Universiteit Leiden

Master Computer Science

Exploring Dutch BERT models for extracting lifestyle characteristics from medical text.

Name:	Hielke Muizelaar
Student ID:	s3365948
Date:	13/07/2023
Specialisation:	Data Science
1st supervisor:	Marco Spruit
2nd supervisor:	Peter van der Putten

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

The HagaZiekenhuis hospital in The Hague gathers a lot of data in order to maintain and improve treatment of their patients, including textual data in the form of unstructured clinical text. Analysing these textual data with the purpose of patient lifestyle classification is severely underanalysed. The application of lifestyle classification could aid medical professionals in identifying health risks in patient behaviour. In 2022 Heath showed that relatively non-computationally intensive methods such as string matching can obtain an accuracy of 63% on classifying discharge letters on the basis of patients' current smoking status. Furthermore, numerous research papers have been published that make use of more complex methods on unstructured text in the medical domain. The most popular trend has been applying Bidirectional Encoder Representations from Transformers (BERT) models. These deeper models are able to grasp context within sentences and can be finetuned on a multitude of tasks, including classification. In this paper, we apply BERT-like structures to clinical texts with the goal being to be able classify them on the basis of the patient's smoking status, alcohol use status and drugs use status. By performing these tests we research whether deeper transformer models can be used to improve upon string matching and classical machine learning approaches on the task of lifestyle classification in Dutch free text clinical texts.

We conducted a systematic literature review in which we explored the existing literature on BERT models used within the clinical domain, as well as Dutch BERT models. Following this literature review, we decided to train an ALBERT-like model from scratch using the full collection of input texts. We furthermore experimented with pretraining on top of three existing Dutch models; RobBERT, belabBERT and MedRoBERTa.nl. Lastly, we decided to experiment with translating input texts to English to finetune English clinical BERT models ClinicalBERT and BioBERT. For this we used the neural translation model *opus-mt-nl-en*. Our approach of translating Dutch texts to input to English BERT models can be considered a scientific novelty.

Our input dataset consists of a total of 148.000 Dutch clinical texts, ranging from consult notes to clinical letters between medical professionals. Initially, every text was labelled automatically using a string matching query on the basis of smoking, drinking and drug usage statuses. As these labels were deemed unreliable due to the constraints of string queries we labelled a set of 4.700 texts manually to serve as ground truth. Every model is evaluated on a uniform test set.

Ultimately, the model that was pretrained on top of MedRoBERTa.nl ranked first among Macro F1-scores on both the smoking (0.93) and drugs (0.77) classification tasks, whilst performing second best on the alcohol task (0.79). Interestingly, the ClinicalBERT model finetuned using translated data outperformed every other model apart from string matching on the alcohol task (0.80) and only performed slightly worse than MedRoBERTA.nl on smoking (0.92). Because of the novelty of combining neural translation with BERT classification we show that translation is a possible option to consider for future classification tasks. Furthermore, as we only translated the 4.700 hand labelled texts, higher performance can be expected when all 148.000 input texts are translated to English.

Acknowledgements

This thesis is the product of 9 months of extensive research and programming work. I would like to thank everybody who played a part in its completion. Firstly, Marco Spruit for suggesting this topic and providing the first line of supervision during the course of this thesis project. Our biweekly meetings provided me with the necessary structure and motivation to make the best out of this work and I could not have done this without you, I furthermore want to thank Peter van der Putten for accepting to be my second supervisor and for his critical view at the research methods and validity of my thesis setup, which helped me greatly in creating and executing a concrete plan. I want to thank Koert van Dortmont from HagaZiekenhuis, who delivered me the necessary data timely for successful completion of this thesis. I enjoyed our meetings concerning the ins-and-outs of the data and everything else. A big thanks as well to Jan Pronk from HagaZiekenhuis, who brought the right people together at the start of the process.
A big thank you to Sietse Muizelaar and Egbert de Vries, who accepted to be annotators for my dataset and delivered their parts at very short notice. Finally, I would like to thank anyone who displayed interest in my progress, your interest spurred me on and made me power through, especially my parents, my grandparents and my friends. Thank you all.

Contents

1 Introduction

The HagaZiekenhuis hospital in The Hague gathers a lot of data in order to maintain and improve treatment of their patients. They make use of CTcue, a data platform specifically tailored to the storage of and access to medical data. Within CTcue there exist numerous kinds of data concerning patients. These data range from personal information like full names and age to consult notes, patient-filled forms and medicine data. A lot of patient data are textual and in free-form, as medical professionals are nowadays required to take digital notes during consultation, which are saved on the platform. As the platform is mostly used to extract statistical trends on patient information, these textual data are generally severely underanalysed. As these texts could be used in many ways to increase the efficiency of medical professionals and improve patient healthcare overall, it is important to research the possibilities of utilizing textual data in a responsible, helpful way.

Numerous research papers have been published that make use of Natural Language Processing (NLP) methods on free (unstructured) text in the medical domain. Kormilitzin et al. created Med7, a transferable clinical NLP processing model specifically tailored to electronic health records (Kormilitzin, Vaci, Liu, & Nevado-Holgado, 2021). This model was firstly pretrained using a collection of 2 million free texts from patients' electronic health records and then finetuned on a named-entity recognition (NER) task. This finetuned model performed extraordinarily well on recognising the presence of seven categories concerning drug data. Other NLP methods applied to relatively highly specific medical free text classification tasks perform similarly well, such as classifying axial spondylarthritis and nonvalvular atrial fibrillation (Zhao et al., 2019; Zheng et al., 2023). It should be noted that the usage of either or combined structured and unstructured electronic health records differs greatly among papers. Some compare the use of the two separately whilst others only use one form of textual data.

NLP methods have also been used in order to extract lifestyle characteristics from (structured) electronic health records. For example, electronic health records of patients with Crohn's disease were analysed in order to find predictors for a relapse of the disease (Gomollón et al., 2021). This was done by applying a form of logistic regression to tens of thousands of independent variables in order to assess their importance in relation to each other and ranking them likewise. A similar approach has been applied to finding lifestyle risk factors that could indicate Alzheimer's (Zhou et al., 2019). Here the lifestyle habits and dietary factors of patients with Alzheimer's and unimpaired subjects were extracted from free text and compared to each other. Other studies focus on a smaller subset of characteristics and some focus on individual ones, such as smoking or alcohol usage but also factors like homelessness and sexual orientation (Palmer, Hassanpour, Higgins, Doherty, & Onega, 2019; Feller et al., 2020). Naturally, for a smaller scope the model that is trained becomes more specialized to a smaller range of outcomes, often compromising its ability to classify more general input.

In 2022, in their master's thesis that made use of discharge letters from HagaZiekenhuis, Heath showed that relatively non-computationally intensive methods such as string matching are able to obtain an accuracy of 63% on classifying Dutch discharge letters on the basis of patients' current smoking status (Heath, 2022). Their research on extracting smoking status from discharge letters from HagaZiekenhuis produced decent results by applying string matching but not with classical machine learning and deeper BERT-like approaches. To advance

research on this topic, it is interesting to examine whether a different experimental setup and the usage of different model setups could improve upon string matching within this domain. Overall, the domain of Dutch lifestyle classification on clinical texts is largely unexplored within the current literature beyond Heath's research. One earlier thesis focused on the task of extracting smoking status (Reuver, 2020), while no studies have been conducted into the classification of alcohol use and drug use statuses in Dutch clinical texts. In this thesis, we assess whether the application of more sophisticated NLP methods can provide valuable insight into whether these methods can improve upon string matching on the problem of Dutch clinical text lifestyle classification, specifically on smoking, alcohol use and the usage of drugs. We provide this angle with the goal of furthering the research in this domain.

Within this thesis, we conduct a case study using the data from HagaZiekenhuis. We set up an extensive collaboration with experts from HagaZiekenhuis in order to organise the data extraction process. Because the format and labelling of the input data posed problems for Heath within their thesis, action plans with requirements were made for data extraction for this thesis. Weekly meetings were held to discuss progress from their side and to adjust the data to best fit our research questions.

Ultimately, it was shown to be possible to extract 148.000 medical texts, which include consults and clinical letters among other sorts. Furthermore, all of these texts were labelled using string matching. The labels indicate whether the contents of the text include information that indicates the patient being a smoker, the patient drinking alcohol and the patient using (non-medical) drugs. If a text does not include smoking status a separate label is given indicating the lack of information on this topic, which also occurs in the alcohol and drug label field when there is no information on those. We test in this thesis whether these automatically assigned labels are sufficient to serve as input for our models. Because we have a relatively large amount of data, it is likely that large NLP architectures can be trained on a large chunk of the data and greatly improve upon string matching. For example, a BERT model could be trained which would be themed around Dutch medical texts and would likely be very effective at classifying medical text on the presence of lifestyle characteristics because of its depth in feature representations.

BERT models are considered to be state-of-the-art currently in the natural language processing domain and have seen success within medical domains, such as with the Med-BERT implementation on electronic health records (Rasmy, Xiang, Xie, Tao, & Zhi, 2021). They are generally trained using unsupervised free text and learn numerical vector representations of words and their context within larger text. We could therefore finetune a BERT model on the labels on smoking, alcohol and drug usage. As a BERT model would be able to grasp contexts of words within sentences, it would also be better in handling negation than string matching. This is beneficial as handling negation is a problem in the text analysis domain, especially in Dutch and medical contexts. By developing a BERT model we could therefore contribute to the field of negation handling within the domain of Dutch clinical text analysis as well as contributing to the field of Dutch lifestyle classification of clinical texts. A BERT model on Dutch medical texts could also be used for other purposes, such as classifying other aspects like using written symptoms to identify a disease, which could be very beneficial for Dutch hospitals. However, as BERT models are computationally intensive to train and finetune, it might be that more shallow, standard classical machine learning approaches are be a better

option. For the purpose of this thesis, considering its duration and in order to obtain actionable results within this time, we test multiple classical machine learning approaches. We furthermore create multiple BERT models and finetune them on the task of lifestyle characteristics classification, specifically on smoking, alcohol and drug usage. We also compare the results to the string matching from Heath's thesis and the string matching queries we used to label the entire bulk of texts. Ultimately, we provide a model that performed the best within the context of our experiments and suggest it to be used as a baseline for further research into Dutch clinical text lifestyle classification.

1.1 Project Goal

In this thesis, we focus on the problem of extracting lifestyle characteristics from free text Dutch medical texts. We propose to investigate the applicability of NLP methods that were used in similar contexts, especially classical machine learning approaches and BERT models, and our goal is to determine whether these approaches could improve upon string matching for the task at hand. The final model will need to be able to process the different writing styles between doctors adequately, as well as tackle the problem of handling negation. The overall goal is to improve upon Heath's string matching method, which is used as a baseline. Our overall goal is to further the research on Dutch free text lifestyle classification.

Whereas Heath focused solely on smoking, the lifestyle characteristics chosen here are smoking, alcohol usage and drug usage, where we create and evaluate models for each lifestyle characteristic. We conduct a case study with data from HagaZiekenhuis, recommending our best-performing model per characteristic to them for lifestyle extraction applications. The resulting models can possibly be utilized practically on medical data in order to make it easier for medical professionals to filter the respective lifestyle characteristics from doctor's notes. Our models could also later be used as a basis for further research on this topic, serving as a baseline for new models, similarly to how we use string matching as baseline. Furthermore, if our model is deemed successful and of a well enough standard there exists the option to, in consultation with HagaZiekenhuis, make our vector models available for medical scientists of other universities to do their own experiments with. Doing so would open a plethora of research possibilities and could have a significant impact on the overall research field of analysing Dutch clinical free text.

1.2 Research Questions

In accordance with the thesis goals and overall description of the thesis, we coined the following research questions for this thesis.

Main research question: To what extent can the most appropriate BERT language model improve upon classical machine learning approaches and string matching on the task of classifying patient lifestyle statuses in free text Dutch clinical notes?

To answer this research question, we tackle the following subquestions:

1. To what extent do labels obtained from string matching queries suffice as input for the task of lifestyle classification of clinical notes?

2. To what extent can the task of lifestyle classification of clinical notes be solved using string matching?

3. To what extent can the task of lifestyle classification of clinical notes be solved using classical machine learning methods?

4. To what extent can the task of lifestyle classification of clinical notes be solved using BERT-like models?

    (a) Which methods can be used to process unstructured texts within the clinical domain to serve as input for BERT models?

    (b) What are the current state-of-the-art Dutch BERT models and how can these models be evaluated?

    (c) Which BERT models have been applied within the clinical free text domain?

    (d) How can BERT models be finetuned for classification either in Dutch or within the clinical free text domain?

To answer these questions, we first explain the workings of string matching, BERT and several classical machine learning approaches in the section 3. We include the first subquestion as we want to explore whether a query can be used to label the entire bulk of texts for smoking, alcohol and drug status and whether the benefits of using such a query outweigh the benefits of labelling the texts by hand. labelling by query would take significantly less time than labelling by hand as the former process is automated.

We include questions 2, 3 and 4 because we want to test how deep transformer models perform at the task-at-hand in comparison to both string matching or classical machine learning approaches, as these methods are regarded to take up significantly less time and resources than pretraining and finetuning BERT models. It could be the case for example that the task is relatively easy in a way that using these simpler methods already achieve near-perfect performance on a uniform test set, which would render an implementation of deep neural networks like BERT redundant. For this reason we perform tests using string matching and classical machine learning approaches.

In order to explore the current literature on BERT we perform a separate systematic literature review using the SYMBALS method. In this systematic literature review we answer questions 4a to 4d.

1.3 Outline

The remainder of this thesis is structured as follows. In section 2 we lay out our approach for conducting research in this thesis. In section 3 we provide the theoretical background to this thesis, here we explore the workings of string matching, classical machine learning approaches and BERT. In section 4, we perform a systematic literature review to establish the necessary knowledge of the components of creating and finetuning BERT models. In section 5 we establish whether the use of query labels is sufficient for the task of lifestyle classification on Dutch clinical notes. In section 6 we set up the methods and experiments with string matching, classical machine learning models and BERT-like models with the goal of evaluating them on the task of lifestyle classification. In this section we also lay out our method of labelling the data

by hand and show our inter-rater agreement. Section 7 contains the results of our experiments and in sections 8 and 9 we discuss these results and provide overall conclusions.

2 Research Approach

In this section, we explain which research methods we apply in order to answer our research questions. In this thesis we are dealing with multiple goals considering multiple parties, as explained in section 1.1. In order to achieve these goals in a structured manner we make use of the CRISP-DM method, which we lay out in section 2.1. We elaborate on a high level on our data collection in section 2.2 and our computational experiments in section 2.3. We furthermore provide an overview in section 2.4 of the evaluation methods we use to evaluate the methods used to answer research questions 2, 3 and 4.

2.1 CRISP-DM

CRISP-DM (CRoss-Industry Standard Process for Data Mining) is a standard process model for data mining projects (Wirth & Hipp, 2000). We follow the CRISP-DM method as a guideline for developing the models used in this paper, meaning we do not necessarily execute all of its steps, but rather follow its general order. CRISP-DM is divided into six phases, which can be seen in figure 1. The CRISP-DM steps form an iterative process in which often there exists the possibility to move back and forth between the different process steps. The whole process can be seen as a continuous loop.

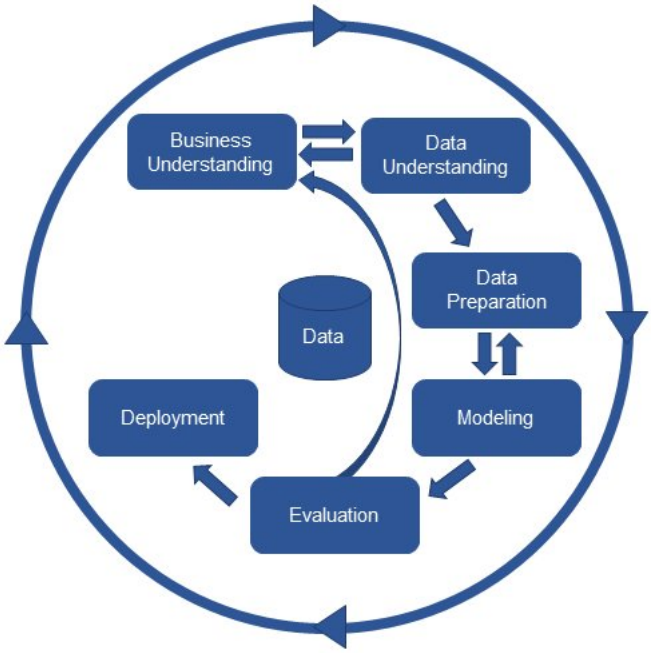


Figure 1: Diagram of CRISP-DM Processes (Wirth & Hipp, 2000)

For the first phase of business understanding we hold several meetings with HagaZiekenhuis in order to map their requirements and ours regarding the case study at hand. We furthermore gather information from literature on the models we use in order to answer our research questions. The next phase consists of using these requirements in order to extract and collect data. In our case our data consists of free-text clinical notes, and we have to make multiple considerations on the basis of data shape and query labelling in this phase.

Once we collect the data we perform analysis to assess whether the data is up to the standards of being used in a case study with the goal of furthering the domain of Dutch lifestyle classification from clinical texts. This is part of CRISP-DM's Data Understanding phase. Furthermore, this "Data Understanding" phase will see us answering research question 1, as we will assess whether applying query labelling resulted in labels that are usable for our models.

Once we establish whether to use query labels or manual labels we move towards the modelling phase, where we apply the aforementioned string matching, classical machine learning and BERT models. In this phase we alter the models in a way that they become suitable within the context of this thesis, this includes processes like parameter tuning and BERT finetuning, which indicates the synergy with the "Data Preparation" phase. We also generate our test design here and create experiments in order to be able to evaluate our models. In this evaluation phase we record performance for all of our models, answering our research questions. We also provide insights in the output of our models in order to explain where some models trump other models and what aspects make the best model better than the other models. For the deployment phase, we do not actually deploy our model due to time constraints. Instead, we propose the model that performed best on our tests for deployment to HagaZiekenhuis, as well as for the Business Understanding phase of future research into the domain of Dutch clinical text lifestyle classification.

Following our CRISP-DM approach, section 3, 4 and 5 include our Business Understanding phase, section 5 also includes our Data Understanding phase. Then, in section 6 we explore our models and tests, corresponding to the Modelling phase. Finally, in section 7 we provide an overview of the results of our tests, which we discuss in section 8 for our Evaluation phase.

2.2 Data Collection

Within the context of this thesis, it is important to note that we do not make use of structured data, such as patient information like age and family history. Our Business Understanding phase therefore consists of gathering knowledge with a free-text context in mind. We furthermore regard every document as the entire context that is available to us for the respective patient. This means we do not incorporate document history into our datasets, but rather have our dataset contain individual, standalone documents which we regard unrelated to each other. We choose to shape our data this way in order to set a clear research framework which can be replicated straightforwardly in future research. We believe replication is more straightforward this way as the availability and quality of structured patient data might vary greatly among research contexts, whereas the analysis of unstructured documents contains far fewer differences.

We collect data from the CTCure platform in accordance with experts from HagaZiekenhuis. We obtain full clinical texts to which no alterations have been made before extraction. Obtaining contextualized phrases from texts could result in valuable information getting lost, which is why we opt to obtain full texts. For each text, we also obtain labels based on smoking, alcohol use and drugs use. We want these labels to be separate from each other in order to be able to split the dataset for each respective lifestyle. The labels need to be assigned automatically, as the total amount of texts is too large to label manually. This means a query needs to be constructed beforehand that both extracts and labels each text. All in all, this results in us receiving a collection of full texts that each have been labeled on the basis of

smoking status, alcohol usage status and drugs usage status.

2.3 Computational Experiments

Within this thesis, We define the best-performing model on a lifestyle classification task to be the model that achieved the highest Macro F1-score on that respective task. We regard extracting smoking status, extracting alcohol usage status and extracting drug usage status as separate tasks, meaning we are dealing with 3 classification problems. We choose this approach as we are interested in finding the best-performing model per lifestyle, rather than averaged over all lifestyles. For example, it could be the case that string matching performs best on smoking, while a BERT model performs best on alcohol and drugs. This would mean that we recommend the respective string matching model for smoking extraction and the respective BERT model for the alcohol extraction and the drugs extraction tasks, both to HagaZiekenhuis and for future research into the domain. This way, if a model is performing the best on smoking and alcohol, but relatively badly on drugs we make sure not to recommend this model for the task of extracting drug usage. Furthermore, by regarding the smoking task separately to the alcohol and drugs tasks we can more directly compare our models to the string matching method used in (Heath, 2022).

In order to be able to answer research question 1 regarding the feasibility of query labels we conduct a feasibility study of these labels compared to manually assigned labels. The results from this study determine whether we use query or manually assigned labels further in this thesis. The further experiments include evaluating our models on a uniform test set extracted from our full dataset.

The equation for Macro F1-score is given in section 2.4. The Macro F1-score is measured on a uniform test set in order to allow for fair comparisons between models. It is important that the contents of this test set have not been seen before by any of our trainable models, as we do not want our models to simply replicate exactly what they have learned in their training phase, but rather want to test the models on their ability to classify new samples. The test set needs to be extracted from the entire dataset, where the remainder of the dataset serves as a training set. Within our literature review, we explore which ratios of splitting the dataset into training and test sets are used and apply the most commonly used split ratio within our experiments.

2.4 Evaluation Method

All of our models will be evaluated using Macro F1-score. In order to understand this metric we first lay out the concept of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) (Sharma, Chatterjee, Kaur, & Vavilala, 2022).

	Actual True	**Actual False**
Predicted True	TP	FP
Predicted False	FN	TN

Using these metrics, we can define Precision and Recall as:

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

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

The F1-score is a popular metric for evaluating performance of classification models (Lipton, Elkan, & Naryanaswamy, 2014). It is defined as the harmonic mean between precision and recall and is used often in situations where both precision and recall should be regarded as equally important. Its formula is given by

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

In this thesis, in consultation with HagaZiekenhuis, we regard every class within a lifestyle extraction task to be equally important. For example, we believe our model should be as good at finding smokers as at ruling out that a patient smokes. Furthermore, as we believe there might be a class imbalance in both the input data and within the real-life context we do not want to reward our model for performing well on a common class but not on uncommon classes. For this reason, rather than using F1-score over all entries in the test set, we calculate the mean F1-score over the classes. This technique is known as Macro F1-score and it disregards the amount of samples per class, making performance on each class equally important (Lipton et al., 2014). Its formula is given by

$$MacroF1Score = \frac{\sum_{i=1}^{n} F1Score_i}{n}$$

Here, n denotes the amount of classes.

3 Theoretical Background

In this section we lay out the groundwork for the rest of our thesis. We discuss the current literature on text classification, string matching, several machine learning approaches and BERT, as this aids us in answering our research questions for free text information extraction. This section corresponds to the "Business Understanding" phase of CRISP-DM.

3.1 Text Classification

Within the field of Natural Language Processing, which is the area of research and application that explores how computers can be used to understand and manipulate language (Chowdhury, 2003), text classification aims to assign labels or tags to textual units such as sentences, paragraphs and documents (Minaee et al., 2021) The input for text classification can be constructed either by applying manual annotation or automatic labelling to a collection of texts. Generally, text classification methods can be divided into two groups: rule-based methods and machine learning based methods that are data-driven. Rule-based methods are used to classify texts to specific categories using a series of predefined (textual) rules. These rules often need to be relatively extensive in order to grasp everything surrounding different classes and creating them requires a deep knowledge of the domain at hand. An example of a rule-based method is string matching, which we will discuss in the next section.

Machine learning based approaches differ from rule-based methods as they learn to classify text on the basis of data by learning associations between texts and respective labels. Classical machine learning methods typically follow a two-step text classification procedure. The first step entails extracting handcrafted features from the collection of input texts, which are fed to a classifier model in the second step. This model then makes a prediction about the label of the input text. Popular examples of classical machine learning methods include Logistic Regression, Naive Bayes, Support Vector Machines and Random Forests (Minaee et al., 2021), which we will explain and apply further in this thesis.

Drawbacks of the two-step procedure applied in classical machine learning are that designing the features to extract from text can be a time-consuming process due to the need for feature selection and analysis. Furthermore, these models are unable to take full advantage of large amount of training data because of the fact that features are predefined and tend to not be extensive enough to capture every edge case in a large set.

In order to handle these shortcomings, researchers have explored neural approaches that nullify the need for hand-crafted features by applying the method of word embeddings (Minaee et al., 2021). This technique maps text to low-dimensional continuous feature vectors, which are then used as input instead of hand-crafted features. Over recent years, the bidirectional transformer BERT model has been the state-of-the-art word embedding model over the last few years. BERT models have been applied extensively to text classification tasks, including within the medical domain (S. Yu, Su, & Luo, 2019; Lu, Ehwerhemuepha, & Rakovski, 2022), meaning there is a lot of information available on how to apply BERT models to contexts that are very similar to this thesis's. For this reason, we seek to explore whether BERT-like models can improve significantly over rule-based methods, like the string matching applied by Heath, and the aforementioned classical machine learning methods.

3.2 String Matching

String matching concerns using a textual query with the goal of finding a substring within a text (Heath, 2022). The simplest form of string matching involves finding exact matches, such as finding all instances of the word "dog" in a document. Implementing this method is relatively easy compared to most machine learning methods because it does not require any labeled data. On the other hand, the lack of labeled data could result in fewer "true" matches being found and/or classifying more false negatives in the text. For example, exact string matching cannot deal with spelling errors, not matching "dog" in the phrase "The *dgo* barked". To combat this issue, approximating string matching algorithms were created. These methods attempt to match a query to a sequence in a text using one or more similarity measures. Furthermore, as exact string matching can only return binary results, meaning either the string is there or not, approximate string matching can always return results by listing the most similar instances found. This way spelling errors and word variations can be accounted for.

In Heath's thesis, the exact string matching implementation achieved an accuracy of 63% on the task of classifying patients' smoking statuses from clinical notes (Heath, 2022). In this paper, we use a string matching implementation as a baseline, attempting to improve upon it using deeper, more sophisticated models.

3.3 Classical Machine Learning Approaches

Before the narrative of (medical) text classification shifted to neural approaches to speed up the process of feature engineering, several "standard" machine learning methods were applied in a widespread manner (Minaee et al., 2021). In this paper we apply four of these methods in order to build a general view on whether the task of lifestyle classification on free text clinical notes from HagaZiekenhuis can be solved using these models. If this were the case, this would mean there is no need for deeper, neural approaches like BERT. We picked these four models in particular as they are significantly different from each other in terms of setup and calculations, which we will cover in this section. They were tested against each other in a comparative study also within the medical domain (Dipnall et al., 2022). Furthermore, we apply four models as we believe less would not provide a solid enough overview of the power of classical machine learning methods and more would not fit within the time constraints of this thesis.

