(a) Input fields for structured tumor classification

(b) Unstructured tumor classification in a text document

Figure 1.1:

Netherlands) (IKNL) which we will introduce in more detail later. IKNL is paid by the hospital to register these audits. It takes about 90 minutes per patient to find and register the correct data. Automating the registration process could save money, time and prevent human error.

1.1 Problem statement

Doctors have a lot of freedom in registering patient information in the EHR. The EHR used at MST has been developed a long time ago and over time more forms have been added. The absence of a uniform way of registering patient information creates the following problems when registering the patient information for the audit:

1. Difficulty to extract the correct information for the audit
2. Different writing styles between physicians making it hard to identify the right information
3. Different people registering the audit have different judgements on subjective items

We will explain each of the problems in more detail in the paragraphs below.

The first problem comes from the fact that the EHR at MST offers multiple options for physicians to register patient information. There are forms where detailed information about a patient's health, disease and progress can be registered in free-text format. There are also forms that offer a much more structured input for all this information using multiple text fields, checkboxes and drop-down selects. There is also the option to create a text document and enter all the information in there. Physicians can create a letter or document and enter their findings, patient's disease and progress in there. An example that the same information can be registered in different ways can be found in figure 1.1 where we see registration of the classification of a tumor. Figure 1.1a has a structured approach where cT, cN and cM have their own fields, whereas 1.1b has the classification of all three in one line of text. The different locations where the person registering the audit can find the information makes it difficult to find all information.

The second problem is due to the variety of different physicians. The dental surgery department at MST has 7 physicians, aging from 30 to 60, who see patients and report about them in the EHR. The physicians differ in approach when it comes to registering patient information in the EHR. As mentioned in the previous problem, there are multiple options for registering patient information. A physician that is used to one of the methods of registering patient information might not be willing to adopt a different method. However, this only describes part of the problem, that physicians have

different methods of registering their patient information. Even when physicians would all write text documents, they still have different ways of writing. Just think about the different formulations of writing that a patient quit smoking after having smoked for 5 years.

The last problem is due to opinions of people. The audit contains multiple items that require some form of a judgement call. When does a patient have comorbidity? Comorbidity is the presence of one or more medical conditions besides the initial condition being cancer in the head or neck are for the DHNA. One could argue that having a cold is a comorbidity, but maybe it has no effect on the tumor growth or danger. Experts from IKNL have to make these judgements for each audit.

1.2 Approach

Of the problems described in the previous section we decided to take on the second and third. The first problem we tackle by only using text documents as our source of information. Most patients have a text document containing a lot of the information required for the DHNA. As we will read in section 2.1 information that is registered in the EHR is often also registered in a text document. So we can use text documents from almost all patients even though they have different physicians. To extract information from a text we must understand the structure of the text and its meaning. According to literature natural language processing can be used for this [4]: "Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications." In sections 2.3 to 2.6 we discuss some common techniques we could use.

In preparation of the experiments we collect all patient information from the EHR we are interested in. As we are focusing on the DHNA we only collect information of patients from dental surgery with tumor disorder. We also collect the DHNA audits of the past 3 years. These audits will help to train our Natural Language Processing (NLP) model and validate our results. Similar text documents can also be found for patients with other diseases, which allows for this research to be used for other audits of DICA.

DICA items can be split in categorical and continuous items. Categorical items have a fixed number of possible values, whereas continuous items have infinite possibilities. An example of a categorical item is about a patient's smoking behavior. Values for this item could be: yes, never, quit or unknown. An example of a continuous item is the (exact) length of a patient for which the value is a number. This gives us the task of classifying the value for categorical items and extracting a piece of the text for the continuous items. We will first attempt the classification and extraction on all text in the document. However, we think that performance of the analysis could be improved focusing on specific parts of the text.

We do preprocessing on two levels. With each level we zoom further in on the available data. With the first level of preprocessing we split on headers that mark the beginning of a specific paragraph and use the text of that paragraph. For instance a paragraph with the header 'intoxications' contains information about a patient's smoking and drinking behavior. The second level of preprocessing is labeling data and marking the meaning of the words. That way we can pick out only the words that are related to the drinking behavior of the patient.

1.3 Research questions

The main research question is as follows:

\How well can we extract patient information from natural texts of the EHR with the goal to automatically fill the audit forms of DHNA?"

The following sub-research questions are derived from the main research question:

Sub-question 1: *\How can we best determine the value for a categorical field?"*
Fields in the audit that are categorical, like whether a patient is a current smoker, ex-smoker or non-smoker, are determined by classification using Naive Bayes (NB) and Logistic Regression (LR). Different types of preprocessing are used to improve the results such as regular expressions and labeling of text.

Sub-question 2: *\How can we best determine the value of a continuous field?"*
Values for fields in the audit that require a value, like the length or weight of a patient or the stadium of a tumor, are extracted using automatic labeling.

1.4 Validation

In order to decide the success of the extraction methods we need to check the extracted information against a source of truth. There are audit records from previous years we can use to check how well the extraction process performs. We use records from 2015 to mid 2018 with data from about 180 patients. More information on the dataset can be found in section 5.2. There are also doctors from the MST involved that help to validate the extracted data. In the case where the extraction process has a different outcome we consult a doctor, who is an expert, and determine the error.

1.5 Report organization

Chapter 2 explains the problem in the hospital, describing the current situation and burden of manually registering the clinical audit. It also introduces the required background information on the techniques applied in this research. In chapter 3 we will discuss initiatives to improve automatic registration and other related work. Then chapter 4 introduces the approach to the problem and explains the design choices made. Results are revealed and discussed in chapter 6. Finally in chapter 7 we conclude the research and recommend how this research could be proceeded and implemented at MST.

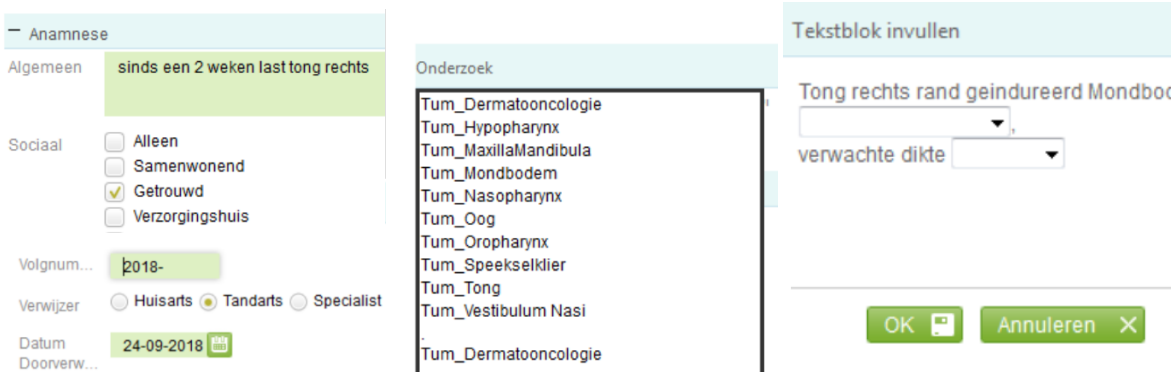
Chapter 2

Background

In this chapter we provide some background information on the problem as well as on Natural Language Processing (NLP). We explain how the Medical Spectrum Twente (MST) currently works and registers patient information. After that we explain what Dutch Institute for Clinical Auditing (DICA) is and what they do. Then we dive into NLP and explain some techniques we could use to interpret texts and extract information.

2.1 Current situation at MST

At MST there are two major systems that keep the patient records, DSV and X/Care. DSV is used as a medical file, which contains diagnoses, reports and conclusions about the patient. X/Care is used for the administrative tasks and contains appointments, doctors the patient has seen, referral letters from other doctors, et cetera. Doctors have a few different options on entering patient information into DSV. As we see in figure 2.1a there are radio buttons, checkboxes, number inputs and text areas. Some text areas have the possibility to be filled using a template, as shown in figure 2.1b, which presents options for the doctor to choose from, like in figure 2.1c. From section 1.1 we know that the EHR systems of MST cannot easily interact with each other, therefore doctors often create a document where they summarize the patients information. To do this they generate a document with X/Care and open it in Microsoft Word (a document editor). Word has a plugin that connects with the databases of X/Care and DSV. This plugin loads information about the patient and fills it in in the template areas of the rich text document. The document is then saved in X/Care. The plugin cannot load all the data from the EHR systems, so doctors and assistants copy and paste data from X/Care and DSV into the document. Similar documents are used during multi-disciplinary meetings, where doctors from different areas, for example dental surgery and radiology, discuss the best course of action for a patient. It is also used as a reference when registering patients with the clinical audit. Even though this information may be available in the EHR, it is easier to find it in this document and saves the person registering the information for the audit a lot of time.



(a) Input possibilities *(b)* Pick a template for text areas *(c)* Options for the template

Figure 2.1: Interface of DSV

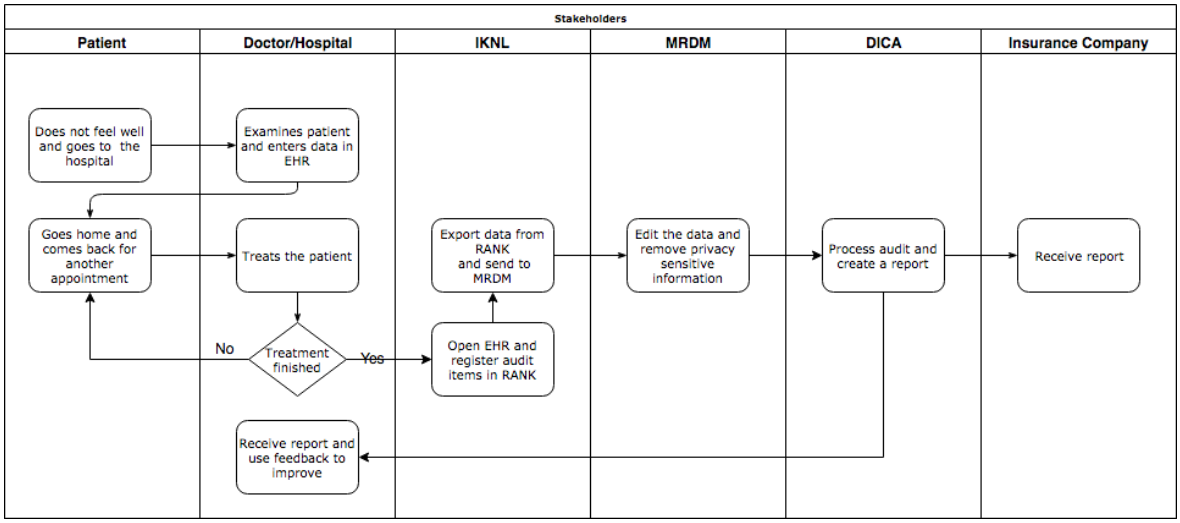


Figure 2.2: Registration process from data input to feedback

Figure 2.2 gives an overview of the registration process and what parties are involved at which point in time. Patient data that doctors entered in the EHR is extracted by data managers from IKNL and entered in RANK. The data is then sent to MRDM, who prepare the data to be used by DICA. DICA and its scientific committees then create analyses and make comparisons between hospitals. Last step is that the data, the comparisons and analyses are returned as feedback to the MST, insurance companies, doctors and patients.

