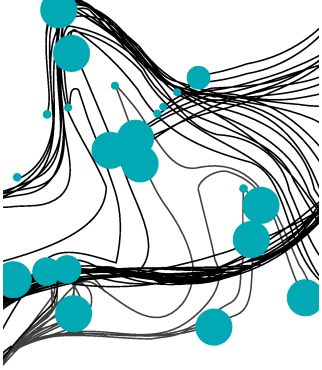
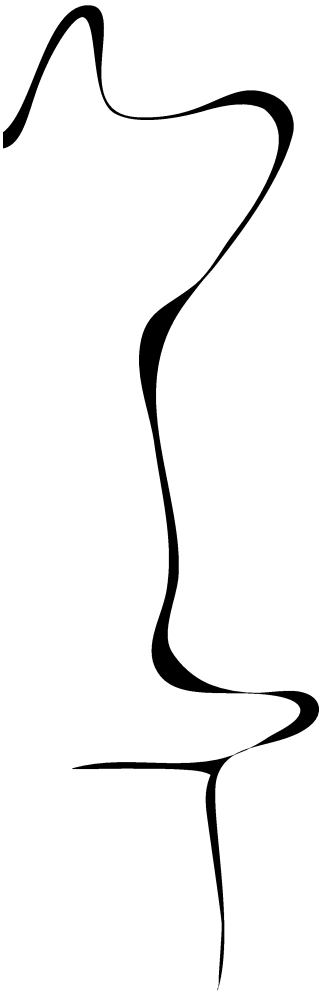


DMB

DATABASE MANAGEMENT
AND
BIOMETRICS

.94487



EXPLAINABLE DIAGNOSES PREDICTION FOR GENERAL PRACTITIONERS

Pieter Zeilstra

MASTER'S ASSIGNMENT

Committee:
dr. ir. Maurice van Keulen (Comittee chair)
dr. Shenghui Wang
Max de Rooij (External from Topicus)
Pien Bouman (External from Digidok)

June, 2023

2023DMB0006
Data Management and Biometrics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Abstract

In this study, a Dutch variant of the RoBERTa language model known as RobBERT was made domain-specific to the domain of Dutch general practitioners(GPs). The model was trained and fine-tuned using 2.2 million user-identified symptoms (S-rules) derived from SOAP notes(Subjective, Objective, Assessment and Plan).

Spreekuur.nl is an online consultation application that requires users to complete a questionnaire before participating in an online consultation. A GP can review the answers to the questionnaire to help them diagnose the patient. Currently, full agendas and heavy workloads burden Dutch GPs because they are the primary point of contact for receiving healthcare in the Netherlands. This study aimed to help alleviate the workload of GPs by predicting diagnoses based on the answers to a questionnaire.

First, the questionnaire data was converted into the text format of an S-rule(user-identified symptoms), resembling an S-rule written by a Dutch GP. The S-rule specifically concerns the **S**ubjective symptoms and patient's narrative without a further medical examination. An S-rule is part of a SOAP note which is the documentation standard for Dutch GP to document a consultation.

ICPC codes are a standard GPs use in consultations to standardise and document a patient's diagnosis and are present in each SOAP note. ICPC codes were used to classify a diagnosis based on the S-rule.

A new classification head was introduced, enabling the separate classification of ICPC symptom codes and ICPC disease/diagnosis codes.

To align with the diagnostic decision-making process of GPs, a threshold function was implemented to determine the number of ICPC codes returned. The threshold function outperformed the simple approach of returning only the top three codes while providing fewer than three codes on average. When predicting ICPC symptom codes, the threshold function achieved an accuracy of 90%; for ICPC diagnosis/disease codes, the accuracy was 88.6%. Notably, when evaluating the model's performance on the generated S-rules from the questionnaire dataset, the accuracy was 87.6% for ICPC symptom codes and 86.5% for ICPC diagnosis/disease codes. A LIME symptom-module was proposed. The symptom-module is an adaptation of the LIME text-module. The symptom-module generates various samples of a text by removing words and tokens. The model then generates output probabilities for these samples, and the probability changes are utilised as training data for a white-box model. The white-box model aims to identify the most important symptoms for each ICPC diagnosis/disease code.

Furthermore, a user study was conducted with five participating general practitioners, of which four participants found that the model contributed to their diagnostic ability by suggesting ICPC codes. Three participants found the explanations generated from the symptom module to improve their diagnostic ability, while further fine-tuning is needed.

Keywords— Diagnosis prediction, Language model, ICPC codes, SOAP notes, LIME symptom module

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Worldwide, ageing populations, pandemics such as COVID-19, and the economy are slowly encumbering countries' healthcare systems[4][5]. Currently, full agendas and heavy workloads burden Dutch GPs because they are the primary point of contact for receiving healthcare in the Netherlands[6]. The pandemic has had its upsides, during the pandemic, new technologies and ideas emerged for taking online consultations[7].

One of these techniques was the *Spreekuur.nl*[1] website. *Spreekuur.nl* is an online triage consultation tool created by *Topicus*[2] and *DigiDok*[3]. The website lets users fill in a question-by-question questionnaire regarding their health, area of complaint and symptoms. At the end of the questionnaire, the user is redirected to a self-help website called *Thuisarts.nl*[4] if their health complaint can be handled by themselves, redirected to a hospital if their health complaint is sufficiently dangerous, or lastly, the user is invited to an online chat with their general practitioner. When the GP accepts the chat, the GP can see a summary of the answers to questions in the questionnaire. *DigiDok* created these questionnaires in regulation with the NHG("Dutch general practitioners association")[8] and NTS("Dutch Triage Standard")[9]. The questionnaires are validated by a team of expert GPs, professors and triagists, giving improvements and commentary. *Topicus* is the developer and creator of *Spreekuur.nl* and other applications including *VIPLive*[5] and *SpoedEPD*[6]. These applications are used nationwide by GPs and healthcare organizations in the Netherlands. *Topicus* has the unique position to use a large amount of data and knowledge gathered, which are normally not available to the general public.

Topicus and *DigiDok* want to use *Spreekuur.nl* to further alleviate Dutch GPs by presenting a list of predicted diagnoses to them based on the answers acquired from the questionnaire. Machine learning is already being used for medical applications across different fields[10] and has state-of-the-art performance in for example cardiology[11], diagnostic imaging[12] and disease prediction[13]. A diagnoses prediction model can help practitioners by finding correct diagnoses codes more efficiently and serve as an extra helping hand when conducting a differential diagnosis.

A major hurdle in machine learning and especially in the healthcare sector, is explainable artificial intelligence(XAI)[14][15]. Best-performing models often use deep learning techniques to predict and learn, which is not readable to humans. Deep learning models are considered a black box. In healthcare, experts want to know for certain if a model is making predictions for the right reasons and if the network is not "cheating" or taking "shortcuts"[16]. The predicted diagnoses should be substantiated with explanations and reasons, such as correlated symptoms. XAI helps GPs understand how the model came to these conclusions and helps GPs crosscheck predictions with their knowledge to make a final diagnosis.

[1] https://Spreekuur.nl
[2] https://Topicus.nl
[3] https://Digidok.nl
[4] https://Thuisarts.nl
[5] https://viplive.nl
[6] https://viplive.nl/viplive-voor-u/huisartsenposten/spoed-epd

The open-ended questions in the questionnaire for patients to report the progression of their health complaints offer valuable information. Hence, a RoBERTa Language model was adapted and trained in this study. The primary research question is thus formulated as follows: **"How can a RoBERTa language model be used for predicting diagnoses based on patient-reported symptoms?"** An additional research question aims to identify the most effective approach for explaining the diagnoses prediction model's generated predictions to GPs. The secondary research question is: **"Which XAI method is most effective at explaining the predictions of a diagnoses prediction model to Dutch general practitioners?"**
In this study, a Dutch variant of the RoBERTa model is trained called RobBERT[17] on 2.2 million user-identified symptoms (S-rule), which were present in written SOAP notes. ICPC codes are used by GPs in consultations to standardize and document a patient's diagnosis; hence they are used as a ground truth for training en predicting diagnoses. To obtain the diagnosis of the health complaints when a patient filled in a questionnaire, 17 thousand filled out questionnaires were connected to the SOAP notes. These 17 thousand questionnaires were converted from tabular data into a textual form mimicking the S-rule in written SOAP notes. One of the key challenges encountered in this study was implementing a hierarchical structure to predict ICPC codes. Multiple ICPC codes may be appropriate for a given S-rule, and GPs may assign different codes for the same S-rule. The variability in ICPC codes reduces the learning ability of the model. It impacts the performance measures of the model, as a conventional loss function does not account for the possibility of multiple appropriate codes for a given S-rule.

To address this variability in ICPC codes, a modification was made to the classification head of the RoBERTa model by creating a separation between symptoms and diagnoses/diseases. The modification allowed the model to predict both ICPC symptom and ICPC diagnosis/disease codes separately. The modification could predict 318 ICPC symptom codes and 361 ICPC diagnosis/disease codes. After training, the model obtained an overall accuracy of **51.4%**. A new performance measure was added: the top-3 accuracy for predicting ICPC symptom codes was **88.6%**, and the top-3 accuracy for predicting ICPC diagnosis/disease codes was **87.5%**. A threshold function was also added, which more efficiently chose how many ICPC codes it should suggest to the GP. The threshold function got an accuracy of **90%** for suggesting ICPC symptom codes and **88.6%** for suggesting ICPC diagnosis/disease codes.

To improve the explainability of the model, a new "symptom-module" is added to LIME[2]. The symptom-module utilised changes in the text to identify which symptoms significantly influenced the model when predicting ICPC diagnosis/disease codes. To validate the model's performance and to measure the interpretability of the explanations, a user study was conducted with five GPs. Each GP rated the suggested ICPC codes and explanations of the models on ten different real test cases. The user study emphasised the key challenge of this study by showing the variability of ICPC codes. The five GPs were first shown five of the same cases and chose 13 different ICPC codes between them. The user study concluded that the diagnoses prediction model and explanation helped the participants' diagnostic ability. The main improvement is in optimising and fine-tuning the explanations and suggested ICPC codes to reduce the number of mismatched keywords, symptoms and suggested ICPC codes. The GP chose a suggested code in **90.0%** of the cases, which signifies that the actual performance of the model is higher than the standard top-3 performance measure may indicate.

1.1 Outline

The outline of this study is as follows:
First, a description of relevant information about the Dutch healthcare system and general practitioners is provided in chapter 2: "Background". In chapter 3: "Related works", relevant papers to this study are explored. chapter 4: "Source data", provides a description and analysis of the source data. The Research questions are formulated in chapter 5: "Research questions". chapter 6: "Methodology", provides the methodology of this study. chapter 7: "Dataset" prepares the dataset for training the model. chapter 8: "Diagnoses prediction model", describes the model's training and validation. chapter 9: "Explainability" adds a new explainability method to explain a diagnosis using symptoms and shows the results of the user study. The results are discussed further in chapter 10: "Discussion" with a brief overview for future works. Finally, a conclusion is found in chapter 11: "Conclusion", which answers all research questions.

1.1.1 Walkthrough guide

- For readers who prefer **a concise overview**, it is recommended to read the following chapters: chapter 8: "Diagnoses prediction model", chapter 10: "Discussion", and chapter 11: "Conclusion". These chapters provide an overview of the study's main findings.
- **Healthcare professionals** are encouraged to at least read chapter 4: "Source data" for its data analysis on ICPC codes, chapter 9: "Explainability" for an explanation of how the output of the machine learning model is used to explain a diagnosis, and *in particular*, section 9.2: "User study" that validates the model's explanations and ability to predict ICPC codes.
- For those **interested in the technology** behind the study, it is recommended to begin with chapter 2: "Background" to understand how the Dutch healthcare system works. chapter 3: "Related works" provides context on how chapter 6: "Methodology" is substantiated. chapter 8: "Diagnoses prediction model" demonstrates how the RoBERTa model is implemented with a new classification head, and its performance and validation. Finally, chapter 9: "Explainability" explains how the last linear layer of neurons can be split up to explain a model via symptoms and diagnosis. chapter 11: "Conclusion" concisely answers all research questions.

Chapter 2

Background

Healthcare systems vary widely per country; this section looks into the Dutch healthcare system and Dutch GP standards. The “Gatekeeping principle” and “triage” of the Dutch healthcare system will be explained. Dutch GP standards such as differential diagnoses, SOAP, and ICPC codes will be described.

2.1 Dutch general practitioners

The Dutch healthcare system differs from other systems worldwide, with a key characteristic being the “gatekeeping principle”. Under this principle, patients must be referred by their general practitioner (GP) to receive hospital or specialist care. This principle makes primary care in the Netherlands more dominant than in other countries. Dutch GP practices can be divided into two categories: a general practice that is open during the day and handles non-urgent care, and a general practice emergency centre that is open during the night and weekends and handles urgent care cases when a regular general practice is not available or unable to meet the patient’s medical needs.
Dutch GPs have broader profiles and specialities because of the “gatekeeping principle”, which results in most visits not ending with a referral to a hospital or specialist[18]. Dutch GPs are overworked because they need to handle more patients in the same amount of time, making consultation time per patient shorter[19][20].
In the following sections, it will be explained how triage is performed, how a Dutch GP finds a differential diagnosis and how consultations are documented in a *EPD* using *SOAP* notes and *ICPC* codes.

2.1.1 Triage

When calling a GP practice, GP emergency centre or hospital for a health-related issue, a triagist or doctor’s assistant will pick up the phone. The triagist or doctor’s assistant will take the caller through a triage process to calculate the urgency of the caller’s health problem. Triage consists of a series of questions and measures the degree of urgency in patients. Triage causes more severe cases to urgent care sooner by applying a “Treat first what kills first” or ABCD policy[21]. Most countries have their own triage standards or use the International Triage Standard (ITS)[22]. The healthcare system and GPs in the Netherlands use *NTS*[9], which consists of five steps:

1. ABCD-safe - Are complaints life-threatening?
2. Category of complaint
3. Determination of urgency
4. Follow-up action for patient
5. Advice for patient

There are six different urgency levels in the *NTS*:

1. **U0**: Failure of ABCD - reanimation needed.

2. **U1**: Unstable ABCD - Directly life-threatening (seek care immediately).
3. **U2**: Threatening ABCD or organ damage (seek care as soon as possible).
4. **U3**: Reasonable chance of damage (seek care within a few hours).
5. **U4**: Negligible chance of damage (seek care within 24 hours).
6. **U5**: No chance of damage.

DigiDok modified *NTS* to create questions in the questionnaire. Only five of six urgency levels are used because a U0 patient cannot complete a questionnaire. The first questions in *Spreekuur.nl* will always make sure the patient is ABCD-safe. In the case of ABCD-unsafe, the patient will be asked to seek immediate help. During the questionnaire, urgency levels are calculated to help patients in need of urgent care and to alleviate the workload of GPs by giving self-care advice to lower urgency patients.

2.1.2 Differential diagnosis

During a consultation with a patient, the practitioner will try to find a differential diagnosis[23]. To find an appropriate diagnosis, GPs will first listen to medical information, health complaints, and symptoms according to the patient. Practitioners will ask more in-depth questions based on given complaints. This process is called anamnesis and offers practitioners an idea of what actions must be taken next. Practitioners will take a medical examination of the patient, if necessary, based on the anamnesis. Now, a differential diagnosis can be made by practitioners. Practitioners factor in each symptom and evidence from anamnesis and examination to argue for and against diagnoses and correlate them to the most probable diagnosis. Typically the differential diagnosis consists of between one and three different potential diagnoses. The differential diagnosis requires experience and expertise and requires GPs to weigh in medical calculations, per-patient differences, and personal intuition[24]. In cases where a GP is unsure about the diagnosis, a less specific diagnosis or symptom is given instead. With the diagnosis, a prognosis can be given. The diagnosis and prognosis can be aided by the use of resources such as *NHG*[8] and *Thuisarts*[25]. These resources provide guidelines, anamnesis questions, and other relevant information to help with the diagnosis and prognosis.

2.1.3 SOAP notes

The previous section explains how a GP traverses a consultation to obtain a differential diagnosis and prognosis. Consultations are documented in an electronic health record(EHR) or *EPD* in Dutch. *EPD*s serve as a comprehensive health record containing a patient's medical history. A patient's *EPD* can be made available across different healthcare specialists, given the patient's permission. Consultations and diagnoses must be documented precisely and uniformly to keep them readable and organised for multiple organisations and to combat information loss. Dutch healthcare professionals and GPs use a standard called *SOAP*[26] to keep the *EPD* organised and readable. After a consultation, a *SOAP*(or *SOEP* in Dutch) note is documented in the *EPD*. *SOAP* is an abbreviation for **S**ubjective, **O**bjective, **A**ssessment and **P**lan.

1. **S**ubjective: Information and symptoms according to a patient.
2. **O**bjective: Objective symptoms based on medical examinations.
3. **A**ssessment: Differential diagnosis/diagnoses according to GP or health professional.
4. **P**lan: Prognosis and medical advice based on differential diagnosis.

Components/ chapters	A	B	D	F	H	K	L	N	P	R	S	T	U	W	X	Y	Z
1. Symptoms and complaints																	
2. Diagnostic, screening and prevention																	
3. Treatment, procedures and medication																	
4. Test results																	
5. Administration																	
6. Other																	
7. Diagnoses, diseases																	

Table 2.1: ICPC structure and component description.

2.1.4 ICPC codes

A SOAP note is a standardised method to document the details of a patient's visit to a healthcare provider. One challenge with SOAP notes is that different practitioners may interpret them differently. To address this issue, the **W**orld **H**ealth **O**rganisation (WHO) introduced the **I**nternational **C**lassification of **P**rimary **C**are (ICPC) as a standardised system of codes for classifying symptoms and diagnoses in primary care settings. *ICPC* codes have a biaxial structure and consist of 17 chapters; each chapter is divided into seven components as seen in Table 2.1 and Table 2.2[27]. *ICPC* codes cover a wide field of symptoms and diagnoses such as A03 Fever, N93 Carpal tunnel syndrome or Z01 Poverty/financial problems. ICPC codes make further distinctions per code by introducing a hierarchical structure, for example, A96 Death, A96.01 Natural death and A96.02 Unnatural death. *ICPC* codes are used in many healthcare and GP systems worldwide. Dutch healthcare practitioners use their version of ICPC codes, updated regularly by the *NHG*. The version of *NHG* biggest difference is that it only has three components: 1) Symptoms and complaints, 2) Diagnostic, screening and prevention and 3) Diagnoses and diseases. The **A**ssesment of a *SOAP* note typically contains a maximum of one *ICPC* code. The ICPC code or name of the complaint/diagnosis can be inputted onto sites used by Dutch GPs such as *NHG*[8] to help to find differential diagnosing and prognosing a patient or to find an ICPC code.

Letter	Chapter description
A	General and unspecified
B	Blood and blood-forming organs and immune mechanism
D	Digestive
F	Eye
H	Ear
K	Circulatory
L	Musculoskeletal
N	Neurological
P	Psychological
R	Respiratory
S	Skin
T	Endocrine, metabolic and nutritional
U	Urological
W	Pregnancy, childbearing, family planning
X	Female genital
Y	Male genital
Z	Social problems

Table 2.2: ICPC chapters and description.

Chapter 3

Related works

In this section, research relevant to this paper will be laid out. Relevant research includes XAI methods and papers regarding diagnosing patients based on symptoms.

3.1 Symptom checkers

Over the past ten years, there has been an increase in the number of online symptom checkers. Symptom checkers allow users to input their symptoms in various formats, such as free text, a list, or a questionnaire, and provide a list of possible diagnoses, advice, and urgent recommendations. This research shares a notable correlation with symptom checkers, as both focus on the diagnosis of patients based on symptom interpretation. In this study, it is crucial to thoroughly examine the existing research conducted on symptom checkers, as it serves as a valuable source of inspiration. This includes using such research for validation purposes and establishing a baseline against which the findings of this study can be compared.

Symptom checkers often include disclaimers stating that they are “for informational purposes only”[28][[29]] and not intended to constitute professional medical advice, diagnosis, or treatment. The disclaimer allows symptom checkers to avoid legal trouble if a diagnosis is incorrect. However, these systems still provide diagnoses and recommendations[30]. Symptom checkers and the research underlying them are often proprietary, making it difficult to do analysis.

Several papers have compared the performance of symptom checkers. Several methods have been used, to measure the performance, such as comparing the diagnoses of a hand surgeon to those of symptom checkers when provided with the same symptoms[31]. Another widely used method involves using standardised medical vignettes, which are typically used as examination material for doctors in training[32][33][34]. Vignettes contain a description of symptoms and a ground truth. They are used as input to symptom checkers by one or more doctors for validation. Each study uses different vignettes, as there is no standard for evaluation, and each symptom checker takes different inputs. These different performance measures result in different conclusions about which symptom checker performs best.

Most studies evaluate whether the ground truth is the first predicted diagnosis (accuracy-at-1) or is among the top-5 predicted diagnoses (accuracy-at-5). Ceney et al. (2021)[32] measured accuracy using a modified version of the 45 vignettes used in the study by Semigran et al. (2015)[33] and looked at the number of questions needed to have the ground truth diagnosis listed first. For example, *Ada*[35] had an accuracy of 72% with an average number of questions needed of 45.8. The worst-performing symptom checker had an accuracy of 22% and required an average of 9.5 questions to reach a conclusion.

Wallace et al. (2022)[30] conducted a systematic review of the performance of symptom checkers by comparing ten different studies on the topic. The review found that the accuracy of diagnosis and triage among symptom checkers varied significantly between studies and overall had low accuracy. Additionally, the review noted that symptom checkers are not regulated, and many studies did not specify performance measures. These findings highlight the need for standardised evaluation methods, as previously suggested by Painted et al. in their research on the topic[36].

3.2 From text to machine-readable data

This study chose to utilise the BERT (Bidirectional Encoder Representations from Transformers) language model for to diagnose patients on based on their answers in a questionnaire. chapter 6: "Methodology" provides detailed insights into the rationale behind this model selection. Additionally, the related work presented in this section describes the underlying principles of language models, the development of BERT, and its capacity to comprehend natural language.

Words can give humans an immediate impression if the context is known. "Wampimuk", is a non-existent word with many different definitions without context. Nevertheless, given an example sentence: "The Wampimuk climbs in the tree", the human mind can imagine a definition based on this context. A machine learning model does not have this human intuition by default, as a word or sentence is just byte data. **N**atural **L**anguage **P**rocessing (NLP) is the field of artificial intelligence concerned with the processing and analysis of natural language data. In the following sections, multiple NLP techniques are discussed.

3.2.1 RNNs

Recurrent **N**eural **N**etworks (RNNs) are a type of artificial neural network that takes each word in a sentence as a separate input, allowing the model to process sentences of any length. RNNs use the current input and previous hidden states to compute the next hidden state, allowing them to capture context across multiple steps. However, RNNs can suffer from the vanishing gradient problem during training, where the gradient used to update the network's weights shrinks as it backpropagates through time. The vanishing gradient problem results in insufficient weight updates and poor performance. To solve this problem, long short-term memory(LSTM)[37] and Gated recurrent unit(GRU)[38] were introduced. LSTM, initially proposed in 1997, can be seen as a cell that consists of three gates: 1) An input gate that controls the input of information at each step, 2) An output gate controls the output of information, and 3) a forget gate determines which data can be forgotten. GRU, proposed in 2014, is a small neural network at the output of each step with three layers: 1) the recurring layer from the RNN, 2) a reset gate and 3) an update gate which acts as a coupled version of three LSTM gates. These methods allow RNNs to carry context along multiple steps allowing for good performance at NLP tasks. RNNs with LSMT or GRU units resulted in an excellent performance for NLP tasks due to their memory but have the issue that it becomes harder to compute or train for longer word sequences due to sequential calculation.

3.2.2 Attention head

In recent years, significant advancements have been made in NLP, specifically in developing machine-learning models that address the limitations of sequential processing.

One key concept in these advancements is the use of attention mechanisms, which play a crucial role in NLP tasks. Attention heads are components that calculate the attention of each word in a sentence. The attention layer processes each word simultaneously and produces a vector for each word, indicating the relative importance of other words in the sentence to that word.

Each word is pre-processed with a semantic embedding which gets multiplied by three different weights vectors to calculate a query, key and value vector.

To calculate the attention of a specific word, the query vector associated with that word is multiplied by each key vector. The resulting values are then scaled and subjected to a softmax operation, which produces scores representing the relevance of each word to the selected word. These scores are multiplied with all value vectors to calculate the attention. Figure 3.1 illustrates this process,

where q represents the calculated query vector for a specific embedding, and K and V represent all calculated key and value vectors for all embeddings. This calculation can be simplified in the formula $A(Q,K,V) = \text{softmax}\left(QK^T\right)V$. For a visual representation of this process, refer to Figure 3.2.

To further enhance the attention mechanism, this layer can be copied multiple times with different sets of trainable weight vectors for the key, query, and value vectors. This approach is known as Multi-head attention, and it allows the model to capture different aspects of the input data and attend to multiple relevant parts simultaneously.

$$A(q,K,V) = \sum_i \frac{e^{q\cdot k_i}}{\sum_j e^{q\cdot k_j}} v_i$$

Figure 3.1: Attention equation for calculating the attention of a specific embedding using the corresponding query vector and all key and value vectors in the sentence.

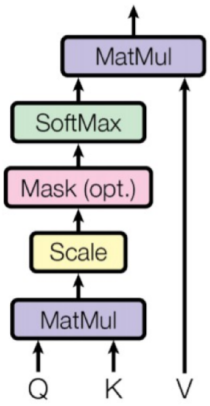


Figure 3.2: Example from Vaswani et al.[1] showing the scaled dot product attention used in a transformer.

3.2.3 Transformer

In 2017, Vaswani et al.[1] introduced the transformer model, which is a neural network architecture consisting of an encoder-decoder framework built upon the concept of Multi-head attention.

The encoder component utilises a Multi-head Attention layer, enabling simultaneous processing of every word in a sentence. To achieve this, each word in the sentence is first provided with a semantic embedding and positional encoding. Subsequently, the Multi-head Attention layer processes all words concurrently. It generates a vector for each word, representing the importance of other words in the sentence with respect to that word. These output vectors are then concatenated and passed through a trainable feed-forward network. If no useful information is provided, the previous outputs are always added to the Multi-head Attention or feed-forward network output and normalised, which is called a residual connection.

The decoder component of the transformer model allows it to perform tasks like language translation. It follows a similar process as the encoder component but adds a Masked Multi-head attention layer. The Masked Multi-head attention layer uses the same input sentence correctly translated into another language as training data. During training, the model is trained by masking subsequent sequence elements of translated sentences using the Masked Multi-Head attention layer. The output of the encoder is used as part of the input of next the Multi-head attention layer to add the representation of the original sentence. The decoder component essentially performs next-word prediction to

train the transformer model, where it is tasked with predicting the next word of the masked sentence. The output representation vector serves as input for a neural network, often with an output size corresponding to the vocabulary used. The output neuron with the highest value represents a word in the vocabulary, indicating the next word in the sentence. Figure 3.3, sourced from the original paper by Vaswani et al., provides an illustration of the encoder and decoder components of the transformer model. Encoders and decoders can be stacked multiple times, with the original paper employing six encoders and six decoders.

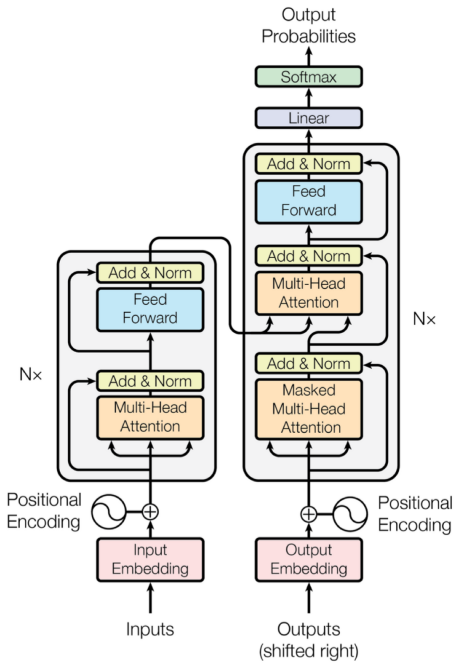


Figure 3.3: Example from Vaswani et al.[1] showing the encoder/decoder structure of a transformer.

3.2.4 BERT

The encoder-decoder structure of the transformer model inspired the development of BERT (2018)[39], which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. BERT is designed to pre-train deep bidirectional representations of natural language data by joint conditioning on all layers' left and right contexts as seen in Figure 3.4.

BERT uses twelve encoder components from the transformer architecture. BERT is trained bidirectionally, resulting in a deeper understanding of language context. Each transformer-encoder layer consists of a Multi-attention head and a feed-forward network. The input sentence is first tokenized into an embedding vector using a trainable tokenizer such as WordPiece. BERT uses a **M**asked **L**anguage **M**odel (MLM) pre-training objective, where some tokens in the input are randomly masked, and the model is trained to predict the original token based on its context. MLM allows the model to fuse the left and right contexts and pre-train a deep bidirectional transformer from unlabelled text data.

In addition to the MLM objective, BERT uses a "next sentence prediction"(NSP) training objective, where sentence pairs are used to predict whether the second sentence is the corresponding next sentence of the first sentence.

BERT is pre-trained on a 3.3 billion word corpus from BookCorpus and Wikipedia. The resulting representations can be used with an additional sequential layer or feed-forward network for NLP

tasks such as general language understanding, question answering, and sentence pair completion while only requiring fine-tuning on labelled data.

BERT achieved state-of-the-art performance on multiple NLP benchmarks, with successors such as RoBERTa (2019) [40]. Among other changes, the most significant change RoBERTa made to the BERT architecture was removing the “next sentence prediction” task in BERT. The authors of RoBERTa show in their paper that the NSP task does not increase performance and even get higher performance with the task removed.

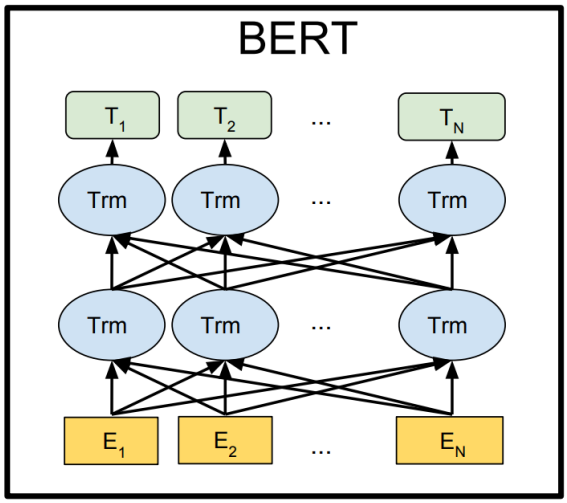


Figure 3.4: Example from Devlin et al. al.[1] showing the architecture of BERT.

3.3 Diagnosis Prediction using machine learning

In this research it was chosen to utilise the BERT language model. However, it is imperative to thoroughly investigate related literature concerning the prediction of diagnoses both with and without the utilisation of language models. Furthermore, this section aims to explore the specific adaptations made to the BERT model to enhance its predictive capabilities of diagnoses.

There has been significant research on using machine learning to predict diagnoses. Many medical datasets are publicly available, and machine learning can be used to uncover patterns in these datasets that may not be readily apparent to humans. In many cases, research on diagnosis prediction focuses on predicting a single disease or diagnosis using binary output or probability predictions (where 1 indicates the presence of the disease and 0 indicates the absence of the disease). However, this study focuses on the problem of predicting many diagnoses simultaneously. In this section, existing research will be reviewed on diagnosis prediction and discuss how these approaches can be applied to the problem of multi-diagnosis prediction.

Medicine is one of the oldest research fields, with millions of studies now available online from resources such as *PubMed*[1]. A machine learning algorithm is typically built from scratch and trained on a dataset. Zhou et al.(2014)[41] used a large-scale biomedical literature database to construct a symptom-based human disease network. They looked at diseases and symptoms in the *MeSH* terms of over 800 thousand studies. From these terms, symptom–disease relationships could be extracted, resulting in 147,978 connections between 322 symptoms and 4,219 diseases. This research also highlighted a challenge in diagnosis prediction, as many symptom-disease relationships are nearly

[1] https://pubmed.ncbi.nlm.nih.gov/

identical. Diseases and symptoms have three causal structures: symptom *S* can be a **direct** cause of disease *D*, *D* does not directly cause symptom *S* but is correlated to a common cause *R* or third, *S* is a direct cause of *D* with addition to a latent common cause *R* is also present. Richens et al. (2020)[42] used these causal structures to derive a counterfactual diagnostic algorithm. A counterfactual is much like a contrastive. In contrastive explanations, a comparison of two outputs demonstrates why an outcome occurred. On the other hand, counterfactual explanations use the current output to explain why outcomes did **not** occur. Richens et al. show that existing diagnosis prediction approaches are based on association and suffer from sub-optimal and dangerous diagnoses. Their approach, however, identifies diseases that correlate most with a patient's symptoms. A Bayesian Network(BN) is used, which models relationships between hundreds of diseases, risk factors, and symptoms as BNs are interpretable and explicitly encode causal relations between variables. Diseases, symptoms, and risk factors are binary nodes and can be either on or off.
Their counterfactual algorithm is trained on symptoms extracted from medical vignettes. It uses a diagnostic measure for ranking the likelihood that a disease *D* is causing a patient's symptoms given evidence *E*. The diagnostic measure looks at the number of symptoms that need to be switched off to cure *S* and the number of symptoms that would persist if all other causes of the patient's symptoms are switched off. The model achieved expert clinical accuracy. The diagnostic measure correlates strongly with how a doctor procures a diagnosis. While the doctors achieved an average diagnostic accuracy of 71.40%, the model achieved an average accuracy of 77.26%, placing in the top 48% of doctors in their cohort.

BioBERT (2019)[43], is a domain-specific BERT model pre-trained from scratch on Pubmed and PMC biomedical text data (18 billion words in total). BERT already achieves state-of-the-art performance on biomedical tasks such as biomedical named entity recognition, biomedical relation extraction, and biomedical question answering. BioBERT further improves upon its performance by adapting BERT to the biomedical domain. Pubmed and PMC medical texts contain words not present in the corpus used to train BERT, which are tokenized into stemmed words or individual characters. BioBERT can correctly tokenise these words by pre-training from scratch and learn a domain-specific context that the pre-trained BERT model could not accurately represent.

Van Aken et al. (2021) [44] further pretrained and fine-tuned BioBERT by adding training objectives to learn relationships between admissions and outcomes. This objective, CORe, is similar to the "Next sentence prediction" task in the original BERT paper [39]. Instead of predicting whether a sentence is the next sentence in a sequence, CORe predicts whether an admission is the follow-up admission of the first admission. Van Aken et al. also applied the same strategy to medical articles and case reports, predicting whether a treatment, prognosis, or diagnosis results from symptoms or risk factors in a symptom-outcome pair. With the additional training objective, Van Aken et al. achieved state-of-the-art performance on diagnoses, procedures, in-hospital mortality, and length-of-stay prediction on the MIMIC 3 dataset, which consists of electronic health records. The dataset contains 1266 unique ICD-9 codes, which could be predicted with an AUROC of 83.54%.

3.4 Explainable AI

As computing power and deep learning techniques have improved over the years, the performance of many medical tasks has also increased. However, increased model complexity and structure can result in "black box" models with internal inference processes that humans cannot interpret. BERT is an example of a "black-box" as it decision making can not be derived from looking at the parameters

and structure of the model itself.

E**x**plainable **A**rtificial **I**ntelligence (XAI) is a field of machine learning that focuses on increasing the transparency and interpretability of AI-driven decisions without sacrificing performance[45]. XAI has been increasingly adopted in the healthcare industry due to its ability to enhance the accuracy of clinical decision-making and reduce the risks associated with incorrect diagnoses or treatments. By providing clear explanations for AI-driven decisions, XAI can help to reduce bias, improve fairness, and increase trust in machine learning models. XAI can also help healthcare practitioners understand the logic behind AI-driven decisions and make more informed decisions.

There are two approaches to explaining a machine learning model: intrinsic and post-hoc. An intrinsic explanation refers to a model that is self-explaining or "transparent" and is not considered a "black box" but a "white box". A post-hoc explanation, on the other hand, is an explanation that is generated by post-processing the model's output and structure to fabricate an explanation. Arrieta et al. (2019)[45] provide a detailed overview of the different methods and approaches used in XAI. They note that XAI has different categories of explainability and goals depending on the target audience. The primary aim of XAI is to increase the trustworthiness of AI-driven systems. Additionally, there is a distinction between local and global explanations. A local explanation focuses on explaining a single prediction, while a global explanation is concerned with explaining the behaviour of the entire model.

When designing an XAI system, it is essential to consider the target audience and the goals of the AI. The target audience will influence the methodology and techniques to make the AI explainable. Danilevsky et al.(2020)[46] performed a survey regarding XAI for NLP models and identified five primary explainability techniques:

1. Feature importance: Use the importance score of different features to derive an explanation.
2. Surrogate model: Predictions are explained by an explainable proxy model. The proxy model can have a different mechanism leading to concerns about the fidelity of the model[2].
3. Example-driven: Explain the output by presenting other semantically similar examples[47] [48].
4. Provenance-based: Explain by illustrating the prediction derivation process.
5. Declarative induction: Induce human-readable representations such as rules and trees to make the model more explainable.