For each model we provide a brief overview and provide examples of the models used within the context of medical text classification. Within this section we also cover Stochastic Gradient Descent, as it can be used to optimize the machine learning models, reducing the overall loss. We furthermore cover ways to convert raw text inputs to feature vectors that are usable to serve as input for machine learning.

3.3.1 Logistic Regression

Logistic Regression is based on the logistic function. The logistic function maps an outcome variable y as a sigmoid function of predictor value x, incorporating mathematical constant e, which is the base of natural logarithms (H.-A. Park, 2013).

$$f(x) = \frac{1}{1 + e^{-x}}$$

This function has an upper limit of 1 and a lower limit of 0. By fitting the data to this logistic curve, the probability of an event happening can be estimated. Logistic Regression is generally divided into two subcategories, these being Binary Logistic Regression and Multinomial Logistic Regression (H.-A. Park, 2013). Binary Logistic Regression can be used when the dependent variable is dichotomous and the independent variables are either continuous or categorical. When the dependent variable can be one of more than two categories multinomial logistic regression is applied.

To explain how Logistic Regression function functions, we first go over Linear Regression, then explain how Linear Regression can be changed such that probabilities can be estimated. The standard equation for Linear Regression is $y = \beta_0 + \beta_1 x$ where y denotes the outcome, β_0 the intercept, β_1 the slope and x a predictor. This equation does not work for probabilities as values can go over 1 and below 0, which is unwanted behaviour. For this reason, to estimate probabilities we instead incorporate the odds of y, given that we know y must be a probability. Odds are the probability of events occurring over the probability of those events not occurring. In logistic regression, the natural logarithm of odds is used as, unlike odds, it is not restricted to being between 0 and infinity. Incorporating the log odds results in the equation $\log(\frac{P}{1-p}) = \beta_0 + \beta_1 x$. As we are ultimately interested in estimating probabilities p rather than log odds we can incorporate exponents likewise to the logistic function. This results in the following equation

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

When there are more than 1 predictors this equation changes to

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 ... \beta_n x_n)}}$$

Here, n denotes the total amount of predictors.
Within multinomial logistic regression, a probability of the input belonging to each class is given as output. Generally, the class that is predicted is the class that received the highest estimated probability. Logistic regression has seen widespread use in text classification (Hassan, Ahamed, & Ahmad, 2022). In order to be able to use logistic regression on textual data, words are converted into TF-IDF vectors. We explain this method in section 3.4.

Within classification, the function that measures the distance between the predictions of the respective models and the actual outcomes is called the loss function. Naturally, we want to keep the loss as low as possible, as this means our model makes predictions that are as close as possible to the ground truth. For logistic regression log loss is regarded as the standard loss function and is defined by the following equation

$$J(\theta) = -[\sum_{i=1}^{m} y^i \log h_\theta(x^i) + (1 - y^i) \log(1 - h_\theta(x^i))]$$

Within this equation, for simplicity's sake, we combine every model parameter in variable θ. h_θ denotes the equation for obtaining probabilities from log odds. In order to minimize this loss function, we make use of the Stochastic Gradient Descent method, which we describe in section 3.3.5. We apply this method as it has shown promise in combination with logistic regression (Gaye, Zhang, & Wulamu, 2021; Tian, Zhang, & Zhang, 2023).

Logistic Regression has been applied extensively in text classification, and naturally in medical text classification as well (Wong & Akiyama, 2013; Shah, Patel, Sanghvi, & Shah, 2020; Boateng & Abaye, 2019; Yao, Mao, & Luo, 2019; Almazaydeh, Abuhelaleh, Tawil, & Elleithy, 2023). Input features are mostly built using the bag-of-words and TF-IDF models, which we touch upon in section 3.4.

3.3.2 Multinomial Naïve Bayes

Naive Bayes is a learning algorithm that has been employed frequently to tackle text classification problems (Kibriya, Frank, Pfahringer, & Holmes, 2004). It is regarded to be computationally very efficient and easy to implement. Furthermore, Naive Bayes provides robust results when there is a relatively low amount of available data and/or there is low variance amongst the data. The Naive Bayes classifier is based on the Bayes Theorem of Probability. This theorem is based on conditional probability, which answers the question: "How does the probability of an event change if we have extra information?" (Teitelbaum, 2022). Let us say there are two events, event *A* and event *B*. Conditional probability of *A* knowing that *B* occurred is written as $P(A|B)$. The conditional probability of *A* given *B* can be defined as:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Provided that $P(B) \neq 0$.
Bayes' theorem inverts this conditional probability, meaning it finds $P(B|A)$ from $P(A|B)$. This is defined as:

$$P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$$

Bayes' theorem is distinguished by its use of sequential events, where additional information that was acquired later impacts the initial probability (IBM, n.d.). These probabilities are known as the prior and posterior probabilities, where the prior probability $P(A)$ denotes the initial probability of an event without context and the posterior probability $P(B|A)$ as the probability of an event after observing a sequence of data.

Naive Bayes classifiers apply Bayes' theorem while holding the key assumption that every predictor is completely conditionally independent from any other feature in the model. This assumptions results in these classifiers being named "naive". This makes the classification problem computationally light, as each variable in the model only requires one probability.

Given that we have a vector of features $X = (x_1, x_2, ..., x_n)$ and a class variable y. By applying Bayes' theorem we can calculate the posterior probability of y given X, $P(y|X)$, from the likelihood of X over y, $P(X|y)$, and the prior probabilities $P(X)$ and $P(y)$. Normally, to be able to define $P(X|y)$ we would need to apply the chain rule and include every probability of every individual feature given every other individual feature, which would be computationally expensive and grows exponentially with feature vector length. However, because of the conditional independence assumption, we would only need to include the probability of every feature given outcome y and leave out the probabilities of the individual features given every other individual feature. This means we can define $P(X|y)$ as

$$P(X|y) = P(x_1|y) \cdot P(x_2|y)...P(x_n|y)$$

And therefore $P(y|X)$ can be defined as

$$P(y|X) = \frac{P(x_1|y) \cdot P(x_2|y)...P(x_n|y) \cdot P(y)}{P(x_1) \cdot P(x_2)...P(x_n)}$$

Here, the denominator is constant for all outcomes, as only the probabilities of the features are included. This means we can define the posterior probability to be

$$P(y|x_1, x_2, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

With this, for every possible outcome y a probability can be determined. Most commonly, the classifier chooses to classify the feature vector based on which outcome yields the highest probability.

Naive Bayes classifier models can generally be divided into multivariate Bernoulli models and multinomial models. The multinomial models generally outperform multivariate ones (Kibriya et al., 2004) and have also been found to perform comparably to more specialized models. Within the context of text classification, Multinomial Naive Bayes calculates the probability of a document d belonging to class c, $P(c_k|d_j)$ with the following equation (Kibriya et al., 2004)

$$P(c_k|d_j) = P(c_k) \prod_{i=1}^{k} P(t_i|c_k)$$

Here, corresponding to Naive Bayes, $P(t_i|c_k)$ denotes the probability of word t_i appearing in class c_k. $P(c_k)$ denotes the prior probability any document appears in class c_k. $P(c_k)$ and $P(t_i|c_k)$ are computed based on the training data. The "multinomial" part stems from that we assume the probabilities for each class follow a multinomial distribution, which is known to work well for data which can be turned into counts, such as word counts in text.

Within the medical domain, Multinomial Naive Bayes has been applied to several text classification tasks, such as to classifying sentences from biomedical texts into rhetorical categories (Agarwal & Yu, 2009), classifying external-cause-of-injury-codes (Nanda, Vallmuur, & Lehto, 2018) and automating the tracking of global health spending (Dixit et al., 2022).

3.3.3 Support Vector Machine

Support Vector Machine, often abbreviated to SVM, is an originally binary classification algorithm (H. Wang, Xiong, Yao, Lin, & Ren, 2017). It attempts to map non-linear vectors to a high-dimensional feature space with the goal of constructing a linear hyperplane that serves as a divider between classes. Finding this divider is a problem that seeks to find a plane in the feature space that divides the classes with the maximal margin size, meaning the largest possible distance between itself and the nearest points of each class. SVM is intuitively easy to understand when it concerns two classes that are linearly separable. In this case, we can define the optimal hyperplane as

$$w^T x + b = 0$$

in which w refers to the weight vector, x the x-coordinate in the feature space, and b the threshold. In order to find important points of both classes this equation is shifted 1 to the left

and right. Input vectors that lie within this margin are regarded as support vectors, meaning that when these points were to change position the position of the entire hyperplane would shift. Then, we can maximise the separation gap by incorporating the length of the weight vector via $\frac{2}{||w||}$, often written as $\min(\frac{1}{2}||w||^2)$. This (constrained) optimization problem can be solved via the Lagrangian multiplier method, for which we refer to other sources (H. Wang et al., 2017; Balasundaram & Kapil, 2010) for its implementation. SVMs can also be altered to classify classes that are not linearly separable. For this the data is approximated via more functions, increasing dimensionality and making the classes linearly separable again. These added functions are known as kernel functions. There are multiple different kernel functions and the choice of which one to apply in specific contexts depends on the characteristics of the data.

SVMs have seen widespread use in medical text classification (Matwin & Sazonova, 2012; Farhadian, Shokouhi, & Torkzaban, 2020; S. Huang et al., 2018). Applications include learning cancer genomics, diagnosis of periodontal disease and medical abstract classification among many others. In this thesis, we optimize the loss function using stochastic gradient descent, which has shown promise when applied to SVMs (Z. Wang et al., 2018; Mutlu & Acı, 2022).

3.3.4 Random Forest

Random Forests are ensemble models, meaning they use multiple classifiers in order to make predictions (Biau, 2012). In particular, Random Forest consists of a collection of tree-structured classifiers, otherwise known as decision trees. Decision trees can model arbitrarily complex relations between inputs and outputs. They consist of nodes that form a rooted tree, meaning a root node is present with no incoming edges (Rokach & Maimon, 2005). All other nodes have exactly one incoming edge. A final node with no outgoing edges is known as a decision node or leaf and it determines the predicted class when reached. Within a decision tree, each internal node splits the instance space into two or more sub-spaces according to a discrete function that incorporates the values of the input attributes. Input instances are classified by navigating them from the root of the tree down to a leaf, using the discrete functions of the internal nodes to take paths to other internal nodes.

Within Random Forests, each decision tree in the collection is formed by first selecting a small group of input coordinates at random at each node to split on (Louppe, 2014). Secondly, the best split is calculated on the training set on the basis of these features. It uses L tree-structured base classifiers $h(X, \Theta_n), N = 1, 2, 3, ...L$, where X denotes the input data and Θ_n is a family of identical and dependently distributed random vectors. Each tree in Random Forest is grown by firstly sampling N amount of cases from the training set, N is set beforehand and the resulting set is used as the training set for the trees of the respective iteration (Louppe, 2014). Then, for a prespecified M number of input variables, we select an m amount of variables at random and use each of these variables to construct a tree. Each tree contains the best split for that particular variable. By using a collection of independently trained decision trees that are uncorrelated to each other, drawbacks of decision trees like bias and overfitting are accounted for, making the model more accurate than using a singular decision tree.

Within text classification, like is the case with many other models, Random Forests are often

applied to TF-IDF input features generated from text (Jalal, Mehmood, Choi, & Ashraf, 2022; Islam, Liu, Li, Liu, & Kang, 2019), which we explain in section 3.4.

3.3.5 Stochastic Gradient Descent

Gradient Descent is a technique which is mostly used for optimization in machine learning (Tian et al., 2023). It learns the optimized values of the parameters of a model at each iteration in order to minimize a cost function. Given training data (x_i, y_i), where x is the input data and y the outcome, the general form of the optimization problem is

$$\min_w \ell(w) := \frac{1}{n} \sum_{i=1}^{n} \ell_i(h(x_i; w), y_i),$$

Here, w denotes the weights of the model, h the prediction function and ℓ_i the loss function. We can furthermore define ζ as a random seed which is used to represent a sample or group of samples and f as the combination of the loss and prediction functions, meaning we can rewrite the loss to be $f(w, \zeta)$ for every given pair (w, ζ).

The gradient descent technique computes the gradient of the loss function in order to update the weights of the model. A learning rate η is used in order to control the impact of every iteration. The goal of gradient descent is to reach a level for the parameters from which a step in any direction results in a higher loss, meaning we converge to a local minimum of the loss function. Batch Gradient Descent, also known as Complete Gradient Descent, uses all available samples to update the parameters until convergence. It performs a series of iterations until this convergence is accomplished. An iteration is defined as:

$$w^+ = w - \frac{\eta}{n} \cdot \sum_{i=1}^{n} \bigtriangledown f(w; \zeta_i)$$

Here w denotes the result of the previous iteration and the $\bigtriangledown$ symbol denotes that the gradient of $f(w; \zeta_i)$ is calculated, which we sum. One cycle of passing through the entire dataset is called an epoch and batch gradient descent stops training at the end of all training epochs, which is specified beforehand.

Stochastic Gradient Descent is a variant of Gradient Descent that does not use the full bulk of samples per iteration, but rather uses only one random sample. This means that instead of computing the gradient of $1_n \sum_{i=1}^{n} f_i(w; \zeta_i)$ exactly, we use one random sample to estimate gradient ζ_i of $f_i(w; \zeta_i)$ in each iteration. By using a random sample rather than the entire batch redundant calculations of related samples relative to the current sample are left out, which saves memory and makes Stochastic Gradient Descent easier to use.

3.4 Bag-of-Words and TF-IDF

During this literature review, we came across multiple methods for transforming text into feature vectors which can be fed into machine learning models. In this section we lay out the most popular method, bag-of-words with TF-IDF.
Bag-of-Words, commonly referred to as BoW, is a simplified text representation model often used in natural language processing (Qader, Ameen, & Ahmed, 2019). In this model, texts like sentences or documents are represented by their word frequencies. The model maps a

document to a vector as $v = [x_1, x_2..., x_n]$, where x_i denotes the occurrence of the ith word in basic terms (Qader et al., 2019). These basic terms are collected from the datasets, which are usually the top n highest frequency words. Word frequencies can either be binary, meaning the word is present (1) or not (0) or based on their term frequency (TF), meaning the amount of times the word shows up in the document. Although high-frequency words might give a representative idea about the document, there are a lot of words that occur frequently but do not convey much meaning, such as 'the' and 'and', these words are called 'stopwords'. For this reason, the term frequency is supplemented with the inverse document frequency, which include how many documents in total include the term. By dividing the term frequency by the inverse document frequency, the importance of stopwords is nullified and a higher importance is given to words that occur often in the document but not often in the grand scheme of all documents (Sarica & Luo, 2021). This process is known as TF-IDF. Within TF-IDF, term frequency is not only the amount of times a word occurs in the document but is rather the relative frequency to the total amount of unique words, given by the following equation

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

Here, $f_{t,d}$ is the number of times the term t occurs in document d. The denominator is the total amount of terms in the document. The inverse document frequency is given by the next equation

$$idf(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Here, N denotes the total amount of documents in the corpus D. $|\{d \in D : t \in d\}|$ denotes the number of documents in which term t appears. As the denominator can be equal to 0 when the term does not occur in the corpus, generally this is solved by changing the denominator to $1 + |\{d \in D : t \in d\}|$.
TF-IDF is then defined as the product of the two equations above. Generally, the usage of TF-IDF in BoW outperforms the usage of word frequencies alone (Abid et al., 2022; Akuma, Lubem, & Adom, 2022). For this reason, in order to transform text to useful input features for our machine learning approaches, we apply TF-IDF in this thesis.

3.5 Bidirectional Encoder Representations from Transformers (BERT)

BERT is a language model which aims to learn deep bidirectional language representations using unlabeled text data (Devlin, Chang, Lee, & Toutanova, 2019). The bidirectionality refers to the context which is learned both backwards and forwards from parts of each sequence of text. Within BERT, a sequence refers to an input token sentence stemming from using embeddings from the input text using the WordPiece subword segmentation algorithm. This algorithm creates a vocabulary of tokens by starting with every individual character in the alphabet and then iteratively adding the most frequent combinations of character pairs, then triads and so on until a predefined limit of characters is reached. The usage of embeddings in favour of plain text makes it easier for machine learning to learn context, as words are converted into feature vector representations. This means semantically similar inputs appear closer to each other than dissimilar inputs within the embedding space. In BERT, the first token of each sequence is always a special classification token *[CLS]*. Another special token *[SEP]* represents a separator between sentences. The rest of the tokens are words or parts of words. Tokens

that do not appear in BERT's 30.000 sized token vocabulary are split into smaller parts named subwords. This way obscure words can still be learned. Consider a vocabulary only including the tokens 'work' and '##ing', where '##' represents any subword. In this scenario the word 'working' will be tokenized into "work" and "##ing" separately. BERT's maximum sequence length is 512 tokens, often spanning multiple sentences. Next to the embeddings of these tokens in a sequence, BERT's input consists of segmental and positional embeddings, see figure 2.

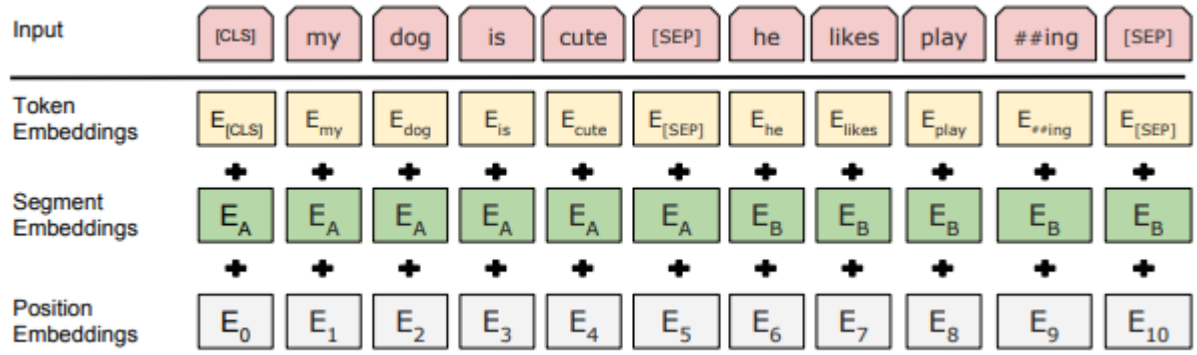


Figure 2: Embeddings that make up BERT's input.

Segmental embeddings are embeddings based on which sentence the respective token belongs to, as can be inferred from the image. Lastly, positional embeddings capture the order of the tokens in the sequence. To build BERT's input, these embeddings are summed.

With this input, BERT can start its (pre)training phase. This phase consists of two tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). Before the introduction of bidirectional language models, conditional language models were mostly learned from left-to-right, meaning valuable information was lost as context of previous words was not learned. Merely adding a right-to-left model would not yield acceptable results, as this would allow each word to include itself in its context, making it trivial for a machine learning method to predict the word, its input already includes the desired output meaning the model does not learn anything. In BERT this problem is tackled by incorporating masked tokens. This entails that, at random, 15% of the WordPiece tokens in BERT's input sequence are converted to masked tokens. The MLM objective is to predict which word belongs to this masked token, which is normally denoted as '*[MASK]*'. As *[MASK]* only shows up during pretraining and not during the finetuning a mismatch between the two phases could arise. To mitigate this, the authors of BERT chose to have the 15% of the tokens either masked to *[MASK]* with a probability of 80%, a random token at 10% and to not change it at all also at 10%. Then, the objective is to predict the original token, with the goal of minimizing the cross-entropy loss.

The NSP pretraining task was added to benefit downstream tasks as Question Answering and Natural Language Interference, which profit from understanding the relationship between sentences. This task is binarized, for every pretraining example containing consecutive sentences *A* and *B*, 50% of the time sentence *B* is the actual next sentence from *A*, and 50% of the time the sentence is a random sentence sampled from the corpus. The goal is to minimize the classification loss of predicting whether each sentence is supposed to follow its predecessor.

BERT's model architecture can be described as a multi-layer bidirectional transformer encoder. A general transformer includes two separate mechanisms, an encoder which processes the text and a decoder which produces predictions. As the goal of BERT is to generate a language model and not to perform predictions it does not include a decoder. As described previously, the encoder reads the entire sequence at once so that all of the surrounding words of a word are included in that word's context. The amount of transformer blocks, or layers, in BERT depends on the size of the model. The $BERT_{BASE}$ version includes 12 layers, with 768 hidden units per layer and 12 attention heads. The authors furthermore introduce a bigger $BERT_{LARGE}$ model with 24 layers, 1024 hidden units per layer and 16 attention heads, this model has above 3 times as many parameters as the base model. BERT is pretrained on the BooksCorpus, which contains English novels, and a corpus of text from English Wikipedia, combining into a total of 3.3 billion words.

In this thesis we are dealing with the task of lifestyle classification. Finetuning BERT for classification is done by adding an output classification layer on top of the BERT encoder. Then, the entire model is trained end-to-end with its own custom loss function and optimizer.

Heath compared string matching to BERT on the task of classifying smoking status from clinical notes (Heath, 2022). In particular, the smaller, distilled Dutch model BERTje (de Vries et al., 2019) was used as there was no access to the necessary computing power to train a model within their time constraints. As they were dealing with a severe lack of input data they were unable to finetune BERTje and opted to pretrain on top of BERTje instead with the goal of increasing the domain-specific knowledge of the model. These factors most likely massively contributed to string matching severely outperforming BERT, with string matching achieving an accuracy of 63% and BERTje only achieving an accuracy of 37%. For this reason, it is very interesting to explore whether the usage of general BERT models in favour of distilled models, actually finetuning the model on more training data and having access to more computing power could result in BERT outperforming exact string matching on this task.

4 BERT Systematic Literature Review

We aim to explore BERT in such a way that we are able to apply it to our task of lifestyle classification. This corresponds to research questions 4a to 4d. For the continuation of our Business Understanding phase of CRISP-DM we conduct a review of the current scientific literature on BERT. We make use of the SYMBALS method for systematic reviews. Our literature review can therefore be regarded as a systematic review, albeit some parts were cut to fit the context and constraints of our thesis.

Within this literature review we aim to explore the form and function of BERT models, as well as how they adapt to different types of inputs and how they can be finetuned to classification tasks within the clinical free text domain. As BERT is 4 years old at the time of writing we also explore alterations made to its architecture that improve performance. These alterations could be on the amount of layers or model parameters.

We furthermore want to discover which Dutch BERT models exist in literature and how a BERT model can be adapted to our Dutch input text. We compile results in two tables, one with source papers introducing deeper context models and one with papers applying these models on classification tasks. From these tables we will further infer methods for data alteration, data verification, model pretraining and finetuning, as these methods will aid in answering the research question.

Within the context of this thesis we apply SYMBALS in a manner that is similar to how it is applied in its source paper, with the goal of extracting information that can help answer several research questions. In this section we first go over the workings of SYMBALS, then give an elaboration of the execution of SYMBALS within this thesis for each step.

4.1 SYMBALS overview

To obtain the information necessary to be able to answer research questions 4a to 4d we firstly need to evaluate the existing literature on relevant topics. In order to cover the existing research in an efficient, effective way we conduct a literature review. This literature review can be regarded as a systematic review. A systematic review seeks to collate evidence which fits prespecified eligibility criteria. The goal of doing a systematic review is to answer a specific research question using existing literature (Uman, 2011). Generally, a systematic review consists of framing a research question, identifying relevant work, assessing quality of the work, summarizing evidence and interpreting findings.

To answer the research questions a query needs to be constructed which yields potentially relevant results. It is very likely that not every result is topical and interesting enough in regards to the research questions. This means we are in need of a method that can filter through results with the goal of reduction and quality protection. For this reason we make use of SYMBALS. SYMBALS blends the traditional method of backward snowballing with the machine learning method of active learning (van Haastrecht, Sarhan, Ozkan, Brinkhuis, & Spruit, 2021). Active learning is a machine learning method where an algorithm chooses relevant data points to learn from, with the goal of finding the most informative samples. This is especially helpful within the quality assessment phase of systematic reviews, as reviewers tend to not

desire to evaluate every identified paper due to time constraints. By having active learning choose potentially relevant papers, papers that are likely irrelevant can be discarded quickly, sharply reducing the time needed for quality assessment. In the context of SYMBALS, this comprises of constructing a dataset containing at least the titles and abstracts of a collection of papers.

The initial sample fed to the active learning algorithm needs to include at least one relevant and one irrelevant paper. The initial sample should also not be too large, as there is no time advantage to be gained in this stage. After the algorithm has learned from the initial samples it will put out the deemed most informative paper first, which the researcher judges as being relevant or irrelevant. This judgment is fed back to the algorithm so it can adjust its output to it. It is deemed difficult to choose an appropriate active learning stopping criterion, which denotes when enough relevant papers have been found and the active learning phase is considered to be completed.

The stopping criterion is denoted as p and, in the original paper, is left to the desk researchers to decide with the details of the relevant thesis in mind. The idea is that we stop once a percentage of the estimated number of relevant papers R has been marked relevant, given the formula below.

$$R \simeq N \times \frac{r}{r+i}$$

Here, R denotes the estimated number of relevant papers, N the total number of papers, r the amount of relevant papers found at that particular stage and i the amount of irrelevant papers found at that stage.

Within SYMBALS active learning is supplemented with the process of backwards snowballing. Backwards snowballing entails obtaining potentially relevant sources from the reference lists of sources obtained by applying active learning. By including this backwards snowballing step, so-called 'grey' literature is more likely to be found (Uman, 2011). This includes older, more fundamental papers which can provide more understanding on the workings of specific methods, as they are often the papers that introduce these methods.

Figure 3 shows an overview of the method. SYMBALS was shown to accelerate the process of title and abstract screening by a factor of 6. The authors of SYMBALS also showed that it was able to outperform the state-of-the-art method FAST in three benchmarking experiments, showing that the inclusion of backwards snowballing increased recall in finding relevant results by as much as 10 %. Only on a dataset including systematic reviews SYMBALS underperformed, likely because systematic reviews contain a deeper information pool than regular research papers.