Integraal Kankercentrum Nederland (Integral Cancer Centre Netherlands) (IKNL) is the party that processes all the data from the EHR to the DICA register. Data managers from IKNL have access to X/Care, DSV and the medication system of the MST. IKNL have their own registration system, Registratie Applicatie Nederlandse Kankerregistratie (Registration Application Dutch Cancer Registration) (RANK), in which they do the registration process. RANK then creates an export which is sent to Medical Research Data Manager (MRDM). MRDM processes the data before it is send to DICA ensuring anonymity. Data managers from IKNL extract data from the MST EHR systems and register this in RANK. The collected data is verified by the data managers by cross-referencing this with other data in the EHR systems of the MST. Difficulties with collecting data from the EHR systems, for the clinical audit, have to do with the way doctors register information about their patients. The EHR contains all the information, however finding all the information requires going through all the different appointments at which doctors registered information about the patient. In all those appointments doctors register some information multiple times and data managers have to cross-reference to ensure the correct values are registered in the audit. For instance, a patient could have quit smoking after the first appointment, so the audit should reflect this. It could also be that the size of the tumor was first a class 1, but later adjusted to be a class 2, so then a class 2 should be registered.

IKNL data managers verify that the data entered into the register is valid and consistent. Validity in this context means that the data entered into the register is a correct representation of the patient. Meaning that data is not copied incorrectly to the register. A patient can be diagnosed multiple times. The data manager ensures that the correct diagnosis is entered in the register.

2.2 DICA

DICA is an authority that gives insight in health care by creating reliable comparisons between patient care in different hospitals and analyzing patient care in hospitals. Goals for DICA are:

to increase patient satisfaction

to improve existing audits

to keep health care affordable

Together with scientific committees, DICA creates a trustworthy measuring system to give useful feedback to hospitals. As of 2018, DICA counts 22 clinical registrations for hospitals to use. The complete list can be found in appendix A. Each registration focuses on its own area of disease, for instance head and neck tumor, lung cancer, hip fractures or Parkinson. For example, information that DICA collects consists of patient identification, patient condition, treatment plan and post-treatment plan. DICA was established and financed by hospitals, medical specialists and health insurance companies. In 2017 and 2018 DICA was completely financed by Zorgverzekeraars Nederland (Health insurers Netherlands) (ZN) a group that serves the interests of the health insurers. With the analyses and comparisons from DICA, health insurers have more information about the quality of healthcare at the hospitals.

MRDM is a company that, according to their website[1], "processes medical data on behalf of organizations in healthcare". DICA is one of those healthcare organizations and MRDM collects the data from the clinical audits. MRDM edits this data to ensure that privacy sensitive information cannot be traced back to individual patients.

DHNA contains over 150 items, which are registered by hand at the MST. There are basic registration items like patient identification, length and weight, alcohol usage and smoking, other illnesses, heart problems, drug usage and others. Then there are items about the treatment plan, cuts or dissections, state and position of the tumor, additional (un)planned surgeries, pre-treatment and post-treatment actions. An example of an item can be found in figure 2.3. If alcohol has been answered with "Gestopt

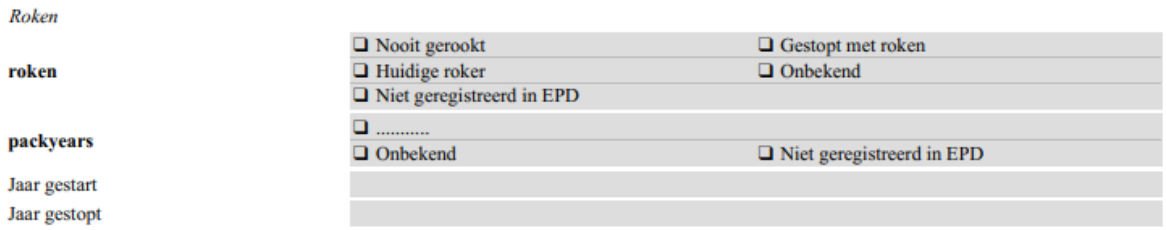


Figure 2.3: Questions about alcohol and smoking in DHNA

met roken" (Quit smoking) or "Huidige roker" (currently smoking), then questions "packyears", "Jaar gestart" (year started) and "Jaar gestopt" (year quit) should be answered. Type of answers in the register are listed options (like the example) and open answers (text, numeric, date). The open answers can have a required format, for instance when it should be a date or postal code.

[1] https://mrdm.nl/en/

2.3 Rule-based approach

Rule-based approaches to NLP are the oldest approaches. They are still used today and are proven to work well. Often they are used in combination with machine learning techniques as shown in this paper where radiology reports are read automatically to detect the stage of cancer [5]. An example of a rule-based approach, and also used in this research, is a regular expression. A regular expression is a sequence of characters that form a search pattern. They are often used to find string patterns in text or to validate form input (e.g. an email address on a website). Rule-based approaches require knowledge on the domain you want to apply it to. Developing rules requires a domain expert and someone to craft the rules. Rule-based approaches have some disadvantages. Often multiple rules are combined to complete a task and rules grow very large and complex. This makes maintenance difficult and time intensive. Also, rule-based approaches perform generally very well for specific use cases. However, when the context changes the rules don't apply any more and performance degrades [6].

2.4 Machine learning approach

Machine learning is a way in which computers can learn patterns and perform a specific task without any rules. Computers can learn these patterns or tasks by evaluating sample or training data and after training they can make predictions based on that. There are two forms of machine learning: supervised and unsupervised. The main difference between the two is that with supervised machine learning we can give the computer examples of input and output and have it learn the pattern and with unsupervised machine learning we do not have these output examples.

With supervised machine learning you have input variables and an output variable and you use a function to learn the mapping from the input to output. The output variable is also known as a label or class and is provided by humans for the corresponding input variables. The algorithm that attempts to learn the mapping function has different variables that are adjusted with each input-output pair it sees. The function is optimized when it can correctly predict the label for unseen data.

Unsupervised machine learning analyzes data and detects the underlying structure of data without having to rely on human provided labels. It is harder to evaluate the performance as there are no labels. This type of machine learning can identify patterns in data and detect anomalies in new data samples.

2.5 Classification

Classification is the task of determining the category for a new input. In our research, the doctor's text document is used as input and the classification task is to determine the output category ["patient smokes", "patient never smoked", "patient quit smoking", "unknown"]. The category with the highest posterior probability, that is after all the input is processed, is selected as the label for the input. Classification is a type of supervised machine learning since you have input and labels.

Binary and multi-class are two types of classification problems. In the example we had 4 categories (multi-class), but you can also have only 2 categories (binary). In binary classification the output

consists of two classes and with multi-class the output consists of n classes. Multi-class should not be confused with multi-label classification where you predict multiple labels for one input sample. Multi-class classification problems can be brought back to binary classification problems. For each class you treat it as one class (positive) and all the other classes are also treated as one class (negative). This is also known as the one-vs-rest method. There is also the one-vs-one method in which $n(n \quad 1)/2$ classifiers are trained, one for every class 'matchup'. When predicting the class a voting scheme is applied and the class with the most votes gets selected.

One algorithm for classification is Naive Bayes (NB), which is a supervised machine learning algorithm. It is based on Bayes theorem with the assumption that features are independent, meaning that each feature contributes independently of other features and there is no correlation between the features. NB calculates the probability of the class given the features of the problem [7]. Given a set of features X NB states the probability of Y is:

$$P(YjX) = P(YjX_1) \quad ... \quad P(YjX_n)$$

Using Bayes Theorem and assuming that the features are independent we can say this.

$$P(YjX_i) = \frac{P(X_ijY) \quad P(Y)}{P(X_i)}$$

$P(X_ijY)$ is the probability that X_i occurs when Y is true. $P(Y)$ is the probability that Y is true. $P(X_i)$ is the probability that X_i occurs.

Another supervised machine learning algorithm for classification is Logistic Regression (LR). It is used for classifying categorical items. The LR function is defined as following:

$$P(YjX) = \frac{1}{1 + e^{-}(\beta_0 + \beta_1 \quad x_1 + ... + \beta_n \quad x_n)}$$

Where x_i are features and β_i are parameters that can be tuned during training of the model to make certain features more or less important. The output of the function is always between 0 and 1. If the output is larger than 0.5 than we can classify the outcome as 1 otherwise as 0.

There are also other classification algorithms, but since we only use NB and LR for this research we do not explain other algorithms. Some other popular algorithms are: Linear Regression, Support Vector Machines, Decision Trees, Random Forests and (Deep) Neural Networks [8].

2.6 Text sequence labeling

Sequence labeling is the task of assigning labels to a sequence of variables. For instance labeling the part of speech (verb, noun, etc) labels to a sentence [9]. It can be seen as a classification problem for each item in the sequence. There are a limited number of labels that can be assigned to each item. Sequence labeling is a binary or multi-class labeling problem, depending on the number of labels. Any of the previously mentioned algorithms can be used, but much better accuracy can be reached by taking the features and labels of the neighbours of an item into account. Conditional random fields (CRF) is a statistic modeling method that predicts labels while taking neighbouring labels and features into account.

Chapter 3

Related work

3.1 Attempts to improve DICA registrations

There has been research to investigate the completeness of DICA data. For the Dutch Lung Cancer Audit for Surgery (DLCA-S) around 90% of the data was complete [10]. Completeness of registrations is important for DICA as it improves their research and reporting.