There is, however, yet to be a consensus on how to evaluate the explainability of an AI, as explainability differs per use case and is subjective. Nauta et al.(2022)[3] who conducted a systematic review of XAI, defined 12 explanatory quality properties called the Co-12 properties(e.g. correctness, consistency, confidence) and present an extensive quantitative overview of XAI evaluation methods. An example of one of the most well-known XAI methods is LIME. In 2016, Ribeiro et al.[2] proposed a **L**ocal **I**nterpretable **M**odel-Agnostic **E**xplanation (LIME) method for generating interpretable explanations of black box machine learning models. LIME trains an interpretable white box classifier, such as a decision tree or linear classifier, by optimising the loss function on local data surrounding a given prediction. LIME can explain any machine-learning model and is an example of a surrogate model. LIME can explain text by generating slight variations of an input sentence and using the new probabilities as training data for an interpretable classifier.

Chapter 4

Source data

For this study, the data is supplied by *Topicus*. The data includes the questionnaire data from *Spreekuur.nl* and data containing SOAP notes from general practitioner practices from *SpoedEPD*. The supplied data is described and analysed in this section.

4.1 Questionnaire data

Spreekuur.nl provided questionnaire data in .XLSX format from 3 March 2021 to 23 December 2022. Totalling 79,215 data entries when combined and when duplicates were removed. Each entry is a "started" questionnaire, which can contain more than 100 columns. Many data entries can be filtered because questionnaires that are not completed or instances where users are triaged out (high-urgency) cannot be connected to a ground truth. Each question has an identifier; e.g. ATC_100, BKH_080, HOO_280. Each question can have three possible question types, open(string), input(int) and choice(int). Each answer and question identifier can be linked back to the text of the question via a lookup table. Integer values and definitions of an answer are not consistent across questions. A value of 0 may mean "No" for one question and "Yes" or "Maybe" for another. The dataset matrix is dense; each question has a value even if not seen or filled in by the user (presented by value "999999"). An entry example can be found in Table 4.1. The first question in *Spreekuur.nl* separates the questionnaire into 24 different categories based on input complaints like "skin complaints", "coughing" or "throat sore". Choosing the input complaint most relevant to the patient's health complaint will result in the most relevant questions being asked. For example, choosing "Throat sore" results in questions about the patient's throat. Figure 4.1, shows how many questionnaires are completed each month in the questionnaire dataset. **Note**: Time-stamp 1970-01-01 is a value for a questionnaire that is not completed, which happens almost 20.000 times.

Entry/Identifier	ABC	BRW_005_6_	BRW_005_020	BUI_190_010	BUI_200_1_	DBS_090	.....	dag_awn	klacht_keuze
1	0	999999	999999	0	1	Ik heb last van symptomen	.....	2	4

Table 4.1: Example of one data entry of a questionnaire.

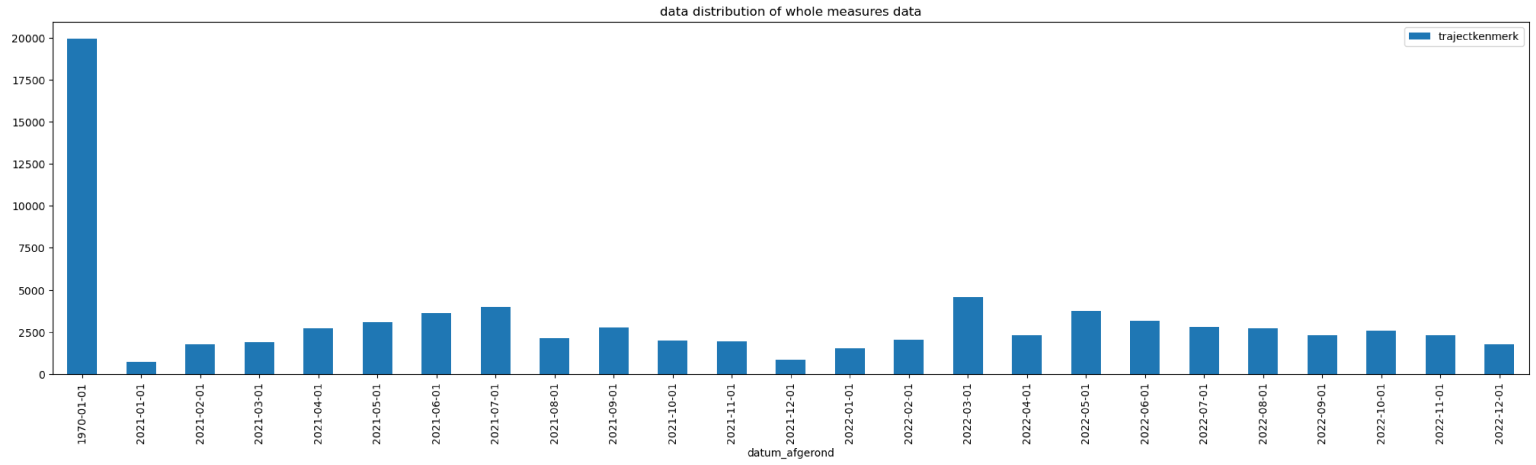


Figure 4.1: Amount of questionnaires per month

4.2 SOAP notes data

Table 4.2 shows a cleaned example of one data entry in SpoedEPD. Normally one data entry is 68 columns wide, but only important columns are shown for simplicity. The S-rule(S-regel) includes the subjective symptoms and patient's narrative. In SpoedEPD there are two S-rule columns, "sda_regel", "sha_regel". The triagist mentions in the "SDA" field the story and symptoms of the patient. The GP copies the "SDA" field into the "SHA" field and supplements it with extra information after consultation(in the dataset, not always the case, but how it should be done). The A-rule(E-regel) includes the diagnoses in textual form. There is also an "icpc_code" column which can be used as a ground truth for training a machine learning model.

The data of one patient belongs to the patient's GP while being saved in the database of *Topicus*. Data is collected by requesting permission per GP practice to use anonymised versions of all their patient's data. It resulted in three cooperating general practice emergency centres and **2.273.077** data entries. The centres are made anonymous per request of *DigiDok* and *Topicus*. **492.590** entries come from GP centre "GP centre A", **1.415.207** entries come from GP centre "GP centre B" and **365.430** entries come from GP centre "GP centre C".

However, it is important to note that all data entries come from GP emergency centres. An emergency post is only used when a health complaint is sufficiently severe. Diagnoses can have a different distribution compared to normal general practices potentially introducing biases. In the next section, data analysis will be performed to analyse the data and discover an eventual bias.

Column name	Example input
organisatie_naam	Centrale Huisartsendienst Drenthe
organisatie_id	HASH
sda_regel	Patiënt klaagt over toenemende pijn in de rechteronderbuik, voelt zich steeds beroerder en pijn niet te houden, zeker tijdens de rit naar de post.
sha_regel	Patiënt klaagt over toenemende pijn in de rechteronderbuik, voelt zich steeds beroerder en pijn niet te houden, zeker tijdens de rit naar de post.
o_regel	temp 38,5 c (rect), buik gespannen
e_regel	Rectaal bloedverlies
p_regel	Doorverwijzen SEH
icpc_hoofdcode	D
icpc_code	D88
patient_leeftijd	70
patient_geslacht	Man
spreekuur_koppel_id	HASH

Table 4.2: Example of a patients data entry in SpoedEPD

4.3 Data analysis

The dataset consists of **2,273,077** entries, representing 1,305 different ICPC codes. When the hierarchy of the codes is removed (e.g. U71.01 to U71), 776 unique ICPC codes remain. Removing the hierarchy simplifies the problem by reducing the amount of possible ICPC structure. After the prediction model performs well, the amount of ICPC codes can be increased. Figure 4.2 shows that the categories “L - Musculoskeletal”, “A - General and Unspecified”, “D - Digestive”, “R - Respiratory” and “S - Skin” are the most frequently occurring categories. The categories “X - Female Genital”, “Y - Male Genital”, “W - Pregnancy, Childbearing, Family Planning”, “B - Blood, Blood Forming Organs and Immune Mechanism”, and “Z - Social Problems” are the least occurring categories. The distribution of categories is uniformly the same as in other GP practices, as a scientific article by Nivel shows by measuring the occurrence of each category across multiple GP practices[49]. Figure 4.3 and Figure 4.4 present stacked bar charts of the top 100 occurring ICPC codes, with and without hierarchy. The most frequently occurring code is ”S18 - Laceration/cut,” appearing in 4% of the data entries, followed by ”A03 - Fever” (3.6%), ”D06 - Abdominal pain localised other” (3.1%), and ”U71 - Cystitis/urinary infection other” (2.9%). Notably, most of the codes represent symptoms rather than diagnoses. GP often records an ICPC symptom code when uncertain of the diagnosis. The distribution of ICPC codes is uniformly the same compared to other GP practices as shown in the scientific article conducted by “Huisarts en wetenschap”, which accounted for 957.636 consults in 2011, crossing nine GP practices [50].

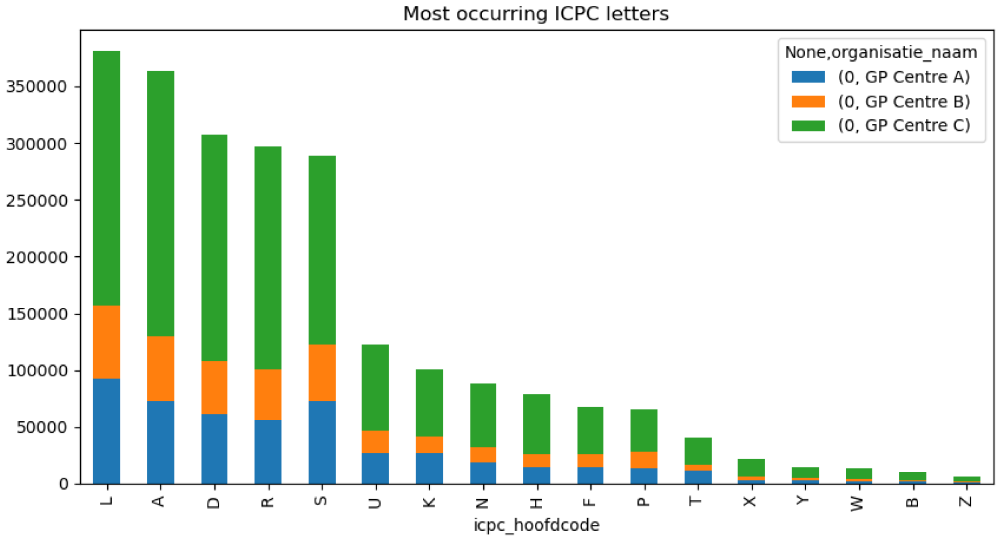


Figure 4.2: Most ICPC categories.

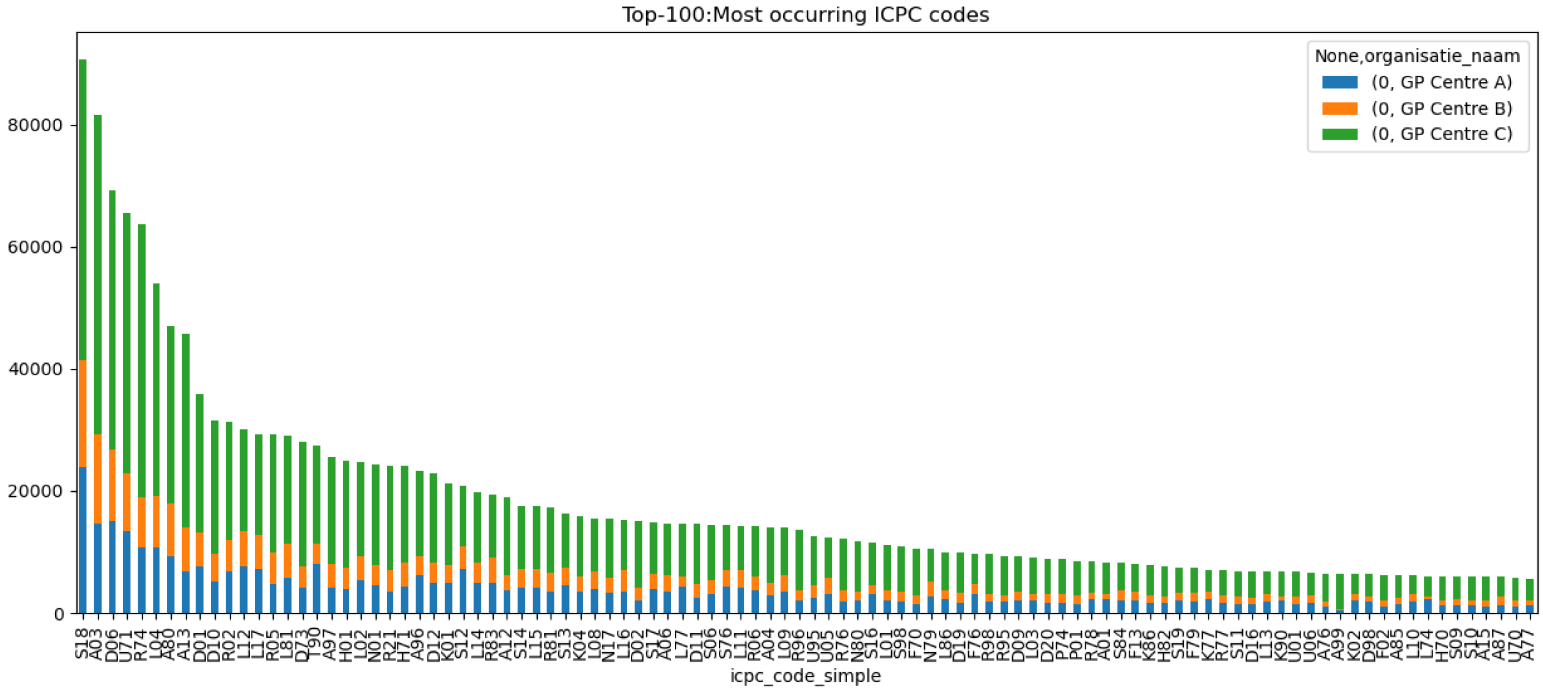


Figure 4.3: Hundred most occurring ICPC codes in the dataset.

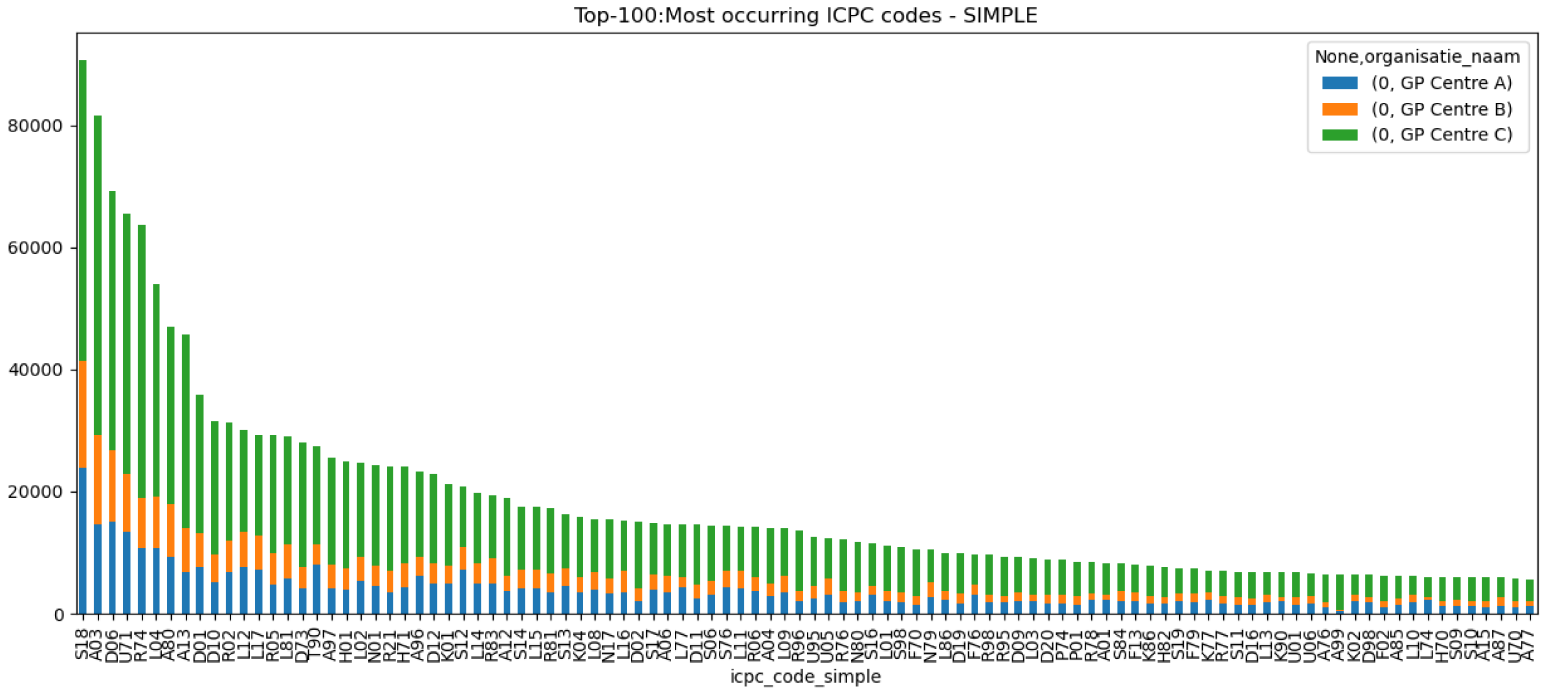


Figure 4.4: Hundred most occurring simplified ICPC codes in the dataset.

4.3.1 S-rule

To accurately represent a complete ”S-rule,” it is necessary to combine the ”SDA-rule” and ”SHA-rule” when they do not contain the same information. In this study, the average number of words in the ”SDA-rule” was 69, while the average number in the ”SHA-rule” was 35. It is noteworthy because the GP is expected to copy the ”SDA-rule” into the ”SHA-rule” and add complementary notes. However, this only occurs in some cases. The average number of words in the combined ”S-rule” is 84. It is also worth noting that the rule often begins with the same prefix sentence: ”Klacht/beloop:” (translated as ”Complaint/course:”).

4.3.2 Bias

In Figure 4.5, the percentage of each ICPC code is shown alongside its sum. The data shows that the top 10 codes represent ∼25% of the total codes, the top 20 codes represent ∼38%, the top 50 codes represent ∼61%, and the top 100 represents ∼79%. This total indicated that the remaining 677 codes only represent ∼21% of the total dataset.

Figure 4.6 shows the dataset’s 100 least frequently occurring ICPC codes. Many codes are recorded only once, twice, or thrice, which may indicate that they represent rare diseases or potentially incorrect or misspelt codes. It is essential to consider whether these codes should be included as potential diagnoses. The codes’ low frequency may negatively impact the model’s performance in predicting more commonly occurring codes.

In Appendix A, further data analysis is done on the age and sex distribution in the dataset.

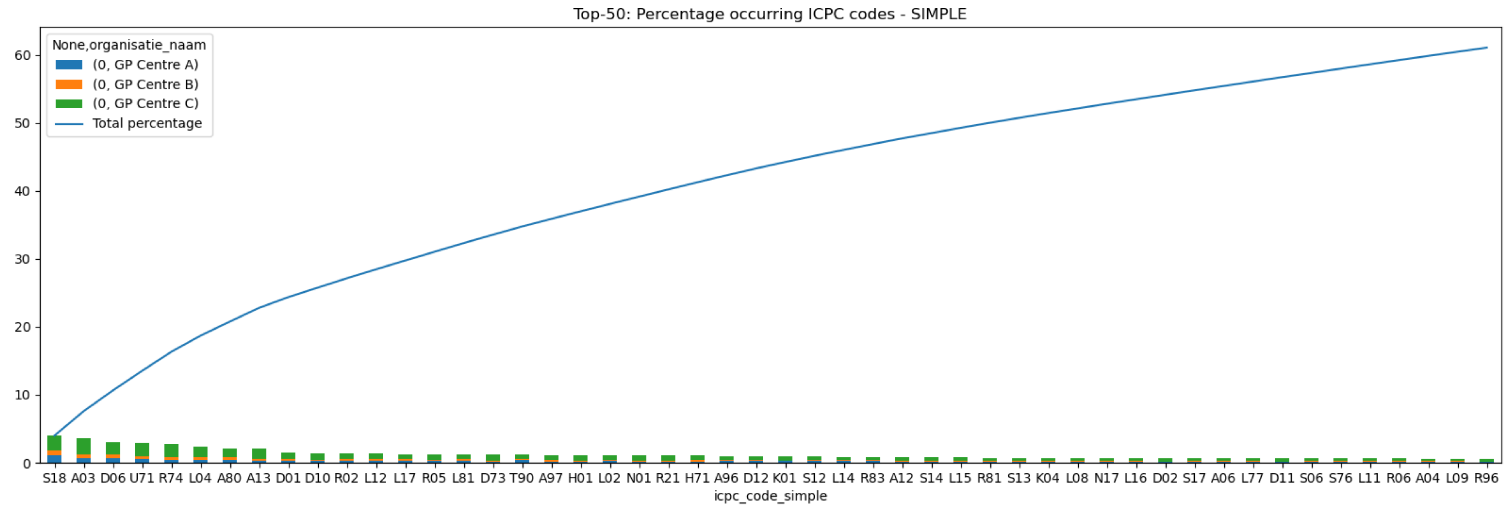


Figure 4.5: Distribution and the sum of ICPC codes in the dataset.

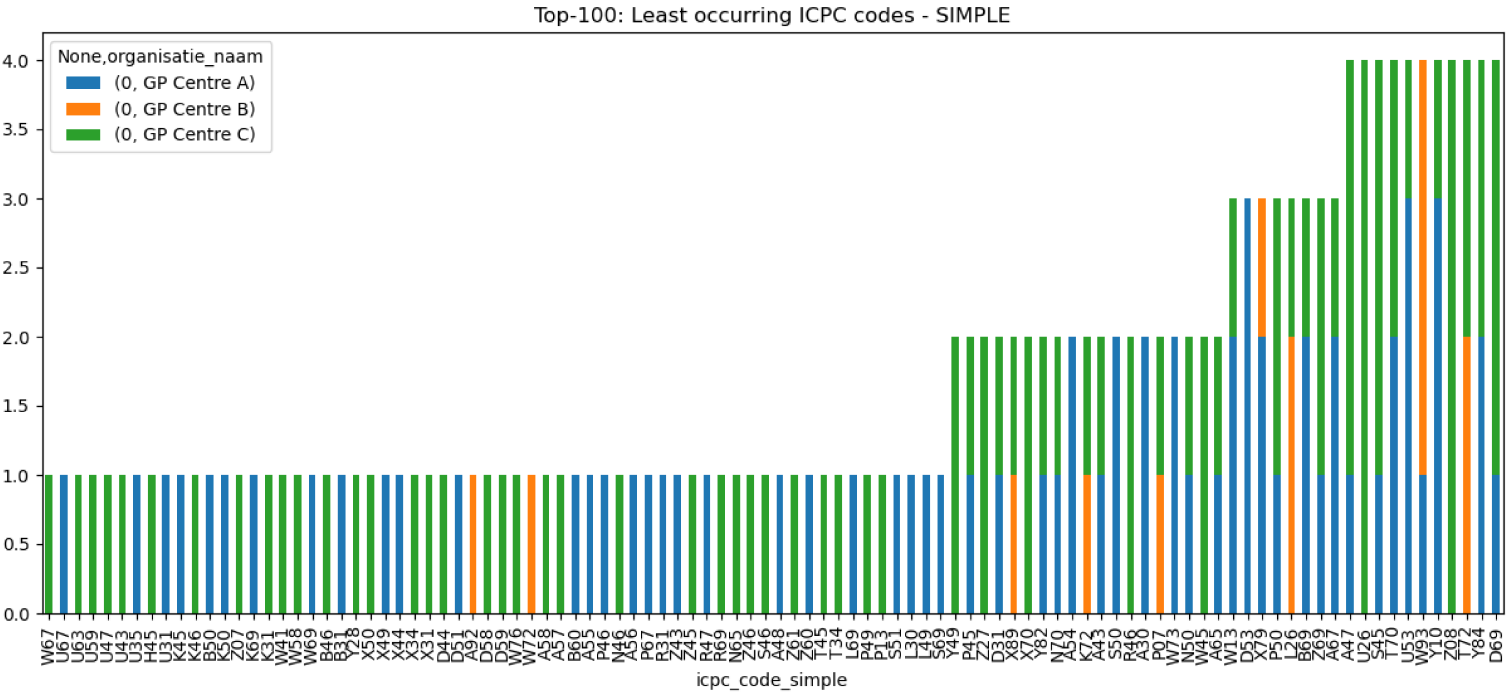


Figure 4.6: Hundred least occurring ICPC codes in the dataset.

Chapter 5

Research questions

Previous sections provide context and background information on the problem statement being addressed and identify some research gaps and practical issues. These issues can be reformulated as research questions, which can be further refined into sub-questions to aid in answering or validating the main research question. The main research question for this study is: ***"How can a RoBERTa language model be used for predicting diagnoses based on patient-reported symptoms?"*** Sub-research questions have been formulated to help address the main question.

1. **What is the most effective method for transforming SOAP notes and questionnaire data to train a RoBERTa diagnoses prediction model?**

   The existing data comprises SOAP notes presented in natural language, while the questionnaire data is structured in a tabular format. It is essential to devise an optimal method for transforming the questionnaire data into natural language, maximising the inclusion of valuable information while minimising the extent to which the model needs to undergo fine-tuning to achieve high diagnostic performance.

2. **How can data from the SOAP note dataset link an ICPC code to the anonymous questionnaire data?**

   Before training, it is necessary to determine the eventual diagnosis/ICPC code assigned by a GP based on the questionnaire and consultation. Fortunately, the SOAP note dataset includes the SOAP note from the respective questionnaire. Previously, an ICPC code/SOAP note could be linked to a questionnaire through a ID field, but the field has since been removed. To validate or train the model on a as large as possible dataset it should be investigated how ICPC codes can be linked to questionnaires without the use of the removed field.

3. **What is the performance of the diagnoses prediction model against established baseline models?**

   The RoBERTa language model's complexity may exceed the requirements of the current task, and a simpler model could achieve comparable results with lower computational costs. Training and validating baseline models on the same dataset as the RoBERTa model can assess whether the enhanced performance of the RoBERTa model justifies its use over the baseline models.

4. **What strategies can improve the model's performance to predict ICPC codes?**

   As observed in section 3.3: "Diagnosis Prediction using machine learning", various strategies and architectural modifications have been implemented on BERT to enhance its performance in specific tasks. The hierarchical structure of ICPC codes presents an opportunity for potential model improvement if appropriate modifications are made to the architecture or training procedures.

5. **How can the model's performance be validated for diagnosing patients using ICPC codes?**

   The diagnostic process employed by Dutch GPs is only partially captured by providing a single diagnosis. As discussed in chapter 2: "Background", GPs conduct a differential diagnosis,

considering multiple possible diseases or diagnoses. However, the ground truth data only comprises a single ICPC code. During the validation process, it is essential to acknowledge that the model's predicted diagnosis could be considered equally valid compared to the single ICPC codes. Two GPs may potentially employ different ICPC codes with nearly identical definitions for the same S-rule.

6. **What is the relationship between the model's performance and the inclusion of specific questionnaire questions and answers as input features for the model?**

   A questionnaire typically can consist of a large number of questions and answers. However, not all questions hold the same level of significance for a GP. Similarly, in the context of machine learning models, specific questions can significantly enhance the diagnostic capability of the model. In contrast, excluding specific questions may not impede its diagnostic performance or even improve it. Removing questions and answers may also simplify the complexity of the input data of the model which can have a positive impact on the performance.

7. **To what extent can current knowledge of diagnoses, symptoms, and causes in the medical field be used for predictions?**

   The available data primarily comprises general practice consultations, comprising substantial information regarding the relationship between symptoms, diagnoses, and diseases. These consultations hold the potential to establish connections between a patient's narrative and a specific diagnosis, thereby potentially enhancing the model in diagnosing patients with similar narratives. In this study, it is essential to employ a method that effectively uses the entirety of this data and knowledge to maximise its information for the model.

8. **Which XAI method is most effective at explaining the predictions of a diagnoses prediction model to Dutch general practitioners?**

   The presence of "black boxes" and non-interpretable machine learning models poses a challenge in the medical domain. GPS must have access to explanations regarding the decision-making behind a diagnosis prediction. Such explanations enable GPs to gain insights into the contributing questions and answers, serving as a valuable second opinion. Moreover, these explanations can cause the GPs to understand the underlying decision-making behind the model's predictions, facilitating a more informed decision-making process for the GPs themselves.

9. **How does the use of the diagnoses prediction models impact the efficiency of Dutch GPs?**

   As shown in chapter 3: "Related works", numerous studies highlight the potential disparity between a medical prediction model's performance as suggested by a validation set and its actual performance. To accurately measure the model's performance, it is crucial to evaluate it using real-world use cases and involve Dutch GPs in the assessment process. Additionally, the model's explainability plays a crucial role in the diagnostic process of GPs. This aspect can also be measured by examining whether the model effectively enhances the GP's diagnostic process and if the provided explanations accurately represent the model's decision-making.

Chapter 6

Methodology

Research questions were formulated in chapter 5: "Research questions" that address the problem statement and research gaps. This chapter contains the methodology on which and what is needed to answer these questions. The three following chapters (chapter 7: "Dataset", chapter 8: "Diagnoses prediction model" and chapter 9: "Explainability") implement the methodology and provide an in-depth overview of the techniques that are proposed. These chapters also contain the results of the study.

The provided data for this study comprises 2.2 million textual consulation, specifically in the form of SOAP notes, as observed in chapter 4: "Source data".

In chapter 3: "Related works" it was noted, that training a language model to be domain-specific can enhance its performance. Hence, for this study, a BERT-based language model was selected. The section also specified how BERT works and how to train a BERT model which will be further investigated in the following sections. Moreover, previous studies have emphasised the significance of explainable predictions, particularly in the context of medical prediction models. However, it is worth noting that XAI methods often lack proper validation as shown by Nauta et al.[3]. To address this gap, this research aims to conduct validation, for instance, through the implementation of a user study.

To provide an overview of the following sections, a summary of the steps can be found in Table 6.1, serving as a roadmap for the study.

Steps
1. Connect ICPC codes to questionnaire data
2. Transform questionnaire data to natural language
3. Pre-train model to be domain-specific
4. Fine-tune model to predict ICPC codes
5. Validate the model
6. Make the model explainable
7. Conduct user study

Table 6.1: Identified roadmap

6.1 Connecting ground truth

The questionnaire data has no ground truth (ICPC code) by default. In the SOAP note dataset, the online consultation following the *Spreekuur.nl* questionnaire should be present. A ground truth is needed to train and validate a machine-learning model. In total, there are 58,651 questionnaire entries. Till March 2022, the SOAP notes and questionnaire data could be connected via a "spreekuur_koppel_id", but this feature was removed. Topicus decided to remove the feature to anonymise the questionnaire data and create a separation of concerns between applications. The questionnaire data is anonymised but potentially can be linked to the corresponding SOAP note in the SOAP note dataset by person-related information such as sex, age and time of completion.

Furthermore, data made available for this study consists of SOAP notes and questionnaire data. Questionnaire data differs from SOAP notes as it only consists of tabular questions and answers, and SOAP notes are text. First, the questionnaire data must be processed to textual data corresponding

to the S-rule as it represents the subjective patient-identified symptoms. The S-rule allows the model to train on both datasets, making training the model on the questionnaire data less complex as the data dimensions are significantly reduced.

The details of how the questionnaire data is connected to a ground truth, how the questionnaire data is transformed into S-rules, and how the datasets for this study are generated are described in chapter 7: "Dataset".

6.2 RoBERTa

In this study, a variation of BERT, a state-of-the-art natural language processing model, will extract medical representations from input data to predict a diagnosis. The use of BERT is justified by its strong performance on medical benchmarks for predicting diagnoses or diseases[44]. As seen in chapter 4: "Source data", most of the data consists of text which would require a language model to process. The large amount of SOAP notes in the dataset can be utilised to extract existing medical information and knowledge from consultations. To diagnose a patient based on the answers of *Spreekuur*, the tabular data from a questionnaire must be transformed into natural language for the language model to process it. BERT can produce better representation than other transformer-based language models as it does not focus on translation or generation. BERT can provide representations which can have a deeper understanding of the context of a sentence because BERT looks at the input sentence bidirectionally. In some cases, BERT is surpassed by domain-specific implementations such as BioBERT. More recent variations of BERT, such as RoBERTa, show increased performance compared to BERT overall. The available data in this study consist of SOAP notes of Dutch GPs and questions/answers of patients from the questionnaires. A domain-specific implementation of BERT/RoBERTa can increase the model's diagnostic ability by creating better representations for medical data.

There are two options for training a BERT/RoBERTa model on medical data: 1) Train on an existing pre-trained model such as BERT base, BioBERT, or RoBERTa base, or 2) Pre-train a BERT / RoBERTa model from scratch. Pre-training a model from scratch requires a large amount of data to train and update all randomly initialised parameters, and the available data in this study may not be sufficient to do so as pre-trained models such as BERT, BioBERT, and RoBERTa are trained on billions of words and millions of lines, providing realistic representations.

Another option is to train on an existing pre-trained model. Trained BERT models are language specific as they are mostly only trained on data from one language. Domain-specific BERT variants such as BioBERT only perform well on English medical data. Therefore, Dutch medical data would require a BERT model trained on a Dutch corpus. RobBERT[17], a RoBERTa-based language model trained on the Dutch OSCAR dataset [51], is currently the state-of-the-art performing Dutch BERT model on Dutch NLP benchmarks. RobBERT can be further pre-trained with the medical data in this study to make it domain-specific to the medical field of Dutch GPs.

There are two options when pre-training a RoBERTa model further on medical data: 1) Use the original vocabulary and extend pre-training on domain-specific data, or 2) Extend pre-training on domain-specific data but use a new vocabulary. Using the original vocabulary allows the use of the same trained tokenizer. However, it has the limitation that out-of-vocabulary words (OOVs) may not be correctly tokenized and may be split up into sub-words or characters, changing the meaning and representation. Using a new vocabulary can have the caveat of losing how the new token embeddings are linked to the presentations in the pre-trained model.

Verkijk et al. (2021) [52] created a domain-specific RobBERT model using Dutch electronic hospital notes. The authors used approximately 10 million notes from two Dutch hospitals. They trained

two different models: A trained RoBERTa model from scratch and a pre-trained RobBERT model with a new domain-specific vocabulary and frozen transformer layers. They found that when sufficient domain-specific data is available, a model pre-trained from scratch yielded the best performance. The model was unavailable; otherwise, it could have been used as a basis for this study.

Chalkidis et al. (2020) [53] created a legal domain-specific BERT model by using over 350 thousand legal documents. They also trained two models (one from scratch and one further pre-trained) and concluded that the model's performance depended on the task, and no clear model was the best.

It is chosen to explore the same strategy as Verkijk and Chalkidis et al. and further train RobBERT to be domain-specific to the Dutch medical fields. chapter 8: "Diagnoses prediction model" shows how the final "diagnoses prediction model" was made and what architectural decisions were made.

6.2.1 Vocabulary transfer

In their studies, Mosin et al. (2022)[54] [55] investigated the use of vocabulary transfer for improving the performance of language models in biomedical texts. The need for vocabulary transfer arises when the dataset used for fine-tuning contains rare words or word fragments that are not present in the pre-training dataset. Implementing a new specific tokenisation scheme can enhance the model's performance by adequately tokenising and representing these rare words. Mosin et al. showed that using vocabulary transfer for biomedical texts can improve the performance of medical text dataset benchmarks by up to 10%. The authors experimented with two different token initialisation heuristics: 1) If a token in the new vocabulary coincides with a token in the old vocabulary, they assign the old token's embedding to the new token. 2) For new tokens that cannot be directly mapped to old tokens, they split the new token into a partition of several tokens from the original tokenisation. For each such partition, they calculate the minimum number of tokens and choose the partition with the smallest number of tokens. In the case of ties, they choose the partition that contains the most extended token. All token embeddings in the chosen partition are then averaged to produce a single average embedding for the new token. Overall, Mosin et al.'s results indicate that vocabulary transfer can be an effective approach for improving the performance of language models on biomedical texts. The results and implementation of the tokenizer can be found in section 8.2: "Tokenizer".