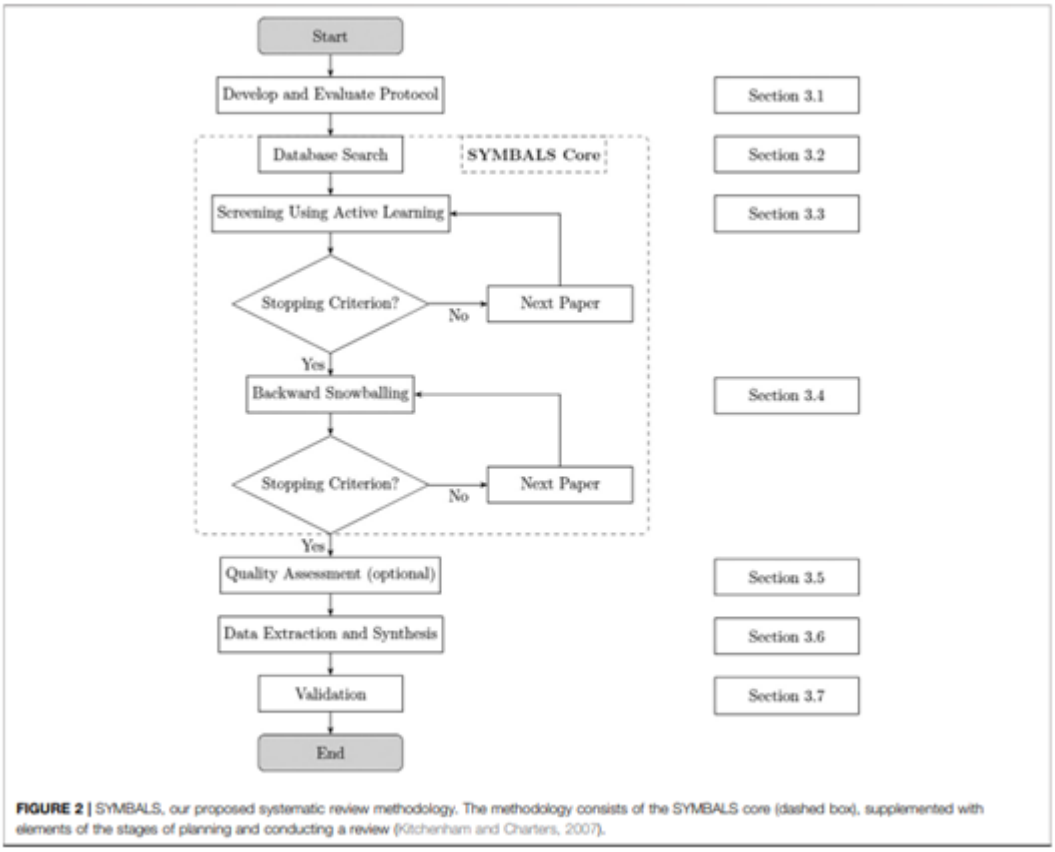


Figure 3: The SYMBALS workflow (van Haastrecht et al., 2021).

After obtaining search results using a search query the process of active learning is applied to the screening of the results. In the context of SYMBALS, this comprises of constructing a dataset containing at least the titles and abstracts of a collection of papers. The initial sample fed to the active learning algorithm needs to include at least one relevant and one irrelevant paper. The initial sample should also not be too large, as there is no time advantage to be gained in this stage. After the algorithm has learned from the initial samples it will put out the most informative paper first, which the researcher judges as being relevant or irrelevant. This judgment is fed back to the algorithm so it can adjust its output to it. It is deemed difficult to choose an appropriate active learning stopping criterion, which is another reason to include backwards snowballing.

The stopping criterion is denoted as p and, in the original paper, is left to the desk researchers to decide with the details of the relevant project in mind. Within SYMBALS backwards snowballing results in the stopping criterion being recommended to stop when in the last N_r references the number of new relevant additions r_r is less than some constant C, given that the number of snowballed papers is at least S. For example, if the set of inclusions contains 100 papers and the minimum number of papers to snowball S is 10, once 10 papers have been snowballed we stop when the last $N_r = 100$ references obtained using backwards snowballing contained less than $C = 5$ additions to our inclusions.

Following the backward snowballing phase, quality assessment is performed if deemed necessary within the context of the relative research project. This phase generally consists of comprising quality criteria based on reporting, rigour, credibility and relevance. Similarly to active learning, machine learning techniques can be applied to speed up this process. Last

steps include analyzing the prior processes to assess whether bias is present and whether the obtained results are sufficient to be able to answer the research question at hand.

4.2 Protocol

The objective of this systematic review is to explore existing literature concerning deep machine learning techniques applied to free text data, mainly Bidirectional Encoder Representations from Transformers (BERT) models and preferably applied to Dutch and the medical domain. The ultimate goal is to apply the methods which are deemed to be the best fitting to the context of our thesis, which is lifestyle classification applied to Dutch clinical unstructured text. For this, we set the following research questions, which correspond to research subquestions 4a to 4d in section 1.2:

4a. How do BERT models function and adapt to different types of inputs?

4b. Which Dutch BERT models exist in literature and how can they be evaluated within the context of this thesis?

4c. Which BERT models have been applied within the clinical free text domain?

4d. How can BERT models be finetuned to do classification either in Dutch or within the clinical free text domain?

In correspondence with these research questions, after obtaining a collection of relevant documents using SYMBALS, we will compile two result tables which we will base the rest of the research thesis upon. These result tables consist of a table showing papers with candidate architectures for the (deep) models used in lifestyle classification and a table showing papers with candidate methods for classification.

We decided on incorporating 4 major search engines for scientific papers. These are PubMed for its collection of biomedical literature and the ACM Digital Library, IEEE Xplore and Scopus libraries for their extensive collections of papers in a wide range of scientific fields. An overview of the query we use to search these libraries can be found in section 4.3.

4.3 Database Search

In order to be able to answer the research questions we constructed the query below, which is in generalized syntax for the purpose of illustration. The query consists of two larger parts and the second part consists of three subparts. Every individual part is numbered and the reason for their inclusion is elaborated on below the query.

(1) "*RobBERT*" OR "*BERTje*" OR *belabBERT*
OR
(
(
(2.1) "*BERT*" OR "*RoBERTa*" OR "*transformer*" OR
"*bidirectional encoder*"
)
AND

(
(2.2) "*medical*" OR "*health*" OR "*clinical*" OR "*lifestyle*" OR "*life style*" OR "*life-style*" OR "*diagnosis*" OR "*dutch*"
)
AND
(
(2.3) "*free text*" OR "*free-text*" OR "*unstructured*"
)
)

In this query:

1. We chose to separate *RobBERT*, *BERTje* and *belabBERT* as they are Dutch BERT models. As we are dealing with Dutch input data in this thesis, we want to find every paper using one of these models in order to build a view on how to handle Dutch text and which models are used on Dutch text. This means every available paper that has the name of these models in either its title or abstract will be included in the result set.

2. 2.1. If a paper does not include one of the Dutch BERT models we want to make sure that it does at least contain a transformer-like model, preferably BERT. For this we are using two popular names for general BERT structures, this being the standard BERT, and the Robustly Optimized BERT pretraining approach: RoBERTa. We also add "bidirectional encoder" and "transformer" to this subquery as we are also interested in structures that are older and/or less complex than BERT and which might include interesting methods and models for classification.

   2.2. Next to the model architecture, we want to make sure that the paper is set either within the medical domain, makes use of lifestyle in any way or concerns Dutch data. As we consider these aspects to be equal in value in light of the research question they are grouped together.

   2.3. If we do find papers using subqueries 2.1 and 2.2 we want to make sure they make use of free, unstructured text. By excluding papers that focus on structured text we limit the amount of results in a way that the total amount of papers becomes more manageable within time constraints, furthermore we increase the chance of the paper being relevant as we are using free text in this thesis.

Using this query, we extracted 98 papers from PubMed, 38 papers from ACM DL, 32 papers from IEEE Xplore and 55 papers from Scopus, totalling 223 papers. These papers serve as our first batch and will be fed into the active learning phase. Here we decide whether each of these papers is relevant within the context of this thesis.

4.4 Active Learning

We evaluate papers on only the title and abstract of the source. In order to be able to evaluate whether each paper is relevant within the context of this thesis we formulated the criteria that are stated below. These criteria serve as relevancy guidelines in both the active learning and backward snowballing phases.
We consider a source to be relevant when:

a) The source uses one of the Dutch BERT models in any sense.
   As mentioned previously, we are interested in how Dutch BERT models function and how they handle Dutch input data, therefore each source that uses a Dutch BERT model in any sense is deemed relevant.

b) The source makes use of English BERT models within the context of unstructured medical texts relating to patients, preferably also incorporating finetuning and classification/prediction tasks.
   We also want to make sure that, due to BERT's broadness of application, we only deem a source that uses BERT relevant when the source has a similar context to ours.

c) The source is **not** solely a named entity recognition (NER) task. The only exception being if the NER task is merely a part in a whole classification task.
   Relatively many papers apply BERT to an NER-like task. As we want to limit the amount of papers to fit time constraints and to minimize redundancy we exclude papers that use BERT merely this way. This creates more room for papers that use classification, which we deem more fitting.

d) The source compares BERT against other (deep) methods, given that the task at hand is one of classification/prediction.
   We consider comparative studies between BERT and other models relevant when they are compared on a classification or prediction task.

e) The source is a standalone paper, no systematic review nor a (full) book.
   We exclude these kind of sources as they are too broad to analyse within the constraints of this literature review.

f) The source is an original source paper on an English or Dutch BERT model that could be useful to train and evaluate our future model with.
   If the paper introduces a BERT model, either in Dutch or English, we consider it to be relevant by default as the paper could aid us in possibly pretraining our own BERT-model.

g) The source proposes a new way to alter the BERT architecture in an attempt to improve its performance.
   We furthermore consider a source to be relevant when it introduces alterations to BERT and shows that these alterations result in a higher performance.

Next to NER tasks, Relation Extraction and Phenotyping tasks are prevalent amongst the result papers. We only consider these sources to be relevant if elements of classification and or prediction are present in the title or abstract in order to keep the amount of papers manageable within this thesis project.

We reviewed 20 papers in order to define a stopping criterion. From these 20 papers, 5 were deemed relevant based on the criteria above, which equates to 25%. Then, using the active learning stopping criterion formula from section 4.1, we estimated the total amount of relevant papers to be 56. We defined the stopping criterion as us finding 100% of these relevant papers, as we deemed the total of 223 abstracts to be manageable to evaluate. We ultimately end up with 55 relevant papers, just over a 75% reduction, which we use in the backward snowballing phase.

4.5 Backward Snowballing

We perform backward snowballing by evaluating the relevance of each reference in a paper that was deemed relevant. For the 55 relevant papers we have at this stage, after 5 papers have been reviewed, we define the stopping criterion p as the last 100 references containing less than 5 unique additions to the inclusions. We use the same relevancy criteria as stated in the previous section. The order of the papers that get snowballed is determined alphabetically on the basis of the name of the authors. Applying backward snowballing resulted in an additional 610 titles/abstracts getting reviewed, which culminated in 32 unique relevant additions, totalling 87 relevant papers overall. An overview of the backward snowballing process can be found in table 1.

Paper number	**Paper name**	**Amount of references**	**Amount of relevant papers**	**Amount of unique additions**	**Amount of unique additions since paper 5 ratio**
1	Identification of asthma control factor in clinical notes using a hybrid deep learning model	35	5	5	-
2	Analysis of Language Embeddings for Classification of Unstructured Pathology Reports.	21	4	1	-
3	MLT-DFKI at CLEF eHealth 2019: Multi-label Classification of ICD-10 Codes with BERT	40	2	0	-
4	Classification of Medical Image Notes for Image labelling by Using MinBERT	46	4	1	-
5	Natural Language Processing Model for Identifying Critical Findings-A Multi-Institutional Study	14	2	2	- (start counting)
6	Highly accurate classification of chest radiographic reports using a deep learning natural language model pre-trained on 3.8 million text reports.	27	3	1	1/27
7	Emotional RobBERT and Insensitive BERTje: Combining Transformers and Affect Lexica for Dutch Emotion Detection	33	5	2	3/60
8	A Pre-trained Clinical Language Model for Acute Kidney Injury	8	4	1	4/68
9	Multi-label classification of symptom terms from free-text bilingual adverse drug reaction reports using natural language processing.	32	4	1	5/100 (restart)
10	Predicting Postoperative Mortality With Deep Neural Networks and Natural Language Processing: Model Development and Validation	36	2	0	0/36
11	Automatic ICD-10 Coding and Training System: Deep Neural Network Based on Supervised Learning.	20	5	1	1/56
12	Transformer-based models for ICD-10 coding of death certificates with Portuguese text.	44	7	5	6/100 (restart)
13	Multi-label Classification for Clinical Text with Feature-level Attention	31	1	0	0/31
14	RobBERT: a Dutch RoBERTa-based Language Model	51	8	5	5/82 (restart)
15	RobBERTje: a Distilled Dutch BERT Model	38	10	6	6/38 (restart)
16	Comparison of state-of-the-art machine and deep learning algorithms to classify proximal humeral fractures using radiology text.	44	3	0	0/44
17	Extracting clinical named entity for pituitary adenomas from Chinese electronic medical records.	28	2	1	1/72
18	Natural Language Processing for Automated Classification of Qualitative Data From Interviews of Patients With Cancer.	62	1	0	1/134 (stopping criterion met)
	Total	610	72	32	

Table 1: Overview of our backward snowballing phase. In total 18 papers were snowballed until the stopping criterion was met, leading to 32 unique additions to our relevant paper set.

The further Quality Assessment, Data Extraction and Validation steps of SYMBALS are generally used to reduce the amount of relevant papers. As we deem the 87 total papers to be manageable to analyse within the available time we do not perform these steps in this thesis.

4.6 Outcomes Systematic Review

For each of the 87 relevant sources we compiled titles and abstracts and created a full source list, assigning a number to each paper. From two of the papers the full text was unobtainable, leading us to disregard these papers further, reducing the total relevant paper size to 85. We then examined each paper's full text with the aforementioned goals in mind. These goals converge into their respective result sections. The goals are to compile model architectures like BERT and ways to pretrain and finetune these models, compile methods for classification and compile methods for data alteration, data annotation and data verification. Our full tables concerning overviews of papers that use text classification and papers that make use of BERT models were too large to include in this thesis and can be found in our code repository. The rest of the results can be found in the next sections.

4.7 Data Preprocessing

In terms of dealing with raw input text data, as is the case within this thesis, there exist a lot of methods within the papers of our literature review to preprocess text with the goal of obtaining the best possible input for our models, i.e. the input that produces the highest evaluation scores. In this section we lay out each individual data preprocessing method we found. For every method we note the first source we encounter that mentions the method, and explain the method using the elaboration of that respective method in the respective paper. We give a short elaboration per point about the relevance of the method to this thesis. At the end, we decide which methods to apply within our experiments.

Applying weakly labeled data via distant supervision (Agnikula Kshatriya et al., 2021): Weakly labeled data is data that has not explicitly been labeled by hand. In particular here weakly labelling is performed using distant supervision, which entails here that a rule-based system is used to automatically label texts. As we have already labeled our data using a string query, we already make use of weakly labeled data and therefore do not apply it further in this thesis.

Usage of post-hoc rules (Agnikula Kshatriya et al., 2021): This would result in a combined model of string matching and BERT, as we would use string matching to fix BERT's possible errors like how this method is used in the paper. In this thesis, for the sake of time constraints we do not experience with post-hoc rules, but recommend it for future research.

Removing stop words (Allada et al., 2021): The removal of stop words is a measure that is included in a plethora of papers that were reviewed. The effect of removing stop words is debatable, as BERT models rely on context to perform well and stop words like negation words are crucial towards retaining context. However, as this method is so prevalent we conduct tests with removing and keeping stop words and measure and compare performances.

Translating texts to English (Amin et al., 2019): Within the respective paper, translating input texts from German to English increased BERT's performance significantly. As the general theme of the data in the paper was not related to the clinical domain, it would be interesting to see whether translating the texts to English and then using them as input for models like BioBERT or ClinicalBERT would yield similar performance increases. For this reason we experiment with translating our input from Dutch to English so we can use English language models and compare their performance to Dutch ones.

Segmenting sections based on header (Banerjee et al., 2022): This method is mostly applicable to structured health data, to which we have no access to in this thesis. We therefore do not apply this method.

Best-worst scaling (De Bruyne, De Clercq, & Hoste, 2021a): The best-worst scaling in this paper was used to annotate a set of tweets on the basis of sentiment. Instances were converted into 4-tuples and distributed among annotators, the anootators then had to indicate the best and worst example for each of the respective dimensions. As tweet sentiment is way more subjective than smoking status this approach makes sense in the former but not in the latter, as there is way less ambiguity in how smoking status is described compared to what sentiment is expressed within a tweet. We therefore do not consider this method further.
Converting general concepts like dates, times and locations into special tokens (D. Pan et al., 2020): In the paper, they mask general concepts with tokens like replacing '01-01-2023' with '{date}'. While the effect of applying this method in particular is not measured, overall, the bulk of preprocessing methods in this paper showed a marginal increase in performance in an ablation study where it was compared to performance with data that was not preprocessed. As it is not clear if applying this method yielded any performance increase we do not apply it in this thesis.
Shuffling the input dataset (Delobelle, Winters, & Berendt, 2022): In this paper, as RoBERTa models drop the next sentence prediction task and use shuffled input data, the authors want to test what the effect of this is on their model RobBERTje in comparison with the old method of not shuffling the data. Ultimately, not shuffling the data appears to give rise to a better Masked Language Modeling head in the distilled pretrained model. Contrarily, not shuffling the data resulted in a better performance on sentiment analysis. As we experiment both with RoBERTa-like and general BERT-like models, the data is shuffled and not shuffled at different times.
Defining an Info-Preservation model with negation (Jaiswal et al., 2021): In this study, to ensure all radiology-related concepts are valued, an info-preservation module is used which includes terms from a radiology ontology. Furthermore, a 30-word negation keyword corpus is used to flip outcomes. Similarly, keywords that indicate uncertainty are also used for alteration of labelling. As this paper makes use of an architecture which we do not implement and the fact that we are training BERT models in an unsupervised manner we do not adapt this method in our thesis. It could be the case that our BERT-like approaches still struggle with handling the problem of negation, even though they learn the context of the texts. In this case, we recommend researching more sophisticated methods of handling negation in Dutch clinical text lifestyle classification for future research.
Swapping character-level byte-pair encoding with larger byte-level vocabulary (Y. Liu et al., 2019): A byte-level vocabulary contains subwords rather than single characters for encoding. This way, a larger amount of parameters can be achieved without performing any additional preprocessing or tokenization. For this reason, we apply this technique in our thesis.
Removing noise (doctor's signatures, inline spaces etc.) (Mu et al., 2021): Contrarily to the removal of stopwords, we will most likely implement this method in our thesis due to widespread use and decrease in tokens per sentence/document. Whiteline spaces contribute very little to context and doctor's signatures and such are not needed for the task at hand nor for any conceivable task.
Deidentifying patient data (Richter-Pechanski, Geis, Kiriakou, Schwab, & Dieterich, 2021): The data we use in this thesis has already been deidentified by CTCue, therefore there is no need to deidentify it further.

All in all, we measure the effect of translating Dutch text to English for English BERT models. We use the byte-level vocabulary from RoBERTa. We also experiment with removing noise

such as inline whitespace and doctors' signatures.

4.8 Data/Results Evaluation

In this section we lay out several methods to evaluate the usefulness of the data for the need for a sophisticated model. For example, it could be the case that the data is so straightforwardly interpretable for the task of lifestyle classification, that a string matching or simpler neural network model outperforms sophisticated models like BERT.

Visualizing word vectors generated by Word2Vec (B. Yang et al., 2023): In this paper, the authors used the t-SNE method to visualize word vectors generated by Word2Vec. Word2Vec was one of the baselines used in this paper. They visualized the vectors to understand how the word embedding maps crucial keywords. By inspecting the projections, they found that Word2Vec created clusters of words with prominent semantic meanings. Here, concepts related to the respiratory tract and the cardiovascular system were gathered into separate groups. The fact that these groups were comprised without any a priori knowledge shows a possible explanation for why embedding-based models outperform classical NLP methods. Using t-SNE could very similarly be applied to BERT models in order to view the embeddings in relation to each other, especially to BERT models that were pretrained from scratch as they solely make use of the input data at hand.
Using the LIME visualization tool to explain how BERT models use embedding features to predict text labels (B. Yang et al., 2023): In the same paper, the authors use the LIME tool to illustrate how each feature of a text contributes to the overall prediction, which gives an overview of how the model functions. Here in particular, the authors visualize the impact of the features on the task of coronary heart disease probability prediction. Overall, they show that a combination of word embedding and deep learning yields weakly correlated keyword features which can be used for the precise inference of text content. The standard neural networks such as Random Forest and Word2Vec were explained to be lacking in this regard, which in turn explains their lower performance on the task at hand. As this method can be used easily to yield explanatory visuals in regards to the results we apply it in this thesis.
Using a wordcloud representation to show word distribution statistics (Banerjee et al., 2022): In this paper they show which words show up most frequently per dataset and class by using a wordcloud representation. The larger the font of the word is in the representation the more often it shows up. As this paper works with more specialized datasets there are clear differences to be seen between the wordclouds. Ultimately, the results of this wordcloud were used to justify finetuning their model on more generic reports rather than specialized datasets. Because of the ease of implementation and interpretation we make use of wordclouds to show the most common words per class.
Showing dataset characteristics such as median age, % male, median BMI etc. (Chen et al., 2022): In this paper the authors have access to structured features such as patient information, from which statistics are calculated and showed in a table. As we have no access to structured data we do not make use of this method.
Showing the amount of errors per semantic category for each model (Fu et al., 2022): Here, using bar plots, the authors show the amount of errors per model per semantic category. The semantic categories included negation and homonyms among other more topic-specific categories. Overall, machine learning models outperformed rule-based approaches on negation but not on identifying fall risk. A combined model of machine learning and rule-based

approaches was able to benefit from both upsides, performing well on negation and on identifying fall risk. As it is unclear which semantic categories should be selected for this thesis we choose not to implement this method.

Showing examples from the dataset and highlighting which words were most important in classifying the document (W. Hu & Wang, 2022): In this paper, the authors collect 2 correctly classified and 2 incorrectly classified examples from the test set and highlight the words that had the most influence on the decision. Feature importances were extracted by leveraging the attention-based architecture of BERT. This way the words that were most highly attended to could be identified and these were regarded as the most important features. We could apply this method to highlight words that caused a correctly predicted text by BERT that was not correctly classified by string matching and vice versa. This way, words that throw off the models can be identified. Comparative methods like these could provide valuable insights into the workings and shortcomings of models, for this reason we choose to conduct comparison experiments between different models in a similar manner as to how they were conducted in this paper.

Showing distribution of negation types in dataset (C. Lin et al., 2020): Here, the authors show their dataset's distribution of 5 common negation cue terms from the Negex system. Within our context, this could help visualizing how prevalent the problem of negation is in our input dataset, which in term likely indicates how much BERT models improve upon the baseline. Although the problem of negation is apparent in these kinds of studies we believe other visualization tools like LIME and t-SNE are easier to use and produce comparably accurate explanations when comparing BERT output to more shallow models' output.

Visualizing attention mechanisms using a warm-to-cool colour spectrum (C.-H. Lin et al., 2021): The authors of this paper used a warm-to-cold colour system to create an image that indicates which words received higher and lower attention scores between different BERT models on the same text. This image was then used to explain differences between the output and workings of the models. In our context, we can use this method to show which words are focused on and can then explain if this focus is a benefit towards the performance compared to the baseline of string matching. Due to us already using multiple other methods to visualize BERT embeddings we choose not to implement this method.

Semantic type embedding visualization using UMAP (Michalopoulos, Wang, Kaka, Chen, & Wong, 2021): In this paper, in order to demonstrate the effect of the semantic types on the input embedding, the authors apply UMAP dimensionality reduction to compare input embeddings of two BERT models. The terms are colour coded based on their semantic groups. As we already use t-SNE to reduce dimensionality we do not implement this method.

Showing the top-5 words models rely on for label prediction (Mu et al., 2021): Here, the authors show per label the top 5 words that contribute to the model classifying that exact label. The words are colour coded based on their L2-normalized importance score. It shows that some words in the top 5's are not semantically similar to the label at hand. Considering we already use LIME to visualize the most important terms we do not implement this method.

Showing dependency connections from BERT model (Rasmy et al., 2021): The authors of this paper used this method to show the attentions of their Med-BERT model on a finetuned task. The attentions are shown between terms between the finetuning layers, where a thicker line indicates a higher attention weight. These kinds of images show that finetuned models express distinct task-dependent patterns across multiple layers, meaning the model is capable of generalizing and adapting to real-word scenarios. As we finetune all of our models we do not compare finetuned models to non-finetuned models, making this method not appli-

cable to this thesis.

Using a Random Forest classifier to determine which variables added the most value to binary predictions (Spruit, Verkleij, de Schepper, & Scheepers, 2022): In this paper, the authors use the Linguistic Inquiry and Word Count (LIWC) tool as one of the evaluated models. In order to aid interpretation of its output, a random forest classifier was used to determine which 10 features added the most value to the binary predictions made. This was applied to the SpaCy tool as well, which was also one of the evaluated models. This method can be used directly after preprocessing our data in order to gain insight in important features in the dataset. As we do not make binary predictions, and we are already using easier methods that show which features contribute the most we do not implement this method.

Showing the amount of classification errors per model (Spruit et al., 2022): In this paper, the authors show the number of unique misclassifications for different model combinations, that being the amount of errors that were made only by the respective model. This can show how ensemble methods can reduce on the amount of errors made. Because we are more interested in which model can estimate the ground truth the best and not so much into the shortcomings of every lesser model we do not use this method.

Visualizing attention patterns and weights using the BertViz toolkit (Yao, Jin, Mao, Zhang, & Luo, 2019): This method is very similar to the visualization method used in (Rasmy et al., 2021), as attention of features between layers is visualized using colour coded lines that have a thickness that corresponds to the size of the respective weight. For the same reasons that we do not show dependency connections we also do not show attention patterns and weights.

In conclusion, we use t-SNE to visualize word embeddings and their place in relation to each other, LIME to show the most important features per class and word clouds to show the most prevalent words per class.

4.9 BERT-like Models

In this section we lay out every BERT model that we encountered while going through our relevant papers. For each model we report the pretraining objectives, the focus of the model (general model, health domain, etc.) and the size and language of the input text data. Lastly, we report the performance of the model. This entails that we record how many papers evaluate the respective model and what the performance was in regards to the other models. In this columns, italic numbers indicate that this paper is the source paper of the model, bold means the model performed the best on the displayed task in the paper and regular typeface means the model's performance was measured but it was outperformed by another model. In terms of the pretraining objectives, MLM stands for Masked-Language Modeling, NSP stands for Next Sentence Prediction and SOP stands for Sentence Order Prediction.