To address these issues and improve registration of data and the quality of the data DICA collaborates with hospitals and other institutes. Registration at the Source (Registratie aan de Bron) is a national initiative that focuses on capturing the data clearly and only once[1]. The project is an initiative by hospitals and other health institutes. It enables the data to be reusable in multiple situations without having to register the data again. At the Radboud University Medical Center (RadboudUMC) they worked on a project to restructure their EHR. RadboudUMC uses an EHR called Epic in which they added all the items the Dutch Head and Neck Audit (DHNA) requires at logical places for doctors. During the restructuring of the EHR they looked at the medical care path of the patient. This path needs to be uniform across the entire medical department. The IT department and the doctors together looked at the medical care path of the patient and determined which item from the DICA should be registered at what time. Not only are all 150 items for the DHNA registered, but also in a way that doctors can still use the EHR efficiently. Guido van den Broek, doctor and initiative taker for the 'registratie aan de bron' project at RadboudUMC, said that the project has been a success according to the people who use the EHR and people who are handling the registration of DHNA.

DICA itself is also working to make it easier to register data. Medical Research Data Manager (MRDM) and Nederlandse Vereniging voor Radiotherapeutische Oncologie (Dutch Association for Radiotherapeutic Oncology) (NVRO) created a database where data about colorectal (intestine and rectal) breast cancer is saved in a Findable Accesible Interoperable Reusable (FAIR)[2] manner. The registration of the radiotherapeutic variables used to be extracted manually from the report and entered in the registration. This costs 10 to 20 minutes per patient. Using the data from the radiotherapeutic systems all the relevant information is extracted automatically. The cost of registration is reduced to approximately half a minute per patient.

3.2 Natural Language Processing in EHR's

With the increasing amount of data in Electronic Health Record (EHR) systems, new challenges arise. A lot of the content in those systems are texts or images. This data contains a lot of information that can contribute to improve the health care. One organisation that does this is Dutch Institute for Clinical Auditing (DICA) and this research is a proof of concept specifically for the DHNA. As it is impossible for humans to analyse the enormous amounts of data and extract the relevant information and patterns, computers are used. From the previous sections we know that text in the EHR is stored in a lot of places and most text is unstructured. This format is not directly suitable for

[1] https://www.registratieaandebron.nl

[2] https://mrdm.nl/fair-implementatie-en-minder-registratielast-voor-radiotherapeutische-kwaliteitsregistraties-dica/

automated quality tests, clinical advise and research (except for researches like our own). Therefore, a lot of research in natural language processing has been done to automatically identify and extract information.

Natural language processing (NLP) is a part of computer science and artificial intelligence concerned with the analysis and interpretation of natural language by computers. It is often related to terms like text mining and information extraction [11]. In 2017 over 7000 publication were reviewed and a list of 71 natural language processing systems was composed using NLP to capture and standardize unstructured clinical data [12]. The review identified many NLP systems capable of processing clinical free text and generating structured output.

Pakhomov et al. [13] developed a machine learning method for identifying foot examination findings in unstructured text of clinical reports. They did a classification using Support Vector Machines (SVM) on 3 items: neurological, vascular and structural findings with each 3 categories. Measuring performance using accuracy they got respectively 87%, 88% and 81% accuracy.

Natural language processing was applied by Raju et al. [14] to identify adenomas (benign tumors of glandular tissue). First using NLP to identify if colonoscopy reports described that a screening took place. In 7 steps this is done using rule based NLP. After that the identification of adenomas is done by rules as well. Using this method 91% of screening reports were found correctly and in those reports 99% of adenomas were identified correctly. The identification accuracy using NLP was higher than when done manual with an accuracy of 88% and 98% respectively.

Research by Jonnalagadda et al. [15] showed a rule based system that identified patients with heart failure with preserved ejection fraction (HFpEF) from free text. They used regular expressions to identify inclusion criteria for patients. They also conclude that while testing, the algorithm selected 113 patients of which 67 did not qualify. This was not due to failure of the algorithm, but because of additional criteria that were not in the algorithm. It shows that finding certain patient characteristics can be found using rules, but also that a rule based system is not very flexible. Similar tooling developed by CTCue can analyze EHR databases[3]. This should be an easy tool for doctors and researchers to query the database for any patients matching certain criteria. This tool uses natural language processing techniques to retrieve information from the unstructured data. Unfortunately at this time MST did not have this tool in use and could not experiment with it.

Closely related and recent research was conducted by Pathak [16] to automatically check the quality of radiology reports on breast cancer using NLP. She applied a nested CRF to identify top level structures and label the text in each of the structures. Top level structures got a F_1 score of 0.97 and content of the report a score of 0.94. The automatic labelling of the report got a F_1 score of 0.78.

[3] https://ctcue.com/

Chapter 4

Technical setup

This chapter describes the technical layout of the research. We explain which techniques are used in the experiments and how the different proposed solutions are set up. The setup is explained using graphics in figures 4.1, 4.2 and 4.3. We will then first explain what is seen in the graphics and later explain why this particular architecture was chosen. In the final section we will describe a possible workflow and implementation.

4.1 General setup

There are 4 methods we used in the experiments. Every method is graphically shown in figures 4.1 to 4.3. The figures all start with a text document. The arrows indicate that some action is performed. After each arrow the output of the action is shown. Methods are constructed from different actions that use a specific technique. Using the letters of the actions we composed the name of the method, making it easier to refer to a specific method. Figure 4.3 has a gray dashed box around the last part, indicating an optional step.

There are two tasks we want to evaluate:

1. Classification
2. Value extraction

The first task is for situations where we want to determine if one of the predefined options applies to the text. There are items about smoking and alcohol that have 4 categories to choose from: 'yes, currently', 'never', 'in the past' and 'unknown'. There are also items, like length and weight that require a value.

Figure 4.1 describes our basic and naive approach to classification. A text document, a report from a doctor, is vectorized and used as input for a classifier. We use two different classifiers, naive bayes **NB** and logistic regression **LR**, to test how well a method performs. We call this method NSNB or NSLR, which is a combination of **N**o-preprocessing **S**imple concatenated with the abbreviation of the classifier.

In an attempt to improve on NSNB/NSLR we reduce the amount of input and increase the amount of relevant data. This is done by preprocessing the text document before vectorizing and classifying it. Preprocessing is done by rules identifying headers that indicate a paragraph containing relevant

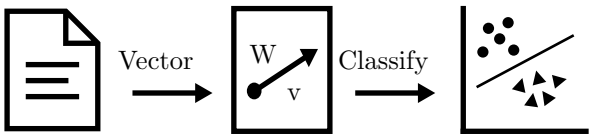


Figure 4.1: Solution 1: NSNB/LR

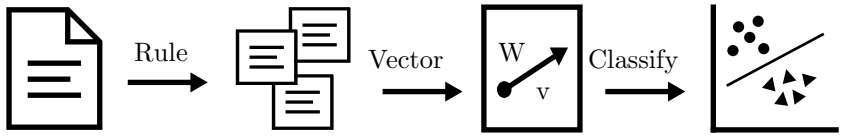


Figure 4.2: Solution 2: PSNB/LR

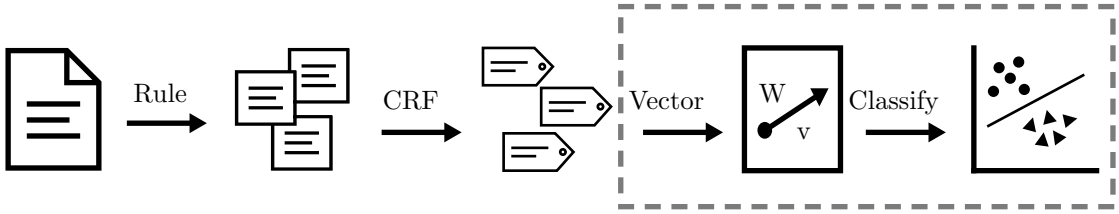


Figure 4.3: Solutions 3 and 4: PCV and PCNB/LR

information. The heading and paragraph are identified using a regular expression. These methods are called PSNB and PSLR where the P comes from **P**reprocessing.

The methods in figure 4.3 can be used for both tasks described earlier. Like the previous method the entire text document is reduced and the amount of relevant data increased. Then we apply a CRF that tags each word with a specific label. These labels have two purposes. The first is value extraction, where a label marks a word that is directly extracted as value for the audit registration. The second is to reduce input and increase the amount of relevant data for the classifier. The method with the first purpose for outcome we call PCV for **P**reprocessing **C**RF and **V**alue extraction. When using the classifier with this method we call is PCNB or PCLR.

4.2 The document

Every solution has the same starting point, which is a text document. This document is a text document from the Electronic Health Record (EHR) where doctors wrote down the conclusion of the multi-disciplinary counsel. Appendix B shows an example of a document that was used. In the EHR, these documents are stored in a rich text format, a format created by Microsoft for exchanging formatted text [17]. The documents are converted to plain text using a library from Sautinsoft that can convert rtf files[1]. The output is stored in files with the txt extension.

4.3 Rule-based preprocessing

Using the 'rule' step information is removed that we deemed 'less relevant' from the documents and thereby we increase the amount of relevant data. The 'less relevant' part was determined together with a doctor. The preprocessing is done using a regular expression (regex). A regex is a sequence of characters that form a search pattern. The documents we have contain headings followed by content and we are interested in content belonging to specific headings. This is the regex we used to identify headings and capture the content.

[1] https://www.sautinsoft.net/help/rtf-to-html-net/Index.html

```
^(voorgeschiedenis:?(\r?\n)?)((.+\r?\n)+(?=(\r?\n)?))?
```

Should start with the word 'voorgeschiedenis' (prehistory)
Optionally match a ':'
Optionally match a new line nr is optional for matching newlines on DOS based systems
Match all characters until we spot 2 newlines

- Match any number of characters followed by a newline
- Repeat this process until the newline is followed by another newline

This way the title 'voorgeschiedenis' and its content is matched and then extracted. This process is repeated for a total of 9 headings. Here we list all of them, including what content they usually have.

ptnm: pathological tumor classification
voorgeschiedenis (prehistory): earlier illnesses or visits to the hospital or doctors
korte ziektegeschiedenis: short description on how a patient got to where he is now
algemene gezondheid: health status of the patient disregarding the tumor
medicatie: medication
intoxicaties: patients use of alcohol, drugs and smoking
klinisch onderzoek: clinical research on the patient
ctnm stadiring en localisatie: clinical tumor classification and localisation
conclusie: conclusion of the doctor on the course of treatment

Python with the module 're' is used for the script to extract the information from the document. The regex is executed with flags to ignore case-sensitivity, use multiline and use unicode encoding.

4.4 CRF preprocessing

Automated labeling of text is done using CRF (conditional random fields). The labeling is used both for preprocessing the text for the classifier and to extract values directly.