6.2.2 Pre-training

As specified in chapter 3: "Related works", BERT uses two training objectives, "Masked Language Model" (MLM) and "Next sentence prediction"(NSP), to learn its representations. Other works extending BERT have shown that additional training objectives for domain-specific tasks can improve performance. Among other changes, the most significant change RoBERTa made to the BERT architecture was removing the "next sentence prediction" task in BERT. The authors of RoBERTa[40] show in their paper that the NSP task does not increase performance and even get higher performance with the task removed. Van Aken et al. [44] added a training objective called CORe to BioBERT which predicted if diagnoses and treatments were indeed part of the symptoms or risk factors in a health record. With the CORe task, they managed to get state-of-the-art performance in ICD-9 diagnoses prediction tasks. They showed that it is possible to predict diagnoses based on symptoms using 1266 possible ICD-9 codes. CORe, like NSP, separates an input sequence into two parts. CORe splits sequence t into $t_{N,1\ldots k} \in A_N$ and into $t'_{N1k} \in D_N$ where A is the admission of the sequence D is the discharge note of the sequence. They train the model to maximise $P\left(P_N \mid X_{N-N}\right)$ where $X_{N\_N} = \mathrm{Enc}\left(t_{N,1\ldots k}, t'_{\mathbf{N},1\ldots k}\right)$ an P_N is the patient corresponding to sequence t. Just like NSP, they use negative sampling for 50% of the examples. The model's final implementation and results after pre-training can be found in section 8.3: "Masked Language modelling/ pre-training".

6.2.3 Classification

BERT and RoBERTa output a representation vector for each token in the input sequence. It is important to note that the special tokens [CLS] and [SEP] are added to the input. [CLS] is added in front of every input example, and [SEP] is a separator token used for the NPS task(not used in RobBERT and RoBERTa) or question answering. The representation vector for the [CLS] token can be used for classification as it represents the entire sequence of tokens. The representation vector can be fed as input to an additional sequential layer or neural network for classification. For predicting ICPC codes, the neural network output would be the same as the number of ICPC codes. The output must pass through a "soft-max" layer to produce probabilities for each code. section 8.4: "Classification", shows how the pre-trained model is trained on SOAP notes and how it can classify ICPC codes. It also explores multiple architectures to classify ICPC symptoms and ICPC diagnosis/disease codes separately.

6.2.4 Explainability

BERT and RoBERTA are considered "black-box" models because their inner workings and the reasoning behind its predictions are not easily interpretable. As explained in chapter 3: "Related works", LIME can be used to explain any black-box model. LIME works by training a locally faithful, interpretable model (also known as a "white-box" model) to explain the predictions of the black-box model. LIME can be used to highlight the specific words, answers, or symptoms that had the most impact on the model's prediction, providing doctors with valuable insights into the model's decision-making process and allowing them to assess the reliability of the prediction, as seen in Figure 6.1. To obtain these explanations, LIME minimises a loss function by finding the values of the locally faithful model that best fit the prediction made by the black-box model by using Figure 6.2. The loss function is minimised so that probability $f(x)$ belongs to a particular class where $\pi_x(z)$ is a proximity measure between instance z to x to define the locality around x. The complexity of the explanation is measured by a function called $\Omega(g)$, which can vary depending on the specific interpretable model being used (e.g., the depth of a decision tree). G is a class of interpretable models. LIME was shown to explain the predictions of BERT models[56] effectively.

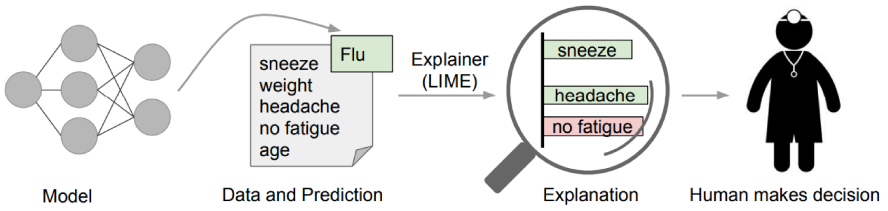


Figure 6.1: How LIME can be used to explain a diagnosis. Taken from the original LIME paper from Ribeiro et al.[2].

$$\xi = \underset{g \in \mathcal{G}}{\arg\min} L\left(f, g, \pi_{x'}\right) + \Omega(g)$$

Figure 6.2: Minimising LIME loss function

In chapter 9: "Explainability", the LIME text-module is applied to ICPC codes and a modified LIME method is proposed and validated for explaining diagnoses to GPs.

6.3 Validation

To validate the model in this study, three dataset splits for training, testing, and validating the model should be created. To achieve a fair validation, the ICPC codes the dataset should be distributed equally among each set. The testing set should be composed of 10% of the entire dataset, while the remaining 90% is split into training, and 10% should be taken for a validation set. The training set is solely used for training. The model is never updated on the validation or test set to ensure the model does not overfit, and the performance metrics are as accurate as possible.

Accuracy-at-x is a commonly used performance metric for evaluating diagnosis prediction models and symptom checkers. It refers to the accuracy when the ground truth diagnosis is among the top x predictions. The metric is often used in diagnosis prediction and symptom checkers papers because it aligns with the method used by medical professionals to diagnose patients. Other measures should consider that multiple ICPC codes may be suitable for a single S-rule as there may not be enough information in the S-rule to make a single diagnosis. When training a BERT model, it is important to consider its accuracy and loss, as the final neural network layers are updated based on the loss. In this study, it is necessary to conduct experiments on the model's loss function, as ICPC codes have a hierarchical structure. As mentioned earlier, the data used in this study may contain biases that could impact the model's performance. The model may achieve good results because it has learned to rely on these biases or struggle because of them. To address these biases, the loss function can be made uniform and, for example, increased for rare ICPC codes. The performance of these methods should be analysed by examining the model's confusion matrix, precision, recall, and f1 scores to identify biases and the actual performance of the model. section 8.6: "Performance" specifies which performance measures were used precisely in this study and the performance of the final model on these measures.

6.4 User study

The standard method of evaluating physicians and other clinicians on their diagnostic abilities is using standardised medical vignettes. Vignettes are created for testing purposes and use non-existent patients, consisting of a description of symptoms and medical information with a diagnosis as the ground truth. Existing symptom checkers are often proprietary and systemic reviews show that these checkers often have no evaluation method. When applying medical vignettes to these existing symptom checkers, the performance greatly decreased compared to the performance specified in the respective paper. This research used a user study to evaluate the model's performance in real scenarios. The user study presents GPs with an S-rule and a list of predicted diagnoses. The GPs are then asked to provide the corresponding ICPC code for the presented S-rule. The chosen ICPC codes can then be used to measure the real-life model's performance.

Moreover, the user study also aims to measure the model's explainability. Nauta et al. [3] highlighted that only a tiny percentage of XAI research papers conduct user studies to evaluate model explainability in real-world scenarios. section 9.2: "User study" presents the user study, including the question formulation, the results, and the conclusion.

Chapter 7

Dataset

In this section, the data that was obtained and detailed in chapter 4: "Source data" is processed to use it to train, validate and test the RobBERT model. For the initial step, the questionnaire dataset needs to be connected to a ground truth to know what the concluded diagnosis was after a patient filled in a questionnaire.

7.1 Connecting via ID

Until March 2022, the SOAP notes and questionnaire data were linked through a field called "spreekuur_koppel_id" in the SOAP notes. However, this feature has since been removed. To connect a SOAP note to a questionnaire, the "spreekuur_koppel_id" field must match the "call_id" field in the questionnaire data. As a result, 5,356 questionnaires have been connected to SOAP notes, providing a ground truth. The number of connected questionnaires each month is presented in Figure 7.1. A set of connected questionnaires is advantageous because it provides a basis for evaluating the performance of the other data connection algorithms for data that does not have a "spreekuur_koppel_id." This dataset is called the "automatically connected dataset."

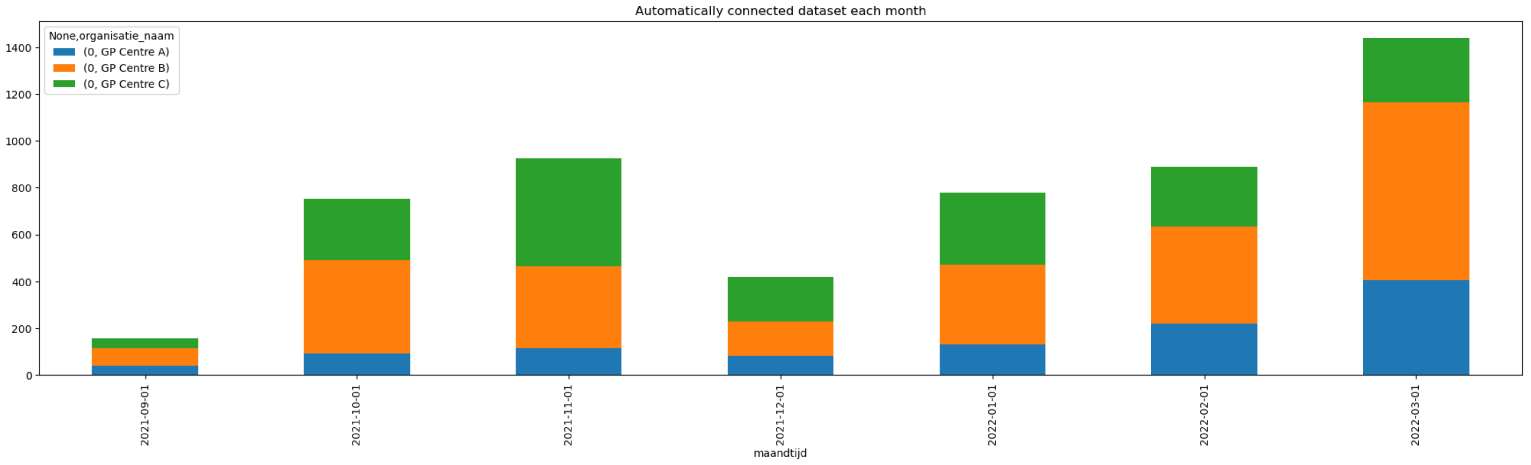


Figure 7.1: Connected questionnaires each month by id

7.2 Connecting to ground truth

The questionnaire dataset consists of 79,215 data entries obtained from 85 GP practices. The SOAP note data is available from the three largest GP practices, which have used the *spreekuur.nl* system for the most extended period. The questionnaire dataset contains only two fields that can match the SOAP data. The first field is the patient's sex, which can be either male (1) or female (2). The second field is the patient's age, which ranges from 0 to approximately 100 years.

Connecting a SOAP note to a questionnaire is challenging since multiple SOAP notes could match each questionnaire entry. To overcome this challenge, the "automatically connected dataset" was

used to identify another matching field named “aannametijd” and “datumAfgerond.” “Aannametijd” or “time of begin triage” in English is a field in SOAP notes where the GP centre is called to plan a consultation. When a GP centre uses the *spreekuur.nl* system, patients are redirected to fill in a questionnaire to start a chat. “DatumAfgerond” or “dateCompleted” is a timestamp in the questionnaire dataset that indicates when the questionnaire was completed.

Analysing the time difference between these two fields in the automatically connected dataset shows that the average time difference is approximately 20 minutes, with a maximum time difference of about 19 hours and a minimum time difference of around 17 seconds.

The 2.2 million SOAP notes available in the dataset include not only consultations initiated via the *spreekuur.nl* system but also regular consultations. To filter out the regular consultations, the “initiele_actie” or “initial_action” field in the SOAP notes was used, which specifies the action that led to the consultation’s creation. Figure 7.2 shows the distribution of the different “initial_actions” in the “automatically” connected dataset. The Figure shows what “initial_actions” are used by GPs when patients use the *spreekuur.nl* system. The four most commonly used “initial_actions” (“Spreekuur.nl ANW” (renamed to “Digi-HAP ANW” after 2022-04-01), “2-DigiHAP ANW”, “Digiconsult Huisarts”, “Digiverwijzing”) account for more than 99% of the data each month.

Filtering the SOAP note dataset on these four “initial_actions”, and removing empty values for sex and age decreased the size of the SOAP note dataset from 2.2 million to only 31,857. Similarly, the questionnaire dataset was filtered to remove entries with an empty sex or age field or incomplete questionnaires, decreasing the length of the questionnaire dataset from 79,215 to 59,261. Figure 7.3 provides an overview of the conditions used to filter each dataset.

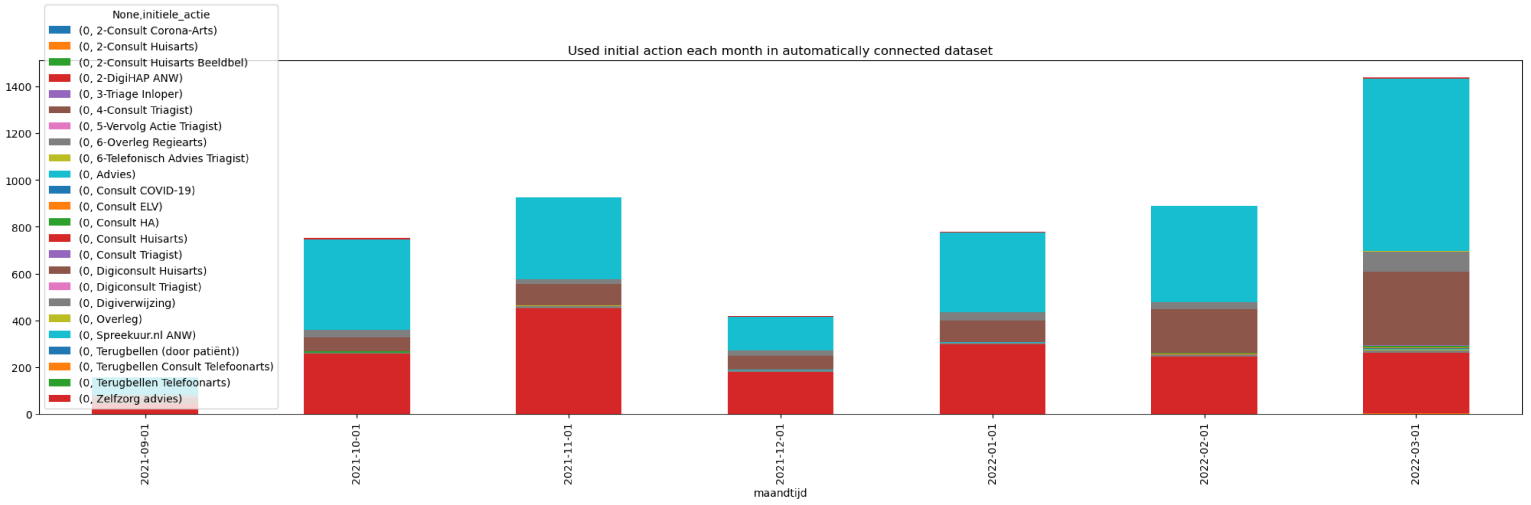


Figure 7.2: Distribution of initial actions in the automatically connected dataset.

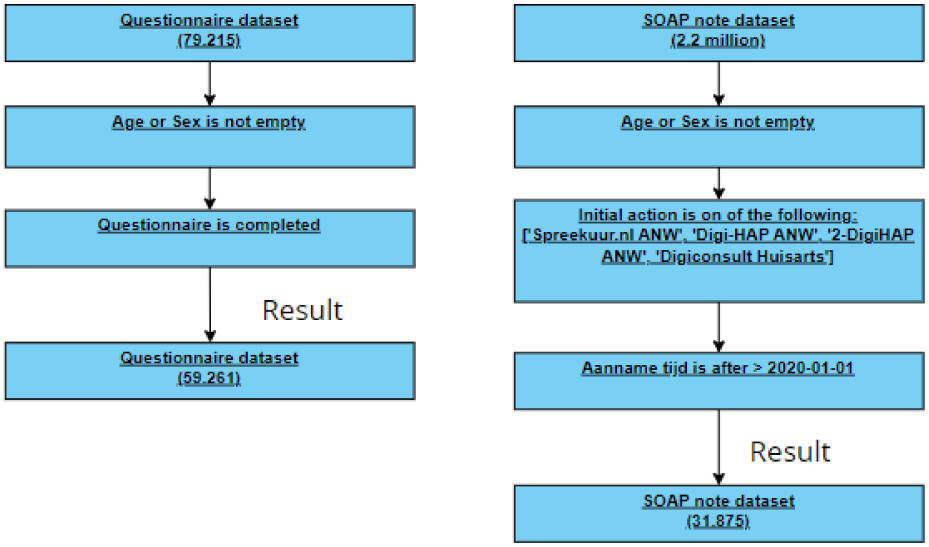


Figure 7.3: Conditions on which the datasets are filtered and the length of the filtered dataset.

The filtered datasets of 59,261 questionnaires and 31,857 SOAP notes were merged using the fields sex, age, and “Aannametijd/datumAfgerond” with a maximum time difference of 60 minutes as depicted in Figure 7.4. This merging process resulted in 15,445 matched entries, as shown in Figure 7.5.

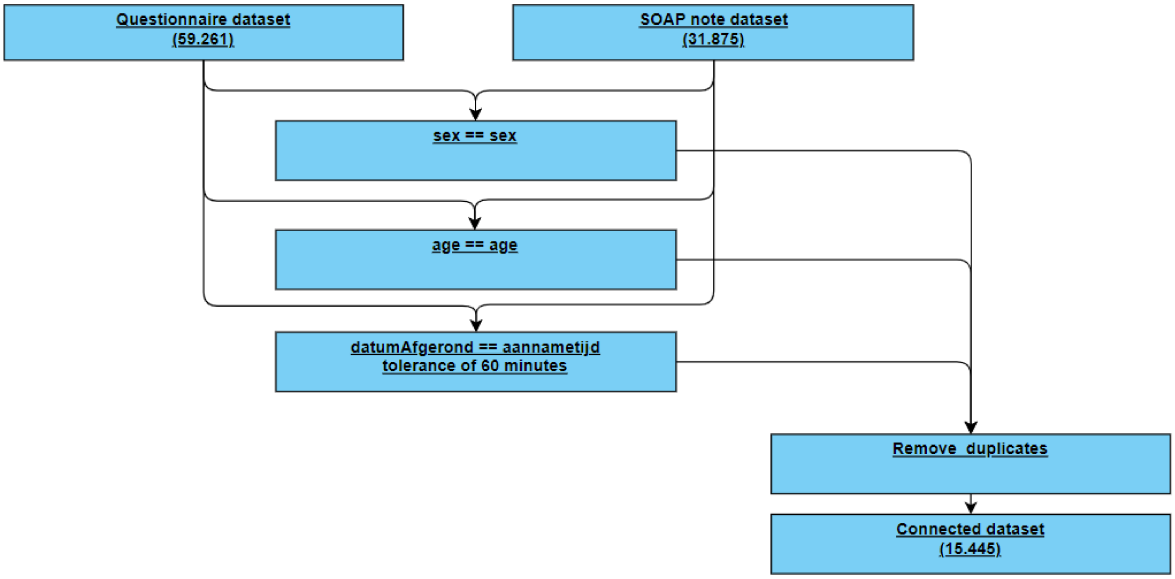


Figure 7.4: Merging conditions of both filtered datasets.

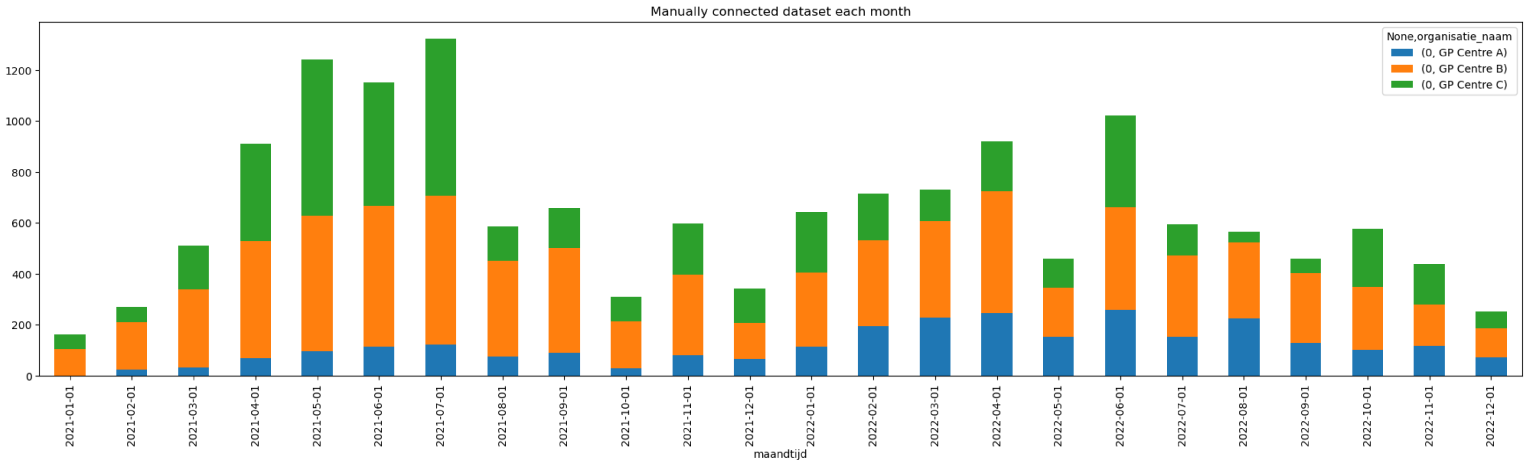


Figure 7.5: Manual connected questionnaires each month.

7.2.1 Quality of connection

Goal + setup

It is important to ensure the connection quality in the manually connected dataset. To achieve this, the automatically connected dataset was used as a reference. By cross-checking the entries that are present in both datasets, the quality of the manually connected dataset can be assessed.

To determine the accuracy of the manually connected dataset, it is necessary to consider the period during which the questionnaire could still be connected via the "spreekuur_koppel_id". The two datasets can be compared by examining the intersection of the manually connected and automatically connected datasets. This analysis measures the two datasets' ratio, precision, and intersection. For a connection to be correct, both the SOAP note id and the questionnaire id should be in the same entry of the automatically and manually connected datasets.

Results

Table 7.1 shows that a total of 3,451 entries in the manually connected dataset correspond to the period covered by the automatically connected dataset. Among these entries, 2,840 have a corresponding questionnaire or SOAP note id in the automatically connected dataset, of which 2,666 are accurately connected. This comparison indicates that 174 entries have a questionnaire id that matches the automatically connected dataset but is connected to an incorrect SOAP note or vice versa. Among the 5,356 entries automatically connected dataset, 4,450 entries meet the criteria that age and sex information in both the questionnaire and the SOAP note dataset match. The connection of the manually connected dataset is conditioned on this criterion. Thus, a fair comparison requires excluding data in the automatically connected dataset because it could not be connected in the first place. The "Ratio in period" may indicate false positives as more entries are connected in the period than in the automatically connected dataset. However, the high precision suggests that there are few false positives present. The actual performance of the manually connected dataset cannot be measured with certainty. Still, in chapter 8: "Diagnoses prediction model", the automatically and manually connected datasets are used to evaluate the model's performance. Both datasets lead to almost identical performance, which suggests that the quality of the manually connected dataset is high.

Measurement	Percentage	Value	Explanation
Ratio total	18%	2,840/15,445	Manual dataset present in auto dataset / manual dataset.
Ratio in period	82%	2,840/3,451	Manual dataset present in auto dataset / manual dataset in same period.
Total intersection	53%	2,840/5,356	Manual dataset present in auto dataset/ Auto dataset.
Possible intersection	64%	2,840/4,450	Manual dataset present in auto dataset/ Auto dataset that is connectable.
Precision	99%	2,666/2,840	How much is correctly connected

Table 7.1: Performance measure of the dataset.

An additional performance metric is the average time interval between the fields “Aannametijd” and “datumAfgrond” in both datasets. The automatically connected dataset shows an average time interval of about 0.32 hours, while the manually connected dataset has an average time interval of approximately 0.33 hours. These findings suggest a 60-minute tolerance value is appropriate for connecting questionnaire data to SOAP data.

Figure 7.6 displays each month’s total number of connected questionnaires (both automatically and manually). When the two datasets are concatenated, the resulting connected questionnaire dataset has a total length of 17,833.

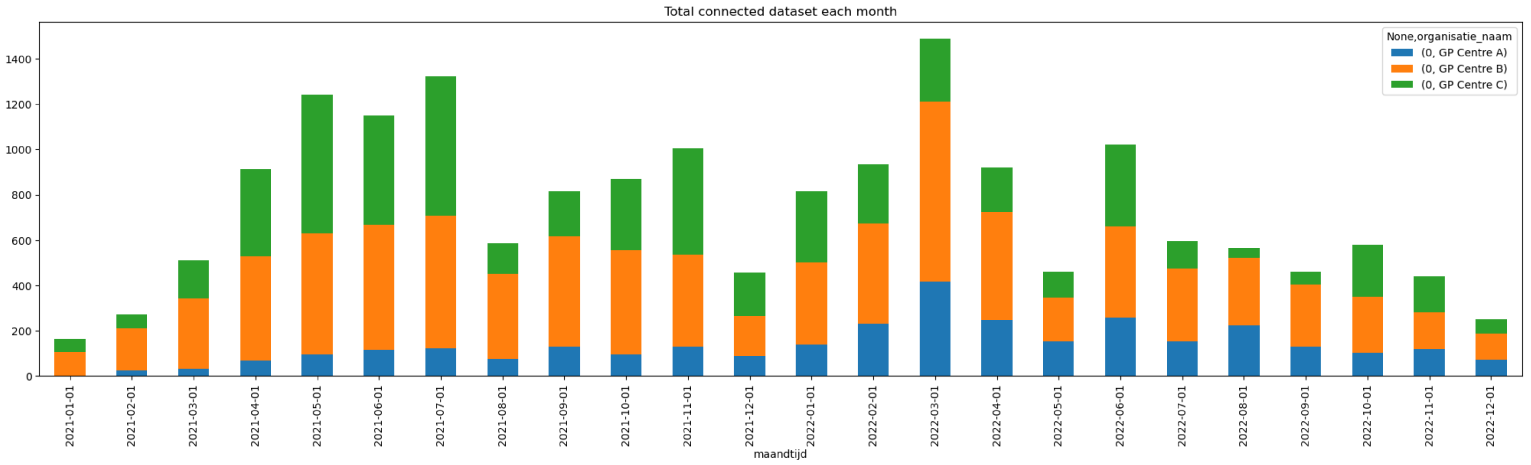


Figure 7.6: Total connected questionnaires each month.

7.2.2 Generating S-rule

The questionnaire data only consists of fields with an identifier of a question, where the value of that field is the answer. For instance, the identifier “GZP_040” corresponds to the question “What is your sex?” with possible values of 1 for “Male” and 2 for “Female”. To use these fields for training, testing and validation in the language model, they must be transformed into textual data as an S-rule. To retrieve the original question and possible answers, a lookup table can be referenced.

DigiDok, the party that created the questionnaire answers and questions made a codebook to transform answers/questions to an S-rule so that GPs do not need to look into every patient’s answer to get information. The generated S-rule is added to the SOAP note written by the GP automatically. *DigiDok* employs doctors and GPs to validate their codebooks. The codebook specifies the section to which each identifier belongs and, for each answer, what should be added to the S-rule. Open-ended questions are always included in their entirety.

Most questions and answers that are not useful for the GP, or contain negative responses, are not added to the S-rule. For instance,“Does your wound itch?” can be answered with a yes or no

response. If the answer is “yes”, the value “jeuk+” (itch+) is added under the “klacht/beloop” (complaint/course) section. If the answer is “no”, nothing is added to the S-rule, as this information is not deemed useful for the GP. For the question “Do you have a fever?” the negative answer is also added to the S-rule, “koorts-” (fever-), because this can be critical information for the GP. The questionnaires and codebooks are updated periodically, and it is crucial to match the version of the questionnaire and codebook when generating S-rules. An example of a generated S-rule to give an idea is

> “initial complaint: skin complaints, fever, moderately ill, 4days-1wk skin complaints, does not recognise complaints, slightly sensitive (score 1), increase in complaints, location: nails, hands, complaints: red, hard spot, pus, swelling, cause: I had last week a cut on my cuticle. The cuticle is torn. That wound is now closed, but my entire fingertip is now swollen and red”.

The advantage of generating an S-rule is that the model has a maximum input size that would be exceeded if the entire question and answer set were taken as input. Furthermore, generating an S-rule removes questions and answers that are not relevant to GPs. Another advantage of generating an S-rule is that it will match reasonably well with an S-rule written by a GP, minimising the fine-tuning required when training on GP-written S-rules and generated S-rules. The present study generated S-rules for 17,202 of the 17,833 total connected questionnaires.

7.2.3 ICPC code selection

In the SOAP note dataset, 1,305 unique ICPC codes are present. However, as explained in chapter 4: “Source data”, these codes can be simplified to only 776 unique ICPC codes. Of these simplified codes, 318 are related to symptoms, while 360 codes are related to diagnoses and diseases. The remaining 96 codes are related to operations and actions at the GP centre. They are removed from the dataset as these codes are not relevant because the codes are seen as “escape codes’ and “meaningless” codes[57] by the *NHG*[8] which is the organisation responsible for updating ICPC codes. 56 of these codes only occur once in the whole dataset and are not present in the connected questionnaire dataset.

As mentioned in section 4.3: “Data analysis”, approximately 60% of the dataset comprises 50 unique codes, and in Table 7.3, it can be observed that a large number of ICPC codes occur less than 0.1% of the time in the SOAP note dataset. In the connected questionnaire dataset, only 382 unique codes appear, as it is much smaller than the SOAP note dataset. Table 7.2, shows that 71 of the 382 codes only occur once in the questionnaire dataset.

Codes that only occur once cannot be divided equally in a training, validation and test set. If the code only occurs in the training set, the model may learn to predict it accurately, but the prediction cannot be validated. On the other hand, if the code only occurs in the test set, the model has not been trained on it and cannot predict it accurately. Therefore, a trade-off must be made between including all codes and splitting them equally among the three sets.

In the SOAP note dataset, 14 ICPC codes that only occur once or twice are removed to allow for equal distribution of the codes in the training, validation, and test sets. These codes represent a negligible proportion of the total dataset. When the model is trained on this dataset, the weights in the sequential layer are updated, and the rest of the RoBERTa model is updated. Even codes that occur only a few times update the word and sentence representations, which may increase the model’s knowledge.

For the questionnaire dataset, no codes are removed. The questionnaire dataset is small compared to the SOAP note dataset, which raises the question of whether the model would benefit from

fine-tuning on the questionnaire data. When the model is trained on all ICPC codes in the SOAP note dataset, it is already trained on ICPC codes that may only occur once or twice in the questionnaire dataset; hence they do not need to be removed from the questionnaire dataset. If the model does not benefit from fine-tuning it on the questionnaire dataset (as seen in chapter 8: "Diagnoses prediction model"), it can still be used for testing the model. In this case, the most fair comparison would be when no codes are removed.

How much does the code occur	How many ICPC codes satisfy this condition
<= 1	6
<= 2	14
<= 5	28
<= 50	149
<= 100	215
<= 500	363
<= 1,000	427

Table 7.2: The amount of codes that occur <= x times in the questionnaire dataset.

How much does the code occur	How many ICPC codes satisfy this condition
<= 1	71
<= 2	112
<= 5	180
<= 50	307
<= 100	333
<= 500	375
<= 1,000	379

Table 7.3: The amount of codes that occur <= x times in the connected SOAP dataset.

7.3 Generating Dataset

This study created two distinct datasets, each consisting of three sets for training, testing, and validating the model. The SOAP note dataset includes all S-rules and simplified ICPC codes, and the questionnaire dataset includes all questions and answers, the generated S-rules, and the simplified ICPC codes. To ensure the model is evaluated on unique data, the questionnaire dataset is mutually exclusive to the SOAP note dataset. The ICPC codes in the SOAP note dataset are divided into different sets in a stratified manner. The distribution ratios of the split sets, and their corresponding lengths, are shown in Table 7.4. To achieve a fair validation, the remaining ICPC codes in both datasets are distributed equally among each set. The testing set is composed of 10% of the entire dataset, while the remaining 90% is split into training, and 10% is taken for a validation set. The total questionnaire dataset can also be used for validation if fine-tuning has no benefit.

Dataset	Split	Train_set length	Test_set length	Validation_set length
SOAP note dataset	∼80%/∼10%/∼10%	1,048,575	221,692	199,523
Questionnaire dataset	∼80%/∼10%/∼10%	13,932	1,721	1,549

Table 7.4: The length of each set in both datasets.

7.4 Conclusion

In this section, a total of 17,000 questionnaires were connected with their corresponding SOAP note and ICPC code. The questionnaires were converted from tabular data into natural language using a codebook provided by *Digidok*. This conversion was necessary to allow the data to be used for the training and validation of the RobBERT language model.

It is important to note that there is no definitive performance measure available to evaluate the quality of the questionnaire connections. To address this limitation, both manually connected and automatically connected datasets from the same time period were compared to each other to assess the quality of the connections. However, it should be acknowledged that when training and validating the model using this data, the ground truth may not be entirely accurate, and the model's actual performance could differ from what the performance measure suggests, either being higher or lower.

Another limitation of the study relates to the utilisation of ICPC codes. Specifically, the decision was made to employ simplified codes and eliminate certain codes that appeared infrequently (occurring only once or twice) within the 2.2 million SOAP notes as they could not be stratified equally between train/test/validation sets. It is crucial to recognise that the model lacks the training with these excluded codes, thus decreasing its ability to make accurate predictions regarding them. Furthermore, the variation in the ICPC code distribution across the SOAP note and questionnaire dataset should be considered during the model's validation process. The difference in distribution causes the model to have great performance on a ICPC code which occurs often in the SOAP note dataset but which is not present as much in the questionnaire dataset.

Chapter 8

Diagnoses prediction model

As described in chapter 6: “Methodology”, the decision was made to utilise RobBERT, a RoBERTa-based language model that was trained on a Dutch dataset. The decision to use a language model was motivated by the substantial amount of natural language data present in both datasets. Furthermore, it was observed that GPs employ ICPC codes for diagnostic documentation. Consequently, ICPC codes were selected as the output of our model.

Previous research revealed that domain-specific RoBERTa-based language models outperformed the standard RoBERTa language model[44]. Hence, the model should first be pre-trained on the 2.2 million S-rules in the dataset, which closely represent the questionnaires and encompass subjective symptoms and patient narratives. Figure 8.1 provides a comprehensive outline of the two training stages employed for this model. It should be noted that the maximum input size of the RobBERT/RoBERTa model is 768 and the vocabulary size is 40.000 which is also the output size for Masked Language Modelling(MLM).

The initial stage involves pre-training, during which the model is trained in the domain of Dutch GPs. The second stage, fine-tuning, leverages the domain-specific model and its updated representations to predict ICPC codes.

In the following sections the training and the validation of the model will be described in more detail.

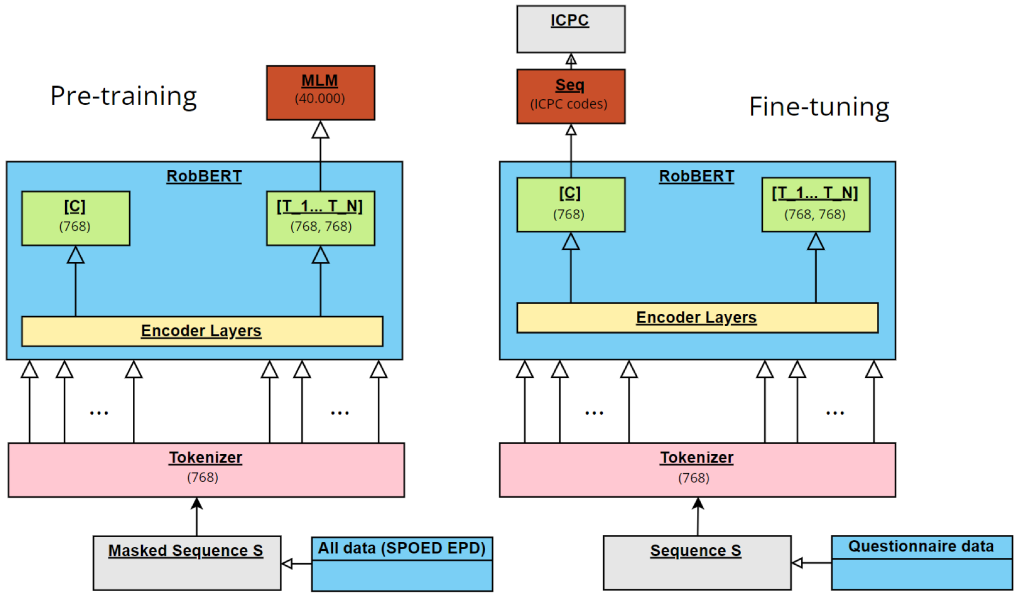


Figure 8.1: Simplified example of pre-training and fine-tuning.

8.1 Training environment and settings

For the following experiments, a single NVIDIA T4 16GB was used. Custom code created with Python, Pytorch and HuggingFace was used to load, develop, train and validate the model. For training the model with Masked Language Modelling (pre-training) and afterwards training the classification head of the model (fine-tuning), the same settings as initially used in the RoBERTa and RobBERT papers [40][17] were taken. The settings enable Gradient checkpointing and gradient accumulation steps to allow having a batch size of 64 for pre-training and a batch size of 64 when fine-tuning. The settings used can be seen in Table 8.1.

Gradient accumulation steps accumulate gradients over several batches and only steps the optimiser after a certain number of batches have been performed. This requires less memory allowing for larger batch sizes. Having a batch size of 32 and a gradient accumulation step of 2 is the same as having a total batch size of 64 because it will process two batches of 32, before stepping the optimiser.