Name	Pretraining Objectives	Focus	Training size	Language	Performance
XLM-RoBERTa	Multilingual MLM	General	2.5 TB	100 languages	**(Chaichulee et al., 2022)**
BERTje	MLM with consecutive masking and SOP	General	12 GB	Dutch	*(de Vries et al., 2019)*, (Delobelle, Winters, & Berendt, 2020), (Delobelle et al., 2022), **(Van Olmen, Van Nooten, Philips, Sollie, & Daelemans, 2022)**, (Verkijk & Vossen, 2021)
RobBERT	MLM	General	39 GB	Dutch	*(Delobelle et al., 2020)*, **(Delobelle et al., 2022)**, (Spruit et al., 2022), (Verkijk & Vossen, 2021), (Wouts, 2020)
RobBERTje	MLM	General	1 GB	Dutch	*(Delobelle et al., 2022)*
BERT	MLM and NSP	General	3.3 billion words	English	*(Devlin et al., 2019)*, **(Adhikari, Ram, Tang, & Lin, 2019)**, (Agnikula Kshatriya et al., 2021), (Amin et al., 2019), (Coutinho & Martins, 2022), (D. Pan et al., 2020), (de Vries et al., 2019), **(Fink et al., 2022)**, **(Han et al., 2022)**, (D. Hu et al., 2021), **(W. Hu & Wang, 2022)**, (K. Huang, Altosaar, & Ranganath, 2020), **(Jiao et al., 2020)**, (Joshi et al., 2020), (Lewis et al., 2020), (Michalopoulos et al., 2021), (S. Park et al., 2022), **(Richter-Pechanski et al., 2021)**, **(Sanh, Debut, Chaumond, & Wolf, 2019)**, (Xie et al., 2022), (Yao, Jin, et al., 2019) **(Z. Yu et al., 2022)**
ClinicalBERT	MLM and NSP	Clinical domain	2 million notes	English	*(K. Huang et al., 2020)*, (B. Yang et al., 2023), (C. Mao, L. Yao, & Y. Luo, 2020), (C.-H. Lin et al., 2021), (Mitchell et al., 2022)
TinyBERT	BERT distillation minimizing loss	General	3.3 billion words	English	*(Jiao et al., 2020)*
SpanBERT	MLM	General	3.3 billion words	English	*(Joshi et al., 2020)*
ALBERT	MLM	General	3.3 billion words	English	*(Lan et al., 2019)*
BioBERT	MLM and NSP	Biomedical domain	21.3 billion words	English	*(Lee et al., 2020)*, **(Amin et al., 2019)**, (B. Yang et al., 2023), (C. Lin, Miller, Dligach, Bethard, & Savova, 2021), (Michalopoulos et al., 2021), (S. Park et al., 2022)
EntityBERT	MLM	Clinical domain	21 GB + 2 million words	English	*(C. Lin et al., 2021)*
RoBERTa	MLM	General	3.3 billion words	English	*(Y. Liu et al., 2019)*, **(Lewis et al., 2020)**, (Xie et al., 2022), (Z. Yu et al., 2022)
BioALBERT	MLM and SOP	Biomedical domain	21.8 billion words	English	*(Naseem, Dunn, Khushi, & Kim, 2022)*
DistilBERT	Minimizing MLM loss, distillation loss and cosine embedding loss	General	3.3 billion words	English	*(Sanh et al., 2019)*
MedRoBERTa.nl	MLM	Clinical domain	13 GB	Dutch	*(Verkijk & Vossen, 2021)*
belabBERT	MLM	General	32 GB	Dutch	*(Wouts, 2020)*

Table 2: Overview of BERT-like models we encountered in our systematic literature review.

Examining table 2 and the other papers from the literature review we do not deem it necessary to apply general English BERT models for finetuning, such as BERT, RoBERTa and ALBERT, and using them as baseline for evaluation. This is because we already have existing and updated string matching frameworks to function as baseline. Furthermore, pretraining on domain-specific data has outperformed general models virtually every time they were compared on clinical tasks, such as for ClinicalBERT (Banerjee et al., 2022; Gu et al., 2022; K. Huang et al., 2020; Michalopoulos et al., 2021; Olthof, van Ooijen, & Cornelissen, 2022; Xie et al., 2022), BioBERT (Amin et al., 2019; Gu et al., 2022; Lee et al., 2020; Michalopoulos et al., 2021; S. Park et al., 2022) and BioClinicalBERT (Agnikula Kshatriya et al., 2021; Michalopoulos et al., 2021; Xie et al., 2022). This shows the benefit of pretraining on domain-specific input data.

From the literature review it can also be concluded that pretraining on top of an existing model yields higher performance than solely finetuning existing models (Lee et al., 2020; C. Lin et al., 2021; C.-H. Lin et al., 2021; Michalopoulos et al., 2021; Mitchell et al., 2022; Naseem et al., 2022; Verkijk & Vossen, 2021). There are significantly less sources that compare pretraining a model from scratch to pretraining a model using another model's weights. Regardless, it has been shown that pretraining using the weights of RobBERT performed better than pretraining a model from scratch for MedRoBERTa.nl (Verkijk & Vossen, 2021).

Dutch models are shown to outperform BERT on tasks with Dutch source text, like is the case for BERTje (de Vries et al., 2019). In terms of BERT architectures trained on the general domain, RoBERTa architectures were shown to outperform standard BERT architectures more often than not (Chaichulee et al., 2022; Delobelle et al., 2020, 2022; Gu et al., 2022; De Bruyne, De Clercq, & Hoste, 2021b; W. Hu & Wang, 2022; Lan et al., 2019; Lewis et al., 2020; Y. Liu et al., 2019; Olthof et al., 2022; Verkijk & Vossen, 2021). This builds a strong case for favouring models that use RoBERTa-like architectures rather than BERT-like architectures. For the reviewed Dutch models, these models would be RobBERT, MedRoBERTa.nl and belabBERT.

It is safe to assume that, just like with BERT, Dutch RoBERTa models outperform English RoBERTa models on Dutch input text. That is however when the text is not translated to English, as translation from German to English showed a significant increase in performance (Amin et al., 2019). This could indicate using a sophisticated translation tool could enable us to use models like BioBERT and ClinicalBERT more effectively, possibly then even outperforming RobBERT and belabBERT due to these latter models' focus on the general domain. A comparison with MedRoBERTa.nl and a RoBERTa model trained from our data from scratch would be very interesting in that regard. This means we will consider the following BERT experiments, where we will compare the performance on finetuning tasks:

1. Experiment 1: Pretraining from scratch using the input data
2. Experiment 2: Pretraining on top of the weights of either RobBERT, belabBERT or MedRoBERTa.nl
3. Experiment 3: Translating the input data, then pretraining on top of BioBERT or ClinicalBERT.

4.10 Pretraining Methods

As we decided on pretraining either from scratch or on top of existing models for BERT experiments 1 and 2 we gathered methods for pretraining and explain which methods we will implement in our thesis. We include the first source we encountered that uses the respective method. For every method we note the first source we encounter that mentions the method, and explain the method using the elaboration of that respective method in the respective paper.

Concatenating TF-IDF vectors to BERT word embeddings (Allada et al., 2021): In this paper, a subbranch of TF-IDF layers was added to the BERT pipeline, where the TF-IDF feature vectors are concatenated to the BERT word embeddings. Removing this TF-IDF branch resulted in a drop of over 8% in performance on pathology reports. This architecture is however not adopted anywhere else and whilst it would be interesting to test whether adding this branch improves performance, limited time and resource constraints lead us to leaving this to future research.

Adding a new task where the goal is discriminating between real and artificial samples (Coutinho & Martins, 2022): This task is very topical to ICD-10 coding, which is also what it is used for in the respective paper. It is safe to assume applying this task is less relevant on lifestyle characterisation, which is why we will not use it.

Apply a better subarchitecture by optimizing inference speed, parameter size and

error rate (de Wynter & Perry, 2020): The subarchitecture proposed in this paper was significantly faster than BERT and RoBERTa and contained significantly less parameters. It was also able to outperform BERT on most GLUE tasks. We consider this architecture further based on the available resources.
Masked-language modelling with token masking (Devlin et al., 2019): As a standard BERT pretraining task which is rarely altered in all research following this paper, we will most likely adapt this task for pretraining of our model.
Next-sentence prediction (Devlin et al., 2019): This other standard BERT pretraining task has been replaced often in newer research, such as in RoBERTa-like models, with noticeable increase in performance, which means we do not adapt this task for pretraining.
Contrastive self-supervised learning (Jaiswal et al., 2021): Contrastive learning seeks to learn effective data representations by maximizing the agreement between two augmentations on one example and minimizing the agreement of augmentations from different instances. The setup and evaluation of a contrastive model would not fit within the constraints of this thesis, therefore we leave it for future research.
Distillation via teacher-student networks (Jiao et al., 2020): Distillation comes in handy if we have limited computing resources available. As this is not the case in this thesis, as we touch upon later, we do not consider this setup.
Masked-language modelling with span masking (Joshi et al., 2020): Using spans of words for masking rather than singular words outperformed BERT on almost all GLUE tasks, for this reason it would be interesting to apply in this thesis. However, this technique is significantly harder to implement than word masking, and we therefore leave it out for future research to compare the two techniques within the domain of Dutch clinical text lifestyle classification.
Consecutive sentence order prediction (Lan et al., 2019): In this paper the proposed model, this being ALBERT, using consecutive sentence order prediction improved upon the same model with next sentence prediction by around 1%. As the SOP task is also used in other papers and it is generally accepted that it is a better pretraining task than NSP we use it as one of our pretraining tasks.
Factorized embedding parameterization (Lan et al., 2019): Factorized embedding parameterization refers to breaking down token embeddings into smaller embedding matrices, reducing the amount of embeddings parameters. This technique, along with the next one, significantly decreased the amount of parameters for BERT without seriously hurting performance, as stated in the paper. Although not significantly, adding this change allowed a more efficient usage of the total model parameters by splitting the vocabulary embedding matrix into two small matrices. This showed in performance too, albeit only slight increases were observed. However, as we can use this technique to reduce the amount of parameters we include it in our thesis.
Cross-layer parameter sharing (Lan et al., 2019): Similarly to factorized embedding parameterization, cross-layer parameter sharing significantly reduces the number of parameters for the model. By sharing all parameters between all layers, the amount of parameters does not grow the deeper the network gets. However, while sharing all parameters significantly decreases the amount of total parameters, it does tend to hamper performance. In our thesis we decide to use cross-layer parameter, as an all-shared parameter set is more scalable and less computationally expensive than the standard BERT measure of non-sharing.
Document corruption (Lewis et al., 2020): This method, designed to replace MLM, was chosen in this paper as it is suitable for a model with multiple end tasks. The model here,

BART, is able to do language generation, translation and comprehension. As BART did not generally outperform RoBERTa and the fact the focus was more divided we do not consider this pretraining task further.
Masked-language modelling with entity masking (C. Lin et al., 2021): To obtain entities, they need to be annotated in each text. As this is not feasible in the context of this thesis, we do not consider entity masking.
Adding BiLSTM and CRF layers (H. Liu et al., 2021): Applying this method increased performance on liver cancer diagnosis, but due to its size it is not feasible to implement this architecture in this thesis given time and resource constraints.
Masked-language modelling with dynamic masking (Y. Liu et al., 2019): In a direct comparison with RoBERTa without dynamic masking and BERT without dynamic masking, RoBERTa with dynamic masking performed better on the same tasks. As RoBERTa-like structures are generally regarded as better-performing than BERT-like structures we implement dynamic masking in this thesis.
Addition of character-based embedding vectors (S. Park et al., 2022): While adding an extra character-based BERT layer could increase performance like it does for ConBERT, due to constraints of resources and time we do not test it here and leave this for further research.

In conclusion, we will perform experiments with the optimal parameters found in (de Wynter & Perry, 2020), adapt the MLM with dynamic masking and consecutive sentence order prediction as pretraining tasks for our models and experiment with factorized embedding parameterization and cross-layer parameter sharing, like in ALBERT (Lan et al., 2019). We apply factorized embedding parameterization and cross-layer parameter sharing to our pretrained from scratch model, and compare this to models that do not use these techniques, such as the models that we pretrain on top of existing models like RobBERT and ClinicalBERT.

4.11 Finetuning

Finetuning will be done on a labelled dataset. This data needs to be either automatically or manually annotated for smoking, alcohol and drug usage status. This makes the task a multi-class classification task on paper. In practice, we separate the task-at-hand into three subtasks, meaning there are three tasks (smoking, alcohol and drugs) per model. As is the case in many of the papers that were reviewed, the BERT models are finetuned by branching out to a single output layer. The amount of nodes in this output layer has to correspond to the amount of possible subclasses (current smmoker, previous smoker et cetera). We use the softmax activation within this output layer (Allada et al., 2021; W. Hu & Wang, 2022), as we want to be able to output probabilities per class in order to show confidence. In section 5 we perform an exploratory data analysis to explore whether to use automatically or manually assigned labels.

4.12 Conclusion

We conducted a systematic literature review using the SYMBALS method. We first laid out methods that had influence on elements of SYMBALS, these methods being string matching and BERT. We then created a query and relevance criteria for our literature review and extracted and evaluated papers. We applied backward snowballing to increase the amount of

relevant papers. From the final set of 85 relevant papers we extracted methods for different aspects of our thesis.

For data annotation we decided to measure the effect of translation to English, removing noise, applying sentence embeddings and removing stop words. In terms of BERT models we experiment with pretraining from scratch using an ALBERT-like architecture, and pretraining on top of Dutch models RobBERT, belabBERT and MedRoBERTa.nl. We furthermore experiment with translating texts to English and then pretrain on top of BioBERT and ClinicalBERT. During pretraining, we use optimal parameters that were found in (de Wynter & Perry, 2020), and adapt Masked-Language Modeling and Sentence Order Prediction as pretraining tasks for our ALBERT-like model, which makes use of factorized embedding parameterization and cross-layer parameter sharing. Lastly, we finetune all BERT models on a uniform labeled set of texts for the task of smoking, alcohol and drugs classification.

5 Exploratory Data Analysis

In this section we complete the Business Understanding CRISP-DM phase and move towards the Data Understanding phase. In the context of Business Understanding we provide an overview of our collaboration with HagaZiekenhuis. HagaZiekenhuis makes use of the CTCue platform in order to save and analyse their data. We establish their needs, use these needs and our needs in terms of research to build a query to extract data from CTCue and provide an initial analysis on the data that was extracted in the context of Data Understanding. Every extracted clinical note is labeled using a string matching query and we test whether these automated labels are sufficient to serve as input for the rest of our thesis.

5.1 HagaZiekenhuis Collaboration

Within this thesis we collaborate with HagaZiekenhuis in The Hague, The Netherlands. They serve as our data provider and have the wish to implement our lifestyle classification model in the future, given that it satisfies their requirements. We conduct a case study with them in order to be able to further the research on lifestyle classification in Dutch clinical texts. In accordance with CRISP-DM, we first held several meetings in which we established their and our needs. We summarize their overall needs using user stories:

a) As data provider, in order to prevent data leaks, we wish the data only gets handled and analysed within a secure environment, such as a secure virtual environment from a hospital.

b) As data provider, we wish to not include any texts in the dataset that contain the word "confidential", as we do not wish that these texts are used to train language models.

c) As health professionals, we wish to have a system with which we can with ascertain with relative certainty what a patient's smoking status is, as this could help in quickly identifying this health risk.

d) As health professionals, we wish to have a system with which we can with ascertain with relative certainty what a patient's alcohol consumption status is, as this could help in quickly identifying this health risk.

e) As health professionals, we wish to have a system with which we can with ascertain with relative certainty what a patient's recreational drug usage status is, as this could help in quickly identifying this health risk.

In accordance with our research questions, we established the following data requirements:
Feature requirements:

a) Smoking

b) Alcohol consumption

c) Drug usage

Gradations in features per text:

a) No user

b) Current user

c) No information given

d) Previous user

In terms of desired data quantities we had the wish to receive every available text from 2016 to 2023, as we wished to get as much input data as possible. CTCue has the option to provide contextualized exempts when performing a string query. This would mean for example that we would only receive a sentence containing the word 'smoking' rather than the full text. As we ascertained that it would likely better to obtain full texts to account for unconventional writing styles we decided not to make use of this option and rather to include full texts. This way we also have more text per note, which means there is more input for the respective models. Corresponding to the reasons we lay out in section 2.2 we regard every document as its separate entity and as the entirety of what we know about a patient, in order to allow for more straightforward comparisons in future research. Although this does not entirely adhere to HagaZiekenhuis's requirements, as they wish to regard the data on a patient-basis, we can contribute more meaningful results to the scientific domain this way.

We set the lower year bound to 2016 because of HagaZiekenhuis' wish to implement the system for current patients. In consultation with HagaZiekenhuis, we feel lowering this bound would not give a representative view of how doctors write their notes in the current age.

With these data requirements, we designed a desired data shape, which can be found in table 3.

Full text	**PatientID**	**Smoking**	**Alcohol**	**Drugs**
Lorem ipsum smoking -	1	0	2	2
Lorem ipsum smoking+, alcohol sometimes, drugs once a week	2	1	1	1
Lorem ipsum smoking +, alcohol -	3	1	0	2
...	...	...	...	...

Table 3: Designed data shape for data from HagaZiekenhuis.

Multiple initial data samples were provided to us, which we analysed and provided feedback to HagaZiekenhuis for. The final query for labelling the texts can be found in section 5.1.1.

5.1.1 String Matching Query

Over multiple weeks, data of several query iterations to label the texts to obtain a ground truth were provided to us. We provided feedback to HagaZiekenhuis with the goal of shaping the data to fit our requirements. Ultimately, we decided on the queries below. We lay out conditions for each class and subclass. The data is grouped in zip files per year, 2016 up to and including 2022. The files are in CSV format. The data totals to around 1.5 GB in size. Ultimately, due to a lack of previous alcohol and drugs users in initial testing, we decided to only include previous users for the smoking category. The query labelling is done using a number of string queries, which can be found in appendix A.

5.2 Data Overview

In this section we provide an overview of the data that was extracted using the query from the previous section. In order to analyse the texts we used Python, specifically the *Pandas* library. Overall, there were some problems with the raw data. While the data was in CSV-format, some of the text reports were not encapsulated with quotation marks, making them hard to view in CSV viewers like Microsoft Excel, as this results in reports spanning multiple rows. For the same reason, the files were not readable by Pandas.

In order to make the files readable several operations had to be performed. At first, in order to distinguish between newlines for new rows and newlines used in the reports we converted every newline character to a tab character. Then, we added a newline character at the end of every line for every report. This was doable as every entry ends with a date and this date was found by matching a regular expression statement.

With every entry now taking up a singular line in each CSV file we loaded the data into Pandas, resulting in one Pandas dataframe for each year. In order to gain insight in the total amount of texts we combined every year's CSV file into one combined file. As each row corresponds to one patient and there are three columns with textual reports, we extracted all unique reports from the combined dataset in order to develop an understanding of its contents. In total there are 148.000 unique texts, totalling a little over 35 million words.

For each lifestyle, we extracted structured fields such as the date of the report and the type of report. We did not use these fields further in this thesis, but rather used these to verify the data in consolidation with HagaZiekenhuis. Future internal research by HagaZiekenhuis could use these kinds of structured fields to perform further analysis within a different experimental setup, but this does not fit within the constraints of our case study. Furthermore, as explained in section 2.2, we wish to not include structured fields in order to make our experiments more reproducible in future research on Dutch clinical text lifestyle classification. We provide an overview of the fields that are present in the received files in table 4.

Field name	Description
pseudo_id	An identifier that can be used to distinguish patients. Within the context of this thesis we only use this field to validate our dataset, making sure every row corresponds to one patient.
roken_answer_label	Label acquired via string matching that indicates whether the report in question states the patient is a smoker.
roken_report_tags	The type of report that is being checked for smoking status. For example a radiology report or a clinical letter.
roken_report_content	The full contents of the report that is being checked for smoking status.
roken_report_start_date	The date of the report that is being checked for smoking status. Another field that qas used to check whether the right data was extracted.
alcohol_answer_label	Label that indicates whether the report in question states the patient drinks alcohol. The same classes apply as for smoking status.
alcohol_report_tags	The type of report for alcohol status.
alcohol_report_content	The text contents of the report that is being checked for alcohol status.
alcohol_report_start_date	The date of the report that is being checked for alcohol status.
drug_answer_label	Label that indicates whether the report in question says the patient uses drugs. The same classes apply as for smoking and alcohol status.
drug_report_tags	The type of report for drug status.
drug_report_content	The text contents of the report that is being checked for drug status.
drug_report_start_date	The date of the report that is being checked for drug status.

Table 4: Overview data fields in received data from HagaZiekenhuis, each of these fields are present in every file we obtain.

Table 5 shows the amount of texts and the distribution of the labels for each class. Note that the labelling was done by string matching queries and could differ from a 'real' (annotated by human) label.

Type of label	Amount of labelled texts	Current users	Previous users	Non-users	No information given
Smoking	148.768	7.015 (4.72%)	32.230 (21.66%)	44.677 (30.03%)	64.846 (43.59%)
Drinking	143.166	16.017 (11.25%)	-	39.119 (27.32%)	87.940 (61.43%)
Drugs	147.999	1.443 (0.98%)	-	53.005 (35.81%)	93.551 (63.21%)

Table 5: Overview and class distribution in query labels from provided data from HagaZiekenhuis.

The reason the total amount of labelled texts differs per class is that sometimes for one entry (i.e. one patient) the same text is used for the smoking, alcohol and drug content column but this is not always the case, resulting in different total values.

5.3 Feasibility Study Query Labels

As Heath managed a 64% accuracy with string matching on classifying Dutch clinical notes on the basis of smoking status (Heath, 2022), it is interesting to see whether other models could improve upon this significantly given the right circumstances. Furthermore, the ground truth only consisted solely of texts that were labeled on the basis of smoking status, more specific current and previous smokers. In total there were 6.560 discharge letters of a total of 480 patients. They labeled the patients' smoking status, rather than labelling on a letter-basis.

We extend the task with the aforementioned alcohol and drugs statuses and also add more subclasses, such as "not a user' and "no information given". By adding these subclasses we believe we can encompass the entire data and do not need to sift out texts that do not comply with the "only current and past smokers" rule that was applied in Heath's research (Heath, 2022). We furthermore shift our focus to texts instead of patients so that we can obtain and use more data. This means we evaluate on note-basis, rather than on patient-basis.

Heath did not automate the labelling process, as string matching is used as a baseline and automating labels can be regarded as a string matching process of its own. Generally, the pay-off of using automated labelling is that it can result in more data but the labels are of lower quality. It would be interesting to see what the difference in quality is between hand-labelling and query labelling on the task of smoking, alcohol and drug usage status and whether query labelling could be a feasible option for answering our research questions. We test this feasibility in this section.

We lay out several flaws with query-labelling by comparing it to a set of hand-labeled texts of edge cases and use these texts in order to craft a hand-labeled set which will serve as input for classical machine learning models. This is in accordance with the "Data Understanding" phase of Crisp-DM, where we verify the quality of the data, in this case the query labels.

5.3.1 Experimental Setup Query Labels

It should be noted that within this thesis we can differentiate between two different methods of label interpretation. As we have the "user", "non-user" and "no information given" classes for each class, we can interpret them as is, or we can group the "non-user" and "no information given" classes together, holding the assumption that when the lifestyle characteristic is not noted that the patient at hand is a non-user. This changes the problem-at-hand from a multi-class classification problem to a binary classification problem.

We apply both methods as it is interesting to explore which features have the biggest impact on the prediction on the labels for both cases. This allows us to gather a deeper understanding of which aspects a potential BERT model will need to master to outperform these more shallow machine learning approaches. For the rest of the query labelling and classical machine learning feasibility studies, we refer to the multi-class classification containing all classes as

task 1 and the binary classification task (users and non-users) as **task 2**. Using this naming convention, we for example refer to the binary classification task for alcohol as "Alcohol Task 2" and the multi-class classification task for drugs as "Drugs Task 1".

To obtain usable features from raw text, we use the term frequency inverse document frequency (TF-IDF) method, which we explained in section 3.4.

We will use the same preprocessing steps as the paper we used to get an idea of classical machine learning approaches that are applied to medical text classification (Dipnall et al., 2022). We will experiment with keeping all stopwords, removing all stopwords and only keeping negation stopwords. Negation words are regarded as stopwords and as negation intuitively plays a central part in classifying whether a patient has certain lifestyle characteristics or not, several studies we found revolve around negation or involve it in some form (C. Lin et al., 2020; Ramachandran et al., 2023; van Es et al., 2023), we experiment with including and excluding these stopwords. This means the preprocessing methods used are:

1. Splitting texts into words
2. Lowercasing
3. Punctuation stripping
4. Using a TF-IDF vectorizer to obtain numerical values for each word
5. Removing stopwords (experimentation), we identified the negation stopwords "niet" (not), "niets" (nothing), "geen" (none) and "zonder" (without).
6. Stemming

We experiment with tuning the ngram size and perform the earlier explained experiment concerning stopwords. The ngram size indicates how many consecutive words are combined into one feature. Within the paper, the authors perform tests with altering the minimum and maximum document frequencies. The minimum and maximum document frequency indicate thresholds for including ngrams based on how often they show up in the dataset. As removing stopwords can be seen as a similar approach to this we do not perform experiments where solely these thresholds are altered. Furthermore, in initial tests on the smoking dataset, altering these thresholds in the same way as the authors did in the paper did only change the performance very marginally. The Macro F1-score even stayed the same among the models with different document frequency thresholds. For this reason, we added the minimal and maximum document frequency as random search parameters.

To also account for the other features of the models, we will use this random search in order to be able to perform this experiment while adhering to time and resource constraints. Within this random search, we make sure to set the same random seed among models that use the same classifier. This results in the same hyperparameters being checked for the same classifier. For example, for Multinomial Naïve Bayes the models with different ngram size and stopwords are being assigned the same parameters randomly, in order to allow for more fair comparisons in terms of performance. We provide an overview of the random search parameters in the next section.

We train the models on a training set, splitting the data to training and test sets with the ratio 80/20 as this ratio has shown to work well with large datasets and was the most occurring split in our systematic literature review, appearing in 24 of the 75 papers that stated using a training/test split. In order to reduce the effect of randomness in splitting the data we make use of k-fold cross validation. Corresponding to our split size, we use 5 fold cross-validation, meaning we use 1 fold (20%) as test set and the remaining 4 folds (80%) as training set. Ultimately, each fold is used as test set once and results are averaged over all test folds.