Before applying the CRF to documents for labeling, a model is trained. The training process is done on the text extracted using the regex. The extracted headings with their content per document are concatenated. Then these documents are annotated by hand using a tool called GATE[2]. This tool makes it easy to add labels to parts of a text. It saves the documents in its own format and can later export the documents with annotations as XML.

Table 4.1 shows the labels added to the texts and how many words have a certain label. The NA label indicated all words that have no special meaning (Not Applicable). The labels t-stadium, n-stadium and m-stadium, length and weight are used for validating the value extraction process. The word having that label is the value that is extracted. The labels smoke-duration and alcohol-amount are used for the classification validation. The label smoke-duration marks only words that describe how long a patient smokes, has smoked or just 'no' when a patient has never smoked. The label alcohol-amount marks only words that describe how much a patients drinks per day or week. The words

[2]https://gate.ac.uk

Label	Label
NA	smoke-duration
head-prehis	alcohol-amount
head-shortill	drugs-amount
head-health	tumor-location
head-social	t-stadium
head-medication	n-stadium
head-intox	m-stadium
head-clinical	weight
head-concltwhht	length
head-conclusion	

Table 4.1: Used labels for preprocessing

having that label are extracted, concatenated and then used as features for the classifier. The labels added to the text were determined at the beginning of the research. It leaves room for other items to be tested, but they are not used in this research. They are left in, because they add additional information to the text. However, as they are not used for validation they are not explained.

Every document is preprocessed before used for training the model. All the words in the document are stripped of punctuation such as !()-[];:'"n,<>./?@#$%&*_~. Words that have no label will receive a label 'NA'. All words are then POS-tagged (part of speech), indicating the function of a word within the sentence. This is done with the nltk package[3]. The following characteristics of the word are added as features: word is uppercase, word is a title (mr, dr) and whether the word is a number or not. We also add the previous and next word as feature to the word and we add the same characteristics for the previous and next word as features. In case the word is the last word of a document we mark it as the last word. Words that are uppercase could indicate some abbreviation or other special word. Words that are titles are most likely followed by a name or person. These features all add information and meaning to a word and help to determine the correct label. With these features we train the model and store this to be used in determining labels for unlabeled text. The training of the model is done using the pycrfsuite[4]. Parameters of the training method were all set to the defaults.

4.5 Vectorization and classification

We use two different classifiers on our data. Before training the classifiers on the data we must first extract features from the text. First we explain the feature extraction and then the classifiers we used.

The words from the texts are the features we use for the classifiers. Naive Bayes (NB) and Logistic Regression (LR) need to have a vector of numbers as features. All words are represented by a unique number. To give importance to the words we could simply count the number of words across all documents. This is called term frequency (TF). This would give importance based on count, but not on how unique or important a word is across all documents. Therefore we consider in how many documents a word occurs. This technique is called term frequency-inverse document frequency (TF-IDF) [18]. We use the TfidfVectorizer, with the default parameters, from the sklearn python package

[3]https://www.nltk.org/index.html
[4]https://github.com/scrapinghub/python-crfsuite

to do this.

The two classifiers we used are NB and LR. We used the Multinomial Naive Bayes, which is typically used for document classification and useful when having discrete features like TF or TF-IDF. Bayes theorem is used to calculate the probabilities with the assumption that the features are independent. Even though the independence is unrealistic for text, NB tends to perform quite well [19]. MultinominalNB, with the default parameters, from sklearn python package was used to build the model. LR is the second classifier we use. It uses the same features as the NB classifiers. From the sklearn python package we used the LogisticRegression model with default parameters.

4.6 Example workflow and implementation

Above we have explained all the techniques that are used. These techniques can be used to build an extraction system where doctors have a simple interface to export all patient information required for a DICA registration. We could use a workflow like this.

1. Look up a patient
2. Click a button to view patient information for the DICA registration
3. Review the information
4. Optional: update information where needed
5. Export and send patient information to DICA

The first step is for a doctor to open the EHR and look up the patient. The doctor then sees the patient file. This could be the place where the button for extracting the patient information for a Dutch Institute for Clinical Auditing (DICA) registration is. Of course there could be more buttons for all other DICA registrations. Clicking the button is step two after which a new screen opens. Here the doctor sees an overview of all 180 items that the DICA registration requires for the Dutch Head and Neck Audit (DHNA). The doctor can also see how confident the computer is that the item has the correct value or possibly even which text is responsible for the answer. In case a doctor does not agree with the system the doctor can change the value. The last step is to export the registration data. From the Medical Research Data Manager (MRDM) website we can get the format to which we should export the registration[5].

Between steps 2 and 3 the extraction system has to predict a value for each item of the DICA registration. This prediction is made by a trained Natural Language Processing (NLP) model as described in the sections above. For categorical items, that is picking one value out of a fixed set of values, a classification model should be used (i.e. NSNB/LR, PSNB/LR or PCNB/LR). For continuous items, items that require a number or string with infinite possibilities, a value extraction method should be used (i.e. PCV).

No matter how good this system will be trained, errors will always occur in the suggestions. To combat these errors we give users the option to review and edit values. We can show users how confident the system is about the suggestion.

[5]https://support.mrdm.nl/registries/dhna/

A separate machine containing a copy of the EHR database should be setup. This prevents the EHR suffering from decreased performance when the model is trained or patient data is extracted. Periodically data from the EHR database should be synchronized to keep the extraction system up to date. After synchronizing the database the extraction system could retrain it's machine learning models.

The extraction system is on a separate machine and must have an application programming interface (API). An API is an interface other systems can talk to, to request information. When a doctor clicks the button in step 2, the EHR makes an API request for the DICA registration data on a patient. The extraction system then retrieves this information and sends it back.

The extraction system uses the medical text files as input. To extract values for continuous items and for one of the proposed preprocessing methods labeling of these files is needed. The machine learning model used by the extraction system does this automatically. However, we do want the ability to edit labels that the system predicted wrongfully and also that the system learns from its mistakes. Therefore, the changes that are made by hand must be send back to the system so the system can learn again from the corrected files.

Chapter 5

Experimental setup

This chapter describes the experiments we have done on the methods as described in chapter 4. First we describe how we measured performance. Then we describe the data that was used to perform the experiments with. Finally we describe how the methods are trained and tested.

5.1 Evaluation metrics

Metrics are needed to evaluate how well a method performs. For this we use a combination of recall and precision. While predicting there are four options for the system.

True Positive (TP): Predicted as positive where it should be positive (correct)

True Negative (TN): Predicted as negative where it should be negative (correct)

False Positive (FP): Predicted as positive where it should be negative (wrong)

False Negative (FN): Predicted as negative where it should be negative (wrong)

Precision (p) is the fraction of true positive predicted items over the total number of positive predicted items. It is the fraction of correctly predicted items for a specific category. Recall (r) is the fraction of true positive predicted items over the total number of positive items. It is the fraction of correctly predicted items over the total number of items that should have been predicted for a specific category. F_1 combines precision and recall in one measure and is the harmonic mean between the two of them.

$$p = \frac{TP}{TP + FP}$$

$$r = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2pr}{p + r} = \frac{2TP}{2TP + FP + FN}$$

In our experiments we have multiple classes to identify, e.g. smoking, used to smoke, never smoked or unknown. TP, TN, FP and FN are used for binary classification. Therefore we use the one-vs-rest method, where one is the current class and the rest the other classes. The performance for the multiple classes as a whole is calculated using the weighted average (F_1^w). Per category the weight is determined by how often that category occurs in the total set of categories.

$$F_1^w = \sum_{i=1}^{n} w_i \quad F_1 i$$

F_1 is a metric we can use both for the classification and value extraction methods. With classification you either predict the right or the wrong answer and there is no margin for error. However, when predicting continuous values, for example weight, the algorithm could predict 84.2kg while the patient weigh 83.9kg. The value given by the algorithm is not 100% right, but it was not far off. But since

Category\Item	*Smoking*	*Alcohol*
Current	47	64
Never	7	4
Quit	23	6
Unknown	21	21
Blank	1	4
Total	**98**	**95**

Table 5.1: Category counts for smoking and alcohol items

Label	**Count**	**Label**	**Count**
NA	12285	smoke-duration	441
head-prehis	140	alcohol-amount	251
head-shortill	248	drugs-amount	30
head-health	171	tumor-location	163
head-social	120	t-stadium	87
head-medication	156	n-stadium	79
head-intox	160	m-stadium	73
head-clinical	234	weight	26
head-concltwhht	80	length	23
head-conclusion	108		

Table 5.2: Label count in all texts

we extract a piece from the text that contains the value we cannot be wrong. Either we extracted the right piece from the text or the wrong piece.

5.2 Dataset

Our reference dataset with DHNA data from 2015 to mid 2018 contains 180 patients. This data was requested from IKNL, the organization that collects this data on behalf of the MST. This dataset is the same as was sent to DICA for analysis. The dataset contains for each patient entry a unique number to identify a patient which was used to retrieve the multi disciplinary counsel documents from the MST's EHR. Of the 180 patients that we have their DHNA registration from, 99 had a multi disciplinary counsel document. This is a text document that contains a lot of information about the patient. We chose this document for the experiment because a lot of patients have such a document. Also the document is for most patients very similar in structure which ensures that we have roughly the same amount of data for every patient. An example of such a document can be found in appendix B. The 99 documents we have from the patients are in rtf format and we converted those to plain text. Section 4.2 describes what rtf is and how we convert it to plain text. These plain text are then annotated with labels. In section 4.4 we already saw which labels are added to the text and table 5.2 shows how many labels all text contain in total.

The data from the DHNA dataset is not 100% complete. There are items that have no answer. We see this in table 5.1 where we have the counts per category for the items smoking and alcohol. We

Category\Item	*Smoking*	*Alcohol*
Current	Huidig roker	Huidige drinker
Never	Nooit gerookt	Nooit alcohol gedronken
Quit	Gestopt met roken	Gestopt met drinken
Unknown	Onbekend	Onbekend

Table 5.3: Available categories per item

Item	**Explanation**
length	The length of a patient in centimeters (cm)
weight	The weight of a patient in kilograms (kg)
t-stadium	Describes the tumor size. A number possibly suffixed by a character.
n-stadium	Describes wheter nearby lymph nodes are affected. A number possibly suffixed by a character.
m-stadium	Describes distant metastasis (spread of cancer from one part of the body to another). A number possibly suffixed by a character.