Usually, all activations are saved during a forward pass to compute the gradients requiring a significant memory overhead. Gradient checkpointing strategically saves activations, so only a fraction of the activations need to be saved to calculate the gradients. Calculating the gradients with gradient checkpointing reduces the memory overhead while slightly increasing training time.

Mixed precision training (FP16 parameter) saves memory. Typically, variables are stored in 32-bit floating precision. With Mixed precision training, the model is allocated twice in the memory(16 and 32-bit). The backwards/forwards pass are still saved in 32-bit precision, but the activations for the gradient computation are in 16-bit precision to save memory. Enabling FP16 can have a slight impact on the training performance, as noted by NVIDIA[1] in their documentation about the subject.

Setting	**Value**
Batch size	32(pre-training) / 64 (fine-tuning)
gradient accumulation steps	2(pre-training) / 1 (fine-tuning)
gradient checkpointing	True
fp16	True
Weight decay	940
Learning rate	$5e{-}^{5}$
Adam epsilon	$2e{-}^{8}$
Warmup steps	250

Table 8.1: Settings used for training the model.

8.2 Tokenizer

The tokenisation process employed in this study uses the byte-level Byte Pair Encoding (BPE) tokenizer, which is loaded with the same vocabulary and tokens as the RobBERT model. The RobBERT tokenizer has a vocabulary size of 40,000, meaning there are 40,000 possible tokens. A custom tokenizer is not created because it would require retraining the RobBERT model from scratch, as the internal representations of RobBERT are mapped explicitly to the same vocabulary and tokens as the RobBERT tokenizer. Using a new tokenizer is equivalent to using an untrained RoBERTa model.

[1] https://docs.nvidia.com/deeplearning/performance/mixed-precision-training/index.html

The byte-level BPE tokenizer from RobBERT is initially trained by pre-tokenising the training set and then splitting words into symbols. The frequency of each possible symbol pair is calculated, and the symbol pair with the highest frequency is merged and added to the vocabulary. The training process is repeated until the desired vocabulary size is reached (40,000). If a word or symbol is absent in the vocabulary, it is replaced by the <unk> symbol. The first and last token of the RobBERT tokenizer always corresponds to the<s> and </s> symbols. In case there are fewer tokens in the input than are required as input for the RobBERT model, extra padding tokens (<pad>) are added until the input size is reached. If there are more tokens than the input size of the RoBERT model, the tokens get concatenated to the necessary length.

8.3 Masked Language modelling/ pre-training

As described in section 6.2: "RoBERTa", in BERT/ RoBERTa language models, pre-training plays a critical role in adapting the model to specific domains, especially when the input data is from a different domain than the original model. In the case of RobBERT, which is Dutch-specific, further pre-training on S-rules in the SOAP note dataset is required to make it domain-specific to the Dutch GPs vocabulary.

The pre-training process begins with loading the weights of the pre-trained RobBERT model into the RoBERTa model from Huggingface. A data collator is then added to the model to mask tokens from the output of the tokenizer randomly. There is a 15% (optimal value according to RobBERT and RoBERTA paper) chance that a token from the tokenizer is masked. In 80% of these cases, the masked token is replaced with the special token(<mask>). In 10% of the cases, the masked tokens are replaced by a random token from the vocabulary. In the 10% remaining cases, the masked tokens are left as is to bias the representation towards the actual observed word. The output is concatenated or padded with padding tokens for the input to corresponding with the input length of the RoBERTA model (512). The output of the RoBERTA model is a representation vector of length 768 for each input token. The representation vector for the masked token is used to predict the original token using a language modelling head added to the RoBERTa model.

The language modelling head can be seen in Figure 8.2 and includes a linear layer with the same size as the representation vector(768), an activation layer, a normalisation layer, and a final linear "decoder" layer with the output size the same as the vocabulary size. For the activation function, GELU[58] is used by the RoBERTa paper. The normalisation layer calculates the mean and variance for each item in a batch of activations and normalises each. Normalisation allows the output to remain generalisable and not reach high values. In the final linear "decoder" layer, each output neuron represents a token in the vocabulary. The softmax function (see Formula Figure 8.4) converts the final output to probabilities. The resulting probability indicates the most probable token that was masked. Optimally the probability for the correct token is 1.00, while the probability for the other tokens is 0.00. The weights in the linear layers and the representation in the model are updated using the Cross-Entropy loss (see Formula Figure 8.3), which is back-propagated to update the representation in the RobBERT model and the weights of the language modelling head layers.

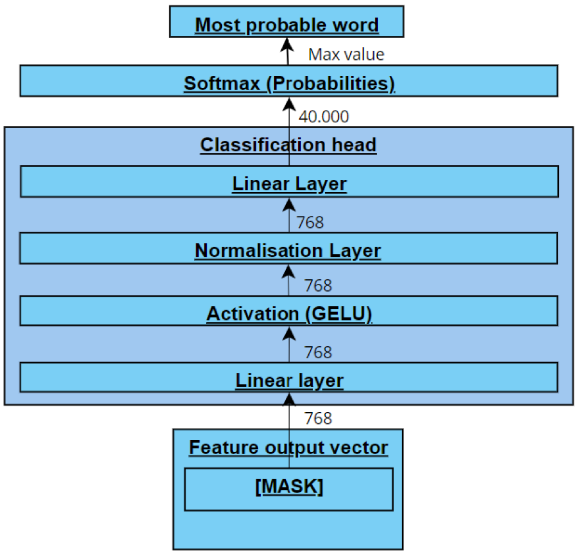


Figure 8.2: Example of the masked language modelling head

$$L_{\mathrm{CE}} = -\sum_{i=1}^{n} t_i \log\left(p_i\right), \text{ for } n \text{ classes}$$

Figure 8.3: Cross-entropy loss:Where t_i is the ground truth and p_i is the Softmax probability for the i^th class

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in R^K$$

Figure 8.4: Softmax function: normalises the outputs and makes the values sum to 1. z_i represents the output of neuron i. Euler's number increases the probability of the biggest score and decreases the probability of the lower scores compared to standard normalisation.

Training the model took three epochs (∼49,000 batches) for the loss to stabilise at approximately 1.00. The MLM word prediction accuracy of the original RobBERT model was 47.3%, while the trained model had an MLM word prediction accuracy of 76.7%. Interestingly, the non-Dutch-specific base RoBERTa model with the base tokenizer had a higher accuracy than the Dutch-specific model of 52.7%. RoBERTa is trained on an English dataset, so why it performs higher on this Dutch dataset is interesting. The English RoBERTa tokenizer causes Dutch words to split up into many tokens. Small tokens are easier to predict than larger tokens. To illustrate how the domain-specific representation of the RobBERT model can help predict ICPC codes, Table 8.2 compares the MLM in the standard RobBERT model and the updated one.

Masked sentence	**Original word**	**Updated RobBERT**	**Base RobBERT**
herkent klachten niet, zeer intense pijn, op: lippen/<mask>/keel	mond	mond, ogen, oog	lippen, lip, riemen
ziet grauw, iets <mask>, jeuk, rond beet, rood, zwelling, kring, insect: teek	benauwd	pijn,pijnlijk,benauwd	dergelijks, je, j
Vanochtend wakker geworden met pijn zijkant van voet. Kan er niet meer op <mask>	lopen	staan, lopen, steunen	reageren, zitten, vertrouwen

Table 8.2: Example masked sentences, with the top-3 predicted tokens from MLM in the RobBERT and the updated RobBERT model.

8.4 Classification

After pre-training, the model can get trained to predict ICPC codes. To predict ICPC codes, the updated base of the model is used. A new classification head is added instead of the language modelling head; see Figure 8.6. The proposed classification head consists of a linear layer of the size of the representation vector (768), a dropout layer, an activation layer, another dropout layer, and a linear output layer where the output has as many output neurons as there are ICPC codes.

The dropout layers prevent over-fitting by randomly setting an input to 0. The value/chance of this happening is the same as being used in the RoBERTA and RobBERT papers (0.1). For the activation function, Tanh (see Figure 8.5) is used in accordance with the RoBERTa paper.

As specified in chapter 7: "Dataset", 14 ICPC codes are filtered out of the dataset because there were insufficient entries to test them correctly. However, it was chosen to have all possible simplified ICPC symptom and diagnosis/disease codes as outputs neurons. There are 678 output neurons, 318 being symptoms and 360 being diagnoses/diseases. Some outputs neurons can never be tested or updated, but for future works, one only needs to continue training with more data to consider all ICPC codes.

For training, the Cross-Entropy loss function (see Figure 8.3) was again used to calculate the loss and update both weights in the classification head and representations using backpropagation.

$$\tanh(x) \doteq \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Figure 8.5: Tanh activation function: was chosen by the RoBERTa paper because it gives higher performance compared to, e.g. a sigmoid function for multi-layer neural networks.

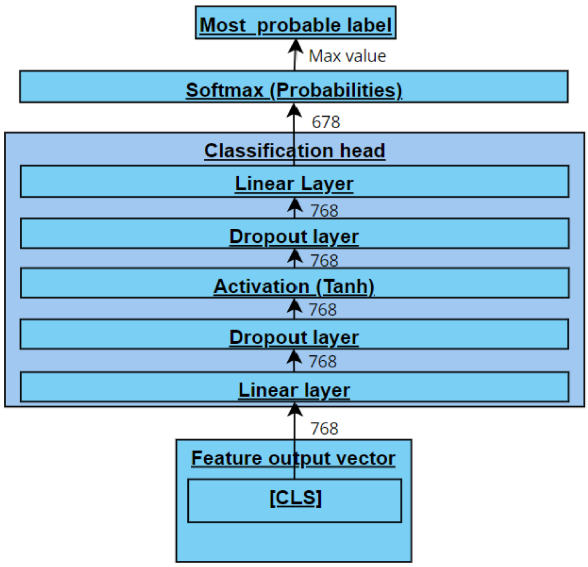


Figure 8.6: Example of the classification head

8.5 ICPC hierarchical structure

Training the model on S-rules with the standard sequential head achieved an accuracy-at-1 of approximately 52.4%. However, accuracy-at-1 may not be an appropriate performance measure in this

context. A significant challenge in this study is to incorporate a hierarchy that emulates the differential diagnosis process of a GP.

GPs may provide a less specific ICPC code when they are uncertain. Ideally, the loss function should not heavily penalise the model when it produces a less specific code than the ground truth. However, it is difficult to establish a hierarchy among less specific and more specific codes in ICPC, as symptoms, diagnoses, and diseases are often correlated. There is also no documentation or research that models a hierarchy. Additionally, GPs may prioritise different symptoms when diagnosing and are not aware of all existing codes. An example can be found in the SOAP note test set where a user reported: "My toe is infected", while the ground truth of the S-rule was "S11: Local infection of/under the skin", the trained model predicted "S09: Local infection finger/toe/cuticle" which seems a better fitting ICPC code for the case. Cases like these, however, can significantly affect the accuracy-at-1 and the model's ability to learn. In the journal of Zwaanswijk et al.[57], it was concluded that GP practices give "meaningful" codes 64.8% of the time. The quality of "meaningful" codes meaning if the most "accurate" is chosen for the diagnosis is not yet researched. In some cases, a further medical examination is also needed to provide a specific code, which is not present in the S-rule. These challenges will be further explored in section 9.2: "User study"

To address these challenges, two symptom/diagnosis architectures are proposed to improve the model's ability to predict a correct ICPC code. section 8.6: "Performance" introduces a new approach to consider the possibility that multiple diagnoses may be valid and a new method to measure the diagnostic ability of the model.

8.5.1 Proposed architectures

To address the challenges outlined in the previous section, two architectures were tested to enhance the model's ability to predict diagnoses accurately. The available hierarchical information only includes whether a code represents a symptom or a diagnosis/disease. Although ICPC codes include a letter indicating a category, symptoms and diagnoses may have different categories while closely related. Two classification heads were created instead of the single classification head used in the previous section to implement a hierarchy. One head, called the "Symptom head," was used to classify symptoms and had 318 output neurons, each representing an ICPC symptom code. The other head, called the "Diagnosis head," was used to classify diagnoses and diseases and had 360 output neurons, each representing an ICPC diagnoses or disease code.

The rationale behind this approach is that if the model can accurately classify ICPC codes that are symptoms and diagnoses, these layers can be interchanged to obtain the most likely ICPC symptom code for a ground truth with an ICPC diagnosis code and vice versa. The following two subsections propose two different architectures that use these two new classification heads.

Both proposed architectures were trained on the S-rules in the relatively small questionnaire dataset to get determine if it was worth training the model on the larger SOAP note dataset for a long period.

8.5.2 Stacked symptom and Diagnosis layer

One proposed architecture involves stacking both the Symptom head and Diagnosis head, as shown in Figure 8.7. With the proposed architecture, symptoms are predicted, and the probabilities of these symptoms can be used to predict a diagnosis. The architecture resembles the counterfactual diagnostic algorithm explained by Richens et al. [42] in chapter 3: "Related works". Richens et al. proposed that symptoms can be a direct or indirect cause of disease and created an algorithm that predicts diseases based on this idea. The primary advantage of the proposed approach, which also applies to

the proposed architecture, is that disease can be directly explained by examining the probabilities of the symptoms. While the proposed architecture seems an ideal solution in theory, several downsides make it infeasible.

One significant problem is that the context inside an S-rule, which can be critical to a diagnosis, is only used to predict symptoms. The only input that the Diagnosis head receives is the output of the Symptom head. For example, consider the sentence, ”My COVID test is positive; I have symptoms X, Y, and Z.” While the sentence would cause the Symptom head to have a high probability for symptoms X, Y, and Z, critical information such as the fact that the patient has a positive COVID test may not be passed through to the Diagnoses head.

Another issue is that calculating the loss of the Diagnoses head will cause the Symptom head to update as well. The updated Symptom head resulted in forgetting the correct weights in the Symptom head and causing low-frequency symptom neurons to update to a different meaning, such as passing through context.

The issue can also be seen when training the model. When training the model on S-rules from the questionnaire dataset, the model performance converges after 1000 steps. The accuracy for the diagnosis head was 39%, indicating that the model could accurately predict a diagnosis/disease code 39% of the time. However, the accuracy for the symptom head was only 11%, indicating that the model had difficulty predicting symptom codes because the diagnosis head also updated the symptom head.

An alternative training method was attempted to address this issue, involving two phases. In the first phase, the symptom head was trained by only training on all ICPC symptom codes. In the second phase, the diagnosis head was added, and the symptom head was frozen to no longer be updated. Only the diagnosis head was then updated. This strategy improved the symptom head accuracy to 58%, but decreased the accuracy of the diagnosis head to 31%. While the model’s performance was better than the previous attempt, it was still not better than the original model.

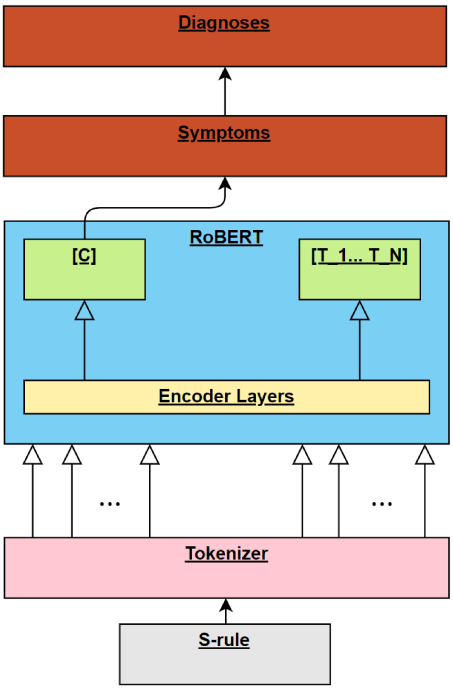


Figure 8.7: Stacked layer model

8.5.3 Side-by-side Symptom and Diagnosis layer

The proposed architecture involves two layers: the first 318 neurons represent the symptom ICPC codes, and the last 360 neurons represent the diagnosis/disease ICPC codes. Unlike the previously proposed stacked approach, these layers are side-by-side, forming one sequential layer, as depicted in Figure 8.8. During training, separate loss functions are used for the symptom and diagnosis/disease neurons. The ground truth information is used to compute the loss function for only the corresponding set of neurons. Specifically, when the ground truth is a symptom, the loss function only considers the symptom neurons, even if the highest activated neuron corresponds to a diagnosis/disease symptom. The separate loss function allows the model to deal with the ambiguity present in the labels by considering the hierarchy of symptoms and diagnoses. The model can also pass context to the sequential layer to predict symptoms and diagnoses/diseases.

The side-by-side architecture still makes a prediction explainable with high accuracy for the diagnoses/disease codes and the symptom codes in the dataset. The highest activated diagnosis/disease codes can be substantiated by examining the highest activated symptom codes and vice versa, as shown in Figure 8.9. The softmax function is applied separately to the symptom and diagnosis/disease neurons, and the resulting outputs reveal that code S09 is the most activated symptom code. In contrast, code S76 is the most activated diagnosis/disease code. According to GPs in the section 9.2: “User study”, both codes are sufficient for a diagnosis, with S76 being more specific than S09 as it corresponds to a diagnosis rather than just a symptom. Further details on the loss function used in training and the model’s performance can be found in the subsequent sections.

The proposed architecture was again trained for 1000 steps on the questionnaire dataset. The resulting accuracy for predicting diagnoses and disease codes was 56%, while the accuracy for predicting symptom codes was 58%. The combined accuracy for predicting either code was 42%. These findings show that the model’s performance is increased by splitting the final layer into diagnoses/diseases and symptoms. This proposed architectural modification was chosen to improve the model’s performance further. In the following section, the model is fully trained on the SOAP note dataset and validated.

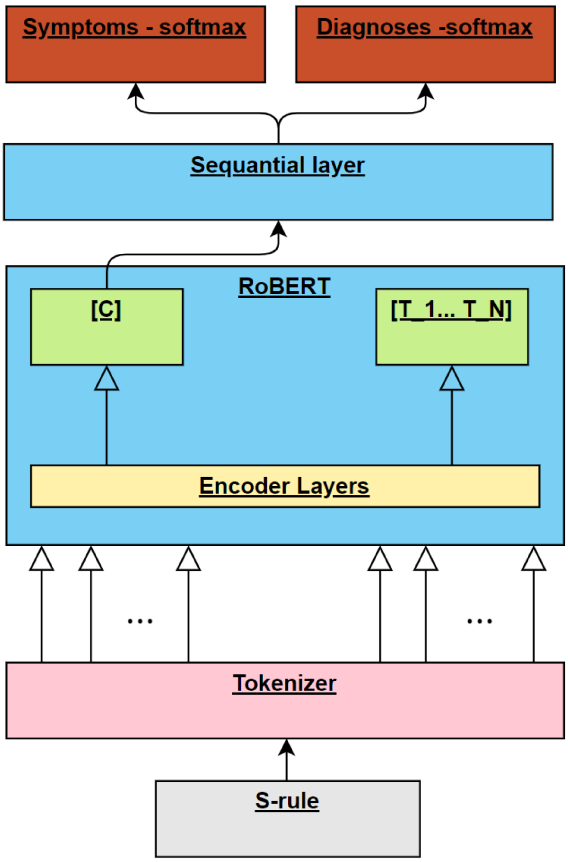


Figure 8.8: Side-by-side layer model

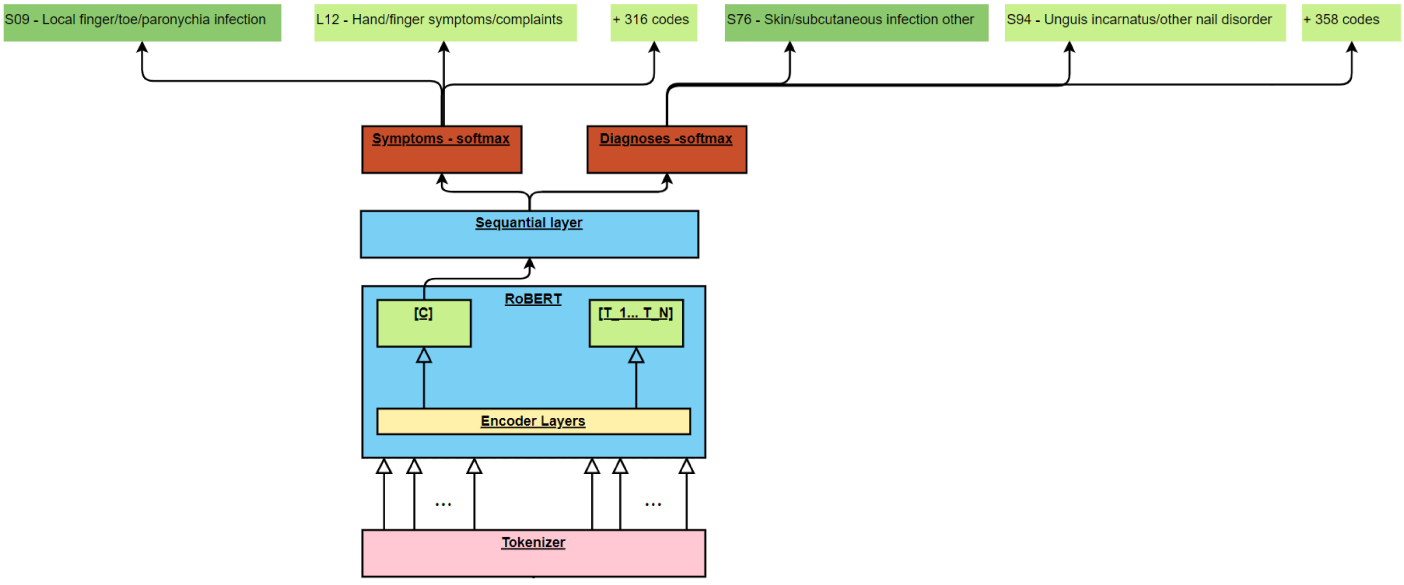


Figure 8.9: Example of output

8.6 Performance

The proposed architectures were only trained on the questionnaire dataset in the previous section. The best-performing architecture was chosen to be improved by first training it on the SOAP note train set, which contains 1.2 million S-rules. The side-by-side architecture is chosen to train thoroughly on the SOAP note train set because of its performance compared to the original "plain" architecture. First, in the following sections, new performance measures are described. Finally, the full performance of the model can be found in Table 8.3. Overall the model's performance improved when looking at the diagnosis and symptom accuracy.

Metric / model(test/val)	Plain classification head (%)	Side-by-Side classification head (%)
Accuracy	**52.4** / **52.6**	51.4 / 51.4
Accuracy top-3	78.2 / 78.2	**78.4** / **78.4**
Accuracy threshold	84.6 (4.0 picked) / 84.5 (4 picked)	83.5(3.7 picked) / 83.6 (3.7 picked)
Macro accuracy	19.9 / 19.9	**22.2** / **22.5**
Macro accuracy top-3	37.1 / 37.6	**42.6** / **42.3**
Macro accuracy threshold	48.2 / 48.2	**52.7** / **53.0**
Symptom accuracy	65.9 / 66.1	**66.9** / **66.9**
Symptom accuracy top-3	87.9 / 87.7	**88.8** / **88.6**
Symptom accuracy threshold	90.0 (3.0 picked) / 90.0 (3.0 picked)	90.0 (2.7 picked) / 89.7 (2.7 picked)
Symptom macro accuracy	27.3 / 27.1	**30.5** / **30.0**
Symptom macro accuracy top-3	46.8 / 46.6	**51.5** / **51.4**
Symptom macro accuracy threshold	56.1 / 56.3	**60.4** / **59.3**
Diagnosis accuracy	63.4 / 63.5	**64.9** / **65.0**
Diagnosis accuracy top-3	85.6 / 85.6	**87.1** / **87.5**
Diagnosis accuracy threshold	89.0 (3.3 picked) / 89.0 (3.3 picked)	88.6 (2.9 picked) / 89.1 (2.9 picked)
Diagnosis macro accuracy	29.4 / 29.5	**36.3** / **36.7**
Diagnosis macro accuracy top-3	47.8 / 47.9	**57.3** / **57.2**
Diagnosis macro accuracy threshold	57.0 / 56.7	**63.3** / **64.0**

*Table 8.3: Model's performance versus the plain classification head. **BOLD** means better performing.*

8.6.1 Symptom accuracy and Diagnoses accuracy

More than using accuracy as a performance measure is often required, as the side-by-side model predicts symptoms and diagnoses/disease codes separately. That is why a separate accuracy is added for ICPC symptom codes and ICPC diagnoses/disease codes. The activations of the first 318 neurons in the last linear layer determine the symptom accuracy. Specifically, for each entry in the dataset where an ICPC symptom code represents the ground truth, the activations of the 318 output neurons are converted into probabilities using the softmax function (refer to Figure 8.4). Notably, the evaluation does not consider the remaining 360 neurons that represent ICPC diagnosis/disease codes. To predict the accuracy of ICPC diagnoses/disease codes, the activations of the last 360 neurons are put in the softmax function. The accuracy function calculates the accuracy, as seen in Figure 8.10. The accuracy function does not take into account the distribution of the ICPC codes which may not correctly represent the actual performance of the model.

As seen in Table 8.3, the model has a symptom accuracy of 66.9% and a diagnosis accuracy of 64.9% on the test SOAP note set. The accuracy is much higher than the normal accuracy of 51.4% because of its preconditions on the knowledge that the ground truth is an ICPC diagnosis/disease or symptom code.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Figure 8.10: where TP = True positive; FP = False positive; TN = True negative; FN = False negative

8.6.2 Accuracy top-k

The accuracy function used in this study evaluates the model's performance based on the highest probability neuron in the output layer. However, in a clinical setting, multiple ICPC codes can represent a patient's diagnosis, and different general practitioners may assign other ICPC codes for the same symptom or condition (as observed in section 9.2: "User study"). As a result, more than a standard accuracy measure may be required as it otherwise would lead to sub-optimal performance. A top-x accuracy metric is employed to address this issue, where k denotes the number of highest probability neurons considered. The top-k accuracy measures whether the ground truth ICPC code is among the k highest probabilities. A top-3 and top-5 accuracy represent a reasonable number of potential ICPC codes that can be presented to a GP without overwhelming them with too many options.

As seen in Table 8.3, the model has a symptom accuracy top-3 of 88.6% and a diagnosis accuracy top-3 of 87.1.9% on the SOAP note test set.

8.6.3 Macro average accuracy

The SOAP note and questionnaire datasets are unbalanced. A minority of the labels (approximately 10% of the total) account for most of the dataset (around 60%). In such cases, it is essential to evaluate the model's ability to predict each label accurately rather than solely considering its overall performance on the entire dataset. The macro average accuracy function, as shown in Figure 8.11, accounts for this issue as it sums the accuracy of each label and divides it by the number of labels.

As shown in Table 8.3, the model has a lower macro accuracy than the normal accuracy top-1/top-3. The outcome was predictable due to the dataset's unbalanced distribution of ICPC codes. Nevertheless, it is a good performance given that 333 out of the total 678 ICPC codes occur less than 100 times in a dataset of 2.2 million S-rules. The decision not to remove the ICPC codes appearing less than 100 times from the dataset was made to provide a fair performance measure and to ensure comparability with future works that may have more data available.

$$\text{Macro Average Accuracy} = \frac{\sum_{n=1}^{N} Precision(n)}{N}$$

Figure 8.11: where N = the amount of classes, for precision formula see Figure 8.12

$$\text{Precision} = \frac{TP}{TP + FP}$$

Figure 8.12: where TP = True positive; FP = False positive

8.6.4 Threshold

After the last linear layer, the output values are passed through the softmax function to transform them into probabilities. The Accuracy top-k function selects the top-k probabilities as the predicted set. However, the method does not consider the probabilities of the selected set. When the model is uncertain, the returned set may need to be larger as there are more possible labels. Conversely, if the highest probability in the model is very high, the model may need to return fewer possible labels as the first label is likely to be the ground truth. Suggesting several diagnoses corresponds directly how a Dutch GP provides a differential diagnosis. When the GP is uncertain about the diagnosis, more possible diagnoses are given. Conversely, only one diagnosis is given when the GP is confident about the diagnosis.

A threshold function is implemented to sum the highest probability neurons until a threshold value is met. These neurons are then considered as possible ICPC labels. The threshold value can be changed to increase performance, with the downside that more ICPC codes are suggested. The most optimal threshold value is chosen(90%), which has better performance than the Accuracy top-3 method but, on average, returns fewer than 3 possible labels. It should be noted that neurons with a probability of less than 1% are not considered for the threshold function.

As seen from Table 8.3, the threshold function yields higher accuracy than the top-3 diagnosis and symptom accuracies. Moreover, the threshold function generates fewer than three ICPC codes on average. The original classification head sometimes performed better but picked more ICPC codes on average than the top-3 accuracy.

8.7 Fine-tuning

The model is trained on 2.2 million S-rules in the SOAP note dataset but needs to be fine-tuned on the questionnaire dataset. Even when different learning rates were employed, the model's overall performance slightly decreased after fine-tuning. These findings suggest that the questionnaire dataset lacks sufficient new information for the model to learn and that the generated S-rules correspond well to those written by GPs in the SOAP note dataset. This observation led to the decision to only use the questionnaire dataset as an additional validation set to determine whether the model's performance on the generated S-rules is comparable to the written S-rules and whether it can be used in practical applications to predict ICPC codes when a patient fills in a questionnaire. Two validation sets were constructed from the questionnaire dataset, one utilising the entire dataset, which has a length of 17,833, and the second using the automatically connected dataset, with a length of 5,356. If the complete questionnaire dataset is correctly connected to the s-rules from the SOAP note dataset, the model's performance on these two sets should be identical. Table 8.4 presents the model's performance on these two sets and with the SOAP note test set as reference. The macro average accuracy is lower in the total set than the auto set because it contains more unique ICPC codes. It can be seen that the model performs slightly worse on the auto and total set compared to the test set from the SOAP note dataset. The total set performs worse than the auto set because some entries are wrongly connected compared to the perfectly connected auto set, as also specified in subsection 7.2.1: "Quality of connection".

Performance measure / validation set	Auto set (%)	Total set (%)	Test set SOAP note (%)
Accuracy top-1	49.4	48.7	51.4
Accuracy top-3	74.7	72.4	78.4
Accuracy threshold	80.1 (3.8 picked)	77.3 (3.7 picked)	83.5 (3.7 picked)
Macro accuracy top-1	20.3	17.6	22.5
Macro accuracy top-3	45.7	39.2	42.6
Symptom accuracy top-1	65.1	63.3	66.9
Symptom accuracy top-3	85.4	82.2	88.8
Symptom threshold	87.6 (2.84 picked)	84.3 (2.84 picked)	90.0 (2.7 picked)
Symptom macro accuracy top-3	58.5	50.9	51.5
Diagnosis accuracy top-1	61.5	59.9	64.9
Diagnosis accuracy top-3	83.4	80.5	87.1
Diagnosis threshold	86.5 (3.1 avg)	83.2 (3.1 picked)	88.6 (2.9 picked)
Diagnosis macro accuracy top-3	62.3	52.6	57.3

Table 8.4: Performance on questionnaire dataset. The avg stands for the average amount of "activated" neurons.

8.7.1 Weight of certain questions

To improve model performance, a common approach is to adjust model architecture or hyperparameters. However, an alternative approach is to modify the input data instead. For the data used for training the model, this is infeasible. GPs write the s-rules used for training in the SOAP note dataset. These S-rules cannot be modified because the context of each S-rule can be completely different. The s-rule in the questionnaire dataset used for validation can be modified. The s-rules from this dataset are generated with the answers to the questionnaire of patients. The S-rule can be modified by removing columns of specific questions/categories and then regenerating the S-rule. The following modified S-rules were generated: 1. Without medication, 2. Without operation, 3. Without open questions. Medication and operation are removed because medication and operations were never a factor when diagnosing a patient by GPs in section 9.2: "User study". Note that the S-rules without open questions also remove medication and operations since they are answered in an open-question field. Table 8.5 compares the performance of these modified S-rules to the original S-rule. The comparison is made on the automatically connected questionnaire dataset to ensure that the S-rule corresponds to the ICPC code. The Table shows no significant improvement in any of the performance measures. It does, however, show that open questions improve the model's performance significantly, which is to be expected because it gives a context not present in closed questions.

Performance measure/ S-rule	**No medication(%)**	**No operations(%)**	**No open questions (%)**	**Original (%)**
Accuracy	49.3	**49.5**	43.2	49.4
Accuracy top-3	74.4	74.6	67.2	**74.7**
Symptom accuracy	**65.4**	65.1	58.3	65.1
Symptom accuracy top-3	85.2	**85.4**	78.3	**85.4**
Diagnosis accuracy	61.2	**62.1**	52.9	61.6
Diagnosis accuracy top-3	83.3	**83.4**	74.2	**83.4**

Table 8.5: The performance of the modified s-rules. Bold indicated the best-performing s-rule for that performance measure. Tested on the automatically connected questionnaire dataset.

8.8 Performance Versus other models

The RoBERTa language model may be overly complex for the task at hand, and a more straightforward model may achieve similar results while being computationally less expensive. Prompting a research question: “Can a simpler model achieve comparable performance to the RoBERTa model for predicting ICPC codes using S-rules?”. Two baseline models, a Naive Bayes model[59] and a random forest classifier[60], were also trained on the same (total) questionnaire dataset. The aim is to compare the model’s performances against the RoBERTa model.

A Naive Bayes classifier was selected as a baseline model because it is a simple language model that utilises word frequencies to calculate the log-likelihood of a word corresponding to a class. The most likely class of a document is determined by summing the log-likelihood of each word in the document per class and selecting the class with the largest sum.

A random forest classifier was also trained as a baseline model to determine whether a language model is appropriate because a decision tree has a similar structure as a questionnaire. The random forest classifier consists of multiple decision trees, where each decision tree predicts the same data entry. The most predicted class is chosen as the most likely class.

Both baseline models showed a drastically decreased accuracy (24.4% and 9.5%) and macro average accuracy (2.8% and 0.4%) and were not considered to improve further. Because of version changes in the questionnaire dataset, many identifiers changed definitions (e.g. BK040 $\rightarrow$ BK050) hence the random forest classifier had too many columns to build a good decision tree.

Performance measure / validation sets	**Naive Bayes (%)**	**Random forest (%)**	**Original (%)**
train set	28.6 (1.8 macro)	8.5 (0.5 macro)	48.6 (17.9 macro)
Test set	24.4 (2.8 macro)	9.5 (0.4 macro)	49.0 (25.3 macro)
Val set	23.0 (2.8 macro)	7.8 (0.2 macro)	47.0 (21.2 macro)

Table 8.6: Baseline accuracy of Naive Bayes and random forest model compared to the baseline on the questionnaire validation sets. (macro is the macro average accuracy)

8.9 Conclusion

The proposed architecture demonstrated overall improvement compared to the standard classification head of RoBERTa. The validation techniques employed in this study aimed to capture the hierarchical structure inherent in the ICPC codes and employ a similar approach to a GPs process conducting differential diagnosis. Although fine-tuning did not enhance the model's performance, evaluating the model using the connected questionnaire dataset did not significantly decrease its performance. This outcome suggests that the generated S-rule and the connection method of the dataset performed effectively. Furthermore, the removal of questions and answers from the S-rule did not substantially impact the model's performance. Notably, the advantage of removing questions and answers is a shorter generated S-rule, reducing the likelihood of it not fitting as input for the model.

In chapter 6: "Methodology", there was a discussion on whether a language model was a suitable choice considering its computational complexity. However, experiments with a simple naive Bayes classifier and random regression tree revealed inadequate performance, reinforcing the selection of the RoBERTa model.

The key question lies in determining whether the model's performance is satisfactory for real-world use cases by actual general practitioners. While the performance appears promising, it is important to note that previous research[16] has highlighted discrepancies between validation set performance and actual performance due to shortcuts or cheating. To measure the model's actual performance, the model's fit for use and detect any potential cheating behaviour, it is necessary to establish model explainability, which is addressed in the subsequent section.

Chapter 9

Explainability

9.1 LIME

LIME includes a text-module that can identify the most influential keywords within a given text. The module generates various samples of the text, each with different tokens removed.

These new samples are used as input for the diagnoses prediction model to calculate new probabilities of the model's output for each sample. By comparing the probabilities of the modified samples with those of the original text, the most influential words or "keywords" can be identified. The number of samples is a variable that can be changed. More samples will provide better keywords but will take longer to compute. For this study, it was chosen to have 5 thousand sample. A white-box model such as Ridge regression is trained to make these explanations locally interpretable using the modified samples and their corresponding probabilities. This approach explains the predicted ICPC codes by showing the keywords that influenced the prediction the most.

During testing, it was found that keywords were often irrelevant, as they were present in every generated S-rule or were stopwords, such as "klacht/beloop" (complaint/course) or "hulpvraag" (question for GP). Each keyword has a score representing how influential it was to the prediction. The scores cause issues when attempting to provide the GP with an explanation. The score of each keyword does not represent a probability and cannot be translated to a reliable metric for deciding whether to include a keyword in the explanation. Determining the optimal number of keywords to be shown to the GP is hard. Showing too many causes non-insightful keywords to be included while showing insufficient keywords creates the opposite effect.