We measure performance using the macro averaged F1-score. We use macro-averaged F1-score rather than micro-averaged F1-score as we regard every class as equally important, no matter how often the class shows up in the dataset. The formula for Macro F1-score is given in section 2.2.

All of our data is processed and analysed using Python on a Linux virtual machine from Leids Universitair Medisch Centrum (LUMC), the university hopsital affiliated with Universiteit Leiden. In order to build the models we make use of the *sklearn* Python library, as this library contains functionalities for each classical machine learning approach we wish to apply in this feasibility study. For Logistic Regression and Support Vector Machines in particular we use sklearn's *SGDClassifier*, which can combine these methods with Stochastic Gradient Descent to optimize loss. Within our random search, we have parameter setups for this classifier that determine whether a linear SVM or a probabilistic Logistic Regression model is used. As described in the previous section, we perform the full experiment on Smoking Task 1 and then use the 4 best performing and 1 lesser performing setups for the other tasks.

In order to fit the studies within the time constraints of this thesis we first perform one full experiment, this being Smoking Task 1. A full experiment entails recording performance for ngram sizes 1, 2 and 3 for keeping stopwords, only keeping negation stopwords and removing all stopwords. We then use the 4 best performing models and 1 lesser performing model from this task to train in the other tasks. Here, the lesser performing model serves as a kind of control group. We define a lesser performing model as the worst model that performs better than 50% on the Macro F1-score evaluation. Per task we provide the results, as well as a word cloud of the features that contributed the most per class. The features are assigned a score by the model that performed the best. We create these word clouds as we are interested in whether the models pick up on features beyond the queries that were used. Word clouds have been used before to show the most-occurring features in a dataset and can provide insights on which features are the most important per class (Banerjee et al., 2022).

5.3.2 Random Search Parameters

In order to implement random search we use the *RandomizedSearchCV* class from the sklearn library. As described earlier, we divide the training set into 5 folds, each iteration 4 of the 5 folds are used as training set and the remaining one as validation set. The model that performed best is evaluated on the actual test set. We set the amount of parameter combinations that are checked to 10, in order to perform a reasonable amount of experiments and to keep within time constraints. We sought to apply random search to as many class variables as possible. This resulted in the following combinations being checked:

TF-IDF

Parameter	Possible values
max_df	0.90, 0.95
min_df	3, 5

Table 6: TF-IDF possible parameter values for random search.

These TF-IDF variables in table 6 refer to the maximum and minimum document frequencies that were applied in (Dipnall et al., 2022).

Multinomial Naive Bayes

Parameter	Possible values
alpha	(0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100)
fit_prior	(True, False)
class_prior	Random sample of size 5 of probabilities for each of the 4 classes.

Table 7: Multinomial Naive Bayes possible parameter values for random search.

Table 7 shows the parameter values that are tested for Multinomial Naive Bayes.

Stochastic Gradient Descent

Parameter	Possible values
loss	('hinge', 'log loss', 'modified huber', 'squared hinge' , 'perceptron', 'squared error', 'huber', 'epsilon_insensitive', 'squared_epsilon_insensitive')
penalty	('l2', 'l1')
l1_ratio	Random float between 0 and 1.
fit_intercept	(True, False)
max_iter	(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)
tol	Random float between 0 and 1.
shuffle	(True, False)
epsilon	Random float between 0 and 1.
learning_rate	('constant', 'optimal', 'invscaling', 'adaptive)
eta0	Random float between 0 and 1.
power_t	Random float between 0 and 1.
class_weight	('balanced', None)
warm_start	(True, False)
average	(True, False)

Table 8: Stochastic Gradient Descent possible parameter values for random search.

Table 8 shows the possible parameter values for Stochastic Gradient Descent. For these parameters, when the 'hinge' loss is selected the model becomes a linear Support Vector Machine. Similarly, when 'log_loss' is chosen the model becomes a logistic regression classifier.

Random Forest

Parameter	Possible values
n_estimators	(10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50)
criterion	('gini', 'entropy', 'log_loss')
max_features	('sqrt', 'log2', None)
max_depth	(2, 3, 4, 5)
min_samples_split	(100, 200, 300, 400, 500, 600. 700, 800, 900, 1000)
min_samples_leaf	(100, 140, 180, 220, 260, 300, 340, 380, 420, 460, 500)
class_weight	('balanced', 'balanced_subsample')
ccp_alpha	Random float between 0 and 1.
max_samples	(0.1, 0.2, 0.3)
max_leaf_nodes	(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)

Table 9: Random Forest possible parameter values for random search.

Lastly, table 9 shows the parameters and their possible values for our Random Forest models.

5.4 Results Query labelling

In this section, we explore the results of Multinomial Naive Bayes, Logistic Regression, Support Vector Machine and Random Forest on query labels for the multi-class and binary classification tasks for smoking, alcohol and drugs.

5.4.1 Results Smoking Task 1

In this section, we lay out the results of our three classical machine learning methods on a uniform test set. We name every model based on which classical machine learning method it uses, whether its input contains stopwords and what the n-gram size is. From the results of this task, we choose 4 of the best-performing models and 1 of the lesser performing models for the other tasks.

1. Multinomial Naive Bayes Performance

Model name	**"No information given" F1-score**	**"Current user" F1 score**	**"Previous user" F1-score**	**"Non-user" F1-score**	**Macro F1-score**
		1.1 Stopwords kept			
1.1.1 Ngram 1	0.87	0.34	0.66	0.70	**0.64**
1.1.2 Ngram 2	0.89	0.36	0.71	0.73	**0.67**
1.1.3 Ngram 3	0.88	0.37	0.72	0.75	**0.68**
		1.2 Only negation stopwords kept			
1.2.1 Ngram 1	0.81	0.29	0.64	0.67	**0.60**
1.2.2 Ngram 2	0.84	0.37	0.70	0.70	**0.65**
1.2.3 Ngram 3	0.87	0.39	0.72	0.73	**0.67**
		1.3 No stopwords			
1.3.1 Ngram 1	0.80	0.33	0.62	0.66	**0.60**
1.3.2 Ngram 2	0.85	0.35	0.70	0.72	**0.65**
1.3.3 Ngram 3	0.87	0.41	0.72	0.73	**0.68**

Table 10: Smoking task 1 Multinomial Naive Bayes performance on the test set. The best models are model 1.1.3 and 1.3.3.

2. Stochastic Gradient Descent Performance

Model name	**"No information given" F1-score**	**"Current user" F1 score**	**"Previous user" F1-score**	**"Non-user" F1-score**	**Macro F1-score**
		2.1 Stopwords kept			
2.1.1 Ngram 1	0.98	0.50	0.89	0.87	**0.81**
2.1.2 Ngram 2	0.99	0.96	0.97	0.97	**0.98**
2.1.3 Ngram 3	0.94	0.87	0.99	0.91	**0.93**
		2.2 Only negation stopwords kept			
2.2.1 Ngram 1	0.98	0.49	0.89	0.87	**0.81**
2.2.2 Ngram 2	0.99	0.98	0.97	0.98	**0.98**
2.2.3 Ngram 3	0.95	0.75	0.89	0.92	**0.88**
		2.3 No stopwords			
2.3.1 Ngram 1	0.98	0.46	0.89	0.87	**0.80**
2.3.2 Ngram 2	0.98	0.98	0.98	0.97	**0.98**
2.3.3 Ngram 3	0.94	0.80	0.89	0.93	**0.89**

Table 11: Smoking task 1 Stochastic Gradient Descent performance on the test set. The best models are model 2.1.2, 2.2.2 and 2.3.2, which all use an N-gram size of 2.

3. Random Forest Performance

Model name	**"No information given" F1-score**	**"Current user" F1 score**	**"Previous user" F1-score**	**"Non-user" F1-score**	**Macro F1-score**
		3.1 Stopwords kept			
3.1.1 Ngram 1	0.98	0.18	0.00	0.43	**0.40**
3.1.2 Ngram 2	0.88	0.97	0.97	0.78	**0.90**
3.1.3 Ngram 3	0.68	0.00	0.80	0.00	**0.37**
		3.2 Only negation stopwords kept			
3.2.1 Ngram 1	0.99	0.18	0.00	0.43	**0.40**
3.2.2 Ngram 2	0.84	0.91	0.79	0.77	**0.83**
3.2.3 Ngram 3	0.61	0.00	0.00	0.00	**0.15**
		3.3 No stopwords			
3.3.1 Ngram 1	0.99	0.18	0.00	0.43	**0.40**
3.3.2 Ngram 2	0.84	0.91	0.79	0.77	**0.83**
3.3.3 Ngram 3	0.00	0.09	0.00	0.00	**0.02**

Table 12: Smoking task 1 Random Forest performance on the test set. The best model is model 3.1.2, which does not outperform Stochastic Gradient Descent's best model.

From the 3 classical machine learning approaches we tested in this study, it is clear that Stochastic Gradient Descent outperforms both Multinomial Naive Bayes and Random Forest on the multi-class classification task for smoking. A possible reason for Stochastic Gradient Descent outperforming the other models is the loss optimization that is being performed, which is absent in Multinomial Naive Bayes and Random Forest. In terms of the stopwords and n-gram sizes, it seems to be the case generally that keeping stopwords results in higher performance than leaving a portion or all of them out. For Multinomial Naive Bayes this is evidenced in table 10 by model 1.1.1, which outperforms 1.2.1 and 1.3.1, model 1.1.2 that outperforms 1.2.2 and 1.3.2 and 1.1.3 outperforming 1.2.3 and 1.3.3. For Stochastic Gradient Descent in table 11 model 2.1.1 outperforms 2.3.1 and model 2.1.3 outperforms 2.2.3 and 2.3.3. Lastly, For Random Forest in table 12 model 3.1.2 outperforms 3.2.2 and 3.3.2 and model 3.1.3 outperforms 3.2.3 and 3.3.3.

Judging the performances among the models with different n-gram sizes, n-gram size 2 outperforms n-gram size 1 and 3 virtually every time they are compared, apart from our Multinomial Naive Bayes experiments where n-gram size 3 yielded the best results, albeit these results are greatly inferior to Stochastic Gradient Descent's performance with n-gram size 2.

From these results, we identified the 4 best performing models to all be Stochastic Gradient Descent models. These models are 2.1.2, 2.1.3, 2.2.2 and 2.3.2. For our lesser performing model we chose the worst-performing Multinomial Naive Bayes model, as it performed the worst while still performing better than 50%. This is the 1.2.1 model. We trained and tested these models for the other tasks. We used the best performing model to create word clouds that show feature importance per class.

Word clouds

Figure 4: Word cloud for class "No information given""

Figure 5: Word cloud for class "Current user"

Figure 6: Word cloud for class "Previous user"

Figure 7: Word cloud for class "Non-user"

From figures 4 to 7 we can ascertain that many of our query parameters show up as important features for our best-performing model. This could indicate that the model merely learns these parameters rather than any new information.

5.4.2 Results Smoking Task 2

Model name	"Current user" F1 score	"Non-user" F1 score	Macro F1-score
		Stochastic Gradient Descent	
2.1.2 (Ngram 2, Stopwords kept)	0.95	1.00	**0.97**
2.1.3 (Ngram 3, Stopwords kept)	0.94	1.00	**0.97**
2.2.2 (Ngram 2, Less stopwords)	0.94	1.00	**0.97**
2.3.2 (Ngram 2, No stopwords)	0.93	1.00	**0.97**
		Multinomial Naive Bayes	
1.2.1 (Ngram 1, Less stopwords)	0.27	0.96	**0.62**

Table 13: Smoking task 2 full results on test set for 4 best-performing and one badly performing model. The 4 best-performing models achieve a Macro F1-score of 0.97.

As was the case for the first smoking task, the performance on the second smoking task is near flawless, this time achieving a Macro F1-score of 0.97, as seen in table 13.

Word clouds

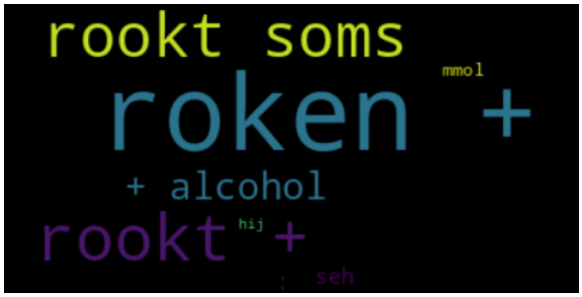


Figure 8: Word cloud for class "Current user"

Figure 9: Word cloud for class "Non-user"

In figures 8 and 9, a lot of the query parameters are again present in the images. This is particularly prevalent in figure 8, where there are not as many important features.

5.4.3 Results Alcohol Task 1

Model name	"No information given" F1-score	"Current user" F1 score	"Non-user" F1 score	Macro F1-score
		Stochastic Gradient Descent		
2.1.2 (Ngram 2, Stopwords kept)	1.00	0.99	0.99	**0.99**
2.1.3 (Ngram 3, Stopwords kept)	1.00	0.99	0.99	**0.99**
2.2.2 (Ngram 2, Less stopwords)	1.00	0.99	1.00	**0.99**
2.3.2 (Ngram 2, No stopwords)	1.00	0.97	0.99	**0.99**
		Multinomial Naive Bayes		
1.2.1 (Ngram 1, Less stopwords)	0.95	0.61	0.69	**0.75**

Table 14: Alcohol task 1 full results on test set for 4 best-performing and one badly performing model. The 4 best-performing models achieved a near flawless Macro F1-score of 0.99.

Table 14 shows even better performance than on the smoking tasks, this time achieving a Macro F1-score of 0.99.

Word clouds

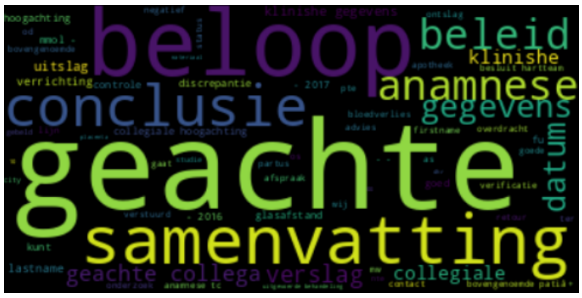


Figure 10: Word cloud for class "No information given"

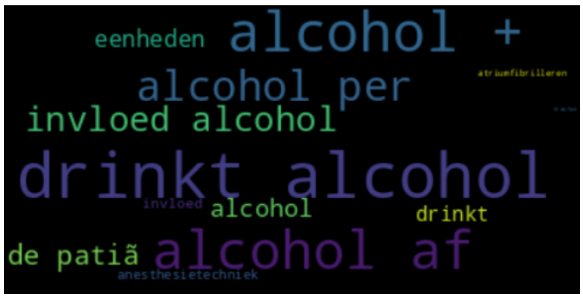


Figure 11: Word cloud for class "Current user"

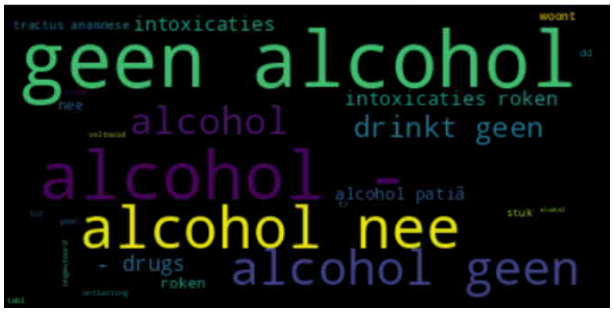


Figure 12: Word cloud for class "Non-user"

Again, figure 11 and 12 show many of our query parameters among the important features, such as "alcohol +", "geen alcohol" and "alcohol -".

5.4.4 Results Alcohol Task 2

Model name	**"Current user" F1 score**	**"Non-user" F1 score**	**Macro F1-score**
		Stochastic Gradient Descent	
2.1.2 (Ngram 2, Stopwords kept)	0.99	1.00	**0.99**
2.1.3 (Ngram 3, Stopwords kept)	0.99	1.00	**1.00**
2.2.2 (Ngram 2, Less stopwords)	0.99	1.00	**0.99**
2.3.2 (Ngram 2, No stopwords)	0.97	1.00	**0.98**
		Multinomial Naive Bayes	
1.2.1 (Ngram 1, Less stopwords)	0.60	0.94	**0.77**

Table 15: Alcohol task 2 full results on test set for 4 best-performing and one badly performing model. The best model is model 2.1.3, which achieves a rounded Macro F1-score of 1.00.

In table 15 model 2.1.3 achieves a Macro F1-score of 1.00, albeit it is not entirely flawless as the current user F1-score is lower than 1.00. It is interesting that this score is higher than for the first alcohol task, which could indicate that this task is more straightforward to grasp for a stochastic gradient descent model.

Word clouds

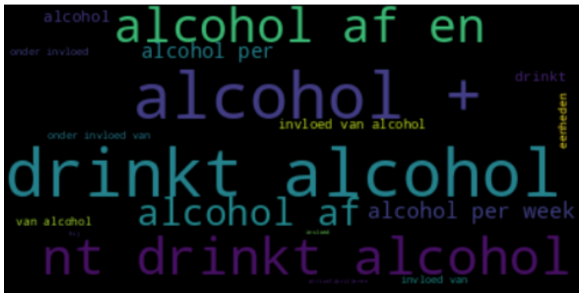


Figure 13: Word cloud for class "Current user"

Figure 14: Word cloud for class "Non-user"

As was the case for the other tasks, we see many query parameters among the top features for this task in figures 13 and 14.

5.4.5 Results Drugs Task 1

Model name	**"No information given" F1-score**	**"Current user" F1 score**	**"Non-user" F1 score**	**Macro F1-score**
		Stochastic Gradient Descent		
2.1.2 (Ngram 2, Stopwords kept)	0.97	1.00	1.00	**0.99**
2.1.3 (Ngram 3, Stopwords kept)	0.97	1.00	1.00	**0.99**
2.2.2 (Ngram 2, Less stopwords)	0.96	1.00	1.00	**0.98**
2.3.2 (Ngram 2, No stopwords)	0.78	0.99	1.00	**0.92**
		Multinomial Naive Bayes		
1.2.1 (Ngram 1, Less stopwords)	0.96	0.17	0.83	**0.65**

Table 16: Drugs task 1 full results on test set for 4 best-performing and one badly performing model. Models 2.1.2 and 2.1.3 perform the best.

As we can ascertain from table 16, models 2.1.2 and 2.1.3 achieve the highest Macro F1-score of 0.99.

Word clouds

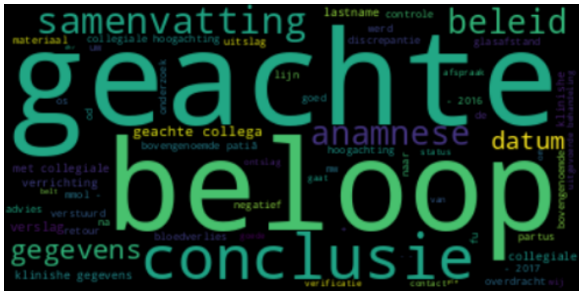


Figure 15: Word cloud for class "No information given"

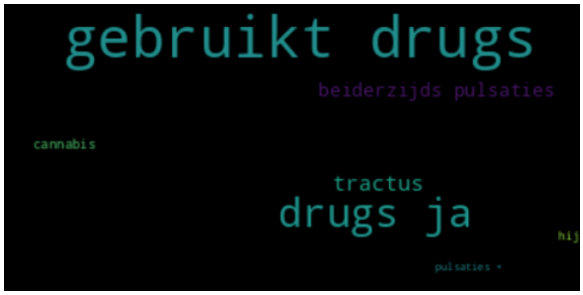


Figure 16: Word cloud for class "Current user"

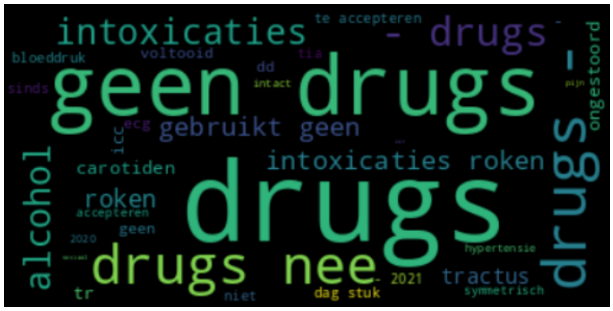


Figure 17: Word cloud for class "Non-user"

From figures 15 to 17 we can, just as with the alcohol and smoking tasks, see many query parameters existing as important features, such as "gebruikt drugs" and "drugs nee". Interestingly, it seems the mere presence of the word "drugs" seems to indicate the patient is a non-user, as evidenced by this word having the largest size in figure 17.

5.4.6 Results Drugs Task 2

Model name	"Current user" F1 score	"Non-user" F1 score	Macro F1-score
		Stochastic Gradient Descent	
2.1.2 (Ngram 2, Stopwords kept)	0.96	1.00	**0.98**
2.1.3 (Ngram 3, Stopwords kept)	0.95	1.00	**0.98**
2.2.2 (Ngram 2, Less stopwords)	0.96	1.00	**0.98**
2.3.2 (Ngram 2, No stopwords)	0.77	1.00	**0.88**
		Multinomial Naive Bayes	
1.2.1 (Ngram 1, Less stopwords)	0.13	0.94	**0.54**

Table 17: Drugs task 2 full results on test set for 4 best-performing and one badly performing model. The best-performing models are 2.1.2, 2.1.3 and 2.2.2.

From table 17, the best-performing models are 2.1.2, 2.1.3 and 2.2.2, which all achieve a Macro F1-score of 0.98. Removing all stopwords significantly hampers the model, judging by model 2.3.2. This effect is more severe here than for the other tasks, albeit for drugs task 1 this negative influence was also significant.

Word clouds

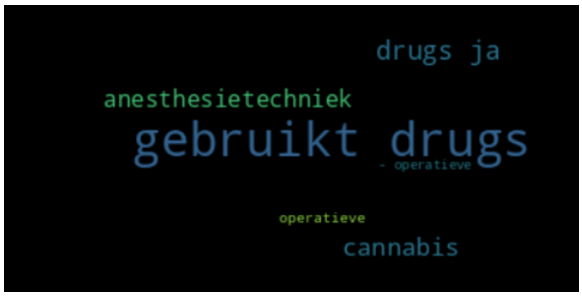


Figure 18: Word cloud for class "Current user"

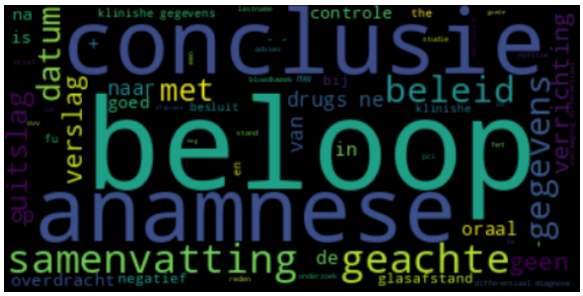


Figure 19: Word cloud for class "Non-user"

The wordclouds in figures 18 and 19 show less query parameters than the other wordclouds for the other tasks. "gebruikt drugs" and "drugs ja" are present in figure 18, but so are "anesthesietechniek" and "cannabis" which are not part of our queries. For figure 19, it seems ordinary words such as "beloop" (course) and "conclusie" (conclusion) are the most important. This could be explained by the fact that there are much less drugs users in our dataset and drug use status is far rarer to be present in a clinical text, that when the "no information given"

texts get grouped with the "non-users" that simply the most common words become the most important features. It could also be the case that these words are often used in specific types of reports that do not mention lifestyle statuses, as they do show up in "non-user" wordclouds for smoking task 2 and alcohol task 2 as well.

5.4.7 Edge Case Study

The near-perfect F1-scores on all tasks in tables 11, 13, 14, 15, 16 and 17 seem to suggest that the tasks are solvable using query-labeled texts and classical machine learning approaches. However, when examining the word clouds the most important features appear to be the query parameters, which are laid out in section 5.2. It is therefore reasonable to assume that the models merely extract the query parameters from the texts, meaning they learn that certain combinations of words lead to certain labels that correspond with our queries. In order to test this hypothesis, we collected the evaluation score per class for every text. This evaluation score is determined by the model that performed the best on that given task. In order to find edge cases, we extract the top 50 texts that were predicted to be one class with the highest score for another class, as we hold the assumption that these texts have the highest chance of having been misclassified. We then label these texts manually in order to test the accuracy of the query-assigned labels. As this is a relatively time-consuming process we only do this for the classes "Current user" and "Non-user" as these are present in both tasks for every lifestyle aspect. We furthermore only analyse the first task for each lifestyle for the same reasons.

Beforehand, in consultation with our supervisors and HagaZiekenhuis, we set a threshold of 90% overlap between the query labels and the ground truth of manually labeled texts. This means that if we find an overall accuracy which is lower than this threshold we move on to labelling texts by hand rather than using labels that were assigned by string matching. We used the best performing model per task, this being the model with the highest Macro F1-score, to assign scores per class to the texts. For each experiment per task we present the class distribution of the hand-labelled ground truth and we provide a sample of 5 edge case texts on which the model assigned the wrong label. For these edge cases, all text belonging to the same row belongs to the same clinical note. Gaps in the text are denoted by "...".

5.4.7.1 Results Smoking

In this section we provide the results of our edge case study on the smoking task. As explained previously, we extract the top 50 texts that were predicted to be one class, but had a high prediction score for another class. We label these texts manually. We calculate accuracy by calculating how many of these edge cases actually belong to the class that was predicted for them by the model. For example, when we extract 50 texts that have been predicted to be non-users and after a manual review only 10 of these texts actually belong to the non-user class the accuracy is equal to 20% (10/50).

Top 50 texts with highest scores for "Current user" class that were predicted "Non-user"

True labels	Amount
No information given	30
Current user	5
Non-user	**10**
Previous user	5

Table 18: Manually picked labels for the smoking task for 50 texts that were predicted to belong to the "non-user" class. These texts were the texts that had the highest prediction score for the "current user" class while still obtaining the prediction "non-user".

Table 18 shows an accuracy of 20%, as only 10 of the 50 texts actually belonged to the class that was predicted for them.

Edge cases from Dutch clinical notes	Translation	True label
... Intoxicaties: roken 2-4 sigaren per dag alleen in de zomer, ...	Intoxications: smokes 2-4 cigars per day only in the summer,	Current user
... Herseninfarct ACP rechts bij risicofactoren hypertensie, roken. ...	Cerebral infarction ACP right in risk factors hypertension, smoking	No information given
... roken (+ (20 packyears, is gestopt); ... Risicofactoren: Familieanamnese (+); roken (-) ...	smoking (+ (20 packyears, has stopped); ... Risk factors: Family history (+); smoking (-)	Previous user
... Roken: 3-4 per dag al jaren lang, vroeger wel meer ...	Smoking: 3-4 a day for years, in the past more	Current user
... Roken: - (20 jaar geleden na 50 py), Cardiovasculaire risicofactoren: roken:+ ...	Smoking:- (20 years ago after 50 py), Cardiovascular risk factors: smoking:+	Previous user

Table 19: 5 exempts from the smoking task from texts that were predicted to belong to the non-user class but were misclassified.