Table 5.4: Value extraction items explained

could say that when the item is left blank in the dataset that the value is unknown. However, we do not know the reason it was left blank. It could have simply been forgotten. So we do not use those patients in our experiment.

5.3 Training and testing plan

There are 6 methods we experiment with for classification. These are NSNB/LR, PSNB/LR and PCNB/LR. We focus on two items from the DHNA each with 4 categories to choose from. These items and their categories can be found in table 5.3. Even though we have 6 methods we will do 8 tests. We explain this in the paragraphs below. For value extraction we have one method, PCV. For this we focus on 5 items: length, weight, t-stadium, n-stadium and m-stadium. More information about the items can be found in table 5.4.

In our experiments we often use the n fold cross validation training and testing technique. This technique allows all data to be used for training and testing. This is particularly useful when there is not a lot of data available which is the case with our experiment. The data is divided in n parts (n is a number) of which (n-1) are used for training and 1 for testing. This process is repeated n times and so all the data is used. Never is data used for training and testing at the same time.

To measure the performance of the NSNB/LR methods we use a 5 fold cross validation.

To measure the performance of the PSNB/LR methods we first apply the rule preprocessing and then use the same 5 fold cross validation. We focus on the items alcohol and smoking which we expect under the heading 'intoxicaties'. Instead of using the entire text for classification we have reduced the amount of text to 1 paragraph.

To measure the performance of the PCNB/LR methods we first apply CRF preprocessing. This

adds a label to each word. We focus on the items alcohol and smoking which should have the labels 'alcohol-amount' and 'smoking-duration' respectively. We then use a 5 fold cross validation.

These last 2 methods, PCNB/LR, use the CRF preprocessing. This technique could give us errors which then propagate into the classifier. To see how these errors impact the performance, we also measure the performance of the technique while pretending that the CRF preprocessing was perfect. We then use the labels that we added ourselves and do not use the CRF to predict those for us. We could call this the PCNB/LR (perfect) method.

To measure the performance of the automated labeling using CRF we use 3 fold cross validation.

To measure the performance of the PCV method we use 3 fold cross validation. For each item we want to test, see table 5.4, we take the word that has that label and validate if it is correct. It could be that we find no word with one of the labels. This is either an error, because the CRF did not label a word correctly, or it is because the text does not contain information on that item and in that case we don't use it in the results.

Chapter 6

Results and discussion

In this section we discuss the experiments and the results. Then we will display and discuss the results of the different methods. First we go over the classification results with the different preprocessing steps. We discuss the differences in the results between the NB and LR methods and the effects of preprocessing with the rule and automatic labeling. Then we discuss the performance of the CRF, which is one of our preprocessing methods. Lastly we discuss the PCV method that we use to extract values.

6.1 Classification methods

Tables 6.1 and 6.2 display the F_1 scores on the smoking and alcohol items. The first column of both tables contains the name of the preprocessing method. The first row contains the name of the classifier that is used. Concatenating both gives the names of the method we defined in section 4.1. In table 6.1 we notice that of the methods using the Naive Bayes (NB) classifier PSNB scored highest whilst of the method using the Logistic Regression (LR) classifier NSLR scored highest. In table 6.2 we notice that of the methods using the NB classifier PCNB (perfect) scored highest whilst of the methods using the LR classifier NSLR scored highest. Last thing we directly notice is that methods using LR have a higher score than methods with the same preprocessing have using NB classification.

When looking at the scores between the smoking and alcohol items we see that every score on the alcohol item is higher. This could be because of the imbalance in the number of items per category for both the smoking and alcohol items. From table 5.1 we see that the alcohol item has 61 patients in the category 'current' of the total 95 patients. For the smoking item 47 of 98 patients are in the 'current' category.

During the last stage of the research we found that the reference dataset contained mistakes. These are that the wrong category is in our reference dataset, while we can manually determine that it should be a different category. Due to time constraints we only checked the smoking item for mistakes. We found 11 cases where the reference set had the wrong category. After correcting these we executed our experiments again. Results of these can be found in table 6.3.

When comparing the different preprocessing method we see that applying more preprocessing, meaning that the further we zoom in on the data, decreases the F_1 scoring. Of all the methods using the NB classifier PSNB does score higher on the smoking item, but on the alcohol item it has almost the same

Preprocessing	**NB**	**LR**
NS	0.61	0.79
PS	0.68	0.75
PC	0.48	0.55
PC (perfect)	0.55	0.65

Table 6.1: F_1 score for smoking

Preprocessing	**NB**	**LR**
NS	0.73	0.89
PS	0.71	0.79
PC	0.69	0.73
PC (perfect)	0.75	0.77

Table 6.2: F_1 score for alcohol

Preprocessing	NB	LR
NS	0.63	0.78
PS	0.69	0.77
PC	0.45	0.63
PC (perfect)	0.60	0.79

Table 6.3: F_1 score for smoking after correction of the dataset

score as NSNB. Of all the methods using the LR classifier NSLR scores best for both the smoking and alcohol items.

This is not the effect we expected that preprocessing would have on classification. Preprocessing is meant to zoom in on the parts of the text that hold the most meaning and leave out pieces of the text that are seemingly irrelevant to the item. The PSNB/LR methods have a lower F_1 score than NSNB/LR except for on smoking where NSNB scores higher.

To get a better understanding why the F_1 score is higher when not using preprocessing we look at what made the classification algorithm decide. In appendix C we have the top 10 most important features for every category per item and per method. Looking at the most important features for the NSNB/LR methods we see that there are no words that have a relation to smoking or alcohol. For the smoking item NSNB some of the words are important for multiple classes. The words 'left' and 'twhht' (abbreviation for Twente Werkgroep Hoofd-Hals Tumoren) are the top words for 2 categories. So we cannot say that this method learned anything about these items. It looks like it found other patterns in our data.

Looking at the most important features used for the PSNB/LR and PCNB/LR we see that the words are related to the alcohol and smoking items. This was to be expected as PSNB/LR uses only text that was in the paragraph with the heading 'intoxications' which contains information on smoking and alcohol and not much else. For PCNB/LR it was also expected, because they only used words that had the label 'alcohol-amount' or 'smoking-duration'. We also see that the most important features per category relate to that category. For instance with PSNB the item smoking has 'gestaakt' (discontinued) as most important feature for the category 'Gestopt met drinken' (quit drinking) and PSNB and PSLR have respectively 'nooit' (never) and 'nee' (no) as most important features for category 'Nooit alcohol gedronken' (never drank alcohol).

Still PCNB/LR methods have a lower score than PSNB/LR, even though we expected this to be higher. Looking at the most important features for each of the classes we find that classes have words in common that are important. The same feature is important for multiple classes, which makes it harder to predict the correct class. Also preprocessing with the labels gives very few words for the classification algorithm to classify on.

We can see that PCNB/LR (perfect) has a higher score than PCNB/LR. The F_1 score of the PCNB/LR (perfect) method on the smoking item has increased much more than on the alcohol item. This is to be expected seeing that the labeling of the alcohol-amount item was already a lot better than on the smoke-duration item.

Label	F_1-score	Label	F_1-score
NA	0.99	smoke-duration	0.65
head-prehis	1	alcohol-amount	0.98
head-shortill	1	drugs-amount	0.8
head-health	0.98	tumor-location	0.81
head-social	0.98	t-stadium	0.86
head-medication	0.99	n-stadium	0.95
head-intox	0.98	m-stadium	0.96
head-clinical	0.99	weight	0.79
head-concltwhht	0.98	length	0.92
head-conclusion	0.96		

Table 6.4: F_1-scores of the CRF method for automatic labeling

Method	length	weight	t-stadium	n-stadium	m-stadium
PCV	0.92	0.63	0.51	0.81	0.91
PCV (perfect)	0.92	0.63	0.51	0.81	0.91

Table 6.5: F_1 scores for extracting information using CRF

6.2 Automatic labeling

Table 6.4 shows the F_1-scores on the automatic labeling using CRF. The label 'smoke-duration' is something found after the 'head-intox' label and most of the time preceded by the word 'roken' (smoking). Same goes for 'alcohol-amount', but this is most of the time preceded by the word 'alcohol'. The label 'smoke-duration' gets a significantly lower score than 'alcohol-amount', but this could be because after the last word that has the label 'alcohol-amount' a newline comes and after 'smoke-duration' other words.

6.3 Extracting method

Table 6.5 shows the results of the value extraction. We notice very different scores with weight and t-stadium a lot lower F_1 scores than scores of height, n-stadium and m-stadium. Also the scores are the same when perfect labels are used. From the previous section we know that the labeling process performs pretty well with F_1 scores of 0.79 for the weight label up to 0.96 for m-stadium label. Even though we did not expect a large increase of the scores we would expect some increase.

A reason for not increasing the score while using better labels could be that the reference dataset is incorrect. Due to time constraints we were only able to evaluate the dataset for the item weight. We found several different errors in the reference data.

One of the errors we found is that some items are registered different by different people. For instance, a doctor could register the weight of a patient as 84.1kg in the document and the person registering this item in the DHNA registration registers the weight as 84kg. We also found instances where the weight of the patient in the document was completely different from the weight in the reference data.

We manually checked other documents in the EHR and concluded that our reference dataset was mistaken.

Redoing the experiment for the weight item using a corrected dataset gives us the results as shown in table 6.6. We now see that the F_1 score is 1.0.

Method	**weight**
PCV	1.0
PCV (perfect)	1.0

Table 6.6: F_1 scores for extracting information using CRF on a corrected dataset

Chapter 7

Conclusion

This section describes the conclusion of our research. First we recap on the problem and go over our proposed solution. Then we discuss our research questions as mentioned in chapter 1 and how well they are answered. Then we go over the limitations of our work and what could be improved on this work. Finally we go over the recommendation for MST, and possibly other hospitals, on what could be done to implement this work.

7.1 Summary

All hospitals in the Netherlands have to report medical operations to Dutch Institute for Clinical Auditing (DICA). DICA is an organization that collects the information with the purpose of improving the quality of the medical care. For the collection of the data DICA created clinical registration forms. Dutch Head and Neck Audit (DHNA) is one of those registration forms. DHNA contains 180 items that can be registered per patient. At Medical Spectrum Twente (MST) the process of registration costs about 90 minutes per patient. This research proposes a technique to automate the registration process.

We proposed a solution that uses Natural Language Processing (NLP) to extract information from the free text of a medical document. For categorical items, items requiring one value of a fixed set of values, we use a classification method. The text document is used as input to determine the category. As this document contains a lot of irrelevant information we have proposed 2 preprocessing methods to zoom in on the text. The first method scopes to a specific paragraph containing information about the item to be registered. The second method zooms in even further where only words related to the item are used.