Furthermore, the LIME text-module only identifies single words as keywords, which does not fully represent how complex language models like RoBERTa make decisions. For instance, RoBERTa considers the context of multiple symptoms together to predict a particular ICPC code. The LIME text-module's output does not cover context but only considers single words, which, for an explanation, may need to provide more confidence to a GP about the model's predictions.

An example can be observed in the following translated S-rule (originally in Dutch and translated to English). The highlighted words indicate the most influential keywords for the top 2 most probable ICPC symptom codes.

> "initial complaint: skin complaints, fever, moderately ill, 4days-1wk skin complaints, does not recognise complaints, slightly sensitive (score 1), increase in complaints, location: nails, hands, complaints: red, hard spot, pus, swelling, cause: I had last week a cut on my cuticle. The cuticle is torn. That wound is now closed, but my entire fingertip is now swollen and red. Self-help: Nothing, because I don't know what I can do best. Request for help: advice on reduction of complaints, diagnosis, treatment"

Code	Keywords
"S09 - Local finger/toe/paronychia infection"	"cuticle" (0.31), "pus" (0.1), "fingertip" (0.1), "nails" (0.07), "reduction" (0.05)
"L12 - Hand/Finger symptoms/complaints"	"fingertip" (0.08), "hands" (0.06), "swelling" (0.03), "treatment" (0.02) and "nothing" (0.02)

Table 9.1: Five most influential keywords with scores.

As seen in Table 9.1, the highest probability ICPC symptom code was "S09 - Local finger/toe/paronychia infection". The five most influential keywords for the symptom code are "cuticle", "pus", "fingertip", "nails", and "reduction". The second highest probability ICPC symptom code was "L12 - Hand/Finger symptoms/complaints", with the five most influential keywords being: "fingertip", "hands", "swelling", "treatment" and "nothing". While the top keywords align well with the symptom, the subsequent keywords provide less meaningful insights.

A new symptom-module is proposed in subsection 9.1.1: "LIME symptom-module" that allows ICPC diagnosis/disease codes to be explained by ICPC symptom codes instead of keywords. Other ICPC symptom codes cannot explain ICPC symptom codes as they would only reference themselves. In this study, it was chosen to explain ICPC symptom codes by the text-module and ICPC diagnosis/disease codes by the symptom-module.

To evaluate whether the LIME text-module or the LIME symptom-module would be a valuable explanation for GPs, a user study was conducted to investigate whether they serve as a valuable explanation for GPs.

9.1.1 LIME symptom-module

In chapter 8: "Diagnoses prediction model", a side-by-side layer was proposed using a side-by-side layer to differentiate between the classification of symptom codes and diagnosis/disease codes. Predicting symptoms and diagnoses/diseases showed a promising performance. With high performance, an S-rule can get given both the highest probability symptom code and the highest probability diagnosis/disease code. The side-by-side layer enables ICPC diagnosis codes to be explained by identifying the most activated symptoms.

This development led to the LIME symptom-module, which shares the same foundation as the text-module. Initially, samples were generated by deleting tokens from an S-rule. Predicting these modified samples resulted in changes to the output of both symptom neurons and diagnosis/disease neurons. These modified outputs were then utilised to identify the most influential symptoms associated with a specific diagnosis/disease code.

The outputs of the symptom neurons and the diagnosis/disease neuron of interest were separately saved for further analysis. The cosine distance between the original sample and each modified sample was calculated to determine the importance of the ICPC symptom neurons for the selected diagnosis/disease neuron. Samples with a larger modification had a smaller similarity. These cosine distances were then employed as sample weights for a ridge model using a exponential kernel function on cosine distance. The sample weight indicates the importance of each sample for trainign the model. The kernel width determines how large the neighbourhood of the local model is. The training data of the ridge model consisted of the outputs of the symptom neurons and the output probabilities of the selected diagnosis/disease neuron that served as a ground truth label. The model had 678 trainable coefficient each representing a ICPC symptom code. The resulting coefficients of the ridge model (which has the same length as the ICPC symptom neurons) signified the importance of that code for the ICPC diagnosis/disease codes. The coefficients with the highest weights were consid-

ered the most influential and were sorted and returned by the symptom-module. An example of two explanations derived from the following translated S-rule:

> "initial complaint: eye complaints, less haze after blinking, sensitive (score 2), does not wear lenses, eye L, 1-3 days, does not recognise complaints, visual complaints: blurred vision, flashes of light, location: eyelid, corner of the eye, complaints: tears, burning, itching, redness, stinging, pus, pain on/around eye, swollen eyelid, additional: herpes face, hay fever."

Code	Symptoms
"F72 - Blepharitis/hordeolum/chalazion"	"F16 - Symptoms/complaints eyelids" (1.18) "F15 - Deviating aspect eye" (0.38) "F01 - Pain eye" (0.06)
"F70 - Infectious conjunctivitis"	"F02 - Red eye" (0.59) "F13 - Deviating feeling eye" (0.13) "F03 - Discharge from eye" (0.09)

Table 9.2: Three most influential symptoms with scores.

As seen in Table 9.2,The highest probability diagnosis/disease code, "F70 - Infectious conjunctivitis", is the most probable diagnosis together with "F72 - Blepharitis/hordeolum/chalazion". the most influential symptoms were checked with a GP and described the clinical picture of the corresponding diagnosis well.

9.2 User study

The primary objective of the diagnoses prediction model in this study is to assist Dutch GPs in diagnosing patients. The LIME model was modified to explain ICPC diagnosis/disease codes based on ICPC symptom codes. Although the modification does not represent the internal decision-making inside the RoBERTa model, it aims to aid GPs in their decision-making process. Evaluating the performance of these explanations requires input from medical professionals who possess the expertise to provide medical validation for an explanation. Since machine learning researchers typically need more medical qualifications to assess a model's real-world performance beyond the dataset and standard performance measures discussed in the previous sections, a user study involving Dutch GPs was conducted. The user study serves two purposes: 1) evaluating the performance of the model's explainability and 2) assessing the performance of the diagnoses prediction model itself.

Throughout the study, it has been highlighted that developing a diagnoses prediction model using ICPC codes presents a significant challenge. Multiple ICPC codes may apply to an S-rule, and the usage of ICPC codes can vary among GPs. These factors influence the model's performance measurement since only one ground truth is available. Achieving high accuracy, particularly in the top-1 accuracy, is considered challenging due to the variability of ICPC codes. The user study helps to accurately quantify the model's performance by providing different GPs for the same S-rule cases. By involving an adequate number of participants, it becomes possible to measure the model's performance on a small subset of S-rules.

As indicated in section 3.4: "Explainable AI", no performance measure is often present in XAI-related studies that evaluate the explainability of a model. Quantifying the accuracy of explanations

can be challenging since these explanations are often based on surrogate models like LIME. It is essential to define the purpose of the explanation, whether it is to assist researchers/developers in assessing the decision-making process of the model (e.g., identifying shortcuts) or to provide valuable insights to third parties such as doctors[45].

Nauta et al. [3], defined 12 explanatory quality properties known as the Co-12 properties (see Figure 9.1), each with a description with a definition and a key idea. It is important to note that a "good" explanation does not necessarily need to adhere to all 12 properties. Instead, the choice of Co-properties should align with the specific purpose of the explanation. Nauta et al. also provide recommendations on how to evaluate each property. Furthermore, they note that most XAI papers do not include a user study that measures the quality of the explanation to the actual users of the model, such as GPs.

	Co-12 Property	Description
Content	**Correctness**	Describes how faithful the explanation is w.r.t. the black box.
		Key idea: Nothing but the truth
	Completeness	Describes how much of the black box behavior is described in the explanation.
		Key idea: The whole truth
	Consistency	Describes how deterministic and implementation-invariant the explanation method is.
		Key idea: Identical inputs should have identical explanations
	Continuity	Describes how continuous and generalizable the explanation function is.
		Key idea: Similar inputs should have similar explanations
	Contrastivity	Describes how discriminative the explanation is w.r.t. other events or targets.
		Key idea: Answers "why not?" or "what if?" questions
	Covariate complexity	Describes how complex the (interactions of) features in the explanation are.
		Key idea: Human-understandable concepts in the explanation
Presentation	**Compactness**	Describes the size of the explanation.
		Key idea: Less is more
	Composition	Describes the presentation format and organization of the explanation.
		Key idea: *How* something is explained
	Confidence	Describes the presence and accuracy of probability information in the explanation.
		Key idea: Confidence measure of the explanation or model output
User	**Context**	Describes how relevant the explanation is to the user and their needs.
		Key idea: How much does the explanation matter in practice?
	Coherence	Describes how accordant the explanation is with prior knowledge and beliefs.
		Key idea: Plausibility or reasonableness to users
	Controllability	Describes how interactive or controllable an explanation is for a user.
		Key idea: Can the user influence the explanation?

Figure 9.1: The 12-co explanation quality properties with descriptions taken from Nauta et al. [3].

9.2.1 Setup

The user study aimed to measure six Co-explanation properties, specifically the presentation and user properties (Figure 9.1). The presentation and user properties were chosen because they can be measured via a user study and can only be validated by some personal opinion. The content property requires further analysis and cannot be measured by a participant. To avoid taking too much of the GP's time, each GP was presented with ten different cases, where each case consisted of a generated S-rule followed by the most likely ICPC codes (symptoms and diagnosis/diseases) using a threshold function as described in subsection 8.6.4: "Threshold". The first five cases were fixed to measure the variability in ICPC codes. Each case was randomly selected from the automatically connected

questionnaire dataset to ensure a fair comparison. Each ICPC code contains an “explanation”. If the ICPC code is a symptom, the explanation contains the top “keywords” according to the LIME text-module. On the other hand, if the ICPC code is a diagnosis/disease, the explanation consists of the top-3 most influential ICPC symptom codes according to the symptom-module. For keywords, there is always a minimum of two keywords shown.
Before the ten cases, background and explanation of the user study were given to ensure all GPs understood the study’s purpose and methodology. Each case was designed with specific questions to measure one of the co-12 quality properties or the model’s performance.

Figure 9.2 gives an impression of the question present in each case. Each question measures one of the co-12 quality properties or the model’s performance. Most questions can be answered on a Likert scale of 1-5, 1 representing “Totally disagree” and 5 representing “Totally agree”. The scale was chosen so that the average of each question could be found across different participating GPs. The questions are as follows:

1. *“Which ICPC code do you think is most appropriate (a code not listed here is also allowed)?”* This question measures the model’s performance and the variability in ICPC codes. This question will also measure the **“Confidence”** quality property.
2. *“Do you think the suggested diagnoses fit the S-rule?”* This question measures the correctness of the returned ICPC codes of the model.
3. *“Do you think the related symptoms fit the suggested diagnoses?”* and *“Do you think the related keywords fit the suggested symptoms?”*. These questions measure the **“Coherence”** quality property of the model and identify which form of explanation GPs found most insightful.
4. The last question of each case is *“Would you consider {ICPC code} an appropriate code?”*. This question measures if the GPs agree with the ground truth of the S-rule. However, it is worth noting that a GP originally assigned the ground truth with more information, such as medical examinations, at their disposal. The participating GPs do not have access to the same information.

Vragen casus 1:

1. Welke ICPC code vind u het meest passend (een code die hier niet bijstaat mag ook)?
   Click or tap here to enter text.
2. Vindt u de **voorgestelde diagnoses** passen bij de S-regel?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

3. Vindt u de **gerelateerde** symptomen passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

4. Vindt u de **gerelateerde** trefwoorden passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

5. Zou u “L17 -Voet/teen symptomen/klachten” een adequate code vinden?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

Figure 9.2: Questions for each case in the user study

After the ten cases, there are six more general questions (see Figure 9.3). These measure the opinion of the GP on the explanation and suggested diagnoses of the model in general. These general questions were open to give more context to the answers to the previous questions.

1. *"Would the suggested diagnoses in their current form help you to find a correct diagnosis more quickly (why/why not)?"*. This question measures if GPs would use the model and find that the model improves their diagnostic ability.
2. *"Could related symptoms in suggested diagnoses help you to find a correct diagnosis faster (why/why not)?"* and *"Could related keywords for suggested symptoms help you find a correct diagnosis more quickly (why/why not)?"*. These questions measure the **"Context"** quality property and how helpful the explanations are to their diagnostics ability.
3. *"What do you think of the number of suggested symptoms and words in an explanation?"* This question measures the quality property of the **"Compactness"** and whether GPs like the number of explanations given for each suggested diagnosis.
4. *"How do you feel about the explanation being shown in the form of keywords and relevant symptoms?"* This question shows if a GP would instead find keywords an insightful explanation, prefer diagnosis codes to be explained by the symptom module, or not find the form of explanation helpful at all. The question measures the **"Composition"** quality property.
5. *"What kind of control would you like over the suggested diagnoses and explanation (e.g. adding and removing your symptoms or adjusting the S rule)?"* This question measures the **"Controllability"** quality property and measures how much control a GP would like over the diagnoses prediction model or explanations. In essence, a modified S-rule can quickly be re-entered in the model to get new predictions and explanations, but this answers the question of what control GPs *would* want over an explanation.

A complete example of a user study can be found in Appendix C: "Example of a full user study".

Algemene vragen

1. Zouden de voorgestelde diagnoses in huidige vorm u kunnen helpen sneller een juiste diagnose te vinden (waarom wel/niet)?

Click or tap here to enter text.

2. Zouden gerelateerde symptomen bij diagnoses in huidige vorm u kunnen helpen sneller een juiste diagnose te vinden (waarom wel/niet)?

Click or tap here to enter text.

3. Zouden gerelateerde trefwoorden bij symptomen in huidige vorm u kunnen helpen sneller een juiste diagnose te vinden (waarom wel/niet)?

Click or tap here to enter text.

4. Wat vindt u van de hoeveelheid voorgestelde diagnoses, symptomen en trefwoorden bij een uitleg?

Click or tap here to enter text.

5. Wat vindt u ervan dat de uitleg word getoond in de vorm van trefwoorden en relevante symptomen?

Click or tap here to enter text.

6. Wat voor controle zou u willen over de voorgestelde diagnoses (denk bijvoorbeeld aan zelf symptomen toevoegen en verwijderen of de S-regel aanpassen)?

Click or tap here to enter text.

Figure 9.3: General questions for each case in the user study

9.2.2 Visualising explanations

The output of the model and explanations are returned as Python dictionaries. However, for end-users such as GPs, interpreting these dictionaries as plain text can take time and effort to understand. To enhance the user experience during the user study, a visual layer was introduced to transform the Python dictionaries into a more easily viewable format. The visual layer aimed to create a clear separation between diagnoses and symptoms and their explanations.

For the keyword text-module explanations, the messaging “Related keywords” was utilised, while the messaging “Related symptoms” was employed for the symptom-module explanations.

For reference, the visual components of existing symptom checkers such as WebMD[29] and Ada[35] were used. Specifically, the symptom checkers do not use numbers or probabilities but indicate the certainty of a diagnosis using a scale such as “Fair match” or “Moderate evidence”.

For this visual layer, the probability of each output neuron was converted into an “evidence” scale. Probabilities higher than 50% were classified as “Strong evidence,” probabilities lower than 50% were categorised as “Decent evidence,” and probabilities lower than 10% were labelled as “Low evidence”. Furthermore, a green bar was added to the output of the ”related symptoms” that indicated the score given by the LIME symptom-module. These two visual additions are aimed at assessing whether they would influence the decision-making process of participating GPs. However, despite including these features during the user study, it was observed that GPs did not utilise them in their diagnostic process. Consequently, these features could have been omitted from the user study. Two examples of a visualised explanation can be seen in Figure 9.4.

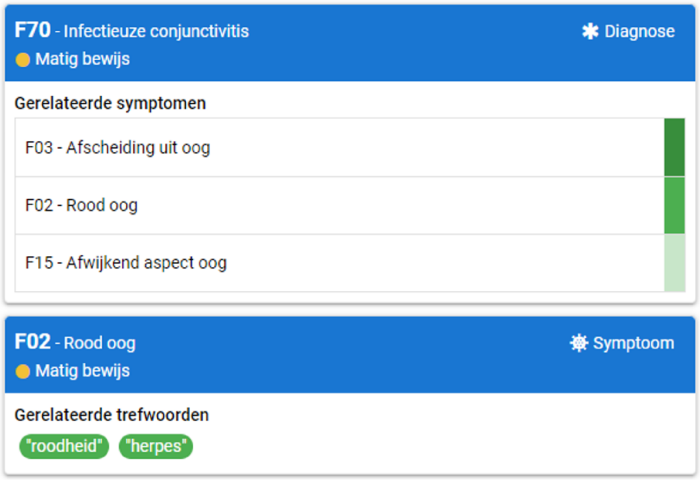


Figure 9.4: Two examples of explaining an ICPC symptom code and ICPC diagnosis/disease code.

9.2.3 Results case questions

Five Dutch general practitioners were involved in the study, collectively responding to over 50 cases. The first 5 of the 10 cases (25/50) were presented to each GP. Table 9.3 displays the variability and answers for each GP. The practitioners selected an ICPC code that matched the output of the threshold function in 44 out of 50 cases. However, in 26 out of 50 cases, a different ICPC code was chosen compared to the ground truth. Among the first 5 cases assigned to each GP, there were 5 distinct ground truths. Surprisingly, the GPs collectively chose 13 different ICPC codes, highlighting the variability issue. The variability in ICPC codes shows that multiple ICPC codes are possible for each S-rule and can vary significantly among practitioners. Furthermore, it emphasises the importance of having the model return more than one ICPC code in the threshold function.

The threshold function initially achieved a performance of 80.1% in section 8.6: "Performance". However, when considering the "new" ground truth derived from the user study, the model's threshold function provided the chosen ICPC code in 84.0% of the cases. When excluding cases where the ground truth ICPC code was in the output of the threshold function but the GP chose an ICPC code not provided by the model, the model's performance improves to ∼98.33%. On one occasion, a code was chosen which was not in the output of the threshold function, and the ground truth was also not in the output of the threshold function.

	Case 1	**Case 2**	**Case 3**	**Case 4**	**Case 5**
Ground truth	L17	F72	S09	X15	L08
Option # in model	1	4	1	Not in model	1
Participant 1	L17 (#1)	F02 (#2)	S09 (#1)	S75 (#1)	L08 (#1)
Participant 2	L17 (#1)	F05 (#X)	S09 (#1)	S06 (#X)	L08 (#1)
Participant 3	L17 (#1)	F02 (#2)	S09 (#1)	S74 (#2)	L08 (#1)
Participant 4	L77 (#X)	F70 (#1)	S09 (#1)	X72 (#4)	L08 (#1)
Participant 5	A80 (#4)	F73 (#5)	S09 (#1)	S74 (#2)	L08 (#1)

Table 9.3: The variance in the first 5 cases across each participant.

The average rating of the scales per participant is presented in Table 9.4. For question 1, the average position of the chosen ICPC code was calculated. If the chosen ICPC code was absent in the model, it was excluded from the calculation. For instance, participant 2 selected the ICPC code with the highest probability in 8 out of 10 cases. However, the participant chose an ICPC code absent in the model in 2 out of 10 cases. As a result, the average chosen option of the participant is 1.00.

The quality of the returned ICPC codes by the threshold function indicates an average rating of 3.62 out of 5.00, as reported by the practitioners. The rating suggests that although the GPs prioritise the ICPC code with the highest probability, they acknowledge the potential relevance of the other presented ICPC codes. It is worth noting that the GPs rated the ground truth (question 5) with an average score of 3.34 out of 5.00. The rating is lower than the rating assigned to the returned ICPC codes by the threshold function. During the user study, all practitioners typically did not find the lower probability ICPC codes probable. These results can be used to optimise and improve the threshold function. The same is true for the "related symptoms" and "related keywords" which showed the same problem. The "related symptoms" had an average rating of 3.5/5, and the "related keyword" had an average rating of 3.32/5. As previously explained, there are always two keywords minimum and three symptoms. During the user study, it was again found that while the keywords and symptoms often contribute to the practitioners' decision-making ability, the lower-scoring keywords and symptoms were often incorrect. The GPs found that incorrect explanations only hindered their diagnostic ability. However, these parameters can still be optimised to increase the models' performance. Rerunning the same user study with a correct amount of keywords, symptoms, and diagnoses would lead to a higher rating.

Average scale	Question 1	Question 2	Question 3	Question 4	Question 5
Participant 1	1.3 (9/10)	3.4	3.4	3.25	3.2
Participant 2	1.0 (8/10)	4.0	3.5	∼3.56	3.3
Participant 3	1.6 (10/10)	3.4	3.2	∼2.56	3.6
Participant 4	∼1.87 (8/10)	3.1	3.4	3.1	3.2
Participant 5	∼2.22 (9/10)	4.2	4.0	∼4.11	3.4
Total average	**∼1.6**	**3.62**	**3.5**	**∼3.32**	**3.34**

Table 9.4: Average rating across ten cases of each participant

9.2.4 Results general questions

The analysis and quantification of the answers to the general questions pose a more significant challenge because they are open-ended. To establish a quantification method, the answers are simplified to a variable of either Yes/No, as presented in Table 9.5. The complete answers to each general question for each participant can be found in Appendix B: "Answers to the general questions"

Describing a consensus for each question is complicated. However, it is worth noting that practitioners generally expressed a positive view regarding the models' diagnostic ability. Of five participants, four indicated that the model's suggested ICPC codes assisted them in their diagnostic ability. The model enabled GPs to identify the correct ICPC code more efficiently.

Three practitioners mentioned that the "related symptoms" provided by the symptom-module enhanced their diagnostic abilities. Frequently, practitioners employed these related symptoms to pinpoint a more specific ICPC code instead of considering them as explanations generated by the model. Two practitioners found the "related keywords" to be insightful. All practitioners observed that the keywords did not contribute significantly to their decision-making process. However, they acknowledged that the keywords effectively reflected the model's decision-making and increased trust in the model. Many times, however, it was noted by practitioners that a lot of weird keywords appeared, such as stop-words. The issue can be addressed by further optimising the system by analysing these wrong keywords' scores.

Practitioners found it challenging to understand that ICPC codes may not appear immediately correct after reading the S-rule. In future iterations, explaining that the model outputs probable diagnoses, including those that often emerge after additional examinations, is crucial. The clarification should extend to explanations as well. In this case, a "related symptom" may be included but is not present in the patient's symptoms and complaints outlined in the S-rule.

Question	1	2	3	4	5	6
Participant 1	Yes (With some improvement)	Yes	No	Good as it is	Yes	Change S-rule
Participant 2	Yes	No	No	3 till 5	Yes	Change S-rule
Participant 3	Yes	Yes	No	2 till 3	Yes	No necessity
Participant 4	No (Yes for triagist)	No	Yes	3	Yes	No necessity
Participant 5	Yes	Yes	Yes	Max 5	Keywords = Yes Symptoms = Maybe	Change S-rule

Table 9.5: Summarised answers to general questions

9.3 Conclusion

In this section, a LIME-based symptom module was proposed that aimed to enhance the explainability of the diagnoses model by explaining the predicted ICPC codes by the most influential symptoms. To measure the effectiveness of the proposed module, a user study was conducted involving five Dutch GPs. The primary objective of the user study was to evaluate the model's actual performance and determine its suitability for use by GPs. Additionally, the study aimed to assess how well the model's explanation performed through the evaluation of six co-explanation properties.

It is important to note that the assessment of co-explanation properties remains subjective, as different GPs may have different preferences regarding the methods of explanation. This subjectivity was observed during the user study, where varying or contradicting opinions were expressed by participating GPs.

The results of the user study indicated that the model has a positive impact on the decision making of the participating GPs and the model had a higher accuracy compared to the validation set. The suspected ICPC variability per GP was proven and it was shown that the predicted ICPC codes by the threshold function encapsulated the variability present per GP well. When the model would be actually implemented in the process of the GP, a more accurate performance can be measured because the user study only existed of five GPs.

To conclude this section, the user study results will be used to review the symptom and text modules, considering the six co-explanation properties.

1. **Compactness**: Participants expressed that an optimal number of 3-5 suggested ICPC codes would be desirable. Moreover, they found the quantity of explanations to be sufficient.
2. **Composition**: Participants acknowledged the adequacy of the composition, but they found the inclusion of both symptom, diagnosis codes, relevant symptoms and relevant keywords confusing. Moreover they found the relevant symptoms to take up too much space.
3. **Confidence**: The participants did not consult the scores associated with the explanations or the overall prediction. While the purpose of these scores was explained at the beginning of the user study, it appears that the scores did not attract enough attention.
4. **Context**: Participants indicated that the symptom-module and text-module demonstrated the decision-making process of the model. However, they did not find it helpful for their diagnostic process but rather useful in identifying a less specific ICPC code.
5. **Coherence**: Given that the participants were medical professionals, they comprehended the explanations effectively and were able to accurately identify incorrect explanations. It was observed that GPs utilise ICPC codes differently per GP, which could impact their perception of the model's performance.
6. **Controllability**: Participants appreciated the availability of a controllable S-rule to modify the model's predictions and explanations. However, they expressed that further control features were unnecessary as that would only increase the time required to arrive at a diagnosis.

Chapter 10

Discussion

10.1 User study

The user study yielded promising results, indicating that 80% of the participating GPs observed an enhancement in their diagnostic capability when utilising the suggested diagnoses. Nonetheless, certain limitations were encountered during the implementation of the user study. One particular limitation relates to the study structure, which may have introduced a potential bias.

In each case, the initial question posed to the participants was, "Which ICPC code do you think is most appropriate (a code not listed here is also allowed)?" This question was asked subsequently to the participants reviewing the S-rule and being presented with the diagnoses suggested by the model. This structural arrangement can potentially create an acquiescence bias by already providing ICPC codes to the GPs before posing the question. Acquiescence bias refers to the tendency of participants to agree or accept suggestions presented to them as defined by Kuru et al. (2016)[61]. The authors investigate how a Likert scale which was also utilised in this study, introduces the bias in social media measurement surveys.

In retrospect, a more representative assessment of the model's performance could have been achieved by asking the initial question *after* the participants had reviewed the S-rule but *before* being presented with the suggested diagnoses. This modification would have eliminated the potential acquiescence bias by solely evaluating the decision-making abilities of the GPs rather than considering both the model's suggestions and the GPs' responses. These possible biases were chosen to ignore in this study because the user study was already nearing completion.

Another possible bias is the "Response Order Effect" bias[62], where participants are more likely to choose choices presented earlier. The diagnoses prediction model presents more probable diagnoses higher than less probable diagnoses. To remove the potential "Response Order Effect bias", the suggested diagnoses could be shuffled among the first five cases.

The user study was completed by five participants. To ensure a robust validation of the model's performance, a larger number of participants would be advantageous. If the diagnoses prediction model were to be implemented by *Topcicus* to improve the decision-making of GPs, a feedback loop could be considered. By monitoring the use of the model in actual scenarios, an accurate assessment of the model's performance could be obtained. The model could be implemented into the GP side of *Spreekuur.nl*, which is used by 60 GP practices and emergency centres, presumably including hundreds of GPs. Consequently, the feedback loop can provide more validation cases than the 50 user study cases. Within the feedback loop, it would be feasible to track the frequency with which the model is utilised by GPs. Additionally, the feedback loop could keep track of how many times an option is chosen by the GP that was suggested by the model. This feedback mechanism provides a robust validation approach which could be used for training the model if enough data is gathered.

10.2 S-rules

GPs write s-rules to document symptoms and patient narratives. The s-rule generated from questionnaire answers demonstrated slightly inferior performance compared to the s-rule written by GPs.

A notable limitation is that an ICPC code is typically assigned after documenting a comprehensive SOAP (Subjective, Objective, Assessment, Plan) note, which includes additional examinations not present in the S-rule. The model's prediction does not incorporate this supplementary information, resulting in suboptimal performance as two identical s-rules can yield different diagnoses based on further examination. User study participants also indicated instances where the model predicted a disease or diagnosis not suggested by the available information. The model can predict the most probable codes, usually only assigned after further examination. These codes can assist GPs by proposing ICPC codes that typically arise after further examination.

Furthermore, the model performed well when applied to written S-rules and was not further fine-tuned using the generated S-rules. This performance suggests an additional use case for the diagnosis prediction model. Since the model's performance was higher when applied to written S-rules than generated ones, it suggests GPs could utilise the model when formulating an S-rule. However, further research is necessary to determine the model's actual performance when predicting ICPC codes for GPs during the s-rule writing process. It is crucial to acknowledge that the dataset comprised three GP emergency centres, each with a limited number of GPs and triage professionals responsible for writing S-rules making the model less generalisable for normal general practices. S-rules from GPs in different centres or practices may reduce the prediction model's performance.

10.3 Linking dataset

This study attempted to link the resulting ICPC code following an anonymous questionnaire. Previously, SOAP notes and questionnaires could be linked via an ID field, which has since been removed. Approximately 17 thousand questionnaires were connected using personal information such as age, sex, and time stamps. These 17 thousand questionnaires exhibited a 3% decrease in performance across all metrics compared to the 4 thousand questionnaires linked via the ID field. This performance decline suggests that the connected ICPC code was incorrect in at least 3% of the questionnaires (potentially even more due to the possibility of fasely linked ICPC codes coincidentally being correct). Out of approximately 50 thousand questionnaires, only 17 thousand could be successfully linked to an ICPC code. Training the model on the generated S-rules did not improve performance, indicating that the dataset needed more data entries. Expanding the dataset size could enhance the model's performance.

Topicus now faces a decision between privacy and utility. They can reintroduce the ability to connect questionnaires to SOAP notes via an ID, enabling a reliable linkage and a larger dataset. The removal of the ID field did not yield the desired results for Topicus, as the information in the SOAP note and questionnaire datasets still allows for reasonably accurate data linkage. Alternatively, if *Topicus* retains the current structure, they must eliminate additional personal information. Alternatively, *Topicus* can add the connection between SOAP notes and notes and questionnaires within a randomly selected subset of new questionnaires. This linkage allows for the continued utilisation of the connected questionnaires for various utility purposes, including validation, while gradually expanding the dataset over time. Notably, the remaining data remains highly protected through this method. However, a drawback of this approach emerges over an extended duration, wherein a substantial majority of patients would likely be present within the connected dataset. This scenario introduces the potential for establishing associations between the unconnected questionnaires and SOAP notes.

10.4 Loss functions

For training the diagnoses prediction model, the Cross-Entropy loss is employed. It is worth noting that various loss functions were explored, but none yielded improved performance. Specifically, the investigation focused on transforming the single-label output into a multi-label one. The dataset's ground truth comprised only one ICPC code per consultation; however, the user study revealed that multiple ICPC codes could be associated with a questionnaire or consultation performed by a GP.

Initially, a Sigmoid activation function was applied with a Binary Cross Entropy (BCE) loss function. This choice was motivated by the substantial variability in the training data, which suggested that multiple neurons could be activated with enough variability. To address the challenge of training a multi-label model on a single-labelled dataset, Cole et al.[63] investigated two strategies, which were also adopted in this study. The first strategy, "assume negative," assumes that the unknown labels (whether present in the ground truth or not) are negative. This approach considers all labels, except the ground truth, as negative. The second strategy, "ignore unobserved negatives," involves excluding the unknown labels from the loss function calculation using BCE.

Applying the "assume negative" strategy resulted in the output neurons exhibiting very small values (approximately 0.00) when using the Sigmoid activation function. The dataset's large amount of possible ICPC codes led the model to achieve a lower loss by not activating any neurons. In Cole et al. paper, the number of possible labels was lower and was weakly labelled instead of single labelled. Conversely, the "ignore unobserved negatives" strategy activated all neurons (approximately 1.00) for every prediction. Furthermore, due to the single-labelled ground truth, the multi-label model cannot be accurately validated.

Other loss functions were also employed to make use of the hierarchy in ICPC codes such as decreasing the loss when the predicted code was of the correct category/letter but this did not result in improved performance.

10.5 Adding tokens

Another potential improvement to the model involved expanding the tokenizer's vocabulary by adding additional tokens(see subsection 6.2.1: "Vocabulary transfer"). An algorithm was devised to introduce 4,000 new tokens into the vocabulary. To identify these tokens, the TF-IDF score was computed for complete words, and the highest-scoring words were added to the vocabulary as tokens until the vocabulary size reached 44,000. The RobBERT model was trained on all S-rules to update the representations using the extended vocabulary. It was observed that the inclusion of the 4,000 new tokens led to significant forgetting and resulted in poorer performance compared to the original model.

10.6 ICPC variability

The primary challenge encountered in this study was the variability between possible ICPC codes. In many cases, multiple ICPC codes could potentially serve as appropriate diagnoses. The absence of a hierarchical documentation structure further complicated the issue. The current approach employed in this study, utilising a side-by-side classification head with a threshold function, partially mitigates the problem by separately predicting symptoms and diagnoses. However, including a hierarchical structure would present a more beneficial solution. A loss function that considers the hierarchical position of the ICPC code could then be efficiently implemented. For instance, the loss could be reduced when the predicted ICPC code falls within the hierarchy of the ground truth code.

Creating a disease/symptom hierarchy poses significant challenges, as it often leads to an infinite structure. Most diseases manifest as various symptoms; conversely, a single symptom can be associated with multiple diseases. The variability in the ICPC codes severely impacts the model's ability to validate and learn effectively. The model may be penalised for assigning a code that another GP, as observed in the user study, would consider correct. Additionally, the validation process is adversely affected due to the dataset being solely single-labelled.

The main reason behind the single-labelled dataset is the GP documentation tools Dutch GP utilise. These software systems only permit a single ICPC code as input per consultation. GPs primarily rely on ICPC codes to quickly locate the corresponding SOAP note for a patient. When multiple ICPC codes apply to a consultation, the GP must duplicate the information into two separate consultations with two ICPC codes, which is often excessively time-consuming. Furthermore, GPs often combine consultations with two similar ICPC codes in one to improve searchability and the number of consultations present for a single patient.

Enabling multi-label inputs could serve as a scientifically robust and forward-looking solution. A multi-label dataset will improve data analysis capabilities and potentially enhance healthcare outcomes by identifying new symptoms and disease patterns or a sudden increase in ICPC code usage. Multi-labelled SOAP notes could also enhance the efficiency of GPs in searching for ICPC codes. Furthermore, utilising multi-label inputs would facilitate the development of a more accurate machine-learning model.

Chapter 11

Conclusion

Spreekuur.nl is an online consultation application that requires users to complete a questionnaire before participating in an online consultation. A GP can review the answers to the questionnaire to help them diagnose the patient. This study aimed to help alleviate the workload of GPs by predicting diagnoses based on the answers to a questionnaire and providing explanations in terms of symptoms and keyword-based evidence for these symptoms.

First, the questionnaire data was converted into the text format of an S-rule (user-identified symptoms), resembling an S-rule written by a Dutch GP. The S-rule specifically concerns the **S**ubjective symptoms and patient's narrative without a further medical examination. An S-rule is part of a SOAP note which is the documentation standard for Dutch GP to document a consultation.

ICPC codes are a standard GPs use in consultations to standardise and document a patient's diagnosis and are present in each SOAP note. ICPC codes were used to classify a diagnosis based on the S-rule.

a Dutch variant of the RoBERTa language model known as RobBERT was made domain-specific to the domain of Dutch GPs. The model is trained and fine-tuned using 2.2 million S-rules. A new classification head was introduced, enabling the separate classification of ICPC symptom codes and ICPC disease/diagnosis codes.

Fine-tuning the model on 17 thousand-generated S-rules did not improve performance, as the size of the questionnaire dataset was insufficient for the model to learn. Instead, the questionnaire data was employed to validate the model's performance.

To align with the diagnostic decision-making process of GPs, a threshold function was implemented to determine the number of ICPC codes returned. The threshold function outperformed the simple approach of returning only the top three codes while providing fewer than three codes on average. When predicting ICPC symptom codes, the threshold function achieved an accuracy of 90%; for ICPC diagnosis/disease codes, the accuracy was 88.6%. Notably, when evaluating the model's performance on the generated S-rules from the questionnaire dataset, the accuracy was 87.6% for ICPC symptom codes and 86.5% for ICPC diagnosis/disease codes.

However, it should be noted that the model's actual performance may be higher due to the variability of ICPC codes across different GPs, as observed in the user study.

Additionally, explaining the model's decision-making process is crucial for GPs. The user study revealed that the LIME symptom-module and text-module provided satisfactory explanations; however, participants found irregularities distracting. Future research can focus on further enhancing the explanation mechanisms for improved usability.

All sub-research questions will first be concluded.

1. **What is the most effective method for transforming SOAP notes and questionnaire data to train a RoBERTa diagnoses prediction model?**
   The S-rule of the SOAP note dataset contains the symptoms and the patient's narrative as described by the patient in a text format. *DigiDok*, has codebooks that are already available to convert questionnaire data into the S-rule's text format. The advantage of transforming questionnaires into S-rule lies in their resemblance to an S-rule written by a Dutch GP. Consequently,

fine-tuning the model on the generated S-rule is reduced, as the model can benefit from the similarities to GP-written S-rules.