Table 19 shows instances where our classical machine learning model fails. For example in the first text, the minus sign is used as an indicator that the patient smokes 2 to 4 cigars per day, but it gets picked up as meaning that the patient is not a smoker, most likely because of the presence of both "roken" (smoking) and "-" in the same text. Furthermore, qualifiers in parentheses are ignored, as evidenced by the third and fifth texts, that should both be classified as previous users.

Top 50 texts with highest scores for "Non-user" class that were predicted "Current user"

True labels	Amount
No information given	25
Current user	**10**
Non-user	9
Previous user	6

Table 20: Manually picked labels for the smoking task for 50 texts that were predicted to belong to the "current user" class. These texts were the texts that had the highest prediction score for the "non-user" class while still obtaining the prediction "current user".

Table 20 again shows a poor accuracy of 20%, with 10/50 correct predictions among this set.

Edge cases from Dutch clinical notes	Translation	True label
... roken -, mee roken + ...	smoking -, passive smoking +	Non-user
... Roken: - (nooit) Passief roken: + (vroeger) ...	Smoking: - (never) Passive smoking: + (past)	Non-user
... Roken -, moeder COPD bij roken + ...	Smoking -, mother COPD with smoking +	Non-user
... roken -, 35 pack years. ...	smoking -, 35 pack years	Previous user
... vader (roken +) ... RF; roken – ...	father (smoking +) ... RF: smoking -	Non-user

Table 21: 5 exempts from the smoking task from texts that were predicted to belong to the current class but were misclassified.

From table 21, we can again ascertain that the queries we used are not all-encompassing, as phrases like "passive smoking +", "mother smoking +" and "father (smoking +)" lead to the patient being classified as a smoker, while this is unwanted behaviour. Also, just like in table 19 the fourth text shows that qualifiers are not taken into account, which possibly shows a need for deeper models that can grasp context better.

These accuracies on the smoking task show how error-prone query labelling can be, of 100 hand-labeled texts only 20% of the texts were assigned a label that is correct. This is a stark contrast with the high F1-scores from the previous section, suggesting that the models solely learn how the queries were set up, which in turn caused them to produce these errors. Edge cases appear to include a lot of flaws with the query, such as classifying a patient as a smoker when "passive smoking +" appears in the text, whilst this is unwanted behaviour, or not recognizing past smokers in the structure "Smoking + (stopped since...)". Naturally, these edge cases are in accordance with the query that was used as these specific circumstances were not accounted for.

5.4.7.2 Results Alcohol

In this section, we conduct the same experiments for the alcohol multi-class classification task.

Top 50 texts with highest scores for "Current user" class that were predicted "Non-user"

True labels	Amount
No information given	2
Current user	26
Non-user	**22**

Table 22: Manually picked labels for the alcohol task for 50 texts that were predicted to belong to the "non-user" class. These texts were the texts that had the highest prediction score for the "current user" class while still obtaining the prediction "non-user".

Table 22 shows an accuracy of 44% with 22 out of 50 correct predictions.

Edge cases from Dutch clinical notes	Translation	True label
... De patiënt drinkt alcohol (bier), af en toe niet altijd. Opmerkingen mbt alcohol: geen alcohol verslaving. ...	The patient drinks alcohol (beer), occasionally, not always. Notes regarding alcohol:no alcohol addiction.	Current user
Alcoholabusus 2014 SEH-presentatie vanwege alcohol-intoxicatie ... 2020-07: trauma capitis na val van trap bij alcoholintox en speedgebruik. ... Zegt zelf vandaag 3 flessen wijn gedronken te hebben. ...	Alcohol abuse 2014 emergency room appearance due to alcohol intoxication ... 2020-07: trauma capitis after falling down stairs in alcohol intoxication and speed usage. ... Says they drunk 3 bottles of wine today...	Current user
... Patiente gebeld, uitleg gegeven over bloeduitslagen en invloed van alcohol op cognitie en voedinstoestand. ... Geeft aan door te gaan met begeleiding van de dietiste en te zullen minderen met alcohol....	Called patient, explained about blood results and influence of alcohol on cognition and nutritional status. ... Indicates to continue with the guidance of the dietician and to cut down on alcohol.	Current user
... Alcohol- Alcohol: ja; Soort alcohol: wijn; Hoeveelheid alcohol: minder dan 2 eenheden per dag; ...	Alcohol- Alcohol: yes; Type of alcohol: wine;Amount of alcohol: less than 2 units per day;	Current user
... Intoxicaties: roken in verleden, drinkt 3 eenheden alcohol per dag ... - advies te stoppen met alcohol. ...	Intoxications: smoking in past, drinks 3 units of alcohol per day ... - advised to quit drinking alcohol.	Current user

Table 23: 5 exempts from the alcohol task from texts that were predicted to belong to the non-user class but were misclassified.

Table 23 shows that the model used is unable to understand the context of these particular texts, not seeing that these texts all belong to current alcohol users rather than the non-user prediction that was assigned. From examining these exempts, it is not entirely clear why the model decides to classify these texts as such, although the fourth and fifth texts do show a minus sign being used in a different way than showing that the patient is a non-user.

Top 50 texts with highest scores for "Non-user" class that were predicted "Current user"

True labels	Amount
No information given	0
Current user	**44**
Non-user	6

Table 24: Manually picked labels for the alcohol task for 50 texts that were predicted to belong to the "current user" class. These texts were the texts that had the highest prediction score for the "non-user" class while still obtaining the prediction "current user".

Table 24 shows an accuracy of 88% (44/50).

Edge cases from Dutch clinical notes	Translation	True label
... Intoxicaties: De patiënt rookte en is gestopt met roken. De patiÃnt drinkt geen alcohol. De patiÃnt gebruikt geen drugs. ...	... Intoxications: The patient smoked and has stopped smoking. The patient does not drink alcohol. The patient does not use drugs. ...	Non-user
... Intoxicaties: De patiÃnt heeft nooit gerookt. De patiÃnt drinkt geen alcohol. De patiÃnt gebruikt geen drugs. ... Alcohol per week: Minder dan 5 eenheden ...	... Intoxications: The patient has never smoked. The patient does not drink alcohol. The patient does not take drugs. ... Alcohol per week: Less than 5 units ...	Non-user
... De patiënt drinkt alcohol (bier), Zit in breijder voor alch detox.Opmerkingen mbt alcohol: Gestopt sinds 2,5 a 3 weken. .De patiënt gebruikt geen drugs. ...	The patient drinks alcohol (beer), goes to Breijder for alch detox. Comments regarding alcohol: Stopped since 2.5 to 3 weeks. The patient does not use drugs.	Non-user
... chronisch alcohol gebruik al vele jaren. fles wijn per dag, soms meer. vanaf zondag opeens niks meer gedronken, is resoluut gestopt. Intoxicaties: roken: , alcohol: 1972 start alcohol +++, sinds drie weken nu gestopt ...	chronic alcohol use for many years. bottle of wine a day, sometimes more. suddenly stopped drinking on a Sunday, resolutely. Intoxications: smoking: , alcohol: 1972 start alcohol +++, stopped since three weeks	Non-user
... Intoxicaties Roken 41PY, Alcohol af en toe, al jaren gestopt Drugs nooit ...	Intoxications Smoking 41PY, Alcohol occasionally, stopped since some years. Drugs: never	Non-user

Table 25: 5 exempts from the alcohol task from texts that were predicted to belong to the current class but were misclassified.

In table 25 we show 5 texts that have been classified as current user, but actually should have been classified as non-users. Interestingly, we find a lot of previous users (texts 3, 4 and 5). As our queries did not encompass previous users for alcohol and drugs they should be regarded as non-users here. Some texts contain statements that seemingly contradict each other. For example, in the second text they wrote that the patient has never drunk alcohol, but later on states an intake of less than 5 units. We speculate the model views this latter statement as an indicator for a current user while it is probably included as 0 units is also less than 5. Texts 3, 4 and 5 furthermore contain similar "inconsistent" phrases, which are obvious when a human reads the text but could pose problems for a machine learning approach that is relatively bad at judging context, as is the case for classical machine learning methods and string matching.

Overall for the alcohol task, the accuracy on alcohol labels is significantly higher than on smoking, even almost reaching our threshold of 90% for current users with a high non-user score. This indicates that the task of labelling on the basis of alcohol is either easier to do than smoking or that our query was more effective in correctly labelling for alcohol.

5.4.7.3 Results Drugs

In this section, we conduct the same experiments for the drugs multi-class classification task as we performed for the smoking and alcohol tasks.

Top 50 texts with highest scores for "Current user" class that were predicted "Non-user"

True labels	Amount
No information given	18
Current user	9
Non-user	**23**

Table 26: Manually picked labels for 50 texts that were predicted to belong to the "non-user" class. These texts were the texts that had the highest prediction score for the "current user" class while still obtaining the prediction "non-user".

Here, table 26 shows an accuracy of 46%, (with 23 out of 50 correct predictions.

Edge cases from Dutch clinical notes	Translation	True label
... Alcohol- Alcohol: nee; Drugs- Drugs: ja; Drugs- Soort drugsgebruik: Cannabis; ...	Alcohol- Alcohol: no; Drugs- Drugs: yes;Drugs- Type of Drug Use: Cannabis;	Current user
... Drugs- Drugs: ja; Drugs- Soort drugsgebruik: Cannabis; Opmerkingen drugs: 1 x per weekend; ...	Drugs- Drugs: yes; Kind of drug: Cannabis; Comments drugs: 1 x per weekend;	Current user
... Drugs: ja, hoeveel: 4x blowen per dag Alcohol: nee ...	Drugs: yes, how much: 4 times a day Alcohol: no	Current user
... [Drugs]-Drugs: ja[Drugs]-Soort drugsgebruik: Cannabis-Hoeveelheid drugs: 2 joints/dag-Toedieningsweg drugs: pulmonaal ...	[Drugs]-Drugs: yes[Drugs]-Type Drug Use: Cannabis Quantity of drugs: 2 joints/day-Method of ingestion: pulmonary	Current user
[Drugs]-Drugs: ja[Drugs]-Soort drugsgebruik: Cannabis-Gebruik sinds: 1990	[Drugs]-Drugs: yes[Drugs]-Type of drug use: Cannabis-User since: 1990	Current user

Table 27: 5 exempts from the drugs task from texts that were predicted to belong to the non-user class but were misclassified.

Table 27 shows that minus signs are often misinterpreted, as is always the case in many of the other exempts we analysed. This corresponds to our string matching queries, as they would also classify these texts as non-users, even though the minus signs should not be interpreted this way at all.

Top 50 texts with highest scores for "Non-user" class that were predicted "Current user"

True labels	Amount
No information given	0
Current user	**50**
Non-user	0

Table 28: Manually picked labels for the drugs task for 50 texts that were predicted to belong to the "current user" class. These texts were the texts that had the highest prediction score for the "non-user" class while still obtaining the prediction "current user".

All of these labels were correct, so the accuracy is 100% (50/50).

The results on the drugs tasks show, similarly to the alcohol tasks, that our query better encapsulates the different forms in which drug usage is indicated in the notes.

Accuracy all tasks: 53% (159/300)

5.5 Conclusions

In this feasibility study, we show that query labelling does not perform up to our standards. It achieved an accuracy of 53%, whilst we set a threshold of 95% for us to consider these labels further in this thesis. The query labels are most often wrong within the smoking tasks, showing that either it is hard for a query to grasp the many different variations doctors tend to report smoking status or our query for labelling smoking was not extensive enough. Examining the edge cases for smoking, we find many different formats for smoking status reporting. Many of the false positive labels were caused by the respective doctor using the minus symbol (-) as a delimiter, which the model seemed to pick up as meaning that a "non-user" label should be assigned. Furthermore, as can be expected from string matching labels, the model did not seem to grasp any context as passive smoking was consistently used as an indicator for a current user whilst this is unwanted behaviour.

Additionally, when it was indicated that one of the patient's parents used a substance the patient was wrongly classified as a user as well. Wrong labels were also assigned to previous users who were noted to having a risk group of smoking due to past use, these patients were almost always classified as current users. For the other tasks, there were many cases where minus symbols used as bullet points also seemed to influence the model into classifying a negative where a positive should be. For the false positives, it appears patients that have stopped using the substance are still being classified as current users often.

It should be noted that the query label models for the alcohol and drugs tasks performed significantly better than on the smoking task. This could indicate these tasks are more straightforward to grasp using query labelling or that the specific queries we used on these tasks are more suited to the problem-at-hand.

As we deemed our query labels unsuitable for this thesis, we move towards creating hand-labeled input. For this, we use the insights gained in our systematic literature review. We explain our hand-labelling process in section 6.1. The query labels are still used however as they serve as our string matching model and are compared to classical machine learning and BERT-like approaches on the manually annotated dataset.

6 Modelling and Experiments

This section corresponds to the Data Preparation and Modelling phase of CRISP-DM. Here, using the knowledge we gained in the previous phases we lay out the methods we use in this thesis, as well as provide an overview of how these models will get tested and how their performance will be assessed. We firstly provide an overview of our manual labelling approach following our negative assessment of query labels, which is part of Data Preparation. We then elaborate on our string matching and classical machine learning setups and provide an overview of our BERT setups. Lastly, we give an overview of our general testing setup, showing what the models are being tested on and how we evaluate performance.

6.1 Manual labelling

We established in the previous section that query labelling does not adhere to the standards we have for our input, meaning we move towards annotating the data by hand. We use the same classes and subclasses as within the query labelling feasibility study, meaning we differentiate between current users, non-users, previous users and users where no information about the respective lifestyle characteristic is given. Contrary to the feasibility study however, as we encountered previous alcohol users during our edge case study we label all lifestyles on previous users rather than only regarding previous users for the smoking lifestyle. This way we hope our models are able to identify previous alcohol and drugs users as well.

When labelling, we view each text as a self-contained comprehensive entity about one patient. This means that a text should be approached as if it contained the full context about one patient and that obtaining more information about this patient outside the text is impossible. The text is thus effectively the complete framework of everything we know about a patient. The text should also be seen as the most recent piece of information available about the patient. For example, if the text states that the patient will stop smoking on May 4, 2019 (and the text dates from before May 4, 2019), then this should be regarded as a future expression of quitting and the patient should be regarded as a current smoker, even when the date of labelling is long after May 4, 2019. As explained in section 2.2, we choose this approach in order to improve reproducibility for future research. For every label we provide an overview below of which patients should get assigned which label and why.

Current user: A patient is a current user if, at the time of writing, they are an active user of the substance. The patient is NOT a current user if the text says they have stopped, but IS when they plan to stop, for example. Frequency of use does not matter, for example, no distinction is made between a usage of once per year and once per day. This also means that people that indicate that they only use the substance "at parties" or the like are regarded as current users. In case of doubt, we rely on the suspicion of the doctor in question, if that can be inferred from the text. For example, if the text states that the patient does not answer questions but the doctor indicates that they smell strongly of alcohol, this means that we regard the patient as an alcohol user.

Previous user: A patient is a former user when it is clearly stated in the text that, at the time of writing, they have stopped taking the substance in question. The time between the moment of stopping and the moment of writing is not important here. For example, a

"Smoker in a distant past" belongs to this class. When a patient says they intend to stop or is advised to stop in the text, we classify this patient as a "Current User". This is also the case for patients who had stopped and started using the substance again. This label is therefore only for patients for whom it can with certainty be inferred that this person has stopped taking the substance and is no longer using it at the time of writing.

Not a user: A patient is not a user if, at the time of writing, they are not an active user of the substance in question and it is not stated in the text that they have stopped using it. If only a negative judgment is given in the text about a substance (for example "does not smoke") and nothing is said in the text about the patient's history with the substance, this label should be assigned.

Nothing found: If nothing can be found in the text about the patient's usage of the substance, we assign this label. For example, it may be the case that a text is about a patient's X-ray scan and does not mention any lifestyle characteristics. It also occurs that texts do contain information about smoking and alcohol but not about drugs, so in that case this label will have to be assigned to "drug usage". Furthermore, for texts with unclear or conflicting indicators, the "Nothing found" label should always be chosen. This is the case, for example, when there is 'smoking +' somewhere in a text and later on it says 'smoking -' and this concerns the same patient, with no further elaboration being done as to why these conflicting statements are present. It also occurs that the substances are listed but not filled in, such as: "Smoking: Alcohol: Drugs: ", this also falls under "Nothing found" for each substance.

We draw these clear distinctions in order to reduce the overlap between classes as much as possible with the goal of it becoming easier for models to learn the differences between the classes. In terms of the lifestyle classes smoking, alcohol use and drug use, we set the following guidelines for our annotators:

Smoking: Smoking means smoking tobacco in every conceivable way. Smoking other substances such as cannabis should fall under drug use. The smoking status of the patients appears in many texts, in many different forms. The shape is often not standardized, requiring a complex model. Most common indicators are "smoking +/-" where a + indicates a current user and a – indicates a non-user and "patient smokes (not)". The dataset also contains much less common notations such as "regularly smokes", "stopped smoking", "hardly smokes anymore" et cetera.

Alcohol use: Alcohol use refers to the consumption of alcoholic beverages such as beer, wine and so on. Most common indicators include "alcohol +/-" and "patient drinks (no) alcohol".

Drug use: With drug use we mean the recreational (non-medicinal, non-prescribed) use of either soft or hard drugs. It is not always the case that drug status is mentioned in the text, most likely due to the low number of users compared to smoking and alcohol. This is also the reason why when drug status is mentioned in the text, it is often negative.

Naturally, for pretraining BERT models these labels do not need to be present as it concerns an unsupervised task. The labels are however essential for our finetuning phase and the

solution to our problem at hand, that being lifestyle classification. For finetuning, judging from the literature review, in order for a BERT model to perform well when finetuned there need to be at least around 1.500 annotated samples, as training set with fewer samples tended to lead to lower performance (Dipnall et al., 2022; Spruit et al., 2022; Zhan et al., 2021).

As there is limited time to obtain annotations and limited availability in terms of people that can annotate there need to be some considerations made regarding obtaining labelled data. There were three human annotators in total involved in this thesis. Next to the author of this paper (Annotator 1), another student (Annotator 2) and a healthcare professional (Annotator 3) were involved. After labelling, Annotator 1 served as reviewer, reviewing both annotators' labels and resolving conflicts between them.

Due to the relative easy nature of the labelling task and the high agreement in the end, which we touch upon later, the amount of annotators and their qualifications are deemed sufficient for this thesis. In total, 4.700 texts were labelled on smoking, alcohol use and drug use status. An overview of the amount of texts per set and annotator can be seen in the table below. For each class, the best "standard" machine learning approach per class was used to predict scores for each type of label, similarly to the query labelling feasibility study. In that study, we showed that the top 50 of scores for the class that was not predicted for the text relatively contain a lot of edge cases. For this reason, we used these sorted lists to comprise parts of the full dataset that was used for labelling.

Set ID	**Annotator**	**# of texts**	**# of current smoker texts**	**# of non-smoker texts**	**# of previous smoker texts**	**# of current alcohol user texts**	**# of non-alcohol-user texts**	**# of current drug user texts**	**# of non-drugs-user texts**	**# of random texts**
cd_1	1	1.000	100	100	100	100	100	100	100	300
cd_1.1	2	*500*	*46*	*45*	*50*	*54*	*49*	*44*	*49*	*163*
cd_1.2	3	*500*	*54*	*55*	*50*	*46*	*51*	*56*	*51*	*137*
cd_2	1	1.000	100	100	100	100	100	100	100	300
Random	1	2.700	-	-	-	-	-	-	-	2.700
Total		4.700	200	200	200	200	200	200	200	3.300

Table 29: Overview manually labeled datasets. This table shows which datasets we have and what the distribution within the datasets are in terms of lifestyles and ranndom texts.

As shown in table 29, we created three datasets in total, where we split Combined Dataset 1 into two subsets for our other annotators. Combined Dataset 1 (cd_1) and Combined Dataset 2 (cd_2) consist of 1.000 texts each. These datasets are comprised of 700 hand selected texts and 300 random texts. The hand-selection process was performed by obtaining the texts that had the highest score for the respective classes, but were not predicted by the best query label model for that particular lifestyle. For example, for the current smoker texts we obtained the top 100 texts on the basis of evaluation score for the "Current User" class, where the model nonetheless predicted the texts to belong to another class. We showed in section 5.5.7 that texts that are obtained this way contain relatively a large amount of edge cases. By including these edge case texts in our dataset, we believe we make our dataset more suitable for lifestyle classification, as more writing styles and contradictions are accounted for.

While comprising the dataset, we only used texts that were not already present in the dataset. This means that when evaluating the top 100 texts for current alcohol users we only considered texts that were not part of the previous smoker, non-smoker and current smoker texts that were already present in the dataset at that point. We supplemented the 700 handpicked texts with 300 random texts to serve as a kind of control group. These random texts, and the random texts in the "Random" dataset, are picked by randomly shuffling the collection of texts that are not yet part of another dataset and extracting the respective amount of texts from this shuffled set.

For cd_1, we had Annotator 1 annotate the whole dataset, with Annotator 2 and Annotator 3 each annotating a half of the dataset respectively. The dataset was split in half randomly and we had Annotator 2 label set cd_1.1 and Annotator 3 labeled cd_1.2. We calculated the inter-rater reliability on cd_1 in order to show the quality of the labels in this dataset. We use Cohen's Kappa to calculate the agreement, which is given by the equation below (McHugh, 2012).

$$k = \frac{p_o - p_e}{1 - p_e}$$

Here, p_o is the probability of agreement, which is calculated by dividing the total amount of agreements between the annotators with the total amount of texts. p_e denotes the hypothetical probability of a random match, given the answers of the annotators. Calculating the random match probability between two annotators can be done via the following formula (McHugh, 2012):

$$p_e = \frac{1}{N^2} \sum_k n_{k1} n_{k2}$$

Where N denotes the amount of observations to categorize, k the amount of categories and n_{ki} the amount of times annotator i predicted category k.
We interpret the results using Krippendroff's alpha coefficient interpretation scale (Krippendorff, 1980). This metric says that a Cohen's Kappa value of less than 67% denotes that the reviewed items should get discarded, a value of between 67% and 80% that the agreement is tentative and a value of 80% and above that there is a good or reliable agreement. The reliability for cd_1 between Annotator 1 and Annotator 2 on dataset cd_1.1, and Annotator 1 and Annotator 3 on dataset cd_1.2 can be found in the tables below.

	Reviewer	**Annotator 1**	**Annotator 2**
Reviewer		97.57%	96.81%
Annotator 1	97.57%		96.80%
Annotator 2	96.81%	96.80%	

Table 30: Cohen's Kappa values between Annotator 1 and Annotator 2 on dataset cd_1.1. As these values are above 80% we can say the agreement is reliable.

	Reviewer	Annotator 1	Annotator 3
Reviewer		98.09%	97.09%
Annotator 1	98.09%		97.09%
Annotator 3	97.09%	97.09%	

Table 31: Cohen's Kappa values between Annotator 1 and Annotator 3 on dataset cd_1.2. As these values are above 80% we can say the agreement is reliable.

Given that the inter-rated reliability is (far) above 0.8 in tables 30 and 31, we consider this labeled dataset to be very reliable. Furthermore, because the reliability is so high, the task-at-hand apparently being relatively easy, and because of the limited availability of our annotators for further annotation we decided to annotate the rest of our datasets ourselves. The other datasets consist of another dataset comprised the same way as cd_1, but with texts that are not included there, named cd_2 and a large dataset containing 2.700 randomly selected texts that were not already part of any of the other datasets.

Ultimately, we combine these datasets into one full dataset containing 4.700 labeled samples, which we use as input for the string matching, classical machine learning and BERT finetuning tasks. As explained previously, every one of these texts is labeled on smoking, alcohol and drug use status, meaning we have 4.700 labeled texts per lifestyle. We therefore split the full dataset into 3 datasets, each corresponding to a lifestyle label. The datasets contain the following class distribution:

Dataset name	Amount of labelled texts	Current users	Previous users	Non-users	No information given
Smoking	4,700	179 (3.81%)	939 (19.98%)	439 (9.34%)	3,143 (66.87%)
Drinking	4,700	616 (13.11%)	19 (0.40%)	744 (15.83%)	3,321 (70.66%)
Drugs	4,700	60 (1.28%)	11 (0.23%)	1,272 (27.06%)	3,357 (71.43%)

Table 32: Class distribution for the texts of each of the lifestyle tasks, these being smoking, alcohol use and drugs use.

Observing the class distribution in table 32, the data seems to be heavily skewed. While this is the case, it was hard to find previous users for alcohol and drugs because this is apparently such a rare phenomenon for a doctor to note given the amount of random texts we included in our dataset. For this reason, we believe our dataset represents the actual situation, meaning there are a relatively few cases where being a previous alcohol or drugs user is actually relevant enough to be included in a clinical note. It is still interesting to explore which model is most robust to a low amount of samples per class and as we are using Macro F1-score to evaluate our models this means it is significantly important for a model to be able to classify on the basis of these rare classes.

6.2 String Matching

We apply string matching models in order to be able to answer our second research subquestion, which is "To what extent can the task of lifestyle classification of clinical notes be solved

using string matching?". For our string matching method, we use our query labels obtained from the query layed out in section 5.2. For every text in our hand-labeled dataset, we acquired its assigned query label and compared it with the manually assigned label, which is the ground truth. In this thesis we make use of a training and test set. Even though a training set is not relevant for string matching, the method does not "learn" anything and therefore does not need a training set, we still only evaluate string matching on our test set for clarity's sake, as it is uniform and the other models are evaluated on the same set. We elaborate more on our evaluation method and the comprising of our training and test sets in section 6.5.

The evaluation is done using the *sklearn* library in Python, which we used before in our exploratory data analysis in section 5. Next to our query labels, we furthermore use the method of labelling of Heath (Heath, 2022), so that we can compare our models to their solution. As their solution only concerned current and previous smokers we only evaluate the method on those aspects. They disregarded texts that we regarded as "non-users" as "previous users", so in the test set we convert every non-user to a previous user. Given that application of their queries does not label every text in our test set we discard the texts that are not labeled. This means the test set for this method in particular is significantly smaller than the test set for the other models. Note that, while these adjustments might not allow for the fairest comparisons due to the significant differences in experimental setups, it is still interesting to compare results to their research as their research contains a HagaZiekenhuis case study with a string matching approach as well. Lastly, as the CTCue query labels do not include previous users for alcohol and drugs we do not evaluate the query labels string matching on those aspects.