To zoom in on the specific words we use a preprocessing technique that adds a label to each word to indicate the meaning of that word. This labeling technique is also used to extract values for the continuous items such as weight and length.

Experiments with these methods have shown promise to extract values automatically. However, mistakes were made during the process. We believe that increasing the amount of training data would improve results. Also using other medical documents could increase the amount of relevant data and possibly improve results. For now the proposed methods could aid during the extraction process, when implemented for all items of DHNA, giving suggestions to the doctor extracting a patients data. We proposed a possible implementation in section 4.6. This implementation gives a value suggestion for each item along with the option to review the suggestion and edit when needed.

7.2 Research questions

We formulated our research questions in section 1.3. We first answer the sub-questions after which we answer the main research question.

Sub-question 1: *\How can we best determine the value for a categorical eld?"*

In this research we proposed a method for extracting values for categorical fields using classification. We proposed two preprocessing methods to improve results. Even though the method using no preprocessing had the highest F_1 scores we do not think that it is the best method as it does not make decisions based on relevant data, but on other unrelated patterns in the features. In the current setup rule-based preprocessing performed overall slightly better than using labeling as preprocessing.

Sub-question 2: *\How can we best determine the value of a continuous eld?"*

Using automatic labeling we added labels to unstructured texts from the EHR to indicate meaning of words. This way we identified 5 items for DHNA. The automatic labeling performed well with F_1 scores ranging from 0.65 to 1.0. The extraction of the values proved to be more challenging. The texts we have do not always contain the values we want to extract. We believe this method is useful when used on a wider variety of texts and not just the document we used in these experiments. When leaving out all the patients for which the data could not be found we got F_1 scores 0.92, 0.63, 0.51, 0.81 and 0.91 for respectively length, weight, t-stadium, n-stadium and m-stadium. We should mention that our reference data contained errors which we had to correct manually after which the score for the weight item increased to 1.0.

The main research question: *\How well can we extract patient information from natural texts the EHR with the goal to automatically ll the audit forms of DHNA?"*

DHNA contains 180 items that need to be registered per patient with the purpose of improving the procedures and health care for patients. We showed that techniques proposed in this research can aid in extracting items for DICA registrations. The proposed preprocessing techniques for zooming in on relevant parts of the data shows promise and even though those did not have the best results they were explainable. Using more training data and other documents we believe that the performance of these methods should improve. When implemented for all items of the DICA registration it should aid in extracting values.

7.3 Limitations

There are some limitations to this research project which we will discuss here. The main limitation is the available amount of labeled data. The preprocessing using automatic labeling and the value extraction method heavily relied on this. We used text documents from the EHR that are not annotated yet, meaning that they do not have labels. Adding the labels is a time intensive process and knowing which labels should be added to what parts of the document is something that requires some medical knowledge.

The rule-based preprocessing step requires one to write a regular expression. This is a manual step that one with knowledge about regular expressions, usually computer scientists, has to do. These rules are also not very flexible and when something in the documents changes that the rule has to process, the regular expression has to be updated. In this project we added labels on the texts that were extracted using regular expressions from the entire text. However, we believe that the extraction of the text using regular expressions is not needed and that the results of the automatic labeling could

be almost the same. The words that were in this project removed by the regular expression would be marked as NA, as they have no label. We believe that after training the CRF with this text it should give the same results.

DHNA contains 180 items for each patient that need to be registered, but we only focused on 7 of these items. Using a wider variety of documents would open the possibility to register more items. However, by using more documents the limitations of the previous paragraphs have to be taken into account.

This project was initiated at MST and data from this hospital was used to train and test the methods. This raises questions as to whether this project is applicable in different hospitals. Until this is tested we cannot say this for sure, but the methods we used for extraction the information from the EHR are quite abstract and could be applied elsewhere. When other hospitals have similar documents the preprocessing methods to zoom in on data using labels can be applied there as well. A machine learning model (CRF) was trained on medical texts which other hospitals have too.

7.4 Future work

There are a number of possible things that could improve this research. Some of these are described in this section.

Currently we used two classifiers to establish if our preprocessing had the desired effect, which was to improve on classifying using the entire document as features. However, there are many more classifiers than Naive Bayes and Linear Regression. Here are some that could be used, but not limited to: Support Vector Machine, Logistic Regression, (Deep) Neural Network and Random Forest.

With the extraction of values we found that in many cases the value could not be found. In that case no match can be found and we register it as a failed detection of the value. However, as we filter out special characters during the tokenization of the words from the texts, also characters like '-' are removed. These character often represent the absence of such a value or indicating that the value is unknown. From section 6.1 texts that discuss the patient's smoking or drinking behaviour are often preceded by the respective words 'roken' and 'alcohol'. The absence of any text with these words could also indicate that the doctor did not ask the patient and that it is unknown to the hospital. We could attempt to incorporate these features as well.

One of the limitations as discussed above is the absence of large amounts of labeled texts that can be used for training. To overcome this problem we could use a technique called transfer learning. This is a machine learning technique where knowledge gained from solving one problem is used to help solve a different problem [20]. In our case we could look for a large source of labeled medical documents and use these to train a CRF. Then we could use this model as a basis for our own CRF model and refine it with our own documents.

7.5 Recommendations

The initiative for the project has come from MST to automate registration for DHNA. This section gives some recommendations that could help to implement the research in practise. Currently the

project has its limitation as we wrote in section 7.3. When some of the limitations are overcome we have the following recommendations for implementing the project.

Throughout the hospital the same EHR system is used and it would be desired to use this project not just for DHNA, but also for all the other registrations, see appendix A. Algorithms for training automatic labeling could be reused using transfer learning as described in the previous section. Abstracting the training part of the classifiers and the CRF could be beneficial for reuse. Also training a classifier with a lot of data is a computationally heavy process and would better not be applied directly on the database of the EHR.

To combat errors and correct them we could allow doctors to correct classifications and labels predicted by the system. This effectively increases the amount of training data. The system should from time to time relearn and update its models with new data. As with any other business in a hospital employees come and go. From section 1.1 we know that different doctors have different writing styles. When a new doctor writes information in a different way than the classifier expects, the doctor can correct it. Then after the model is updated with the new training data it would correctly interpret what the doctor wrote and extract the information.

Bibliography

[1] H. K. Walker, "The Problem-Oriented Medical System," *JAMA*, vol. 236, no. 21, pp. 2397–2398, 11 1976. [Online]. Available: https://doi.org/10.1001/jama.1976.03270220017024

[2] W. E. Hammond, "Health Level 7: an application standard for electronic medical data exchange," *Top Health Rec Manage*, vol. 11, no. 4, pp. 59–66, Jun 1991.

[3] R. H. Dolin, L. Alschuler, S. Boyer, C. Beebe, F. M. Behlen, P. V. Biron, and A. Shabo (Shvo), "HL7 Clinical Document Architecture, Release 2," *Journal of the American Medical Informatics Association*, vol. 13, no. 1, pp. 30–39, 01 2006. [Online]. Available: https://doi.org/10.1197/jamia.M1888

[4] E. D. Liddy, "Natural language processing," 2001.

[5] K. Gupta, R. Thammasudjarit, and A. Thakkinstian, "Nlp automation to read radiological reports to detect the stage of cancer among lung cancer patients," in *Proceedings of the 2019 Workshop on Widening NLP*, 2019, pp. 138–141.

[6] B. Waltl, G. Bonczek, and F. Matthes, "Rule-based information extraction: Advantages, limitations, and perspectives," *Jusletter IT (02 2018)*, 2018.

[7] K. P. Murphy *et al.*, "Naive bayes classifiers," *University of British Columbia*, vol. 18, p. 60, 2006.

[8] C. C. Aggarwal, *Data classi cation: algorithms and applications*. CRC press, 2014.

[9] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," *arXiv preprint arXiv:1603.01354*, 2016.

[10] F. Hoeijmakers, N. Beck, M. W. J. M. Wouters, H. A. Prins, and W. H. Steup, "National quality registries: how to improve the quality of data?" *Journal of thoracic disease*, vol. 10, no. Suppl 29, pp. S3490–S3499, Oct 2018, 30510784[pmid]. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/30510784

[11] M. A. Hearst, "Untangling text data mining," in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, 1999, pp. 3–10.

[12] K. Kreimeyer, M. Foster, A. Pandey, N. Arya, G. Halford, S. F. Jones, R. Forshee, M. Walderhaug, and T. Botsis, "Natural language processing systems for capturing and standardizing unstructured clinical information: A systematic review," *Journal of Biomedical Informatics*, vol. 73, pp. 14 – 29, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1532046417301685

[13] S. V. S. F, P. L. Hanson, S. S. Bjornsen, and S. A. Smith, "Automatic Classification of Foot Examination Findings Using Clinical Notes and Machine Learning," *Journal of the American Medical Informatics Association*, vol. 15, no. 2, pp. 198–202, 03 2008. [Online]. Available: https://doi.org/10.1197/jamia.M2585

[14] G. Raju, P. Lum, R. Slack, S. Thirumurthi, P. Lynch, E. Miller, B. Weston, M. Davila, M. Bhutani, M. Shafi, R. Bresalier, A. Dekovich, J. Lee, S. Guha, M. Pande, B. Blechacz, A. Rashid, M. Routbort, G. Shuttlesworth, L. Mishra, J. Stroehlein, and W. Ross, "Natural language processing as an alternative to manual reporting of colonoscopy quality metrics," *Gastrointestinal Endoscopy*, vol. 82, no. 3, pp. 512–519, 2015, cited By 23. [Online].

Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84939161556&doi=10.1016%2fj.gie.2015.01.049&partnerID=40&md5=8eb96fb950376fe107515580bf67fb32

[15] S. R. Jonnalagadda, A. K. Adupa, R. P. Garg, J. Corona-Cox, and S. J. Shah, "Text mining of the electronic health record: An information extraction approach for automated identification and subphenotyping of hfpef patients for clinical trials," *Journal of Cardiovascular Translational Research*, vol. 10, no. 3, pp. 313–321, Jun 2017. [Online]. Available: https://doi.org/10.1007/s12265-017-9752-2

[16] S. Pathak, "Automatic structuring of breast cancer radiology reports for quality assurance," August 2018. [Online]. Available: http://essay.utwente.nl/76327/

[17] M. Sargent III, "Unicode, rich text, and mathematics," *Microsoft Corporation*, vol. 19, 2016.