2. **How can data from the SOAP note dataset link an ICPC code to the anonymous questionnaire data?**
   Before training, it is necessary to determine the eventual diagnosis/ICPC code assigned by a GP based on the questionnaire and consultation. Fortunately, the SOAP note dataset includes the SOAP note from the respective questionnaire. Previously, an ICPC code/SOAP note could be linked to a questionnaire through a designated ID field, but the field has since been removed. Among the available questionnaires, 5,356 were automatically connected using the now-removed ID field, while an additional 15,445 questionnaires were manually connected using timestamps and personal information. The manually connected dataset was validated using the 5,356 automatically connected questionnaires, resulting in a precision of 99% for the 2,840 questionnaires present in both datasets. However, it is important to note that only 53% of the automatically connected dataset was successfully linked using the manual connecting algorithm. Consequently, the total dataset consisted of 17,833 questionnaires after merging the automatically and manually connected datasets.

   Both datasets were used for validating the model. The total dataset had a 3% lower performance than the automatically connected dataset. This performance difference can be used to indicate how well the questionnaires were connected in the manually connected questionnaires in the dataset.

3. **What strategies can improve the model's performance to predict ICPC codes?**
   The primary challenge encountered in this study was the variability between possible ICPC codes. In many cases, multiple ICPC codes could potentially serve as appropriate diagnoses. Moreover, different GPs had varying levels of efficiency in the user study in their use of ICPC codes. The only available hierarchical information was whether an ICPC code represented a symptom or a diagnosis/disease.

   To address the variability, a new classification head was proposed. The classification head was divided into 318 neurons representing all ICPC symptom codes and 360 neurons representing all ICPC diagnosis/disease codes. The training was conducted using separate cross-entropy loss functions and activation functions. The division slightly improved accuracy while enabling the model to more effectively predict less frequently occurring ICPC codes.

   The diagnosis accuracy of the model increased from 63.4% to 64.9%, and the symptom accuracy improved from 65.9% to 66.9%. The macro accuracy for symptoms increased from 27.3% to 30.5%, and for diagnosis, it rose from 29.4% to 36.3%.

   To better emulate the diagnosis process of a GP, a new threshold function is implemented, which returns a variable number of ICPC codes based on their probabilities. The threshold function selects the ICPC codes with the highest probabilities until a threshold value is met. The approach increased the symptom and diagnosis accuracy to 90% and 88.6%, respectively, while on average, returning fewer than three codes (2.7 and 2.9).

4. **How can the performance of the model be validated for diagnosing patients using ICPC codes?**
   Providing a single diagnosis only partially captures the diagnostic process employed by Dutch GPs. Multiple ICPC codes often apply to the same S-rule, indicating variability in the assigned codes. Dutch GPs often use multiple diagnoses as a differential diagnosis or use less specific ICPC codes when uncertain, such as ”Cough” or ”Fever.” Consequently, the model achieved

an accuracy of only 51.4%. However, when considering the top-3 predicted ICPC codes, the model's accuracy increases to 78.4%.

Using the normal accuracy function as a performance measure is also inadequate in this context because the proposed classification head of the model predicts symptoms and diagnoses/disease codes separately. Therefore, separate accuracy measures are employed for ICPC symptom and ICPC diagnosis/disease codes. The symptom accuracy is determined by the activation of the first 318 neurons, while the last 360 neurons determine the diagnosis accuracy.

A user study was conducted to measure the actual performance of the model. The five participants chose an ICPC code suggested by the threshold function 84.00% of the time in 50 cases. There often needed to be more information in the generated S-rule to indicate the more detailed ground-truth ICPC code. The participants were always shown the same first five cases. The five participants chose 13 different ICPC codes across these five cases. This suggests that the real performance may be higher when considering the variability of ICPC codes.

5. **To what extent can current knowledge of diagnoses, symptoms and causes in the medical field be used for predictions?**
   The current medical knowledge that was available in this study consisted of 2.2 million SOAP notes. As these SOAP notes are from actual documentation from Dutch GPs, they contain domain-specific knowledge from the medical field, specifically the Dutch healthcare field. The S-rules in the SOAP note dataset were used to make the RobBERT model domain-specific by updating its representation using Masked Language Modelling. The MLM accuracy rose from 47.3% to 76.7%. Afterwards, the classification head is trained on the SOAP note dataset. Further fine-tuning the model on the generated S-rules from the questionnaire dataset was unnecessary as it did not increase the model's performance. The generated S-rule dataset was small compared to the S-rules in the SOAP note dataset and did not contain enough new information for the model to learn. Hence, the questionnaire dataset was used for validation. The threshold function got a symptom accuracy of 87.6% (2.82 picked on average) and 86.5% (3.1 picked on average) on the automatically connected questionnaire dataset.

6. **What is the relationship between the model's performance and the inclusion of specific questionnaire questions and answers as input features for the model?**
   The models did not significantly perform worse when removing specific questions/answers and parts of the generated S-rule. The removed parts also did not increase the model's performance. A large section of the S-rule often consists of previous operations or medicine. The section often is unrelated to the diagnosis of the patients. Other open questions did contain informative information. Patients described the course of their complaint, which is often much more insightful than what a multiple-choice question would produce in the generated s-rule. The performance when removing open questions proves the necessity of using a language model for this study.

7. **What is the performance of the diagnoses prediction model against established baseline models?**
   The RoBERTa language model may be overly complex for the task at hand, and a more straightforward model may achieve similar results while being computationally less expensive. Hence a naive classifier and random forest classifier were trained to compare the performance against the complex RoBERTa language model. Both baseline models showed a drastically decreased accuracy (24.4% and 9.5%) and macro average accuracy (2.8% and 0.4%) and were not considered to improve further.

8. **Which XAI method is most effective at explaining the predictions of a diagnoses prediction model to Dutch general practitioners?**
   RoBERTa lacks transparency and interpretability due to its complexity. To address this issue, the LIME symptom-module was proposed. The symptom-module is an adaptation of the LIME text-module. The text-module generates various samples of a text by removing words and tokens. The model then generates output probabilities for these samples, and the probability changes are utilised as training data for a white-box model. The white-box model aims to identify the most important keywords for each ICPC code.

   The symptom-module leverages the same samples but focuses on the changes in output probabilities of the first 318 symptom code neurons. By analysing these changes, the module identifies the most influential ICPC symptom codes corresponding to an ICPC diagnosis/disease code. A user study was conducted to evaluate the effectiveness of these two types of explanations. The survey included questions that measured six out of the twelve CO explanation quality properties specified by Nauta et al. [3].

   The user study results indicated that three out of five participants found a diagnosis's "related symptoms" helpful. In comparison, two out of five participants found the "keywords" associated with a code beneficial. The explanations were found to gain insights into the model's decision-making process. However, the participants found that the explanations increased their diagnostic abilities.

   Participants also noted that less relevant keywords and symptoms often hindered their diagnostic abilities by causing distractions.

9. **How does the use of the diagnoses prediction models impact the efficiency of Dutch GPs?**
   A user study was conducted to measure the performance of the diagnoses prediction model and evaluate its explainability. The study involved five participants, and their experiences were measured to determine the impact of the model on their diagnostic abilities.

   The user study results showed that four out of five participants reported an enhancement in their diagnostic abilities due to the incorporation of the diagnoses prediction model. Specifically, the participants were able to provide more specific ICPC codes.

In conclusion, this study demonstrates the promising performance of the model in predicting S-rules generated from questionnaire answers and those written by GPs. Explaining the suggested ICPC codes is crucial for establishing trust in the model's decision-making process. While the LIME symptom-module and text-module demonstrated potential in offering explanations, further optimisation is required to remove irrelevant symptoms and keywords. By incorporating further improvements and fine-tuning in areas such as the threshold function, suggested explanations, and the visual layer, the model can effectively suggest additional ICPC codes to practitioners. GPs found that suggesting ICPC codes enhanced their diagnostic capabilities in online consultations. It is important to note that the dataset utilised in this study consists of data from only three GP emergency practice centres. Increasing the number of participating general practices would provide a more comprehensive dataset, leading to higher macro average accuracy as many ICPC codes have a low occurrence within the current dataset. Furthermore, a multi-label dataset or hierarchical implementation of ICPC code could drastically improve the model's performance in future works.

Bibliography

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," Dec 2017. arXiv:1706.03762 [cs].

[2] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," Aug 2016. arXiv:1602.04938 [cs, stat].

[3] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. van Keulen, and C. Seifert, "From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai," May 2022. arXiv:2201.08164 [cs].

[4] H. Alderwick, "Is the nhs overwhelmed?," *BMJ*, vol. 376, p. o51, Jan 2022.

[5] Philips, "Healthcare hits reset - 2022 global report." https://www.philips.com/a-w/about/news/future-health-index/reports/2022/healthcare-hits-reset.html.

[6] R. Batenburg, M. Bosmans, S. Versteeg, E. Vis, B. van Asten, and L. Vandermeulen, "Balans in vraag en aanbod huisartsenzorg,"

[7] B. Davis, B. K. Bankhead-Kendall, and R. P. Dumas, "A review of covid-19's impact on modern medical systems from a health organization management perspective," *Health and Technology*, vol. 12, no. 4, p. 815–824, 2022.

[8] N. H. Genootschap, "Nhg-richtlijnen." https://richtlijnen.nhg.org/.

[9] NTS, "Nederlandse triage standaard - homepage." https://de-nts.nl/home/.

[10] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean, "A guide to deep learning in healthcare," *Nature Medicine*, vol. 25, p. 24–29, Jan 2019.

[11] K. W. Johnson, J. Torres Soto, B. S. Glicksberg, K. Shameer, R. Miotto, M. Ali, E. Ashley, and J. T. Dudley, "Artificial intelligence in cardiology," *Journal of the American College of Cardiology*, vol. 71, p. 2668–2679, Jun 2018.

[12] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang, "Artificial intelligence in healthcare: past, present and future," *Stroke and Vascular Neurology*, vol. 2, p. 230–243, Dec 2017.

[13] J. Betancur, F. Commandeur, M. Motlagh, T. Sharir, A. J. Einstein, S. Bokhari, M. B. Fish, T. D. Ruddy, P. Kaufmann, A. J. Sinusas, E. J. Miller, T. M. Bateman, S. Dorbala, M. Di Carli, G. Germano, Y. Otaki, B. K. Tamarappoo, D. Dey, D. S. Berman, and P. J. Slomka, "Deep learning for prediction of obstructive disease from fast myocardial perfusion spect: A multicenter study," *JACC. Cardiovascular imaging*, vol. 11, p. 1654–1663, Nov 2018.

[14] E. Begoli, T. Bhattacharya, and D. F. Kusnezov, "The need for uncertainty quantification in machine-assisted medical decision making," *Nature Machine Intelligence*, vol. 1, Jan 2019. Institution: Oak Ridge National Lab. (ORNL), Oak Ridge, TN (United States); Los Alamos National Lab. (LANL), Los Alamos, NM (United States).

[15] K. Rasheed, A. Qayyum, M. Ghaly, A. Al-Fuqaha, A. Razi, and J. Qadir, "Explainable, trustworthy, and ethical machine learning for healthcare: A survey," *Computers in Biology and Medicine*, vol. 149, p. 106043, Oct 2022.

[16] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, p. 665–673, Nov 2020. arXiv:2004.07780 [cs, q-bio].

[17] P. Delobelle, T. Winters, and B. Berendt, "Robbert: a dutch roberta-based language model," Sep 2020. arXiv:2001.06286 [cs].

[18] M. Kroneman, W. Boerma, M. van den Berg, P. Groenewegen, J. de Jong, and E. van Ginneken, "Netherlands: Health system review," *Health Systems in Transition*, vol. 18, p. 1–240, Mar 2016.

[19] L. van Sadelhoff, "Huisartsen lopen over, dus ze protesteren: "de drukte is extreem"." https://www.rtlnieuws.nl/nieuws/nederland/artikel/5317622/huisartsen-protest-hoge-werkdruk-zorg, Jun 2022.

[20] O. Beukers, "Huisartsendemonstratie in den haag: "wij zijn een spoedgeval"." https://nos.nl/artikel/2434873-huisartsendemonstratie-in-den-haag-wij-zijn-een-spoedgeval, Jul 2022.

[21] P. Mout, C. i. eld, and W. Fraanje, "Het abcde van de acute huisartsgeneeskunde," *Huisarts en wetenschap*, vol. 54, p. 210–214, Apr 2011.

[22] G. FitzGerald, G. A. Jelinek, D. Scott, and M. F. Gerdtz, "Emergency department triage revisited," *Emergency medicine journal: EMJ*, vol. 27, p. 86–92, Feb 2010.

[23] C. Stolper, A. Rutten, and G. Dinant, "Hoe verloopt het diagnostisch denken van de ervaren huisarts?," *Huisarts en Wetenschap*, vol. 48, p. 993–997, Jan 2005.

[24] V. C. L. Tielens, "Het medisch-diagnostisch handelen van de huisarts," *Huisarts en Wetenschap*, p. 5.

[25] N. H. Genootschap, "Thuisarts.nl — betrouwbare informatie over ziekte en gezondheid." https://thuisarts.nl/.

[26] V. Podder, V. Lew, and S. Ghassemzadeh, *SOAP Notes*. Treasure Island (FL): StatPearls Publishing, 2022.

[27] WHO, "International classification of primary care, 2nd edition (icpc-2)."

[28] isabel healthcare, "Symptom checker — isabel - the symptom checker doctors use." https://symptomchecker.isabelhealthcare.com.

[29] "Symptom checker — webmd." https://symptoms.webmd.com/.

[30] W. Wallace, C. Chan, S. Chidambaram, L. Hanna, F. M. Iqbal, A. Acharya, P. Normahani, H. Ashrafian, S. R. Markar, V. Sounderajah, and A. Darzi, "The diagnostic and triage accuracy of digital and online symptom checker tools: a systematic review," *NPJ digital medicine*, vol. 5, p. 118, Aug 2022.

[31] M. G. J. S. Hageman, J. Anderson, R. Blok, J. K. J. Bossen, and D. Ring, "Internet self-diagnosis in hand surgery," *Hand (New York, N.Y.)*, vol. 10, p. 565–569, Sep 2015.

[32] A. Ceney, S. Tolond, A. Glowinski, B. Marks, S. Swift, and T. Palser, "Accuracy of online symptom checkers and the potential impact on service utilisation," *PLOS ONE*, vol. 16, p. e0254088, Jul 2021.

[33] H. L. Semigran, J. A. Linder, C. Gidengil, and A. Mehrotra, "Evaluation of symptom checkers for self diagnosis and triage: audit study," *BMJ*, vol. 351, p. h3480, Jul 2015.

[34] C. Shen, M. Nguyen, A. Gregor, G. Isaza, and A. Beattie, "Accuracy of a popular online symptom checker for ophthalmic diagnoses," *JAMA ophthalmology*, vol. 137, p. 690–692, Jun 2019.

[35] "Health. powered by ada." https://ada.com/.

[36] A. Painter, B. Hayhoe, E. Riboli-Sasco, and A. El-Osta, "Online symptom checkers: Recommendations for a vignette-based clinical evaluation standard," *Journal of Medical Internet Research*, vol. 24, p. e37408, Oct 2022.

[37] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, p. 1735–80, Dec 1997.

[38] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," Dec 2014. arXiv:1412.3555 [cs].

[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," May 2019. arXiv:1810.04805 [cs].

[40] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," Jul 2019. arXiv:1907.11692 [cs].

[41] X. Zhou, J. Menche, A.-L. Barabási, and A. Sharma, "Human symptoms–disease network," *Nature Communications*, vol. 5, p. 4212, Jun 2014.

[42] J. G. Richens, C. M. Lee, and S. Johri, "Improving the accuracy of medical diagnosis with causal machine learning," *Nature Communications*, vol. 11, p. 3923, Aug 2020.

[43] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, p. btz682, Sep 2019. arXiv:1901.08746 [cs].

[44] B. van Aken, J.-M. Papaioannou, M. Mayrdorfer, K. Budde, F. A. Gers, and A. Löser, "Clinical outcome prediction from admission notes using self-supervised knowledge integration," Feb 2021. arXiv:2102.04110 [cs].

[45] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," Dec 2019. arXiv:1910.10045 [cs].

[46] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen, "A survey of the state of explainable ai for natural language processing," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, (Suzhou, China), p. 447–459, Association for Computational Linguistics, Dec 2020.

[47] A. Ross, A. Marasović, and M. E. Peters, "Explaining nlp models via minimal contrastive editing (mice)," Jun 2021. arXiv:2012.13985 [cs].

[48] T. Wu, M. T. Ribeiro, J. Heer, and D. Weld, "Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Online), p. 6707–6723, Association for Computational Linguistics, Aug 2021.

[49] Nivel, “Cijfers huisartsenposten - gezondheidsproblemen.”

[50] H. en wetenschape, “Icpc-codering op de huisartsenpost — huisarts wetenschap.”

[51] P. J. Ortiz Suárez, B. Sagot, and L. Romary, “Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures,” Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, (Mannheim), pp. 9 – 16, Leibniz-Institut für Deutsche Sprache, 2019.

[52] S. Verkijk and P. Vossen, “Medroberta.nl: A language model for dutch electronic health records,” *Computational Linguistics in the Netherlands Journal*, vol. 11, p. 141–159, Dec 2021.

[53] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, “Legal-bert: The muppets straight out of law school,” Oct 2020. arXiv:2010.02559 [cs].

[54] V. D. Mosin and I. P. Yamshchikov, “Vocabulary transfer for medical texts,” Aug 2022. arXiv:2208.02554 [cs].

[55] V. Mosin, I. Samenko, A. Tikhonov, B. Kozlovskii, and I. P. Yamshchikov, “Fine-tuning transformers: Vocabulary transfer,” Dec 2022. arXiv:2112.14569 [cs].

[56] M. Szczepański, M. Pawlicki, R. Kozik, and M. Choraś, “New explainability method for bert-based model in fake news detection,” *Scientific Reports*, vol. 11, p. 23705, Dec 2021.

[57] M. Zwaanswijk and K. Hek, “Icpc-codering op de huisartsenpost,” *Huisarts en wetenschap*, vol. 56, p. 577–577, Nov 2013.

[58] D. Hendrycks and K. Gimpel, “Gaussian error linear units (gelus),” Jul 2020. arXiv:1606.08415 [cs].

[59] I. Rish, “An empirical study of the naive bayes classifier,”

[60] A. Parmar, R. Katariya, and V. Patel, “A review on random forest: An ensemble classifier,” in *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018* (J. Hemanth, X. Fernando, P. Lafata, and Z. Baig, eds.), Lecture Notes on Data Engineering and Communications Technologies, (Cham), p. 758–763, Springer International Publishing, 2019.

[61] O. Kuru and J. Pasek, “Improving social media measurement in surveys: Avoiding acquiescence bias in facebook research,” *Computers in Human Behavior*, vol. 57, p. 82–92, Apr 2016.

[62] S. L. BECKER, “Why an order effect,” *Public Opinion Quarterly*, vol. 18, p. 271–278, Jan 1954.

[63] E. Cole, O. M. Aodha, T. Lorieul, P. Perona, D. Morris, and N. Jojic, “Multi-label learning from single positive labels,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Nashville, TN, USA), p. 933–942, IEEE, Jun 2021.

[64] L. M. Haraldseide, L. Sortland, S. Hunskaar, and T. Morken, “Contact characteristics and factors associated with the degree of urgency among older people in emergency primary health care: a cross-sectional study,” *BMC Health Services Research*, vol. 20, Apr 2020.

Appendix A

Extra data analysis.

Figure A.1 shows that for most consultations, the patient is between 0 and 5 years old (values such as -979 and 220 are likely errors on the part of the GP). Meaning that most ICPC codes are given for this age group, which introduces further bias as symptoms can differ between toddlers, adults, and elderly patients who have a higher risk of heart disease and cancer. Interestingly, for patients over 70, the code “A96 - Death” was most present (see Figure A.4). “T90 - Diabetes” was also among the most commonly occurring codes, along with “A13 - Concern about/fear of medical treatment”, which corresponds with the findings of the study by Haraldseide et al.(2020)[64] investigated the most commonly occurring ICPC codes for people over 70 in a similar dataset.. For people younger than 12 (Figure A.5), ”A03 - Fever” occurred almost 12% of the time, and ”R74 - Upper respiratory infection acute” occurred almost 8% of the time, making them the most commonly occurring codes. Figure A.6 shows that females (”Vrouw” in Dutch) are slightly more represented in the data than males (”Man” in Dutch), with females making up 55% of the data and males making up 45%. As shown in Figure A.3) and Figure A.2, the most commonly occurring ICPC codes change significantly based on sex.

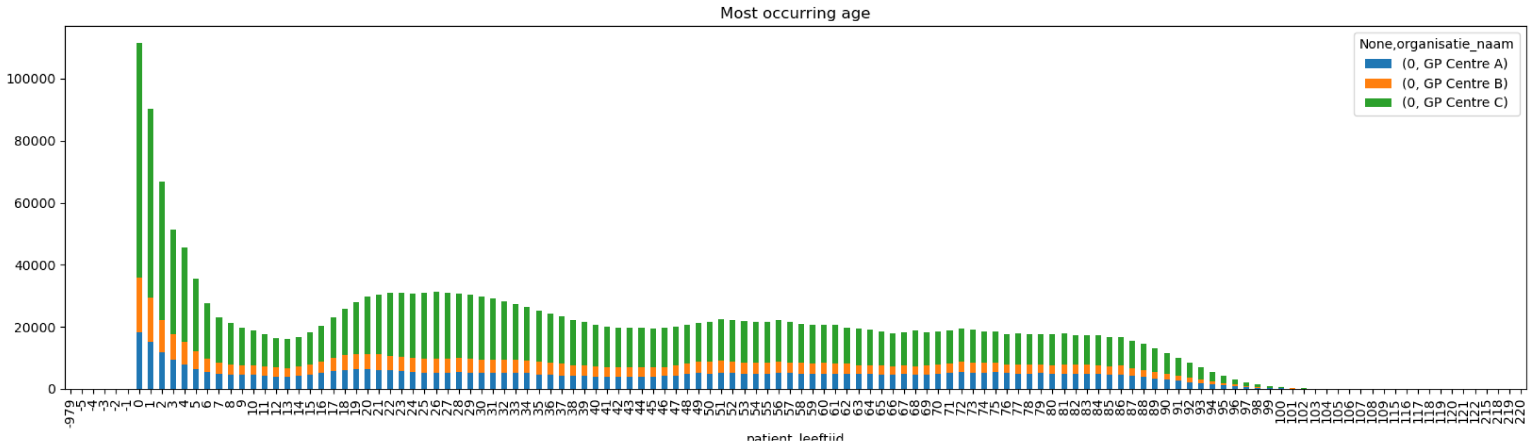


Figure A.1: Count per age of the dataset(strange values are the fault of GP).

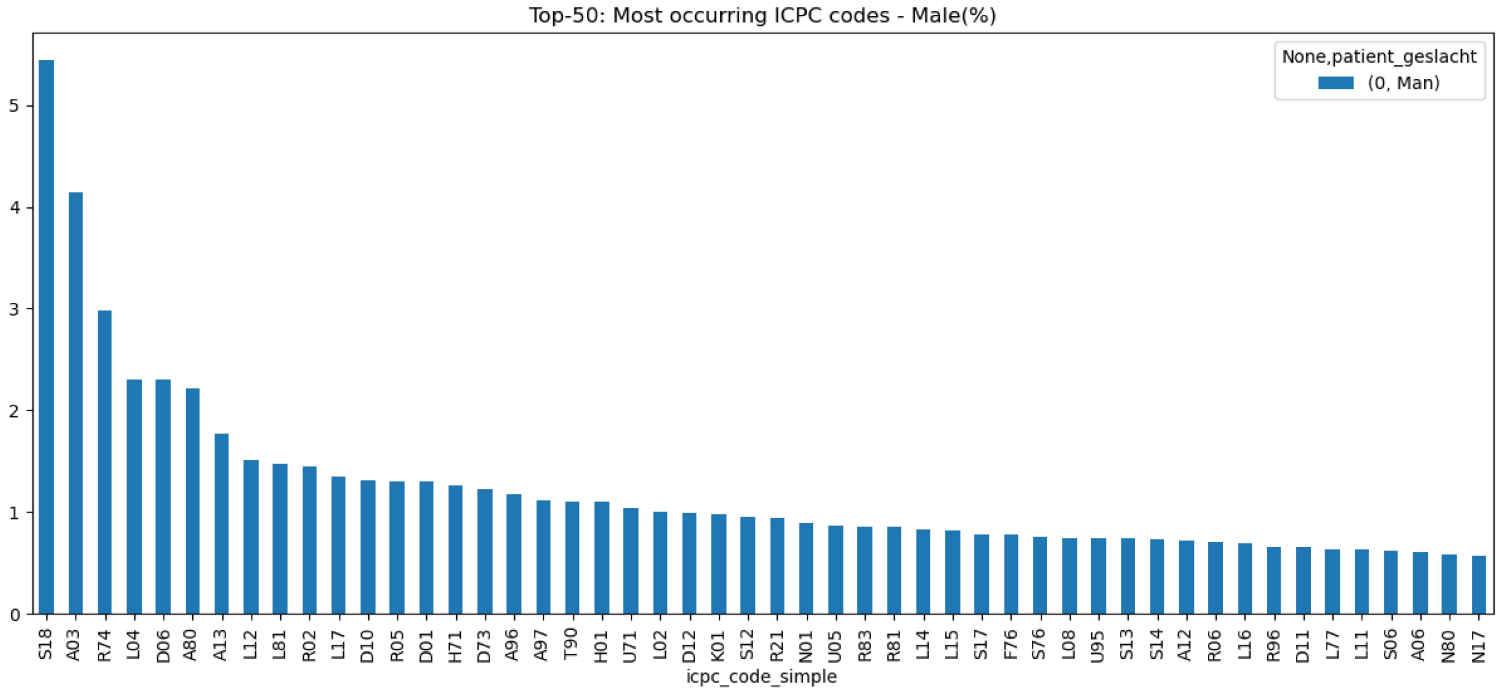


Figure A.2: Most occurring ICPC codes for males.

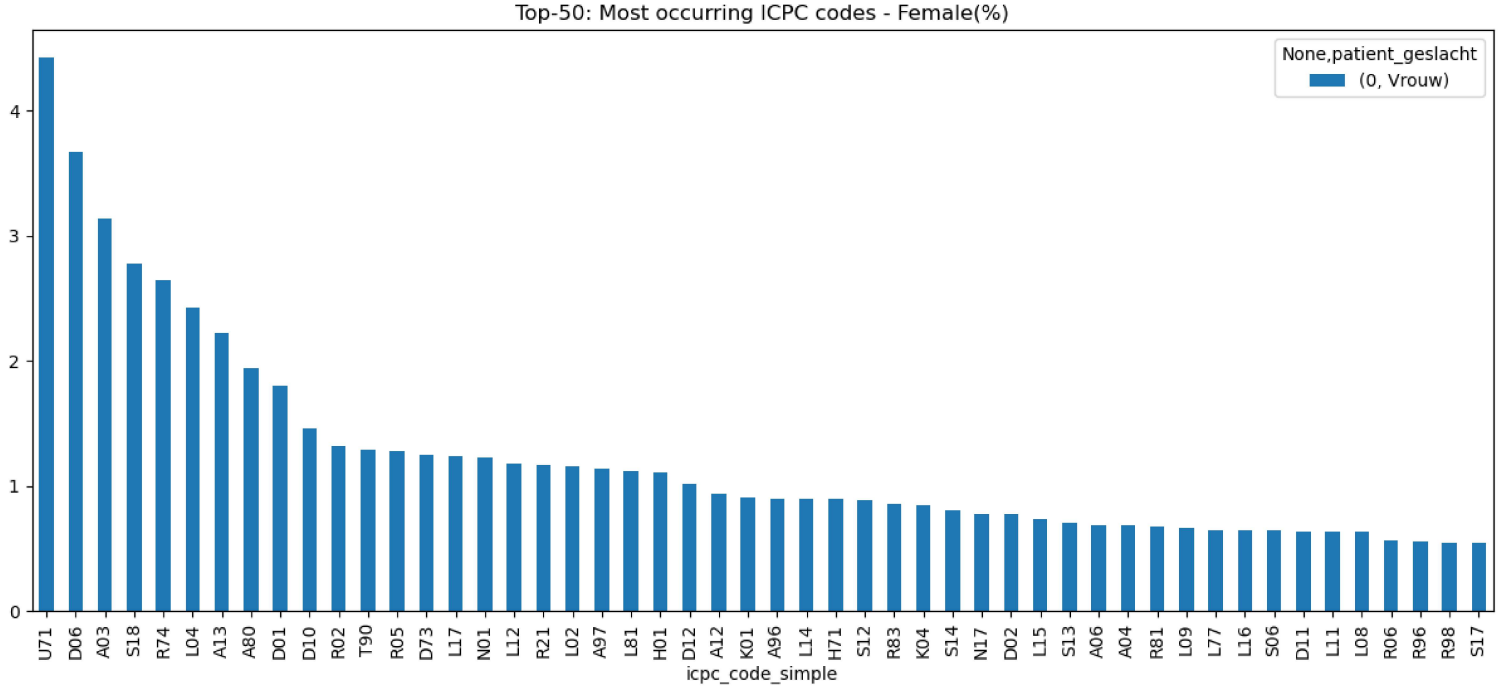


Figure A.3: Most occurring ICPC codes for females.

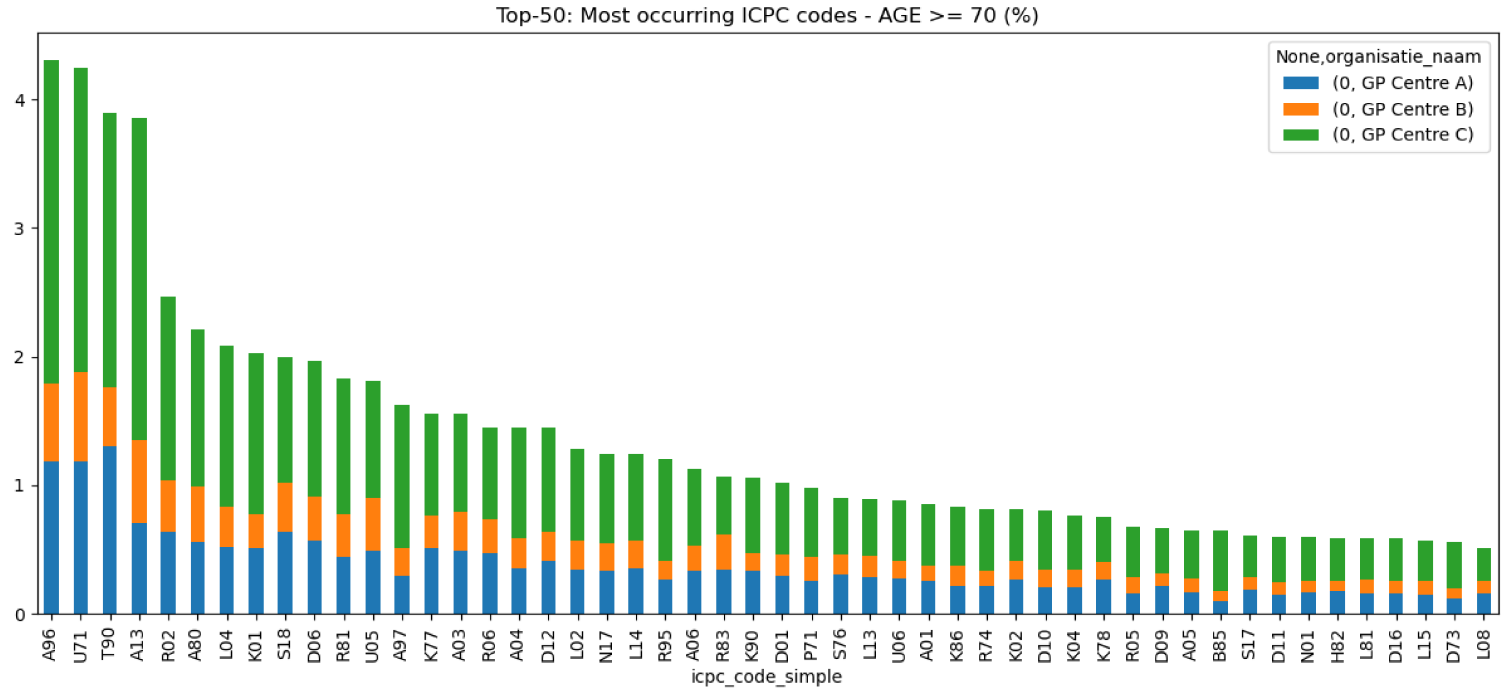


Figure A.4: Fifty most occurring simplified ICPC codes for people over the age of 70.

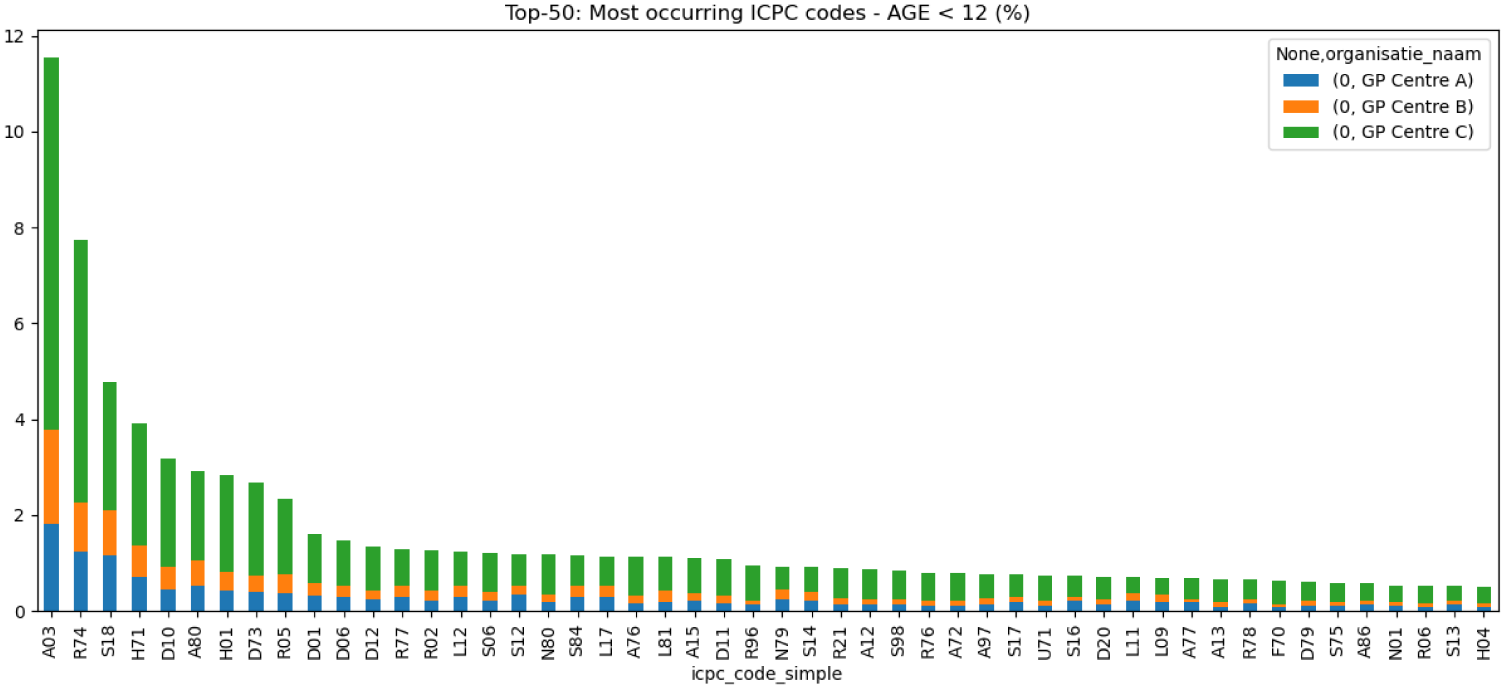


Figure A.5: Fifty most occurring simplified ICPC codes for people younger than the age of 12.