6.3 Classical Machine Learning Approaches

We test classical machine learning approaches in order to be able to answer our third research subquestion "To what extent can the task of lifestyle classification of clinical notes be solved using classical machine learning methods?". For our classical machine learning approaches, we take a similar approach as to our exploratory data analysis. As recreating the full experiment using the hand-labeled dataset would not fit within our time constraints we opted instead for reusing the setups that performed best on the query labels and tested their performance. We do not reuse the exact same models, but choose to rerun the random search using the best performing setup. For example, for smoking the Stochastic Gradient Descent model where we set the ngram size to 2 and kept all of the stopwords performed the best, so in this experiment we use a Stochastic Gradient Descent classifier with those settings and determine the other parameters via random search.

In our query label analysis we did not have "Previous users" for the alcohol and drugs lifestyle tasks. As we do have previous users for these lifestyles in our hand-labeled dataset we use the best-performing task 1 models, which concerned the multi-class classification setups rather than the binary classification setups. Table 33 shows the best performing setups per lifestyle which we use to train on the hand-labeled dataset.

Lifestyle	Type of classifier	Ngram size	Stopword status
Smoking	Stochastic Gradient Descent	2	Kept
Alcohol	Stochastic Gradient Descent	2	Only negation
Drugs	Stochastic Gradient Descent	2	Kept

Table 33: Best performing setups for classical machine learning for each lifestyle.

As stated before, we use random search to test and pick the rest of the parameters of the model. Which random search parameters are used corresponds to those for Stochastic Gradient Descent in the query label study, and these are restated in tables 34 and 35.

TF-IDF

Parameter	Possible values
max_df	0.90, 0.95
min_df	3, 5

Table 34: Possible parameter values for TF-IDF that are being tested in random search.

Stochastic Gradient Descent

Parameter	Possible values
loss	('hinge', 'log loss', 'modified huber', 'squared hinge' , 'perceptron', 'squared error', 'huber', 'epsilon_insensitive', 'squared_epsilon_insensitive')
penalty	('l2', 'l1')
l1_ratio	Random float between 0 and 1.
fit_intercept	(True, False)
max_iter	(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)
tol	Random float between 0 and 1.
shuffle	(True, False)
epsilon	Random float between 0 and 1.
learning_rate	('constant', 'optimal', 'invscaling', 'adaptive)
eta0	Random float between 0 and 1.
power_t	Random float between 0 and 1.
class_weight	('balanced', None)
warm_start	(True, False)
average	(True, False)

Table 35: Possible parameter values for Stochastic Gradient Descent that are being tested in random search.

Similarly to the string matching implementation, we use Python to perform the random search and train the model, and use the *sklearn* library in order to calculate the Macro F1-score.

6.4 BERT

We experiment with BERT-like methods in order to answer our fourth research subquestion "To what extent can the task of lifestyle classification of clinical notes be solved using BERT-like models?" and to ultimately be able to answer our main research question "To what extent can the most appropriate BERT language model improve upon classical machine learning approaches and string matching on the task of classifying patient lifestyle statuses in free text Dutch clinical notes?".

In this section we lay out our BERT implementation, following the systematic literature review we conducted on the topic. Contrarily to Heath's thesis, we have access to high computing power in the form of the SHARK HPC cluster, which we cover in section 6.4.1. We use this cluster in order to pretrain our ALBERT-like model from scratch, the procedure of which is elaborated on in section 6.4.2. In section 6.4.3, we explain our pretraining process where we pretrain on top of BERT models RobBERT, belabBERT and MedRoBERTa.nl, which are the Dutch BERT models we found in our literature review. Section 6.4,4 describes our approach for translating the texts from Dutch to English and using ClinicalBERT and BioBERT and section 6.4.5 describes our common finetuning approach which will be applied to all models.

6.4.1 Elaboration SHARK

In order to be able to pretrain and finetune our BERT models we were given access to the SHARK High Performance Computing (HPC) cluster of Leids Universitair Medisch Centrum (LUMC). A HPC cluster is a group of servers called nodes. Each of these nodes are connected through a fast network. There are multiple different types of nodes with different uses, such as head nodes, login nodes, GPU nodes and storage nodes. To run jobs on SHARK they make use of the SLURM workload manager and scheduler [1]. SLURM is an open source cluster management and job scheduling system which allocates access to resources on compute nodes on SHARK. Using SLURM, we can start, execute and monitor jobs for purposes of pretraining, translation and finetuning for example.

To create these jobs we use their Open OnDemand web interface, which provides an integrated, single access point for HPC resources. From this interface, we use Jupyter Notebooks in order to test our jobs before submitting them to the SHARK. After testing functionalities in a Jupyter Notebook, we create Python scripts which we initialize in SLURM job files. Then, using a shell from which we can access SHARK, we add the jobs to the queue. Within a SLURM file, we can specify which SHARK partition to use. SHARK has many available partitions which all serve their own purpose. Within this thesis we mostly use the "gpu" partition, as it allows for fast GPU calculations. As this partition has a relatively long queue because of its high demand amongst SHARK users, we only submit jobs to it that we have tested before on a smaller, more available partition. We also connect to the HuggingFace API in the SLURM file. This API will serve as our method for saving BERT models online after pretraining/finetuning. We also use this API for tokenizing our texts, which we will explain in the following section.

We mostly make use of the *res-hpc-gpu01* node of SHARK to run our experiments. This node has 48 CPU cores, 512 gigabytes of memory and 3 available GPUs. For the purpose of

[1] https://slurm.schedmd.com/documentation.html

testing our code before we run a full experiment we make use of a smaller execution node which contains 24 CPU cores, 384 gigabytes of memory and no GPUs.

6.4.2 Experiment 1: Pretraining from scratch

As we laid out in our systematic literature review in section 4, we pretrain our from-scratch BERT model using the ALBERT architecture. The ALBERT architecture allows for us to use the optimal parameters found in (de Wynter & Perry, 2020). It also makes sure we use the pretraining tasks of MLM with dynamic masking and the task of consecutive sentence order prediction. ALBERT also applies cross-layer parameter sharing by default.

We refer to our model pretrained from scratch as HAGALBERT, combining the name of HagaZiekenhuis en the name of the ALBERT model architecture. The input that we use to train HAGALBERT consists of the entire bulk of 148.000 texts that we obtained from HagaZiekenhuis. As ALBERT's maximum input size is 512 tokens per document, we had to find a way to split the texts into chunks with a maximum size of 512 characters. We do this by applying a so-called sliding window. This sliding window considers a current sentence and the two sentences that come after it and adds these three sentences in total to a result list, this means the last sentence of the first entry is the second sentence of the second entry and so on. We end up with a large list containing entries of three sentences each, which helps us preserve context within consecutive sentences. In order to be able to record the performance of our models during training we split this large list into a training and test set with the ratio 80/20.

We load both lists in the Shark Python file. We then combine the lists in one HuggingFace Dataset, which is easier to use for HuggingFace tokenizers. Within HuggingFace, tokenizers prepare inputs for models by splitting strings in sub-word token strings, as explained in our section on BERT in section 3.5. Furthermore, tokenizers add special tokens like ¡MASK¿ and line separators. In accordance with our plan to use the ALBERT architecture we make use of the ALBERT tokenizer setup, which can be loaded in using HuggingFace. We train this tokenizer using our sliding window of texts and use a standard vocabulary size of 30.000.

Using the tokenizer, we tokenize the dataset so it can serve as input for HAGALBERT. We then initialize an empty ALBERT model using the *"albert-base-v2"* identifier on HuggingFace. For this model, we set several training arguments in correspondence with the optimal parameters found in paper 21 of our literature review. The rest of the arguments were set to correspond to ALBERT's standard parameters as the base ALBERT setup showed to outperform BERT in its respective paper using those parameters (Lan et al., 2019). An overview of the training arguments used to train HAGALBERT can be found in table 36.

Parameter	Value
per_device_train_batch_size	32
per_device_eval_batch_size	32
evaluation_strategy	"steps"
eval_steps	5.000
logging_steps	5.000
gradient_accumulation_steps	8
max_steps	125.000
weight_decay	0,1
warmup_steps	1.000
lr_scheduler_type	"cosine"
learning_rate	0,00176
save_steps	True

Table 36: Training arguments used to pretrain HAGALBERT from scratch. These parameters correspond to ALBERT's training parameters,

After training, we push the trained model to a HuggingFace repository, in order to be able to load and finetune the model later.

6.4.3 Experiment 2: Pretraining on top of existing models

As explained in section 4.9 of our systematic literature review, we pretrain on top of Dutch BERT models RobBERT, MedRoBERTa.nl and belabBERT. This means we use the models' pretrained weights as a starting point to fit our data on top of. We again use HuggingFace to load tokenizers and model weights. As these models are all RoBERTa-like models we can use HuggingFace's *RobertaTokenizer* class in combination with the unique identifier of the respective model in order to load the tokenizer and model object. We then transform our sliding window input to correspond with the models' maximum token size and tokenize it using the respective tokenizer.

For our pretraining phase, in order to allow for a fair comparison, we set the same training arguments as for HAGALBERT. Again, after training we save the models to HuggingFace repositories. In our case these repositories are named "RobBERT-HAGA", "MedRoBERTA.nl-HAGA" and "belabBERT-HAGA" and we will refer to the models by these names further in this thesis.

6.4.4 Experiment 3: Translating Dutch input data for English clinical domain BERT

Within our systematic literature review we decided to translate the input data to English and then use these translated texts to pretrain on top of English medical BERT models. In this thesis, we first conducted research towards the available translation methods in literature. Here, we identified several industry applications, such as Google Translate, Yandex and DeepL. All of these solutions were however deemed to be too pricey to fit within the constraints of this thesis. As the bulk of our training sliding window sentences totalled to over 2.3 million characters, using an industry solution would cost thousands of euros. We therefore shifted our focus towards free-to-use neural models that can be initialized the same way as BERT models via HuggingFace.

We ultimately decided to use *opus-mt-nl-en*, which is a neural translation model. OPUS-MT models are machine translation models created by the University of Helsinki (Tiedemann & Thottingal, 2020). The models are trained on freely available parallel corpora collected in the large bitext repository OPUS. The general model architecture is based on a standard transformer setup, containing an encoder and decoder with 6 self-attentive layers in both. The encoder and decoder network both contain 8 attention heads in each layer. For preprocessing, the Dutch-to-English model used a combination of normalization with SentencePiece. SentencePiece is an unsupervised text tokenizer which can be regarded as the open source version of the WordPiece tokenizer, which was used to tokenize BERT inputs.

The opus-mt-nl-en model was tested on the "Tatoeba.nl.en" dataset, on which it achieved a chrF-score of 0.749. This chrF-score is a metric which scores the output of a translation model on the basis of its character n-grams overlapping with the ground truth. It incorporates the character n-gram precision and recall arithmetically averaged over all n-grams. We did not find any other models that were evaluated on this dataset however, so there exists a possibility that this score is not representative of the quality of the model. We leave comparing multiple Dutch-to-English translation models to future research, as this does not fit within our time constraints.

Using the opus-mt-nl-en model, we firstly attempted to translate all of our sliding window texts. For this we used a similar approach to pretraining HAGALBERT, that being using the Shark OOD interface in order to create a Python script that loads the translation model from HuggingFace and uses the model to translate the grouped sentences. This Python script is then invoked via a SLURM job, which is scheduled to run on Shark. Within the Python script, translating the texts is done by creating a HuggingFace pipeline. These pipelines make it easier to infer HuggingFace models on a GPU. Before the initialization of this pipeline, we alter the sliding window sentences to adher to the maximum input size of opus-mt-nl-en of 512 tokens, similarly to how the input needs to be truncated to 512 tokens per text for pretraining BERT-like models.

Contrarily to this process however, we translate entire sliding window groups of three sentences, even when the total length of one sequence exceeds 512 tokens. We do this by separating the sequence in chunks with a maximum length of 512 tokens. In order to preserve context, we only split the sequence on space-characters between words, meaning every chunk starts and ends with a full word. We chose to translate the entire text instead of just the first 512 characters as this allows us to obtain more training data. After translating each chunk we save it to an output file, keeping track for each chunk which text it belongs to so that we can recreate the input later.

While running the Python script in a SLURM job we found out that the runtime of translating all the sentence sequences would take a total of between 200-300 hours, even when running completely on the Shark GPU partition. This is likely the case because of the speed of inferring the model, meaning the transformer architecture of opus-mt-nl-en model did not allow for faster calculations. It could also be the case that we did not find an optimal setup for translating the texts and that using another setup would yield a (much) faster total time.

As running a translation job for over 200 hours in total did not fit within our time constraints nor within SHARK's own constraints we decided instead to only translate our hand-labelled input rather than the full bulk of sentence sequences. This reduced the total translation runtime from 200 hours to 3 hours. It does however mean we cannot pretrain on top of ClinicalBERT and BioBERT, as we only have 4.700 translated texts. This means we instead opt for finetuning these English BERT models on the translated manually annotated data. Although we hypothesize that this could potentially lead to a lower performance, we still believe it is interesting to see what the performance of these models is when they are finetuned on text that has been translated from Dutch to English. An overview of our finetuning processes can be found in the next section. We leave translating the entire bulk of input text to future research.

6.4.5 Finetuning

Corresponding to the outcomes of our systematic literature review, we finetune all of our BERT-like models on our hand-labelled dataset. Our dataset is split into a training and test set, which is uniform among all models. We use the training set to finetune our BERT models on. We finetune our models by firstly loading in the models we pretrained from HuggingFace. The English ClinicalBERT and BioBERT are also loaded in from HuggingFace, albeit not from one of our repositories as we use their standard versions. We finetune every model separately for the smoking, alcohol and drugs labels. Instead of the sliding window sentences that we loaded in to pretrain the models we load the hand-labeled texts into a HuggingFace dataset. We then tokenize every text using the respective model's tokenizer, which is again also loaded from HuggingFace. Finetuning is then done by training the models on the tokenized texts and the corresponding labels.

During finetuning, model accuracy is recorded during each epoch. We conducted several prior tests in order to determine an optimal total amount of iterations for finetuning the model. Within these tests, we found that most models stopped improving around 5 iterations. Keeping this into account and applying an error safe we decided to finetune model for 10 iterations. After 10 iterations, we save the finetuned model to a separate HuggingFace repository. Within this thesis, we do not apply other hyperparameter tuning, as this does not fit within our time constraints. We rather apply the same training argument parameters to each finetuning task. These are the standard parameters that are assigned when an instance of the class *TrainingArguments* is created, and can be found in the HuggingFace documentation [2]. We believe performing hyperparameter tuning would be more suitable to find the highest possible performance per model within the context of our experimental setup, but found that it did not fit within our time constraints. We therefore suggest it for future research.

6.5 Evaluation

In this section, we explain our approach of evaluating our models and visualizing their results. We firstly elaborate on our approach of splitting our input dataset into a training and test set. We then go over the metrics we use to evaluate performance. Lastly, corresponding to our systematic literature review, we give an overview of the methods we use to verify the outcomes and provide more insights into the workings of our models. This section corresponds to the end of the Modeling phase and the start of the Evaluation phase of CRISP-DM.

[2] https://huggingface.co/docs/transformers/v4.30.0/en/main_classes/trainer

6.5.1 Evaluation Metrics

We split our dataset containing 4.700 hand-labeled samples into a training and test set with the ratio 80/20, meaning 80% of the full set ends up in the training set and the remaining 20% in the test set. The split is done at random, but with a set random state, meaning the training and test sets are the same for the evaluation of each individual model. After training every model, apart from our string matching model, on the training set we record the Macro F1-score, which serves as the main metric for comparing performance among models, like is the case in our exploratory data analysis.

We record Macro F1-score for each model on each lifestyle (smoking, alcohol and drugs) for each class (current user, non-user, previous user and no information given). We choose Macro F1-score as it gives equal weight to performance on every class, regardless of how many entries of the class exist in the database. This adheres to the requirements set by HagaZiekenhuis and us at the start of the thesis. The equation for Macro F1-score can be found in section 2.2. Next to Macro F1-score we provide the individual precisions and recalls for each model on each class for each lifestyle as we are interested in showing and explaining why particular models perform better than other models on the basis of these metrics. Table 37 shows the amount of entries for each class in the training and test set.

Smoking				
Set	**Non-users**	**Current users**	**No information given**	**Previous users**
Training	356	141	2.532	731
Test	84	38	611	208
Alcohol				
Training	597	474	2.673	16
Test	147	142	648	3
Drugs				
Training	998	50	2.702	10
Test	274	10	655	1

Table 37: Class distributions among the training and test sets for the smoking, alcohol use and drugs use lifestyle tasks.

6.5.2 Results Visualization

Next to Macro F1-score and per-class precisions and recalls we apply visualization methods on the output of our models in order to explain differences between them. As explained in our systematic literature review, we use LIME to show the most important features per class. In particular, in the context of this thesis we will use LIME visualizations to compare our models based on texts where one model classified the text correctly and another incorrectly. LIME can then visualize which features in particular caused each model to classify the text into a specific class and we deem it interesting to inspect these visualizations on the same text for each model to show differences between models. For our LIME visualizations, we perform 5 separate experiments where we aim to show results in a way to show which models succeed where other models fail. We furthermore show the shortcomings of the best performing model. The experiments are:

1. Feature visualization of 5 texts that were misclassified by HAGALBERT that were classified correctly by our best-performing model that was pretrained on top of existing weights.

   We include this visualization as we are interested in finding out where pretraining from scratch fails in comparison with pretraining on top of existing weights, in particular on our tasks. If finetuned HAGALBERT is shown to outperform models that were pretrained on top of existing weights we flip this experiment, meaning we visualize the features of 5 texts that were classified correctly by HAGALBERT and incorrectly by the best performing other model.

2. Feature visualization of 5 texts that were misclassified by our best-performing model.

   As we wish to recommend our best-performing model to HagaZiekenhuis, we wish to show its shortcomings such that points of improvement can be identified and that we can provide knowledge on certain situations where the output of the model cannot always be trusted to be correct.

3. Feature visualization of 5 texts that were misclassified by a model that was pretrained on top of existing weights that were classified correctly by a model that was finetuned on translated input.

   As our translation approach is a scientific novelty, we wish to show differences between models that use Dutch input and models that use input that has been translated to English. Again, if the translated models perform worse than the other models we flip the experiment.

4. Feature visualization of 5 texts misclassified by string matching and classified correctly by our best model.

   We hypothesize that deeper BERT models are able to outperform string matching on the given lifestyle classification tasks. For this reason, we wish to show where string matching fails and our best model triumphs over it.

5. Feature visualization of 5 texts misclassified by classical machine learning and classified correctly by our best model.

   We again hold the assumption that deeper models outperform classical machine learning methods and wish to show scenarios that indicate how this performance increase is achieved.

We perform these experiments for smoking, alcohol and drugs classification, given that we can show meaningful differences between the models. It could be the case that the models perform close to or exactly perfect on the classification of a certain lifestyle, rendering these experiments on that regard meaningless. Next to LIME, we use the t-SNE dimensionality reduction method in order to visualize embeddings per class and show how well each BERT-like model is able to distinguish between the different classes.

In this visualization, we colour every embedding obtained from the text based on its assigned

label. Then, under the assumption that the closer the embeddings of the same class are to each other and the further away they are to embeddings of other classes in the visualization, the better the model is in distinguishing between classes. Using these visualization methods, we provide insights in the performance of the models, which aspects they perform well on and on which aspects there can be improved upon.

7 Results

In this section, we provide the results of the experiments that were explained in the previous section. We provide performance scores on the lifestyle classification tasks, as well as visualizations that show differences between models.

7.1 Lifestyle Classification

In this section, we show the performances of each model on every lifestyle. We show precision, recall and F1-score for every class. Within the performance tables, we denote the amount of test set entries that the model was evaluated on per class in parentheses, for example "Non-user (84)" means that there were 84 entries of the non-user class in the test set for that specific lifestyle.

7.1.1 String matching

Smoking

Non-user (84)			
Model	**Precision**	**Recall**	**F1-score**
CTCue	0.91	0.75	**0.82**
Current user (38)			
Heath	0.30	0.74	**0.43**
CTCue	0.89	0.45	**0.60**
No information given (611)			
CTCue	0.93	1.00	**0.96**
Previous user (208)			
Heath	*0.96*	*0.77*	***0.86***
CTCue	0.99	0.96	**0.98**

Table 38: Results of string matching approaches on the test set of the smoking task. Our CTCue queries outperform Heath's method on every comparable class.

The scores in table 38 clearly show our CTCue string matching queries outperforming Heath's queries on the smoking task. We adjusted our test set to Heath's method, removing entries that were not classified and grouping non-users and previous users together and yet the CTCue string matching query still outperformed the other method on both overlapping classes. For the "No information given" and "Previous user" classes the CTCue method the string matching query approaches a flawless classification on the test set. This does not completely discard Heath's method of course, as their method was constructed to fit within the context of their thesis and our method was created with the limits and constraints of our thesis in mind, with both methods being tested on a task crafted for this thesis.

Another interesting phenomenon is the relatively large difference in performance of both string matching approaches on the current user class compared to the other classes. This seems to correspond with the edge case study we conducted in section 5.4.7, as there were large differences between the accuracy of the labels for the current smoker class compared to the non-smoker class here as well.

Alcohol

	Non-user (147)		
Model	**Precision**	**Recall**	**F1-score**
CTCue	0.97	0.98	**0.98**
	Current user (142)		
CTCue	1.00	0.96	**0.98**
	No information given (648)		
CTCue	0.99	1.00	**0.99**

Table 39: Results of string matching approaches on the test set of the alcohol task. The model performs almost flawlessly.

In table 39, it appears our queries used to distinguish alcohol use perform very close to 100% on our test set. This could indicate that the alcohol status of patients in our test set is either easy to grasp using string matching query and/or the queries we used are particularly well suited to them. Again, these results seem to correlate slightly to the results of our edge case study, as the accuracy of the string matching query on the edge cases was significantly higher than for smoking.

Drugs

	Non-user (274)		
Model	**Precision**	**Recall**	**F1-score**
CTCue	0.99	0.99	**0.99**
	Current user (10)		
CTCue	1.00	0.60	**0.75**
	No information given (655)		
CTCue	0.99	1.00	**0.99**

Table 40: Results of string matching approaches on the test set of the drugs task. The model again performs almost flawlessly, apart from on the current user class.

For the drugs task in table 40, a similar phenomenon occurs as for the alcohol task, this being that using our string matching query almost flawless classification can be achieved. Only for the current users, of which 60% were predicted correctly, the queries do not perform as well. This indicates that there is still significant performance to be gained upon our string matching queries in terms of current drugs users. As there are a relatively low amount of current users it could be the case that adjusting the string matching queries to perform better should be a relatively straightforward task. However, the lower performance on this class could indicate more variety in "real-world" examples, which would mean the queries need to be adjusted significantly or that deeper models need to be applied to this task.

Overall, these string matching scores indicate that on the smoking task there is still a lot of performance to be gained, as the F1-scores on the non-user and current user classes are far from perfect. Contrarily, the scores on the alcohol and on almost all of the classes of the drugs

tasks indicate that these tasks could be potentially solved using string matching alone. Of course, every text classification task is solvable using string matching applied in a contextually extensive manner, but this is often not achievable due to the query becoming relatively huge and the inclusion of conditions and subconditions, which complicate the query hugely.

These scores however show that on our test set our relatively computationally and structurally straightforward queries could achieve near-perfect performance, indicating the tasks-at-hand are straightforward enough such that deeper models are unnecessary. Deeper models require significantly more computing power and management and when string matching performs comparably or even better than the latter should be preferred. Of course it could be the case that these scores are largely the product of the composition of our test set, which we touch upon in our discussion section in section 8.

7.1.2 Classical Machine Learning Methods

Smoking

Non-user (84)			
Model	**Precision**	**Recall**	**F1-score**
SGD	0.69	0.98	**0.81**
Current user (38)			
SGD	0.67	0.68	**0.68**
No information given (611)			
SGD	0.98	0.94	**0.96**
Previous user (208)			
SGD	0.99	0.93	**0.96**

Table 41: Results of our Stochastic Gradient Descent approach on the test set of the smoking task.

Our Stochastic Gradient Descent model for smoking performs comparably to slightly better compared to our CTCue string matching model, as can be ascertained from table 41. It could be the case that the amount of cases in the training and test set causes a break-even point of the performance scores of classical machine learning and string matching and that more training data would propel machine learning above string matching.

Alcohol

Non- user (147)			
Model	**Precision**	**Recall**	**F1-score**
SGD	0.85	0.98	**0.91**
Current user (142)			
SGD	0.97	0.94	**0.96**
No information given (648)			
SGD	1.00	0.98	**0.99**
Previous user (3)			
SGD	0.00	0.00	**0.00**

Table 42: Results of our Stochastic Gradient Descent approach on the test set of the alcohol task.

In table 42, most likely due to the low amount of previous users in the training and test sets, the model is unable to classify previous users. The rest of the classes are similar to string matching in terms of performance.

Drugs

Non- user (274)			
Model	**Precision**	**Recall**	**F1-score**
SGD	0.94	0.99	**0.96**
Current user (10)			
SGD	1.00	0.30	**0.46**
No information given (655)			
SGD	0.99	0.98	**0.99**
Previous user (1)			
SGD	0.00	0.00	**0.00**

Table 43: Results of our Stochastic Gradient Descent approach on the test set of the drugs task.

The performance on current drugs users in table 43 is also lower than string matching most likely due to the low amount of training samples. The model only assigned the current user label to current users, but most of the current users were misclassified. The model is again completely unable to identify the previous user present in the test set.

Summarizing these results, it appears that our classical machine learning method performs overall worse than our CTCue string matching approach. It achieves comparable scores for classes with sufficient training cases and performs abysmally on classes that are occurring infrequently.

7.1.3 BERT Pretraining

Smoking

Non-user (84)			
Model	**Precision**	**Recall**	**F1-score**
HAGALBERT	0.41	0.83	**0.55**
RobBERT-HAGA	0.81	0.98	**0.89**
belabBERT-HAGA	0.67	0.05	**0.09**
MedRoBERTa.nl-HAGA	0.82	0.96	**0.88**
Current user (38)			
HAGALBERT	0.55	0.29	**0.38**
RobBERT-HAGA	0.61	0.82	**0.70**
belabBERT-HAGA	0.00	0.00	**0.00**
MedRoBERTa.nl-HAGA	0.94	0.76	**0.84**
No information given (611)			
HAGALBERT	0.97	0.99	**0.98**
RobBERT-HAGA	1.00	0.99	**0.99**
belabBERT-HAGA	0.84	0.98	**0.90**
MedRoBERTa.nl-HAGA	1.00	0.98	**0.99**
Previous user (208)			
HAGALBERT	0.97	0.60	**0.74**
RobBERT-HAGA	0.99	0.85	**0.91**
belabBERT-HAGA	0.91	0.94	**0.92**
MedRoBERTa.nl-HAGA	0.99	1.00	**0.99**

Table 44: Results of our BERT-like approaches on the test set of the smoking task.