[18] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242. Piscataway, NJ, 2003, pp. 133–142.

[19] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine learning*, vol. 29, no. 2-3, pp. 103–130, 1997.

[20] J. West, D. Ventura, and S. Warnick, "Spring research presentation: A theoretical foundation for inductive transfer," *Brigham Young University, College of Physical and Mathematical Sciences*, vol. 1, p. 32, 2007.

Appendix A

List of DICA audits

Audit	**Disease**
DCRA	Colon cancer
NBCA	Breast cancer
DUCA	Stomach and esophageal cancer
DLCA	Lung cancer
DSAA	Aortic aneurysm
DMTR	Melanome
DACI	Carotis intervention
DHBA	Liver tumors
DPCA	Pancreatic cancer
DGOA	Gynecological oncology
CVAB	CVA Treatment
EPSA	Special child surgery
DHNA	Head and neck cancer
DSSR	Spinal surgery
DATO	Bariatric surgery
DPIA	Parkinson treatment
DBIR	Breat implants
DGEA	Coloscopies
DAPA	Pheripheral arterial disease
DRCE	Endoscopy related complications
DHFA	Hip fractures

Appendix B

Example document from doctor

Leden TWHHT

Datum: 13 11 2017

Kenmerk: 470/M005893/02970075/201707132031

Betreft: mevr. A.B.C. Janssen, 05 02 1969, BSN 123456789
Appelstraat 10
1234 AA Amsterdam

Geachte collega,

BESPREEKFORMULIER HOOFD HALS WERKGROEP

Datum oncologiebespreking : 02 11 2017 TWHHT/UMCU
Hoofdbehandelaar : dr. Jansen specialisme Kaakchirurgie
Case manager : S. Peters
Volgnummer : 2017 111
Versie : 1
Eerste consult : 06 10 2017

Voorgeschiedenis:
Ablatie ivm hartritmestoornissen; sectio 1x. ASA classificatie:

Korte ziektegeschiedenis:
April/mei 2016: Via huisarts naar KNO/ZGT lichen planus tong.
Wegens steken linker oor doorverwijzing naar AVL geen vervolg.
Okt 2016 tandarts doorverwijzing dr. Janssen.
Januari 2017: Partiele excisie tong links waarin een lichen planus. Geen gisten of schimmels aantoonbaar. Nadien onder controle.

Algemene gezondheid:
Lengte: 160 cm Gewicht: 60 kg.

Sociale anamnese:
Getrouwd, 3 kinderen.

Intoxicaties:

Roken:
Alcohol: sociaal
Allergie: nikkelallergie

Medicatie:

Klinisch onderzoek:

CT scan, PET scan, MRI, X thorax, Oesophagoscopie, Laryngoscopie:
Niet verricht.

Echo/Punctie (12 10 2017):
Conclusie: Echografisch geen aanwijzingen voor lymfadenopathie

PA (15 10 2017):
AARD INGREEP: excisie tongrand
ZIJDIGHEID: links (onderzijde)
LOKALISATIE TUMOR: onderrand
TYPE TUMOR: plaveiselcelcarcinoom (oppervlakkig invasief; lastig te onderscheiden van ernstige dysplasie)
DIFFERENTIATIEGRAAD: goed
MAXIMALE DOORSNEDE: ca. 5 mm
INFILTRATIEDIEPTE: ca. 1 mm
SPRIETERIGE GROEIWIJZE: nee
PERINEURALE GROEI: nee
(LYMF)ANGIOINVASIE: nee
MULTIFOCAAL: nee
ONTSTEKINGSREACTIE: matig
RESECTIERANDEN: tumorvrij
MINIMALE MARGE: ca. 3 mm t.p.v. (richting caudaal)
INGROEI IN BOTMERG: niet van toepassing
INGROEI IN SPIEREN: nee
INGROEI IN HUID: nee
CARCINOMA IN SITU COMPONENT BUITEN TUMOR: ja (lastig te onderscheiden van oppervlakkig invasief carcinoom)
CARCINOMA IN SITU COMPONENT RADICAAL: ja
ANDERE BEVINDINGEN: geen
TNM classificatie (7e editie): pT1
TNM classificatie (8e editie): pT1

Stadiring en localisatie:
pT1 PCC laterale tongrand links.

ICD code:
C02.1

Protocol / richtlijn:
Chirurgie.

(Papieren) Bespreking TWHHT UWHHT d.d. 15 10 2017:

Conclusie:
pT1 G1, Pn0, L0, V0, R0 (3 mm t.p.v. caudaal) PCC laterale tongrand onderrand links.

Beleid:
Follow up. Echo FNA 3 maanden.

Aanwezigen:
TWHHT: Jansen, Janssen, Peters, de Vries, de Leeuw,
UWHHT: Rembrand, van Gogh, Appel

Namens de Twentse Werkgroep Hoofd Hals Tumoren.
Mondziekten, Kaak en Aangezichtschirurgie, Keel Neus en Oorheelkunde, Radiotherapie,
Reconstructieve Chirurgie, Radiologie, Nucleaire Geneeskunde, Pathologie, Medische Oncologie,
Chirurgische Oncologie, Logopedie, Di tetiek en Medische Psychologie.

Contactgegevens: tel: 06 12 34 56 78 voor intercollegiaal overleg.

Origineel aan:
Leden TWHHT

Appendix C

Top 10 features for NB and LR

In these tables we see the 10 most important features per category for each item. In the tables on the left you see the top 10 for the Naive Bayes (NB) classifier and the right table for the Logistic Regression (LR) classifier.

C.1 Features for alcohol item

Huidige drinker

Onbekend

Gestopt met drinken

Nooit alcohol gedronken

C.1.1 NSNB/LR

Word	Weight
links	-6,975273
level	-7,150719
twhht	-7,21756
rechts	-7,30677
aantal	-7,354958
d	-7,400264
hals	-7,469938
lymfklieren	-7,630547
umcu	-7,668891
stadium	-7,668975

Table C.1: NSNB for category 'Huidige drinker'

Word	Weight
links	0,899638
nasi	0,475235
dd	0,439179
verdenking	0,416517
glottis	0,416172
vumc	0,410116
long	0,382243
chemoradiatie	0,365254
extranodaal	0,363995
epiglottis	0,341677

Table C.2: NSLR for category 'Huidige drinker'

Word	Weight
rechts	-7,02297
twhht	-7,058717
onderlip	-7,311642
stadium	-7,395833
tumor	-7,436721
d	-7,474735
hals	-7,496361
verruceuze	-7,507743
umcu	-7,543743
hyperplasie	-7,56163

Table C.3: NSNB for category 'Onbekend'

Word	Weight
onderlip	0,98199
verruceuze	0,841859
hyperplasie	0,735537
naresectie	0,579112
slijmvlies	0,453051
verhoornend	0,439116
marge	0,423581
glandula	0,407348
durum	0,399944
trigonum	0,39842

Table C.4: NSLR for category 'Onbekend'

Word	Weight
rechts	-6,671224
mondbodem	-6,712915
aantal	-6,714548
level	-6,79844
st	-6,869391
twhht	-6,941359
lymfklieren	-7,040866
tong	-7,060098
umcu	-7,216504
shunt	-7,231738

Table C.5: NSNB for category 'Gestopt met drinken'

Word	Weight
mondbodem	1,426489
st	1,380592
shunt	0,788088
lap	0,787225
oraal	0,720527
vriescoupe	0,694913
mandibula	0,644074
tablet	0,626889
korsakov	0,587806
pn1	0,552768

Table C.6: NSLR for category 'Gestopt met drinken'

Word	Weight
rechts	-6,553064
level	-6,757749
tong	-6,838479
nasofarynx	-6,93432
twhht	-6,941205
aantal	-6,97948
stapeling	-7,132607
links	-7,137042
fdg	-7,146299
melanoom	-7,14943

Table C.7: NSNB for category 'Nooit alcohol gedronken'

Word	Weight
tong	1,266243
nasofarynx	1,052379
laterale	0,804201
melanoom	0,785346
ct1	0,765721
aantal	0,653073
stapeling	0,631601
rug	0,627713
mg	0,601264
fdg	0,600422

Table C.8: NSLR for category 'Nooit alcohol gedronken'

C.1.2 PSNB/LR

Word	Weight
roken	-4,103833
alcohol	-4,105992
allergie	-4,128543
intoxicaties	-4,228395
dag	-4,241166
sociaal	-4,377606
per	-4,400599
eh	-4,500088
jaar	-4,667568
gestopt	-4,856976

Table C.9: PSNB for category 'Huidige drinker'

Word	Weight
sociaal	1,578336
eh	1,247907
voorheen	0,815077
gerookt	0,804772
allergieen	0,780664
dag	0,755416
abusus	0,649712
3eh	0,593206
pakje	0,566356
bier	0,563225

Table C.10: PSLR for category 'Huidige drinker'

Word	Weight
intoxicaties	-3,602962
allergie	-3,805374
roken	-4,104682
alcohol	-4,202157
gestopt	-4,22155
sinds	-4,297377
fam	-4,551738
gaat	-4,578424
nee	-4,600282
stoppen	-4,602131

Table C.11: PSNB for category 'Onbekend'

Word	Weight
allergie	1,237467
gestopt	1,109561
fam	0,976415
gaat	0,901436
sigaartjes	0,814349
ongeveer	0,814349
medicatie	0,764161
katten	0,763337
honden	0,763337
dexamethason	0,757661

Table C.12: PSLR for category 'Onbekend'

Word	Weight
gestaakt	-3,275425
jaar	-3,708828
daarvoor	-3,878989
fors	-3,890563
diagnose	-3,890563
sedert	-3,912852
gedurende	-3,912852
gebruik	-3,923602
bekend	-4,008419
alcohol	-4,009388

Table C.13: PSNB for category 'Gestopt met drinken'

Word	Weight
gestaakt	2,26404
jaar	1,454124
py	1,145982
daarvoor	1,114613
diagnose	1,085546
fors	1,063573
sedert	0,98068
gebruik	0,933332
gedurende	0,92515
shag	0,841235

Table C.14: PSLR for category 'Gestopt met drinken'

Word	Weight
intoxicaties	-2,904807
nooit	-3,577351
drugs	-3,597931
nee	-3,758776
alcohol	-3,984232
gedronken	-4,093507
aantal	-4,137871
stop	-4,148251
start	-4,148251
gebruikt	-4,158383

Table C.15: PSNB for category 'Nooit alcohol gedronken'