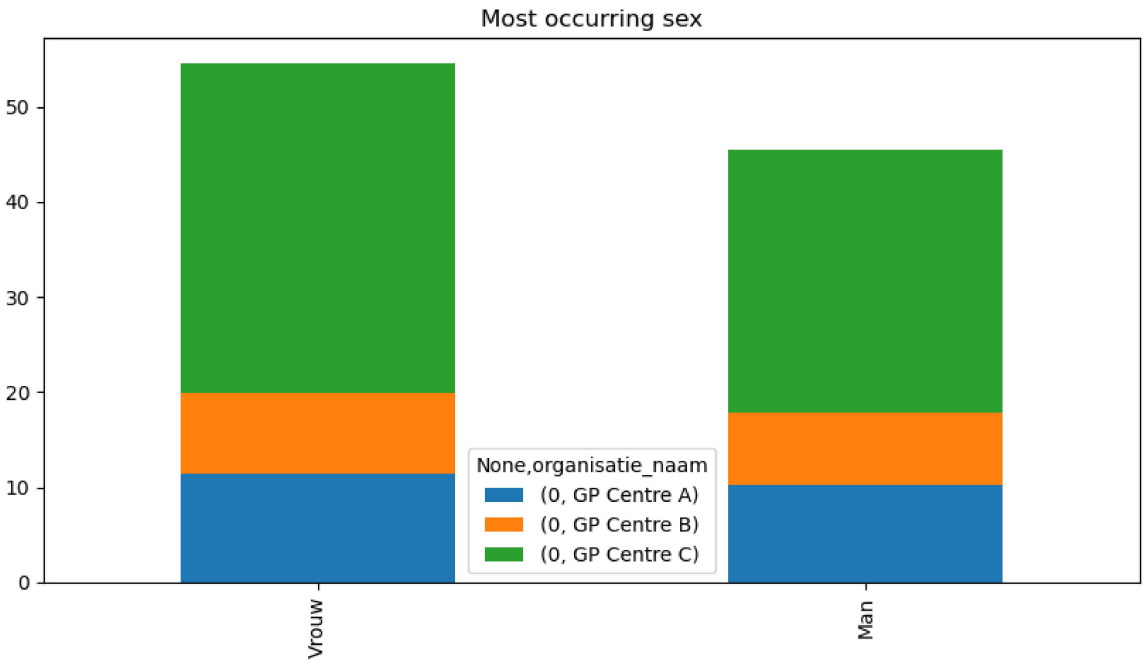


Figure A.6: Male/Female representation of dataset.

Appendix B

Answers to the general questions

Table B.1, Table B.2, Table B.3, Table B.4, Table B.5 provide the answers of each participant to the general questions. The answers are translated from Dutch to English.

Participant 1	**Answer**
Question 1	On the one hand, yes. On the other hand, you also get too many choices and confusion and I usually keep the choice as limited as possible.
Question 2	Yes in most cases
Question 3	No, it usually has a more restrictive effect because you usually come up with the keywords yourself. In a few cases, however, the computer had come up with better words than I could. So it also keeps you sharp. There is, as it were, someone who thinks along with you
Question 4	It's great how the quantity remains limited despite the sometimes very long stories of the patients
Question 5	I find it useful. As a result, the patient's story is compressed
Question 6	The possibility to adjust the s line yourself

Table B.1: Translated answers of Participant 1

Participant 2	**Answer**
Question 1	Yes, because sometimes I'm looking for possible symptom diagnoses of which I don't know ICPC code. This gives me a suggestion in the right direction, so I don't have to enter a number of words in E-rule myself with trial & error
Question 2	Not, because it has sometimes been confirmed that a discovered for eg hand is given while case/diagnosis is about a foot problem. Than is disappointed. Well, as a suggestion when you are not sure about a diagnosis and therefore prefer to choose the underlying(in accordance with ADEPT guidelines)
Question 3	No, for me is noise that distracts from formulating E-rule. They don't always match what I would consider a keyword for a case. For example, the case about 'blisters', AI does not see this as a keyword, but I do.
Question 4	3(-5) is fine by me. Not anymore, because then reading/watching will take too much effort.
Question 5	Nice to be able to 'understand' AI how it arrives at this suggestion, but not for my own thinking because that is more based on clinical reasoning
Question 6	Adjust S-rule yourself and that AI gives new suggestions as a result. Optionally select keywords / keywords yourself that get 'extra weight' in AI (but that I quite laborious)

Table B.2: Translated answers of Participant 2

Participant 3	Answer
Question 1	Yes, I think so because I also have a better picture of the differential diagnosis
Question 2	Yes, with infections redness/warm/swelling with infection. bruises; swelling, whether or not wounds.
Question 3	No
Question 4	2-3 is enough. That is now well understood.
Question 5	Makes it clearer to connect and understand.
Question 6	No need.

Table B.3: Translated answers of Participant 3

Participant 4	Answer
Question 1	no, but possibly the triagist / assistant
Question 2	no diagnosis is already there in my head while reading the case it doesn't change the symptoms.
Question 3	Yes, these do raise alarm symptoms and points for attention. the worst of the complaints come forward more and can help me choose a cure or wait or send in
Question 4	too few keywords not too few diagnoses too many symptoms 3 is a nice overview
Question 5	yes fine the keywords clear color short alarms stand out
Question 6	No need

Table B.4: Translated answers of Participant 4

Participant 5	Answer
Question 1	Yes, pre-sorting already largely selects the correct ICPC codes.
Question 2	Certainly, it shows more options that apply, if there are also ICPC codes with subcodes it would be complete.
Question 3	It's a handy explanation of why the system chooses certain codes, so it's nice for that clarity. As a doctor, I don't go beyond that.
Question 4	Well, I think a maximum of 5 options offers a clear choice.
Question 5	Keywords provide a good insight into the choices of the system to select certain codes. The relevant symptoms too, but are sometimes less specific and may therefore distract from your final choice.
Question 6	I would like to add things myself (or rather that the system does that automatically from the chat > so that you can stand on answers in the chat and that there is the possibility to check it to take over in the S line).

Table B.5: Translated answers of Participant 5

Appendix C

Example of a full user study

See next page for pdf.

Achtergrond

Spreekuur.nl is een online triage consultatie hulpmiddel voor huisartsen. De website laat patiënten een vragenlijst invullen over hun gezondheid, klachtgebied en symptomen. Aan het einde van de vragenlijst kan er een online chat gestart worden met hun huisarts. De huisarts kan een samenvatting zien van de vragen die door de gebruiker in de vragenlijst zijn beantwoordt. Spreekuur.nl voorkomt dat patiënten onnodig fysiek naar de huisarts te gaan door een online consult aan te bieden.

Deze tool is gemaakt door het IT-bedrijf Topicus. Topicus heeft een unieke positie om de grote hoeveelheid verzamelde gegevens en kennis te gebruiken, die normaal niet beschikbaar zijn voor het grote publiek. Door middel van deze data kan Spreekuur.nl gebruikt worden om huisartsen verder te ontlasten door een lijst met voorspelde ICPC codes aan te bieden, gebaseerd op de antwoorden die zijn verkregen uit de vragenlijst.

Doel

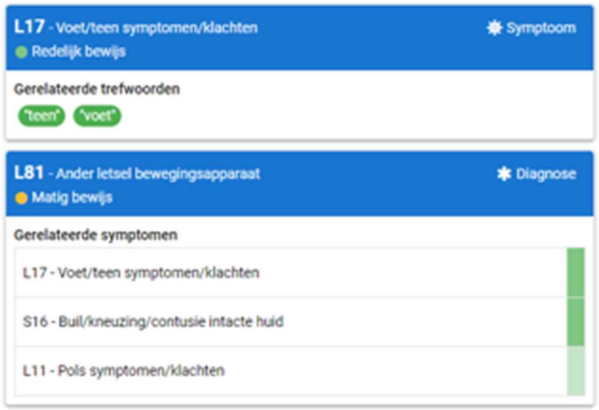
U wordt gevraagd om in deze user study 10 casussen van patiënten te bekijken, elk met dezelfde algemene vragen. Na de 10 casussen, staan nog enkele algemene vragen. Elke casus bestaat uit een ‘S-regel’ (Subjectief van SOEP-notatie) dat automatisch gegenereerd is op basis van de antwoorden van een patiënt op een vragenlijst. Onder de ‘S-regel’ staan verschillende voorspelde mogelijke diagnoses met ICPC codes die het meest relevant zijn. Een mogelijke diagnose kan een symptoom zijn als de diagnose ‘onzeker’ is en wordt gedetailleerder naarmate de huisarts ‘zekerder’ wordt van een diagnose.

Elke voorgestelde mogelijke diagnose heeft een uitleg eronder. Indien de best passende diagnose bestaat uit een symptoom, bestaat de uitleg uit de belangrijkste trefwoorden van de ‘S-regel’. Indien de best passende diagnose bestaat uit een diagnose of ziekte, bestaat de uitleg uit relevante symptomen die passen bij de diagnose én de ‘S-regel’.

Bij elke mogelijke diagnose staat ook een kleur met ‘Matig bewijs’, ‘Redelijk bewijs’ en ‘Sterk bewijs’ die aangeeft hoeveel de voorgestelde mogelijke diagnose past bij de ‘S-regel’. Het is de bedoeling dat de ’S-Regel’ eerst wordt gelezen en dat pas daarna de vragen worden beantwoord.

U mag de volgende site gebruiken voor het vinden van ICPC codes in dien de code niet aanwezig is bij de voorgestelde symptomen: https://viewers.nhg.org/icpcviewer/

Bij eventuele opmerkingen of vragen kunt u contact opnemen met Pieter Zeilstra via e-mail (pieter.zeilstra@topicus.nl) of telefonisch (0639857364).



Voorbeeld van twee voorgestelde diagnoses. Eén diagnose is een symptoom en de andere eén diagnose. De ‘uitleg’ van een symptoom bestaat uit een aantal trefwoorden terwijl de ‘uitleg’ onder een diagnose bestaat uit gerelateerde symptomen.

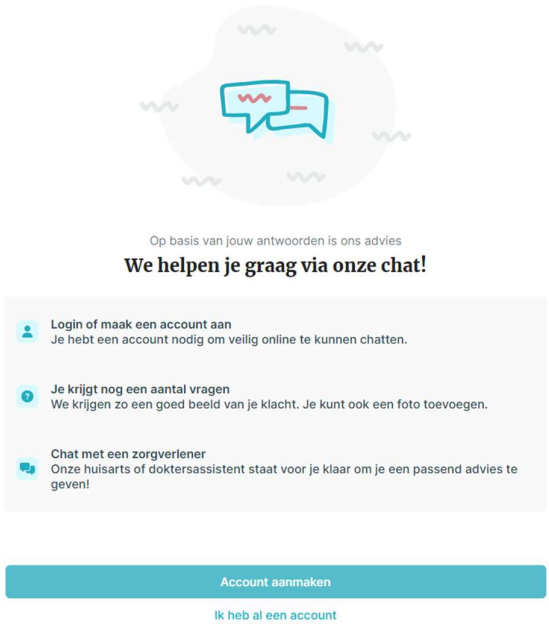
Voorbeeld collage patiënt

Hieronder zijn een paar afbeeldingen om aan te duiden hoe een vragenlijst wordt doorlopen door een patiënt. Een vragenlijst kan tussen de 10 tot 40 vragen bevatten.

Figuur 1 Patiënt kiest een klacht gebied

Figuur 2 Patiënt krijgt vragen

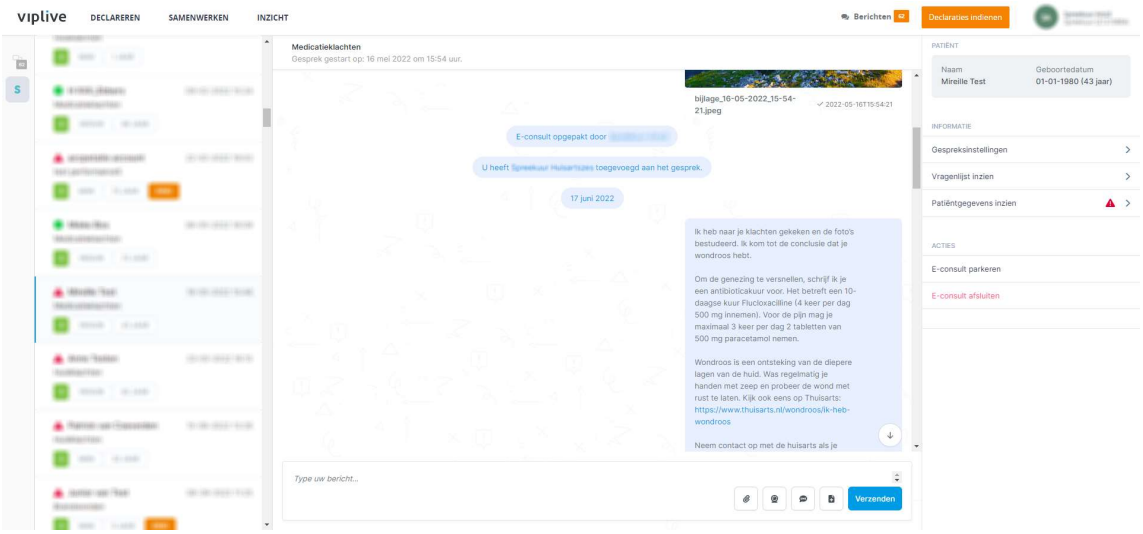
Figuur 3 Patiënt krijgt vragen



Figuur 4 Patiënt na het beantwoorden van alle vragen een chat met eigen huisarts starten

Voorbeeld collage huisarts

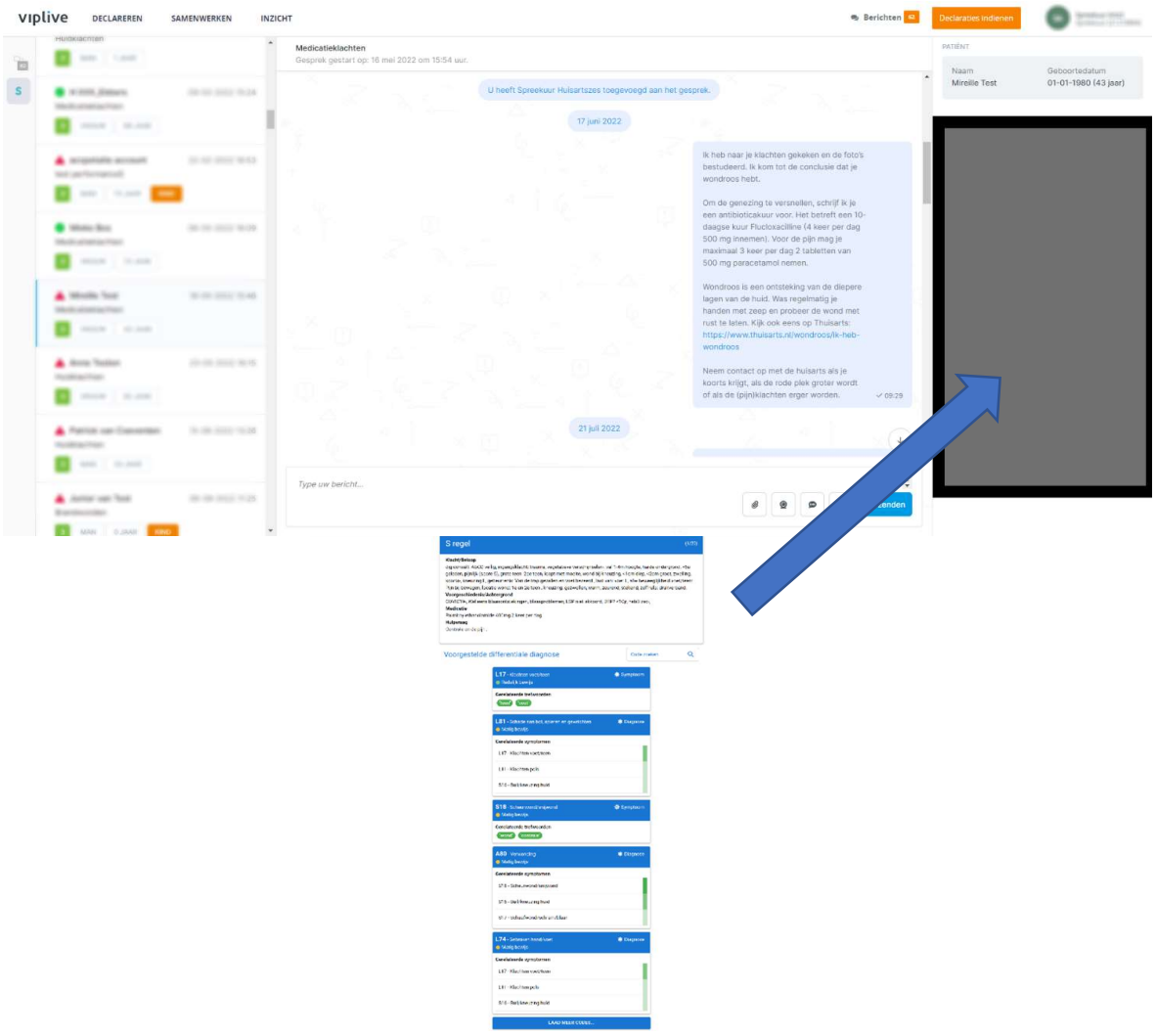
De afbeelding hieronder laat zien hoe een huisarts een vragenlijst kan inzien en een chat kan beginnen.



Figuur 5 Huisarts kan antwoorden van vragenlijst inzien en vragen stellen via chat

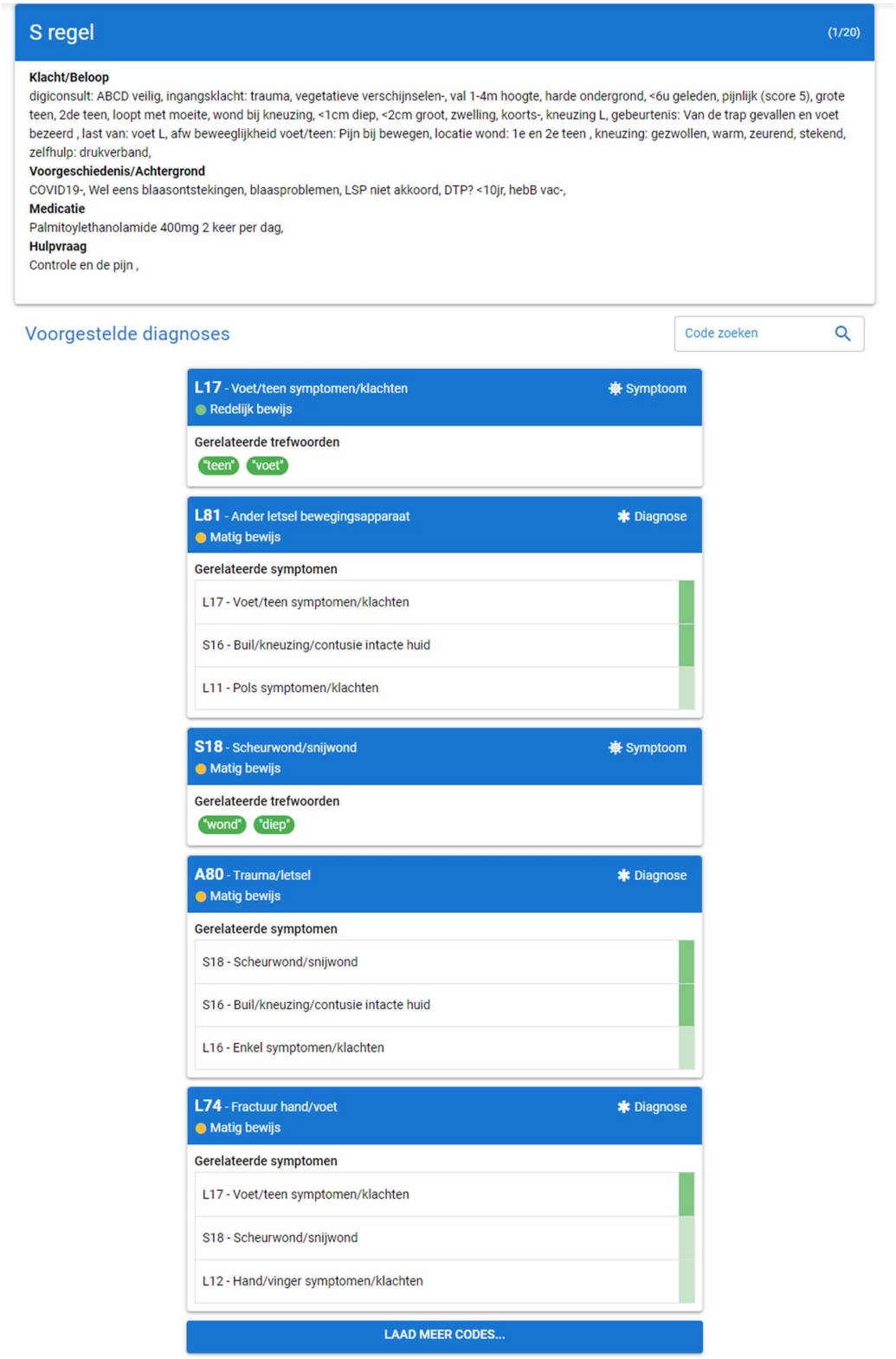
Hoe ziet dit eruit?

Aan het chat-gedeelte van een arts zal naast de chat de gegenereerde ‘S-regel’ met voorgestelde diagnoses te vinden zijn.



Begin vragenlijst

Casus 1



Vragen casus 1:

1. Welke ICPC code vind u het meest passend (een code die hier niet bijstaat mag ook)?

   Click or tap here to enter text.

2. Vindt u de **voorgestelde diagnoses** passen bij de S-regel?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

3. Vindt u de **gerelateerde** symptomen passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

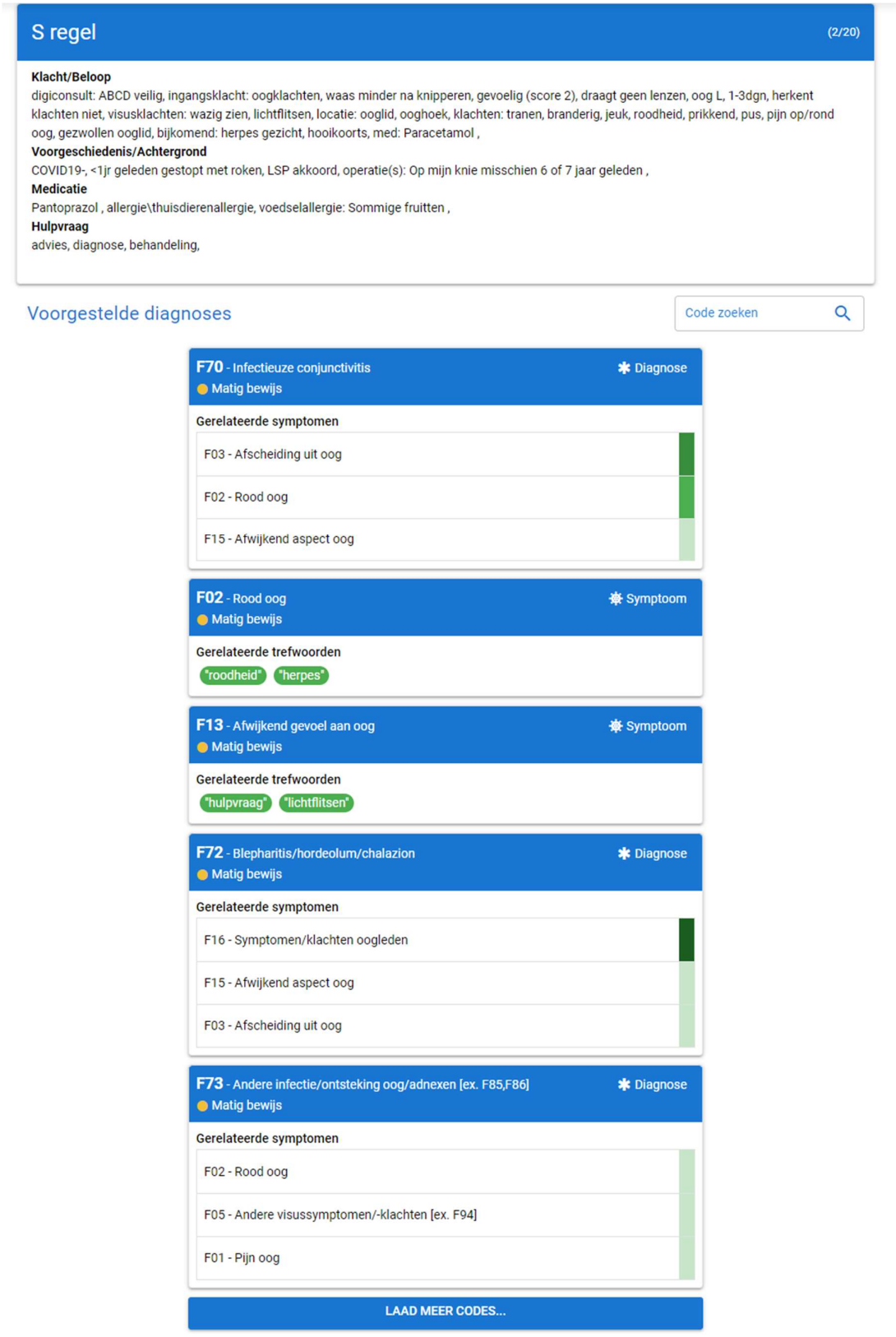
4. Vindt u de **gerelateerde** trefwoorden passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

5. Zou u “L17 -Voet/teen symptomen/klachten” een adequate code vinden?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

Casus 2


S regel
(2/20)
Klacht/Beloop
digiconsult: ABCD veilig, ingangsklacht: oogklachten, waas minder na knipperen, gevoelig (score 2), draagt geen lenzen, oog L, 1-3dgn, herkent klachten niet, visusklachten: wazig zien, lichtflitsen, locatie: ooglid, ooghoek, klachten: tranen, branderig, jeuk, roodheid, prikkend, pus, pijn op/rond oog, gezwollen ooglid, bijkomend: herpes gezicht, hooikoorts, med: Paracetamol ,
Voorgeschiedenis/Achtergrond
COVID19-, <1jr geleden gestopt met roken, LSP akkoord, operatie(s): Op mijn knie misschien 6 of 7 jaar geleden ,
Medicatie
Pantoprazol , allergie\thuisdierenallergie, voedselallergie: Sommige fruitten ,
Hulpvraag
advies, diagnose, behandeling,
Voorgestelde diagnoses
Code zoeken
F70 - Infectieuze conjunctivitis
Diagnose
Matig bewijs
Gerelateerde symptomen
F03 - Afscheiding uit oog
F02 - Rood oog
F15 - Afwijkend aspect oog
F02 - Rood oog
Symptoom
Matig bewijs
Gerelateerde trefwoorden
"roodheid"
"herpes"
F13 - Afwijkend gevoel aan oog
Symptoom
Matig bewijs
Gerelateerde trefwoorden
"hulpvraag"
"lichtflitsen"
F72 - Blepharitis/hordeolum/chalazion
Diagnose
Matig bewijs
Gerelateerde symptomen
F16 - Symptomen/klachten oogleden
F15 - Afwijkend aspect oog
F03 - Afscheiding uit oog
F73 - Andere infectie/ontsteking oog/adnexen [ex. F85,F86]
Diagnose
Matig bewijs
Gerelateerde symptomen
F02 - Rood oog
F05 - Andere visussymptomen/-klachten [ex. F94]
F01 - Pijn oog
LAAD MEER CODES...

Vragen casus 2:

1. Welke ICPC code vind u het meest passend (een code die hier niet bijstaat mag ook)?

   Click or tap here to enter text.

2. Vindt u de **voorgestelde diagnoses** passen bij de s-regel?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

3. Vindt u de **gerelateerde** symptomen passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

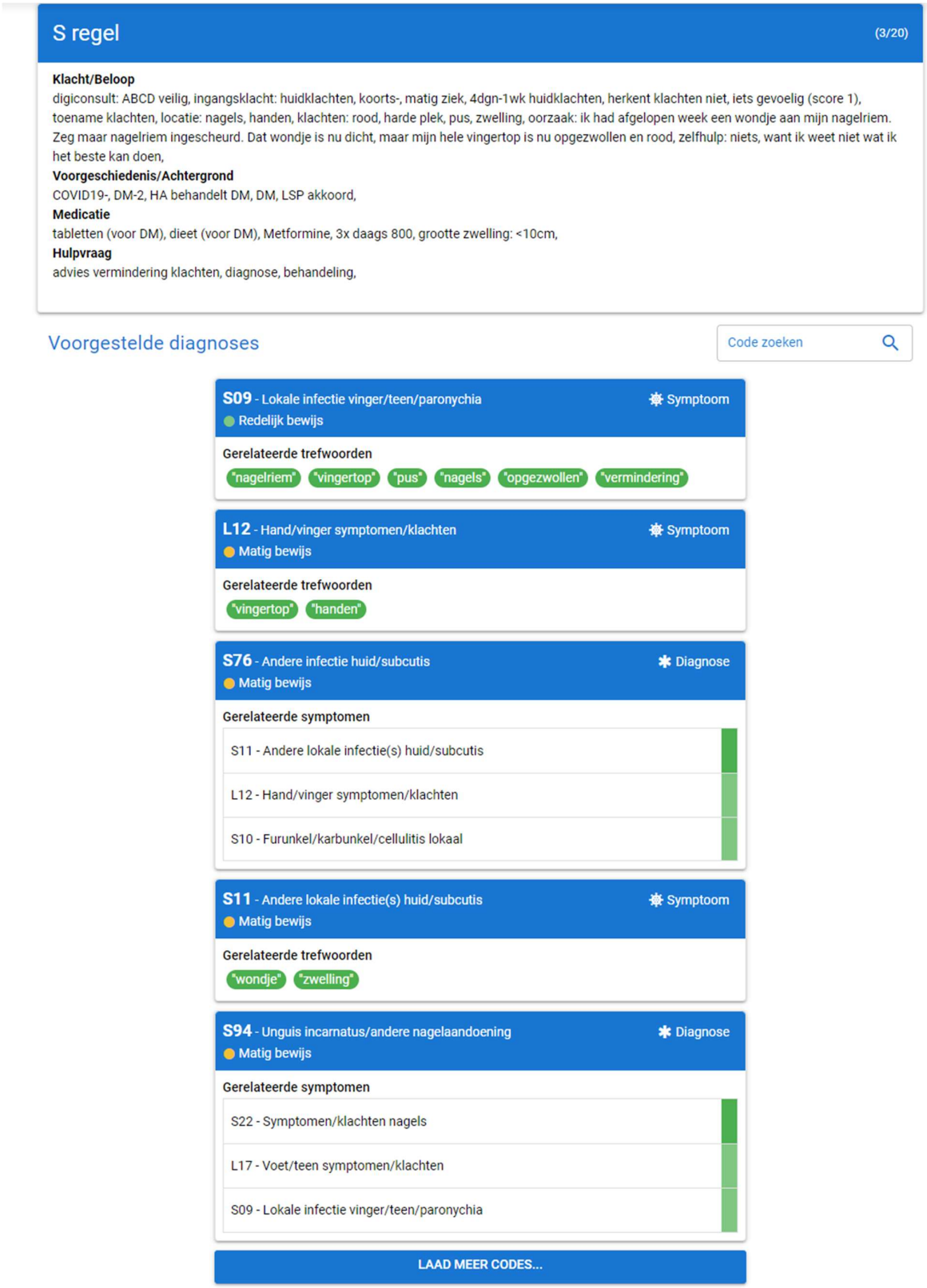
4. Vindt u de **gerelateerde** trefwoorden passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

5. Zou u “F72 - Blepharitis/hordeolum/chalazion” een adequate code vinden?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

Casus 3


S regel
(3/20)
Klacht/Beloop
digiconsult: ABCD veilig, ingangsklacht: huidklachten, koorts-, matig ziek, 4dgn-1wk huidklachten, herkent klachten niet, iets gevoelig (score 1), toename klachten, locatie: nagels, handen, klachten: rood, harde plek, pus, zwelling, oorzaak: ik had afgelopen week een wondje aan mijn nagelriem. Zeg maar nagelriem ingescheurd. Dat wondje is nu dicht, maar mijn hele vingertop is nu opgezwollen en rood, zelfhulp: niets, want ik weet niet wat ik het beste kan doen,
Voorgeschiedenis/Achtergrond
COVID19-, DM-2, HA behandelt DM, DM, LSP akkoord,
Medicatie
tabletten (voor DM), dieet (voor DM), Metformine, 3x daags 800, grootte zwelling: <10cm,
Hulpvraag
advies vermindering klachten, diagnose, behandeling,
Voorgestelde diagnoses
Code zoeken
S09 - Lokale infectie vinger/teen/paronychia
Symptoom
Redelijk bewijs
Gerelateerde trefwoorden
"nagelriem" "vingertop" "pus" "nagels" "opgezwollen" "vermindering"
L12 - Hand/vinger symptomen/klachten
Symptoom
Matig bewijs
Gerelateerde trefwoorden
"vingertop" "handen"
S76 - Andere infectie huid/subcutis
Diagnose
Matig bewijs
Gerelateerde symptomen
S11 - Andere lokale infectie(s) huid/subcutis
L12 - Hand/vinger symptomen/klachten
S10 - Furunkel/karbunkel/cellulitis lokaal
S11 - Andere lokale infectie(s) huid/subcutis
Symptoom
Matig bewijs
Gerelateerde trefwoorden
"wondje" "zwelling"
S94 - Unguis incarnatus/andere nagelaandoening
Diagnose
Matig bewijs
Gerelateerde symptomen
S22 - Symptomen/klachten nagels
L17 - Voet/teen symptomen/klachten
S09 - Lokale infectie vinger/teen/paronychia
LAAD MEER CODES...

Vragen casus 3:

1. Welke ICPC code vindt u het meest passend (een code die hier niet bijstaat mag ook)?

Click or tap here to enter text.

2. Vindt u de **voorgestelde diagnoses** passen bij de s-regel?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

3. Vindt u de **gerelateerde symptomen** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

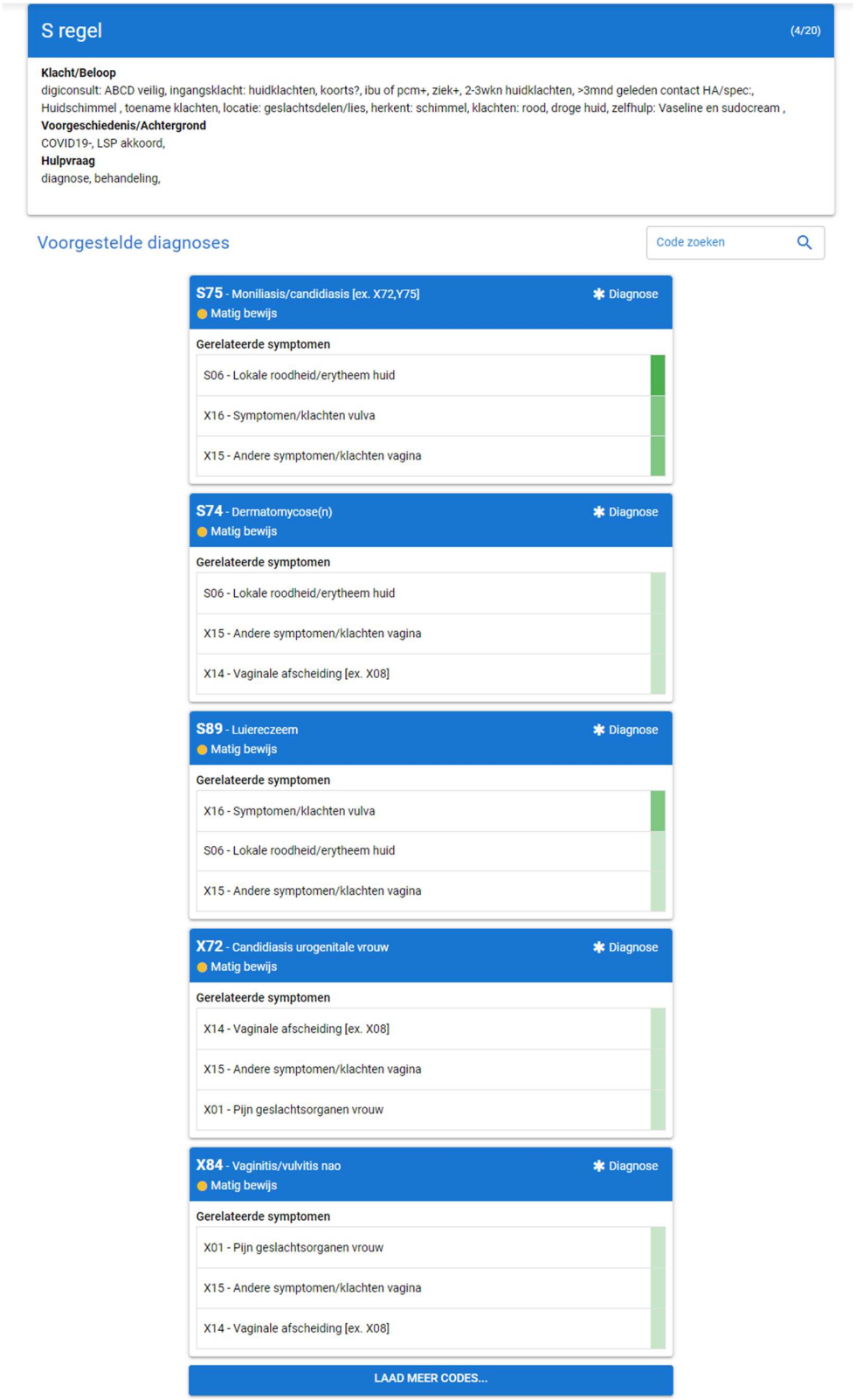
4. Vindt u de **gerelateerde trefwoorden** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

5. Zou u “S09 - Lokale infectie vinger/teen/paronychia” een adequate code vinden?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

Casus 4


S regel
(4/20)
Klacht/Beloop
digiconsult: ABCD veilig, ingangsklacht: huidklachten, koorts?, ibu of pcm+, ziek+, 2-3wkn huidklachten, >3mnd geleden contact HA/spec:, Huidschimmel , toename klachten, locatie: geslachtsdelen/lies, herkent: schimmel, klachten: rood, droge huid, zelfhulp: Vaseline en sudocream ,
Voorgeschiedenis/Achtergrond
COVID19-, LSP akkoord,
Hulpvraag
diagnose, behandeling,
Voorgestelde diagnoses
Code zoeken
S75 - Moniliasis/candidiasis [ex. X72,Y75]
Diagnose
Matig bewijs
Gerelateerde symptomen
S06 - Lokale roodheid/erytheem huid
X16 - Symptomen/klachten vulva
X15 - Andere symptomen/klachten vagina
S74 - Dermatomycose(n)
Diagnose
Matig bewijs
Gerelateerde symptomen
S06 - Lokale roodheid/erytheem huid
X15 - Andere symptomen/klachten vagina
X14 - Vaginale afscheiding [ex. X08]
S89 - Luiereczeem
Diagnose
Matig bewijs
Gerelateerde symptomen
X16 - Symptomen/klachten vulva
S06 - Lokale roodheid/erytheem huid
X15 - Andere symptomen/klachten vagina
X72 - Candidiasis urogenitale vrouw
Diagnose
Matig bewijs
Gerelateerde symptomen
X14 - Vaginale afscheiding [ex. X08]
X15 - Andere symptomen/klachten vagina
X01 - Pijn geslachtsorganen vrouw
X84 - Vaginitis/vulvitis nao
Diagnose
Matig bewijs
Gerelateerde symptomen
X01 - Pijn geslachtsorganen vrouw
X15 - Andere symptomen/klachten vagina
X14 - Vaginale afscheiding [ex. X08]
LAAD MEER CODES...

Vragen casus 4:

1. Welke ICPC code vindt u het meest passend (een code die hier niet bijstaat mag ook)?
   Click or tap here to enter text.
2. Vindt u de **voorgestelde diagnoses** passen bij de s-regel?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

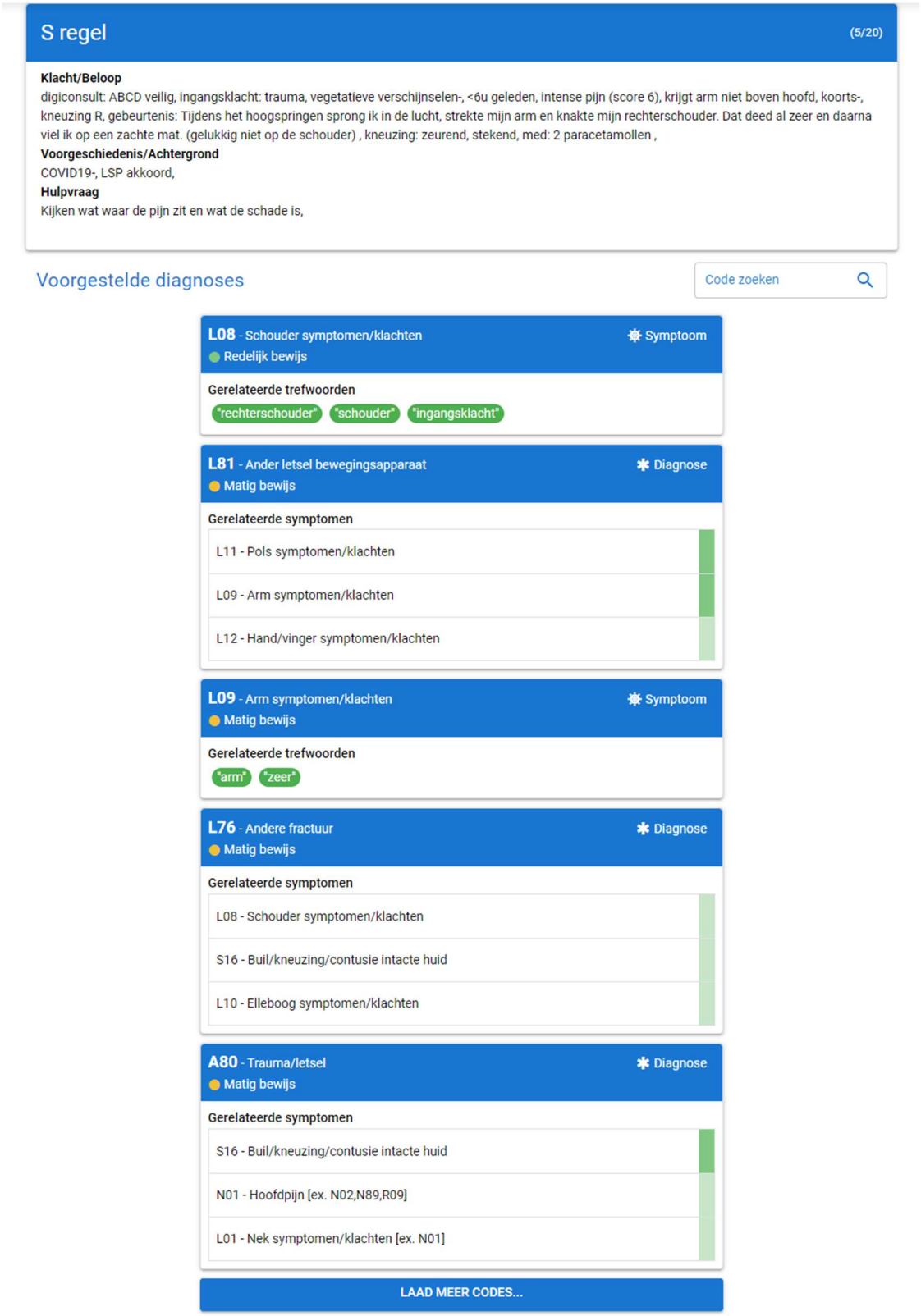
3. Vindt u de **gerelateerde symptomen** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

4. Zou u “X15 - Andere symptomen/klachten vagina” een adequate code vinden?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

Casus 5


S regel
(5/20)
Klacht/Beloop
digiconsult: ABCD veilig, ingangsklacht: trauma, vegetatieve verschijnselen-, <6u geleden, intense pijn (score 6), krijgt arm niet boven hoofd, koorts-, kneuzing R, gebeurtenis: Tijdens het hoogspringen sprong ik in de lucht, strekte mijn arm en knakte mijn rechterschouder. Dat deed al zeer en daarna viel ik op een zachte mat. (gelukkig niet op de schouder) , kneuzing: zeurend, stekend, med: 2 paracetamollen ,
Voorgeschiedenis/Achtergrond
COVID19-, LSP akkoord,
Hulpvraag
Kijken wat waar de pijn zit en wat de schade is,
Voorgestelde diagnoses
Code zoeken
L08 - Schouder symptomen/klachten
Symptoom
Redelijk bewijs
Gerelateerde trefwoorden
"rechterschouder"
"schouder"
"ingangsklacht"
L81 - Ander letsel bewegingsapparaat
Diagnose
Matig bewijs
Gerelateerde symptomen
L11 - Pols symptomen/klachten
L09 - Arm symptomen/klachten
L12 - Hand/vinger symptomen/klachten
L09 - Arm symptomen/klachten
Symptoom
Matig bewijs
Gerelateerde trefwoorden
"arm"
"zeer"
L76 - Andere fractuur
Diagnose
Matig bewijs
Gerelateerde symptomen
L08 - Schouder symptomen/klachten
S16 - Buil/kneuzing/contusie intacte huid
L10 - Elleboog symptomen/klachten
A80 - Trauma/letsel
Diagnose
Matig bewijs
Gerelateerde symptomen
S16 - Buil/kneuzing/contusie intacte huid
N01 - Hoofdpijn [ex. N02,N89,R09]
L01 - Nek symptomen/klachten [ex. N01]
LAAD MEER CODES...

Vragen casus 5:

1. Welke ICPC code vindt u het meest passend (een code die hier niet bijstaat mag ook)?

Click or tap here to enter text.

2. Vindt u de **voorgestelde diagnoses** passen bij de s-regel?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

3. Vindt u de **gerelateerde symptomen** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

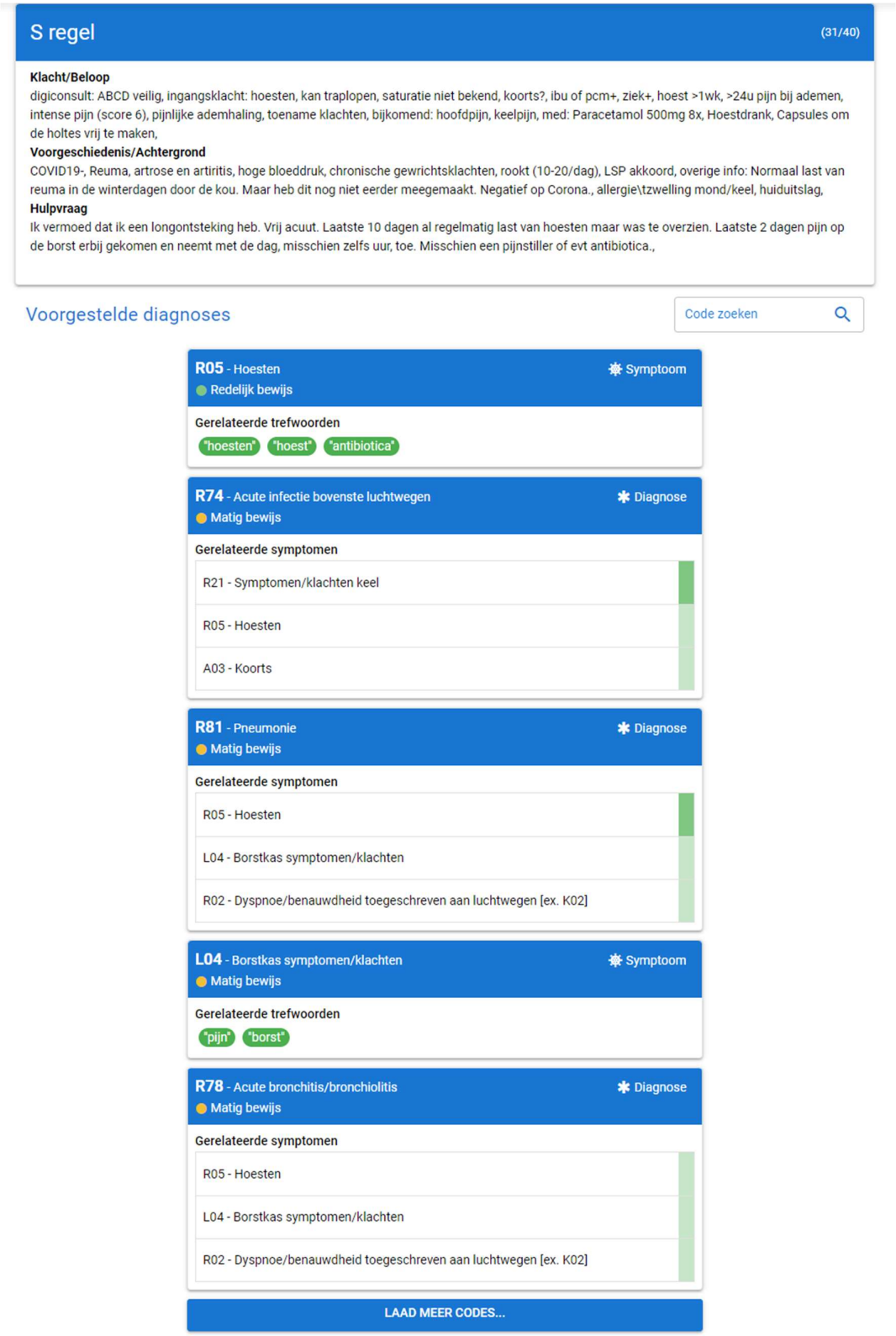
4. Vindt u de **gerelateerde trefwoorden** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

5. Zou u “L08 - Schouder symptomen/klachten” een adequate code vinden?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

Casus 6


S regel
(31/40)
Klacht/Beloop
digiconsult: ABCD veilig, ingangsklacht: hoesten, kan traplopen, saturatie niet bekend, koorts?, ibu of pcm+, ziek+, hoest >1wk, >24u pijn bij ademen, intense pijn (score 6), pijnlijke ademhaling, toename klachten, bijkomend: hoofdpijn, keelpijn, med: Paracetamol 500mg 8x, Hoestdrank, Capsules om de holtes vrij te maken,
Voorgeschiedenis/Achtergrond
COVID19-, Reuma, artrose en artiritis, hoge bloeddruk, chronische gewrichtsklachten, rookt (10-20/dag), LSP akkoord, overige info: Normaal last van reuma in de winterdagen door de kou. Maar heb dit nog niet eerder meegemaakt. Negatief op Corona., allergie\tzwelling mond/keel, huiduitslag,
Hulpvraag
Ik vermoed dat ik een longontsteking heb. Vrij acuut. Laatste 10 dagen al regelmatig last van hoesten maar was te overzien. Laatste 2 dagen pijn op de borst erbij gekomen en neemt met de dag, misschien zelfs uur, toe. Misschien een pijnstiller of evt antibiotica.,
Voorgestelde diagnoses
Code zoeken
R05 - Hoesten
Symptoom
Redelijk bewijs
Gerelateerde trefwoorden
"hoesten"
"hoest"
"antibiotica"
R74 - Acute infectie bovenste luchtwegen
Diagnose
Matig bewijs
Gerelateerde symptomen
R21 - Symptomen/klachten keel
R05 - Hoesten
A03 - Koorts
R81 - Pneumonie
Diagnose
Matig bewijs
Gerelateerde symptomen
R05 - Hoesten
L04 - Borstkas symptomen/klachten
R02 - Dyspnoe/benauwdheid toegeschreven aan luchtwegen [ex. K02]
L04 - Borstkas symptomen/klachten
Symptoom
Matig bewijs
Gerelateerde trefwoorden
"pijn"
"borst"
R78 - Acute bronchitis/bronchiolitis
Diagnose
Matig bewijs
Gerelateerde symptomen
R05 - Hoesten
L04 - Borstkas symptomen/klachten
R02 - Dyspnoe/benauwdheid toegeschreven aan luchtwegen [ex. K02]
LAAD MEER CODES...

Vragen casus 6:

1. Welke ICPC code vindt u het meest passend (een code die hier niet bijstaat mag ook)?

Click or tap here to enter text.

2. Vindt u de **voorgestelde diagnose**s passen bij de s-regel?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

3. Vindt u de **gerelateerde symptomen** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

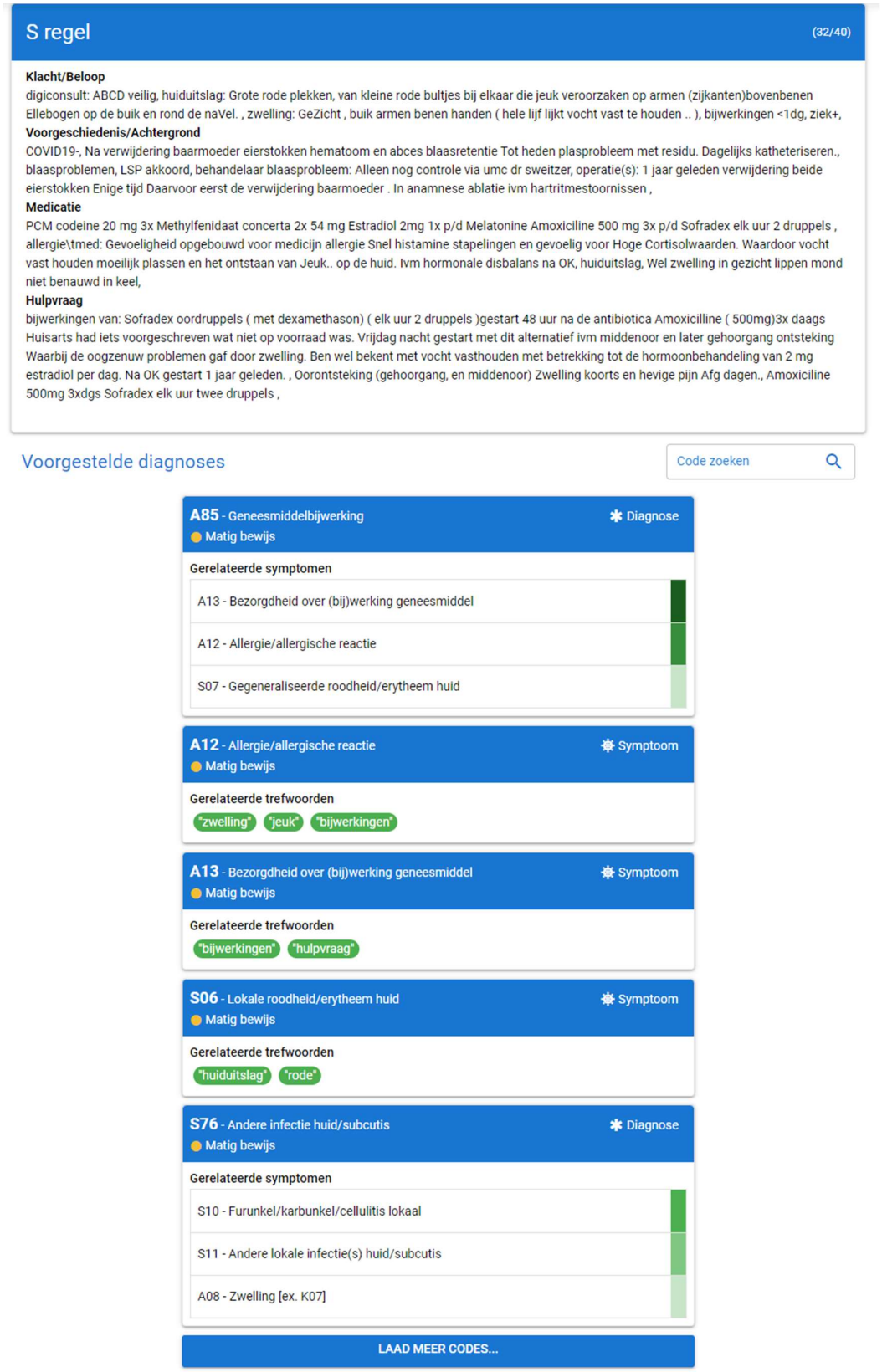
4. Vindt u de **gerelateerde trefwoorden** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

5. Zou u “R05 - Hoesten” een adequate code vinden?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

Casus 7


S regel
(32/40)
Klacht/Beloop
digiconsult: ABCD veilig, huiduitslag: Grote rode plekken, van kleine rode bultjes bij elkaar die jeuk veroorzaken op armen (zijkanten)bovenbenen Ellebogen op de buik en rond de naVel. , zwelling: GeZicht , buik armen benen handen (hele lijf lijkt vocht vast te houden ..), bijwerkingen <1dg, ziek+,
Voorgeschiedenis/Achtergrond
COVID19-, Na verwijdering baarmoeder eierstokken hematoom en abces blaasretentie Tot heden plasprobleem met residu. Dagelijks katheteriseren., blaasproblemen, LSP akkoord, behandelaar blaasprobleem: Alleen nog controle via umc dr sweitzer, operatie(s): 1 jaar geleden verwijdering beide eierstokken Enige tijd Daarvoor eerst de verwijdering baarmoeder . In anamnese ablatie ivm hartritmestoornissen ,
Medicatie
PCM codeine 20 mg 3x Methylfenidaat concerta 2x 54 mg Estradiol 2mg 1x p/d Melatonine Amoxiciline 500 mg 3x p/d Sofradex elk uur 2 druppels , allergie\tmed: Gevoeligheid opgebouwd voor medicijn allergie Snel histamine stapelingen en gevoelig voor Hoge Cortisolwaarden. Waardoor vocht vast houden moeilijk plassen en het ontstaan van Jeuk.. op de huid. Ivm hormonale disbalans na OK, huiduitslag, Wel zwelling in gezicht lippen mond niet benauwd in keel,
Hulpvraag
bijwerkingen van: Sofradex oordruppels (met dexamethason) (elk uur 2 druppels)gestart 48 uur na de antibiotica Amoxicilline (500mg)3x daags Huisarts had iets voorgeschreven wat niet op voorraad was. Vrijdag nacht gestart met dit alternatief ivm middenoor en later gehoorgang ontsteking Waarbij de oogzenuw problemen gaf door zwelling. Ben wel bekent met vocht vasthouden met betrekking tot de hormoonbehandeling van 2 mg estradiol per dag. Na OK gestart 1 jaar geleden. , Oorontsteking (gehoorgang, en middenoor) Zwelling koorts en hevige pijn Afg dagen., Amoxiciline 500mg 3xdgs Sofradex elk uur twee druppels ,
Voorgestelde diagnoses
Code zoeken
A85 - Geneesmiddelbijwerking
Diagnose
Matig bewijs
Gerelateerde symptomen
A13 - Bezorgdheid over (bij)werking geneesmiddel
A12 - Allergie/allergische reactie
S07 - Gegeneraliseerde roodheid/erytheem huid
A12 - Allergie/allergische reactie
Symptoom
Matig bewijs
Gerelateerde trefwoorden
"zwelling" "jeuk" "bijwerkingen"
A13 - Bezorgdheid over (bij)werking geneesmiddel
Symptoom
Matig bewijs
Gerelateerde trefwoorden
"bijwerkingen" "hulpvraag"
S06 - Lokale roodheid/erytheem huid
Symptoom
Matig bewijs
Gerelateerde trefwoorden
"huiduitslag" "rode"
S76 - Andere infectie huid/subcutis
Diagnose
Matig bewijs
Gerelateerde symptomen
S10 - Furunkel/karbunkel/cellulitis lokaal
S11 - Andere lokale infectie(s) huid/subcutis
A08 - Zwelling [ex. K07]
LAAD MEER CODES...

Vragen casus 7:

1. Welke ICPC code vindt u het meest passend (een code die hier niet bijstaat mag ook)?

Click or tap here to enter text.

2. Vindt u de **voorgestelde diagnoses** passen bij de s-regel?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

3. Vindt u de **gerelateerde symptomen** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

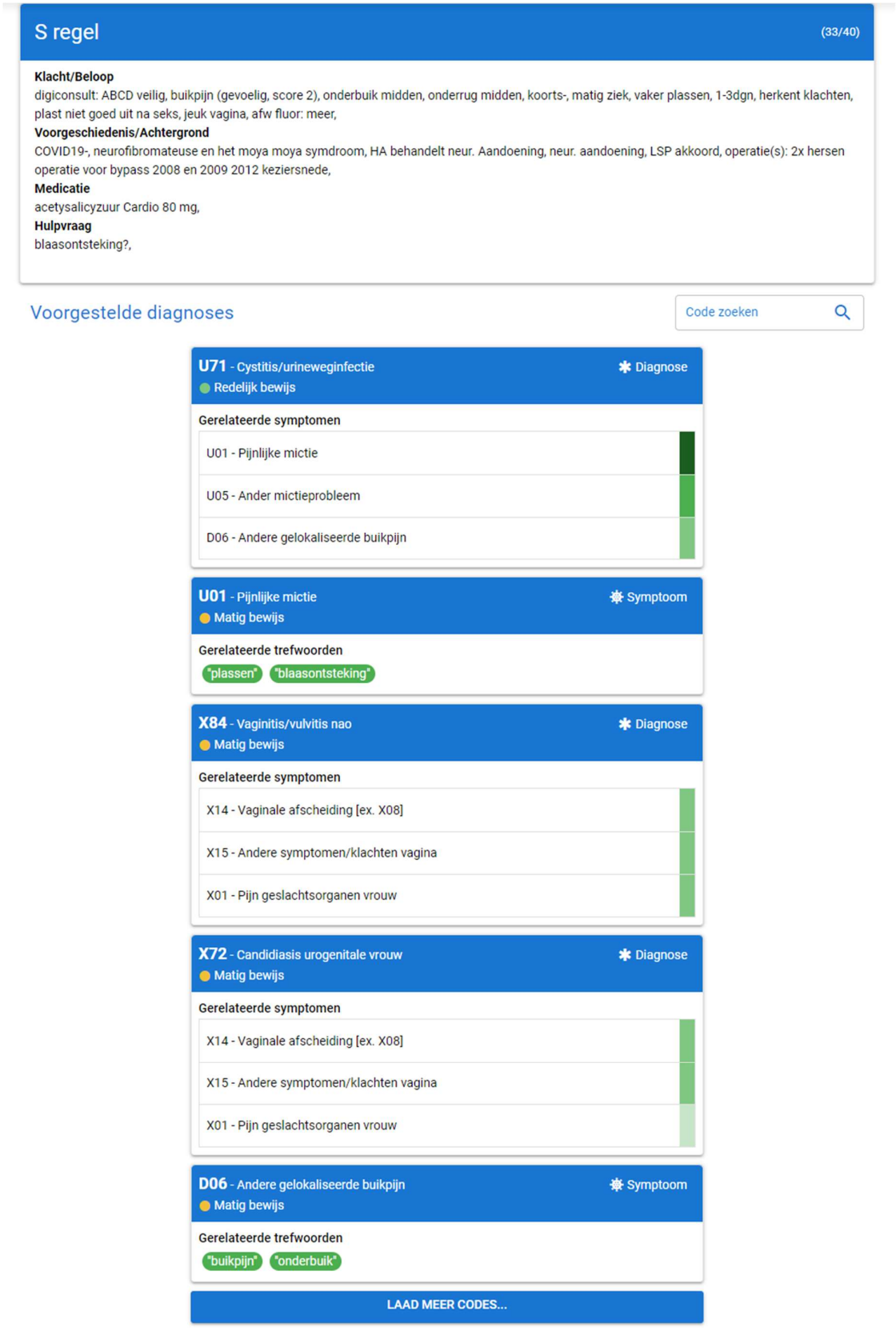
4. Vindt u de **gerelateerde trefwoorden** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

5. Zou u “S04 - Lokale zwelling/papel/knobbel huid/subcutis” een adequate code vinden?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

Casus 8


S regel
(33/40)
Klacht/Beloop
digiconsult: ABCD veilig, buikpijn (gevoelig, score 2), onderbuik midden, onderrug midden, koorts-, matig ziek, vaker plassen, 1-3dgn, herkent klachten, plast niet goed uit na seks, jeuk vagina, afw fluor: meer,
Voorgeschiedenis/Achtergrond
COVID19-, neurofibromateuse en het moya moya symdroom, HA behandelt neur. Aandoening, neur. aandoening, LSP akkoord, operatie(s): 2x hersen operatie voor bypass 2008 en 2009 2012 keziersnede,
Medicatie
acetysalicyzuur Cardio 80 mg,
Hulpvraag
blaasontsteking?,
Voorgestelde diagnoses
Code zoeken
U71 - Cystitis/urineweginfectie
Diagnose
Redelijk bewijs
Gerelateerde symptomen
U01 - Pijnlijke mictie
U05 - Ander mictieprobleem
D06 - Andere gelokaliseerde buikpijn
U01 - Pijnlijke mictie
Symptoom
Matig bewijs
Gerelateerde trefwoorden
"plassen"
"blaasontsteking"
X84 - Vaginitis/vulvitis nao
Diagnose
Matig bewijs
Gerelateerde symptomen
X14 - Vaginale afscheiding [ex. X08]
X15 - Andere symptomen/klachten vagina
X01 - Pijn geslachtsorganen vrouw
X72 - Candidiasis urogenitale vrouw
Diagnose
Matig bewijs
Gerelateerde symptomen
X14 - Vaginale afscheiding [ex. X08]
X15 - Andere symptomen/klachten vagina
X01 - Pijn geslachtsorganen vrouw
D06 - Andere gelokaliseerde buikpijn
Symptoom
Matig bewijs
Gerelateerde trefwoorden
"buikpijn"
"onderbuik"
LAAD MEER CODES...

Vragen casus 8:

6. Welke ICPC code vindt u het meest passend (een code die hier niet bijstaat mag ook)?

Click or tap here to enter text.

7. Vindt u de **voorgestelde diagnoses** passen bij de s-regel?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

8. Vindt u de **gerelateerde symptomen** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

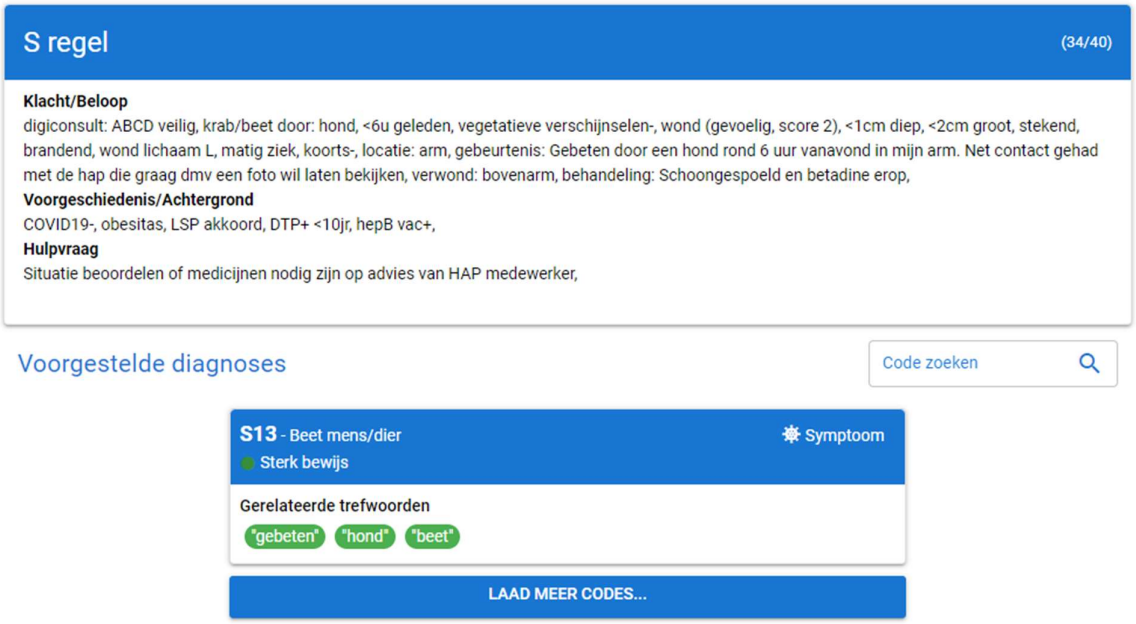
9. Vindt u de **gerelateerde trefwoorden** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

10. Zou u “U01 - Pijnlijke mictie” een adequate code vinden?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

Casus 9



Vragen casus 9:

1. Welke ICPC code vindt u het meest passend (een code die hier niet bijstaat mag ook)?

   Click or tap here to enter text.

2. Vindt u de **voorgestelde diagnoses** passen bij de s-regel?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

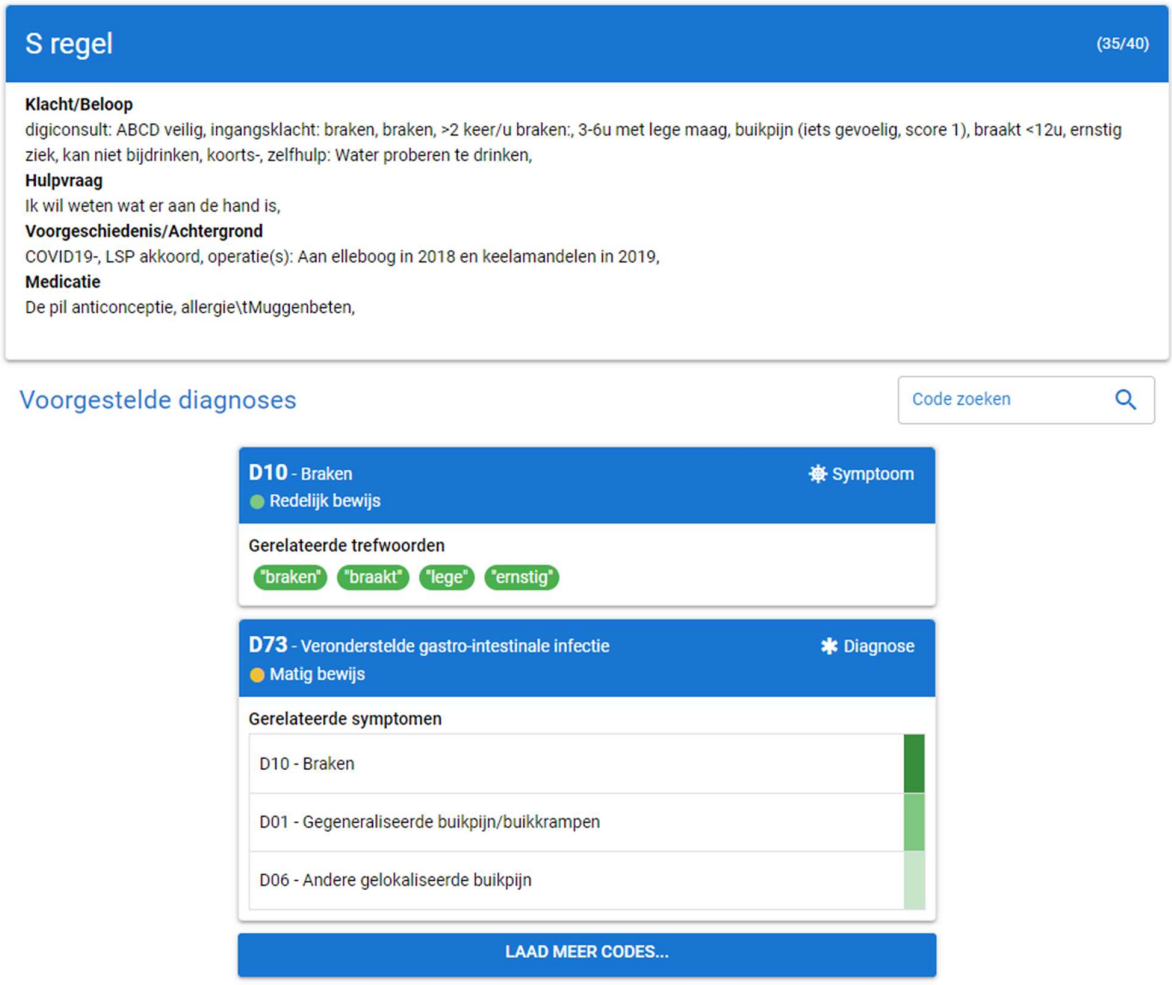
3. Vindt u de **gerelateerde trefwoorden** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☒	☐

4. Zou u “S13 - Beet mens/dier” een adequate code vinden?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

Casus 10


S regel
(35/40)
Klacht/Beloop
digiconsult: ABCD veilig, ingangsklacht: braken, braken, >2 keer/u braken:, 3-6u met lege maag, buikpijn (iets gevoelig, score 1), braakt <12u, ernstig ziek, kan niet bijdrinken, koorts-, zelfhulp: Water proberen te drinken,
Hulpvraag
Ik wil weten wat er aan de hand is,
Voorgeschiedenis/Achtergrond
COVID19-, LSP akkoord, operatie(s): Aan elleboog in 2018 en keelamandelen in 2019,
Medicatie
De pil anticonceptie, allergie\tMuggenbeten,
Voorgestelde diagnoses
Code zoeken
D10 - Braken
Symptoom
Redelijk bewijs
Gerelateerde trefwoorden
"braken"
"braakt"
"lege"
"ernstig"
D73 - Veronderstelde gastro-intestinale infectie
Diagnose
Matig bewijs
Gerelateerde symptomen
D10 - Braken
D01 - Gegeneraliseerde buikpijn/buikkrampen
D06 - Andere gelokaliseerde buikpijn
LAAD MEER CODES...

Vragen casus 10:

1. Welke ICPC code vindt u het meest passend (een code die hier niet bijstaat mag ook)?

Click or tap here to enter text.

2. Vindt u de **voorgestelde diagnoses** passen bij de s-regel?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

3. Vindt u de **gerelateerde symptomen** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

4. Vindt u de **gerelateerde trefwoorden** passen bij de voorgestelde diagnoses?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☒	☐

5. Zou u “D10 - Braken” een adequate code vinden?

Helemaal oneens	Oneens	Neutraal	Eens	Helemaal eens
☐	☐	☐	☐	☐

Algemene vragen

1. Zouden de voorgestelde diagnoses in huidige vorm u kunnen helpen sneller een juiste diagnose te vinden (waarom wel/niet)?

Click or tap here to enter text.

2. Zouden gerelateerde symptomen bij diagnoses in huidige vorm u kunnen helpen sneller een juiste diagnose te vinden (waarom wel/niet)?

Click or tap here to enter text.

3. Zouden gerelateerde trefwoorden bij symptomen in huidige vorm u kunnen helpen sneller een juiste diagnose te vinden (waarom wel/niet)?

Click or tap here to enter text.

4. Wat vindt u van de hoeveelheid voorgestelde diagnoses, symptomen en trefwoorden bij een uitleg?

Click or tap here to enter text.

5. Wat vindt u ervan dat de uitleg word getoond in de vorm van trefwoorden en relevante symptomen?

Click or tap here to enter text.

6. Wat voor controle zou u willen over de voorgestelde diagnoses (denk bijvoorbeeld aan zelf symptomen toevoegen en verwijderen of de S-regel aanpassen)?

Click or tap here to enter text.