Word	Weight
intoxicaties	2,415333
nee	2,042261
nooit	1,634349
gedronken	1,164543
drugs	0,914872
aantal	0,836584
bekend	0,75025
packyears	0,738525
gebruikt	0,729242
stop	0,545625

Table C.16: PSLR for category 'Nooit alcohol gedronken'

C.1.3 CRF preprocessing

Word	Weight
dag	-2,724246
sociaal	-2,776327
per	-3,033332
eh	-3,146739
week	-3,727614
zelden	-3,732005
sporadisch	-3,732005
3eh	-3,847924
borrels	-3,93733
eenheden	-3,939225

Table C.17: PCNB for category 'Huidige drinker'

Word	Weight
dag	1,574916
sociaal	1,4701
per	1,147557
eh	0,809171
zelden	0,713729
sporadisch	0,713729
week	0,533953
jaar	0,497654
abusus	0,458335
matig	0,425404

Table C.18: PCLR for category 'Huidige drinker'

Word	Weight
nee	-1,261938
drinkt	-3,037552
alcohol	-3,528956
1eh	-4,557775
per	-4,557775
sinds	-4,557775
sedert	-4,557775
s	-4,557775
rode	-4,557775
pilsjes	-4,557775

Table C.19: PCNB for category 'Onbekend'

Word	Weight
nee	2,909672
drinkt	1,708629
avonds	-0,0562
s	-0,0562
ongeveer	-0,0562
whisky	-0,0562
glas	-0,0613
wiskey	-0,0613
glazen	-0,06179
alleen	-0,06261

Table C.20: PCLR for category 'Onbekend'

Word	Weight
nooit	-1,745256
gedronken	-1,745256
alcohol	-1,783878
ongeveer	-4,838997
sedert	-4,838997
s	-4,838997
rode	-4,838997
pilsjes	-4,838997
per	-4,838997
1eh	-4,838997

Table C.23: PCNB for category 'Nooit alcohol gedronken'

Word	Weight
nooit	2,079911
gedronken	2,079911
alcohol	1,769323
huizinga	-0,02104
weekend	-0,02104
madelon	-0,02104
liters	-0,02104
alleen	-0,02104
begeleiding	-0,02104
halve	-0,02104

Table C.24: PCLR for category 'Nooit alcohol gedronken'

Word	Weight
gestaakt	-2,258062
sinds	-2,489713
bekend	-2,971495
fors	-2,971495
gebruik	-2,992159
gedronken	-3,183391
nooit	-3,183391
alcohol	-3,215806
stop	-3,699058
drinken	-3,699058

Table C.21: PCNB for category 'Gestopt met drinken'

Word	Weight
gestaakt	2,333762
sinds	1,686036
fors	1,12403
bekend	1,12403
gebruik	1,033899
drinken	0,879597
stop	0,879597
start	0,783697
gestopt	0,771988
liters	-0,04244

Table C.22: PCLR for category 'Gestopt met drinken'

C.2 Features for smoking item

The possible labels for the smoking item:

Huidige roker

Gestopt met roken

Onbekend

Nooit gerookt

C.2.1 NSNB/LR

Word	Weight
twhht	-7,317091
rechts	-7,354735
links	-7,367849
level	-7,491938
d	-7,580415
aantal	-7,602423
hals	-7,642561
umcu	-7,704624
stadium	-7,706058
conclusie	-7,794079

Table C.25: NSNB for category 'Huidige roker'

Word	Weight
st	0,573203
supraglottisch	0,542474
piriformis	0,490912
achterwand	0,486959
mondbodem	0,475816
mandibula	0,353039
1dd1	0,332692
hpv	0,324523
aangetoond	0,300517
drugs	0,296793

Table C.26: NSLR for category 'Huidige roker'

Word	Weight
twhht	-7,041258
aantal	-7,135805
level	-7,279852
rechts	-7,311496
links	-7,319641
nee	-7,322543
onderlip	-7,380613
stadium	-7,433353
mm	-7,449246
tong	-7,452378

Table C.27: NSNB for category 'Gestopt met roken'

Word	Weight
epitheliaal	0,815961
onderlip	0,685526
nee	0,527057
aantal	0,511966
myo	0,489577
avl	0,454655
maxillaris	0,433849
carcinoom	0,408651
tablet	0,380889
adenoid	0,37612

Table C.28: NSLR for category 'Gestopt met roken'

Word	Weight
links	-6,995673
twhht	-7,191956
aantal	-7,301226
level	-7,320273
d	-7,438477
rechts	-7,509091
pa	-7,559354
wang	-7,577552
nee	-7,58412
umcu	-7,59844

Table C.29: NSNB for category 'Onbekend'

Word	Weight
wang	0,825924
stemband	0,653104
ware	0,554577
vumc	0,487438
vestibulum	0,48183
coupe	0,432902
extranodaal	0,409466
tongrand	0,408374
voorheen	0,401814
invasieve	0,401808

Table C.30: NSLR for category 'Onbekend'

Word	Weight
links	-6,811684
parotis	-7,302767
twhht	-7,317152
tong	-7,426433
hals	-7,51246
d	-7,519611
level	-7,523565
onderlip	-7,598002
umcu	-7,632757
pa	-7,659234

Table C.31: NSNB for category 'Nooit gerookt'

Word	Weight
parotis	0,882327
links	0,854879
onderlip	0,737927
tong	0,634565
sternum	0,504167
coupes	0,497919
o	0,494542
duct	0,435635
paramediaal	0,431021
m1	0,419402

Table C.32: NSLR for category 'Nooit gerookt'

C.2.2 PSNB/LR

Word	Weight
dag	-4,114256
roken	-4,283867
alcohol	-4,344808
per	-4,454292
intoxicaties	-4,454862
allergie	-4,547443
jaar	-4,752644
py	-4,831104
drugs	-4,953896
eh	-5,003883

Table C.33: PSNB for category 'Huidige roker'

Word	Weight
dag	1,235669
py	1,045548
start	0,671256
allergieen	0,615122
packyears	0,53964
sigaretten	0,482957
heden	0,466704
per	0,46078
diagnose	0,458497
stoppen	0,416821

Table C.34: PSLR for category 'Huidige roker'

Word	Weight
intoxicaties	-3,488072
nee	-3,888358
allergie	-3,943906
roken	-4,056319
alcohol	-4,101222
nikkelallergie	-4,303284
drugs	-4,43972
bekend	-4,478215
dexamethason	-4,536148
sociaal	-4,571079

Table C.35: PSNB for category 'Gestopt met roken'

Word	Weight
nikkelallergie	1,513886
dexamethason	1,185118
drugs	1,145243
intoxicaties	0,941691
bekend	0,913915
nee	0,812302
pleisters	0,45279
matig	0,433414
s	0,416671
nsaid	0,416671

Table C.36: PSLR for category 'Gestopt met roken'

Word	Weight
roken	-4,175578
sinds	-4,241437
gestopt	-4,25849
jaar	-4,265447
allergie	-4,275557
alcohol	-4,296715
intoxicaties	-4,319262
medicatie	-4,42861
per	-4,557224
dag	-4,63166

Table C.37: PSNB for category 'Onbekend'

Word	Weight
medicatie	1,600157
gestopt	1,336997
sinds	1,31017
jaar	0,932513
voorheen	0,887656
sporadisch	0,844942
jr	0,830117
3eh	0,669939
geleden	0,643572
neen	0,591575

Table C.38: PSLR for category 'Onbekend'

Word	Weight
intoxicaties	-3,624343
nooit	-3,795289
rookt	-3,963812
alcohol	-4,060896
nee	-4,062116
pat	-4,070359
gebruik	-4,129671
allergie	-4,259063
sociaal	-4,322809
roken	-4,440638

Table C.39: PSNB for category 'Nooit gerookt'

Word	Weight
Word	Weight
nooit	2,097931
rookt	1,509842
pat	1,184299
nee	1,116014
gerookt	1,083273
gebruik	0,966542
intoxicaties	0,949829
drinkt	0,743527
eh	0,730521
stof	0,633659

Table C.40: PSLR for category 'Nooit gerookt'

C.2.3 CRF preprocessing

Word	Weight
dag	-2,851726
per	-3,084008
packyears	-3,441409
jaar	-3,487154
sig	-3,583317
sigaretten	-3,708803
pack	-3,719257
years	-3,719257
start	-3,748509
py	-3,752109

Table C.41: PCNB for category 'Huidige roker'

Word	Weight
dag	1,716769
packyears	1,035284
per	0,989846
py	0,707216
sigaretten	0,675861
25pky	0,602312
start	0,56946
sig	0,51617
pack	0,493588
years	0,493588

Table C.42: PCLR for category 'Huidige roker'

Word	Weight
nee	-2,154665
14e	-4,234107
sigaren	-4,234107
packyears	-4,234107
pakje	-4,234107
pakjes	-4,234107
per	-4,234107
pijp	-4,234107
py	-4,234107
roken	-4,234107

Table C.43: PCNB for category 'Gestopt met roken'

Word	Weight
minimaal	-0,037954
echt	-0,037954
14e	-0,060869
tussentijds	-0,060869
stuks	-0,085197
dagen	-0,104214
lange	-0,104214
fors	-0,106565
ongeveer	-0,115216
pijp	-0,115229

Table C.44: PCLR for category 'Gestopt met roken'

Word	Weight
jaar	-2,526051
gestopt	-2,534659
sinds	-3,201743
geleden	-3,373111
roken	-3,439831
jr	-3,580794
pakjes	-3,91743
week	-3,91743
per	-3,929614
voorheen	-3,938063

Table C.45: PCNB for category 'Onbekend'

Word	Weight
gestopt	2,134156
jaar	2,044522
roken	1,014936
geleden	0,88256
sinds	0,872614
jr	0,710215
voorheen	0,696087
neen	0,670805
gestaakt	0,65109
pakjes	0,610005

Table C.46: PCLR for category 'Onbekend'

Word	Weight
nee	-1,482876
nooit	-2,516793
gerookt	-2,600609
rookt	-3,221994
ongeveer	-4,608288
packyears	-4,608288
pakje	-4,608288
pakjes	-4,608288
per	-4,608288
pijp	-4,608288

Table C.47: PCNB for category 'Nooit gerookt'

Word	Weight
nee	2,472293
nooit	1,936217
gerookt	1,615649
rookt	1,541206
tussentijds	-0,030362
14e	-0,030362
echt	-0,033408
minimaal	-0,033408
dagen	-0,043221
lange	-0,043221

Table C.48: PCLR for category 'Nooit gerookt'