From table 44, we can ascertain that on this task our pretraining from scratch approach performs significantly worse than our models that were pretrained on top of existing weights. This is the case for the RobBERT and MedRoBERTa.nl models. The MedRoBERTa.nl model furthermore outperforms RobBERT, possibly because of the fact that MedRoBERTa.nl was trained on medical text data rather than the general training approach of RobBERT. These two models also outperform both string matching and stochastic gradient descent on this task in particular, showing the potential for deeper models and their usefulness in lifestyle classification. Only on the previous user class does RobBERT-HAGA yield inferior performance than string matching, but MedRoBERTa.nl-HAGA does improve upon string matching on every class. Interestingly, HAGALBERT and belabBERT-HAGA perform significantly worse than RobBERT-HAGA and MedRoBERTa.nl-HAGA on every class. Even though belabBERT was trained on a set with comparable size to RobBERT's training set, it performs remarkably worse than it and when compared to string matching and stochastic gradient descent. This could indicate that something about belabBERT's model setup might harm its usefulness in this task in particular.

The same can be said about HAGALBERT, even though it was trained from scratch entirely on our full bulk of texts it gets outperformed by string matching and stochastic gradient descent on every class apart from "no information given". This might indicate problems with HAGALBERT's model setup in relation with our experimental setup. Furthermore, it could be the case that the comparatively small amount of 148.000 input texts is not enough to outperform string matching and stochastic gradient descent on this task in particular. Following this logic, it seems to make sense that MedRoBERTa.nl-HAGA and RobBERT-HAGA do

outperform these more shallow methods as they are pretrained on much more data.

Alcohol

Non-user (147)			
Model	**Precision**	**Recall**	**F1-score**
HAGALBERT	0.55	0.84	**0.67**
RobBERT-HAGA	0.90	0.92	**0.91**
belabBERT-HAGA	0.74	0.78	**0.76**
MedRoBERTa.nl-HAGA	0.87	0.99	**0.93**
Current user (142)			
HAGALBERT	0.92	0.34	**0.49**
RobBERT-HAGA	0.91	0.93	**0.92**
belabBERT-HAGA	0.82	0.82	**0.82**
MedRoBERTa.nl-HAGA	0.98	0.92	**0.95**
No information given (648)			
HAGALBERT	0.97	0.99	**0.98**
RobBERT-HAGA	1.00	0.99	**0.99**
belabBERT-HAGA	0.99	0.98	**0.98**
MedRoBERTa.nl-HAGA	0.99	0.97	**0.98**
Previous user (3)			
HAGALBERT	0.00	0.00	**0.00**
RobBERT-HAGA	0.00	0.00	**0.00**
belabBERT-HAGA	0.00	0.00	**0.00**
MedRoBERTa.nl-HAGA	0.25	0.33	**0.29**

Table 45: Results of our BERT-like approaches on the test set of the alcohol task.

The scores in table 45 are similar to the smoking task, with RobBERT and MedRoBERTa.nl significantly outperforming the other BERT-like models. Interestingly, MedRoBERTa.nl is the only model that is somewhat able to classify previous users, even though there are so few training samples of this category. This could indicate that MedRoBERTa.nl is more robust to smaller sample sizes. Compared to string matching, from the three classes that are present in both experiments, every model performs worse, apart from RobBERT-HAGA on the "no information given" class. Compared with stochastic gradient descent the scores are much closer, with MedRoBERTa.nl-HAGA slightly outperforming stochastic gradient descent on the non-user class by obtaining a higher precision, and slightly getting outperformed by it on the current user class by having a lower recall. However, contrary to stochastic gradient descent MedRoBERTa.nl-HAGA is marginally able to classify previous users.

Drugs

Non-user (274)			
Model	**Precision**	**Recall**	**F1-score**
HAGALBERT	0.94	0.68	**0.79**
RobBERT-HAGA	0.97	0.99	**0.98**
belabBERT-HAGA	0.95	0.96	**0.95**
MedRoBERTa.nl-HAGA	0.96	0.99	**0.97**
Current user (10)			
HAGALBERT	0.00	0.00	**0.00**
RobBERT-HAGA	0.80	0.40	**0.53**
belabBERT-HAGA	1.00	0.20	**0.33**
MedRoBERTa.nl-HAGA	1.00	0.30	**0.46**
No information given (655)			
HAGALBERT	0.88	1.00	**0.94**
RobBERT-HAGA	0.99	1.00	**0.99**
belabBERT-HAGA	0.98	0.99	**0.98**
MedRoBERTa.nl-HAGA	0.99	0.99	**0.99**
Previous user (1)			
HAGALBERT	0.00	0.00	**0.00**
RobBERT-HAGA	0.00	0.00	**0.00**
belabBERT-HAGA	0.00	0.00	**0.00**
MedRoBERTa.nl-HAGA	0.50	1.00	**0.67**

Table 46: Results of our BERT-like approaches on the test set of the drugs task.

Again, it can be ascertained from table 46 that the scores for each model mostly follow the same trends, with MedRoBERTA.nl the only model that can classify previous users. An interesting phenomenon is belabBERT-HAGA achieving much closer performance to the best models than for smoking and alcohol, albeit still worse. Next to the previous user, which is not found by any other model than MedRoBERTa.nl-HAGA, HAGALBERT is also unable to classify current users. Compared to string matching, our BERT-like models are slightly outperformed on the non-user class and significantly outperformed on the current user class. The latter of these phenomena is particularly noteworthy, as none of the models so far have been able to obtain an F1-score of over 0.74 on the current drugs user class. For this class, only RobBERT-HAGA is able to outperform stochastic gradient descent, which performs comparably on the non-user and "no information given" classes.

Ultimately for the BERT-like models, our models that were pretrained on top of existing weights, in particular RobBERT and MedRoBERTa.nl, outperformed our model that was pretrained from scratch on all three lifestyles. For the smoking task, these deeper models outperform both string matching and stochastic gradient descent. For the other tasks string matching is able to grasp the classes with fewer samples better than these models, show-

ing that for these tasks applying deeper models might not be necessary in order to achieve acceptable performance.

7.1.4 Finetuning on Translated Texts

Smoking

Non-user (84)			
Model	**Precision**	**Recall**	**F1-score**
BioBERT	0.92	0.95	**0.93**
ClinicalBERT	0.91	0.96	**0.94**
Current user (38)			
BioBERT	0.71	0.76	**0.73**
ClinicalBERT	0.73	0.87	**0.80**
No information given (611)			
BioBERT	0.99	0.99	**0.99**
ClinicalBERT	1.00	0.99	**0.99**
Previous user (208)			
BioBERT	0.98	0.97	**0.97**
ClinicalBERT	0.99	0.96	**0.97**

Table 47: Results of our translation approaches on the test set of the smoking task.

In table 47, the models finetuned on translated texts surprisingly perform comparably to RobBERT and MedRoBERTa.nl. This is especially noteworthy as the act of translating input text to English to use larger English BERT models can be seen as a scientific novelty. We show here that translation might be a viable option for lifestyle classification tasks. It might even be worthwhile to apply translation to more general medical text classification tasks, solely based on these results, as only the 4.700 texts in the hand-labeled set were translated rather than the entire bulk of 148.000 texts.

Alcohol

Non-user (147)			
Model	**Precision**	**Recall**	**F1-score**
BioBERT	0.91	0.99	**0.95**
ClinicalBERT	0.92	0.99	**0.96**
Current user (142)			
BioBERT	0.98	0.92	**0.95**
ClinicalBERT	0.99	0.93	**0.96**
No information given (648)			
BioBERT	1.00	1.00	**1.00**
ClinicalBERT	1.00	0.99	**0.99**
Previous user (3)			
BioBERT	0.00	0.00	**0.00**
ClinicalBERT	0.25	0.33	**0.29**

Table 48: Results of our translation approaches on the test set of the alcohol task.

As can be ascertained from table 48, the performance of these models on the alcohol task is very comparable to the Dutch BERT models, with the models not being able to classify previous users likely due to the lack of training samples.

Drugs

Non-user (274)			
Model	**Precision**	**Recall**	**F1-score**
BioBERT	0.95	0.99	**0.97**
ClinicalBERT	0.96	0.99	**0.98**
Current user (10)			
BioBERT	0.17	0.10	**0.12**
ClinicalBERT	1.00	0.30	**0.46**
No information given (655)			
BioBERT	0.99	0.99	**0.99**
ClinicalBERT	0.99	0.99	**0.99**
Previous user (1)			
BioBERT	0.00	0.00	**0.00**
ClinicalBERT	0.00	0.00	**0.00**

Table 49: Results of our translation approaches on the test set of the drugs task.

Again, from table 49, translating the texts to English and finetuning the models on those texts achieves similar performance to pretraining on top of Dutch BERT models.

As stated before, it is very interesting how merely translating the hand-labeled texts to English and using this input to finetune English medical BERT models results in comparable performance to using the entire bulk of texts and pretraining existing Dutch BERT models on that set and then finetuning the models on the hand-labeled set. This shows the potential for translated inputs within the BERT domain, which we will go more in depth about in the discussion section in section 8.

7.1.5 Overall Macro F1-Scores

Here, the model that achieved the highest score on the metric specified in the column name has that score stated in bold typeface. As specified earlier, we regard the Macro F1-score as the most important metric, hence we colour it differently in the tables. For models where classes are missing we put the scores in italic typeface, as these models cannot be fully compared to models with no missing classes. The best scores per class are highlighted in green, with the best Macro F1-score being highlighted in darker green.

Smoking

Model	F1-score Non-user (84)	F1-score Current user (38)	F1-score No information given (611)	F1-score Previous user (208)	F1-score Weighted Avg	F1-score Macro Avg
String Matching Heath	-	*0.43*	-	*0.82*	*0.81*	*0.64*
String Matching CTCue	0.82	0.60	0.96	0.98	0.94	0.84
Standard Machine Learning (SGD)	0.81	0.68	0.96	0.96	0.94	0.85
HAGALBERT	0.55	0.38	0.98	0.74	0.87	0.66
RobBERT-HAGA	0.89	0.70	**0.99**	0.91	0.96	0.87
belabBERT-HAGA	0.09	0.00	0.90	0.92	0.80	0.48
MedRoberta.nl -HAGA	0.88	**0.84**	**0.99**	**0.99**	**0.97**	**0.93**
BioBERT (translated)	0.93	0.73	**0.99**	0.97	**0.97**	0.91
ClinicalBERT (translated)	**0.94**	0.80	**0.99**	0.97	**0.97**	0.92

Table 50: Full results of our models on the test set of the smoking task.

For the smoking task, table 50 shows that the MedRoBERTa.nl model that was further pre-trained on our entire bulk of texts and finetuned on hand-labeled texts adhered the best to our requirements, by achieving the highest Macro F1-score. The models that were finetuned on translated input perform almost as good as MedRoBERTa.nl.

Alcohol

Model	F1-score Non-user (147)	F1-score Current user (142)	F1-score No information given (648)	F1-score Previous user (3)	F1-score Weighted Avg	F1-score Macro Avg
String Matching CTCue	***0.98***	***0.98***	*0.99*	-	***0.99***	***0.98***
Standard Machine Learning (SGD)	0.94	0.93	0.99	0.00	0.97	0.71
HAGALBERT	0.67	0.49	0.98	0.00	0.85	0.54
RobBERT-HAGA	0.91	0.92	0.99	0.00	0.97	0.71
belabBERT-HAGA	0.76	0.92	0.98	0.00	0.92	0.64
MedRoBERTa.nl-HAGA	0.93	0.95	0.98	**0.29**	0.97	0.79
BioBERT (translated)	0.95	0.95	**1.00**	0.00	0.98	0.72
ClinicalBERT (translated)	0.96	0.96	0.99	**0.29**	0.98	0.80

Table 51: Full results of our models on the test set of the alcohol task.

For alcohol status, in table 51, even though we do not have a query for previous users, it is clear that our string matching approach outperforms more sophisticated, deeper approaches. It should be noted that the reason for the Macro F1-score being significantly higher for string matching is that there are no previous user evaluations. If we disregard the previous user class and evaluate every model on the other classes the BioBERT and ClinicalBERT models come close in terms of F1-scores, yet these models are significantly more resourceful and harder to train and maintain, with string matching already performing close to perfectly. It can therefore be stated that with these training and test sets, string matching is able to achieve near perfection whilst also requiring the least resources to implement, meaning there is likely no need for sophisticated models in this regard.

Drugs

Model	F1-score Non-user (274)	F1-score Current user (10)	F1-score No information given (655)	F1-score Previous user (1)	F1-score Weighted Avg	F1-score Macro Avg
String Matching CTCue	**0.99**	**0.75**	**0.99**	-	***0.99***	***0.91***
Classical Machine Learning (SGD)	0.96	0.46	0.99	0.00	0.97	0.60
HAGALBERT	0.79	0.00	0.94	0.00	0.88	0.43
RobBERT-HAGA	0.98	0.53	0.99	0.00	0.98	0.63
belabBERT-HAGA	0.95	0.33	0.98	0.00	0.97	0.57
MedRoBERTa.nl-HAGA	0.97	0.46	0.99	**0.67**	0.98	0.77
BioBERT (translated)	0.97	0.12	0.99	0.00	0.97	0.52
ClinicalBERT (translated)	0.98	0.46	0.99	0.00	0.98	0.61

Table 52: Full results of our models on the test set of the drugs task.

As shown in table 52, just like on the alcohol task, string matching is able to achieve near perfection, indicating deeper models are unnecessary in this context.

7.2 LIME Experiments

In this section we show our LIME visualizations for each experiment that was laid out in section 6.5.2. As we showed in the previous section that within the context of this thesis a lot of the models are close to solving the alcohol and drugs tasks, we visualize the smoking task in this section, as we believe that the F1-scores on this task among the models are different enough such that interesting differences between models can be shown. As explained previously, we conduct these experiments as we are interested in showing the differences in how different models classify the same text and how the differences in performance can be explained. As the experiments are quite lengthy and contain a plethora of images, we refer to the full experiments in the appendices of this paper. In this section we provide a summary of these experiments, showing snippets which clearly show the differences between the models.

7.2.1 Experiment 1: 5 texts misclassified by HAGALBERT and classified correctly by MedRoBERTa.nl-HAGA.

Within Experiment 1, we show that MedRoBERTa.nl-HAGA is better at grasping context than our pretrained model from scratch HAGALBERT. For example, in text 1 the subtext "Roken

-" (Smoking -) is interpreted by HAGALBERT to indicate a current smoker, while this is obviously not the case. HAGALBERT also is way less sure in its prediction, assigning a probability of 44% the text belongs to the current user class and a 39% chance the text belongs to the true class, this being the non-user class.

In comparison, MedRoBERTa.nl-HAGA was able to judge the text correctly and assigns a 100% probability to the non-user class solely based on the "Roken -" part. A similar phenomenon occurs in text 2, where the double presence of the word "roken" (smoking) causes HAGALBERT to predict the non-user class. This is unwanted behaviour, as the word is used in the context "stoppen met roken" (to stop smoking), meaning the patient is still smoking and should therefore be classified as a current smoker. For text 4, HAGALBERT falsely regards a collection of unimportant words as predictors for the non-user class, leading it to classify the text as such. This can be seen in figure 20.

HAGALBERT
Prediction: Geen gebruiker (Non-user)

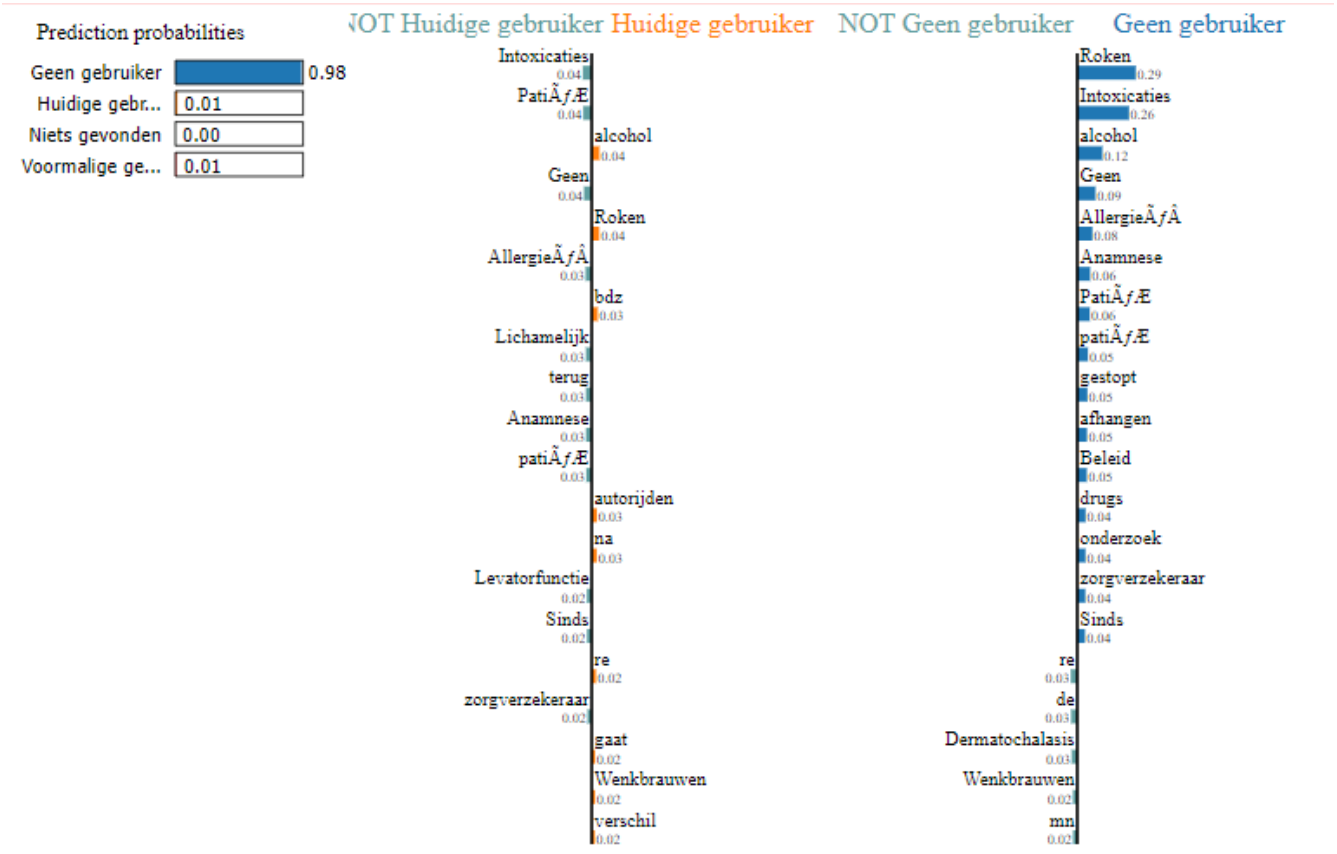
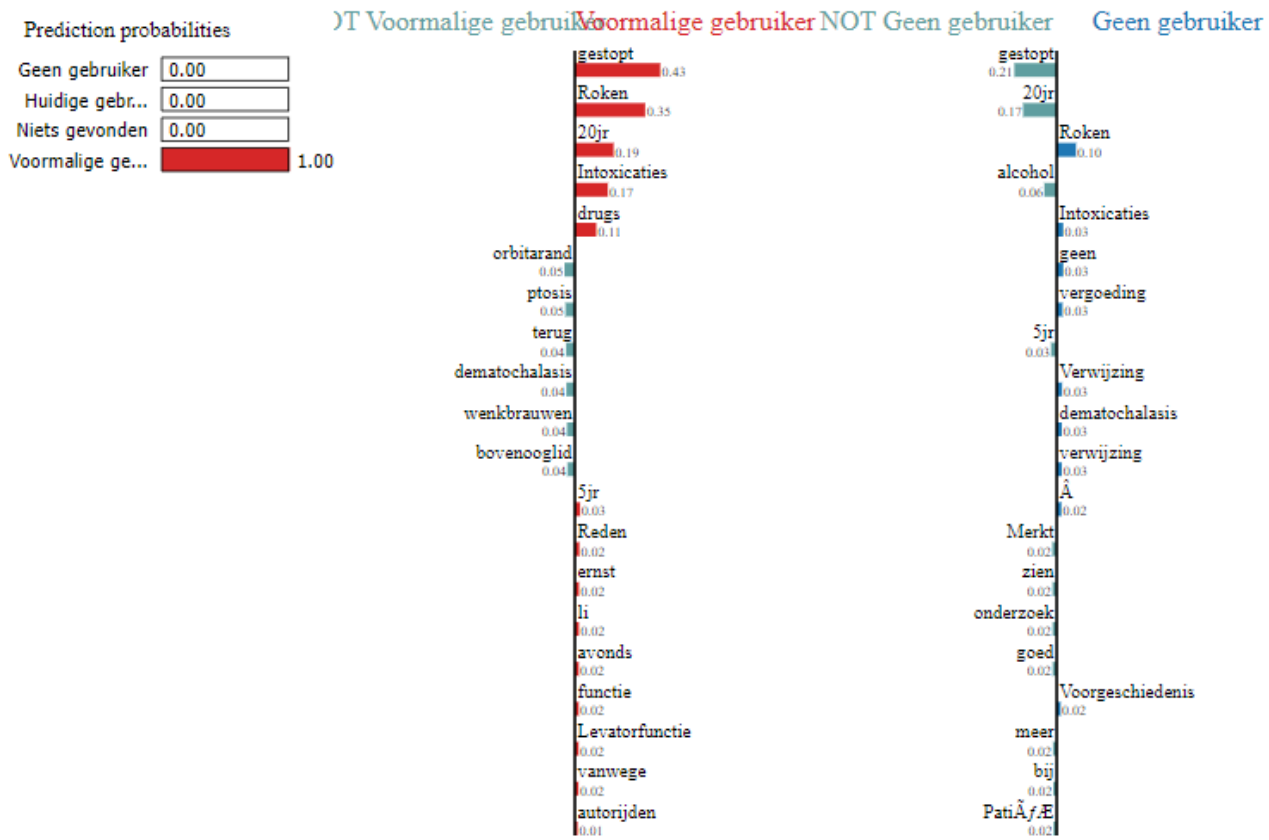


Figure 20: HAGALBERT LIME Visualization of Text 4 of Experiment 1

the presence of the Dutch word for anamnesis is wrongly regarded by HAGALBERT to be a strong indicator for the non-user class. We can ascertain from this image that the words "Intoxicaties" (intoxications), "Geen" (no/none), "Patient" and "Allergieën" (allerguies) further contribute to a non-user prediction. The reason the spelling of the two last words is different in the image is that the model parses the "ë" symbol in that particular way. It is interesting that the word "Intoxicaties" is seen as strong indicator for non-user as the word often denotes that the patient's smoking, alcohol and drug use statuses are stated next, regardless of the patient is an active user of those substances or not.

MedRoBERTa.nl-HAGA is however able to discard these terms as they bear no importance towards smoking classification, as can be seen in figure 21. It assigns substantial weights to less and seemingly better fitting predictors, leading it to correctly classify the patient as a previous user.

MedRoBERTa.nl-HAGA
Prediction: Voormalige gebruiker (Previous user)



Text with highlighted words
Reden van komst / Verwijzing: Reden verwijzing: Oogleden Anamnese: Voorgeschiedenis Blanco Medicatie Geen Anamnese Sinds 0,5jr progressieve hoofdpijn en vermoeidheid na ontstaan afhangen bovenooglid li. Kan hierbij ook iets minder zien, moet mn bij autorijden de wenkbrauwen helemaal optillen. Kan 's avonds niet meer goed lezen vanwege de vermoeidheid. Merkt zelf geen verschil tussen li en re. Intoxicaties Roken 20jr terug gestopt, alcohol-, drugs- AllergieÃƒÂ«n Geen bekend Lichamelijk onderzoek: Wenkbrauwen bdz op niveau van de orbitarand. Dermatochalasis re mild, li tot op de wimpers. Levatorfunctie bdz intact, goede functie. Geen ptosis. Conclusie: Conclusie: Milde dermatochalasis. Beleid: Beleid: Helaas voldoet de ernst van de dematochalasis niet aan de voorwaarden die de zorgverzekeraar stelt aan vergoeding voor een blepharoplastiek. PatiÃƒÆ'Ã‚Â«nt werd op de mogelijkheden gewezen van zelfbetaling. De patiÃƒÆ'Ã‚Â«nt gaat hier over nadenken.

Figure 21: MedRoBERTa.nl LIME Visualization of Text 4 of Experiment 1

The words that were deemed important by HAGALBERT are largely ignored by MedRoBERTa.nl and with good cause, as the model is able to predict the correct outcome with 100% certainty. Contrarily to HAGALBERT, this model views the word "Intoxicaties" as a strong indicator for the previous user class rather than the non-user class.

7.2.2 5 texts misclassified by MedRoBERTa.nl-HAGA

We conduct experiment 2 as we are interested in the aspects on which our best-performing model can still improve. This way, if we recommend the model to HagaZiekenhuis, we can provide clarity on which cases exist where the model's output needs to be checked by hand. Overall, the shortcomings of MedRoBERTA.nl-HAGA can be ascribed to complex sentence structures. For example, in text 1 MedRoBERTa.nl-HAGA is unable to recognise the sequence "rookt niet meer" to be about a previous user, rather it views the patient as a non-user, likely also because the word "niet" (not) is often used in the same text which might contribute negatively to the classification.

For text 2, the sequences "Rookt nog" (Still smokes) and "... als het stoppen met roken niet lukt ..." (if the patient fails to stop smoking) are not enough to convince the model to stray from its prediction of "non-user", even though these should clearly indicate the patient is a current smoker. This is likely due to the uncommon phrasings of these sequences, which we suspect do not occur often in the dataset. In text 4, this shortcoming is present in a clear way. As can be seen in figure 22, the model does not recognise the words "is niet" (has not) as negator for the sequence "gestopt with roken" (stopped smoking), hence it fails in classifying the text correctly. This flaw could potentially be resolved with more training data, specifically training data that includes several forms of negation.

MedRoBERTa.nl
Prediction: Voormalige gebruiker (Previous user)

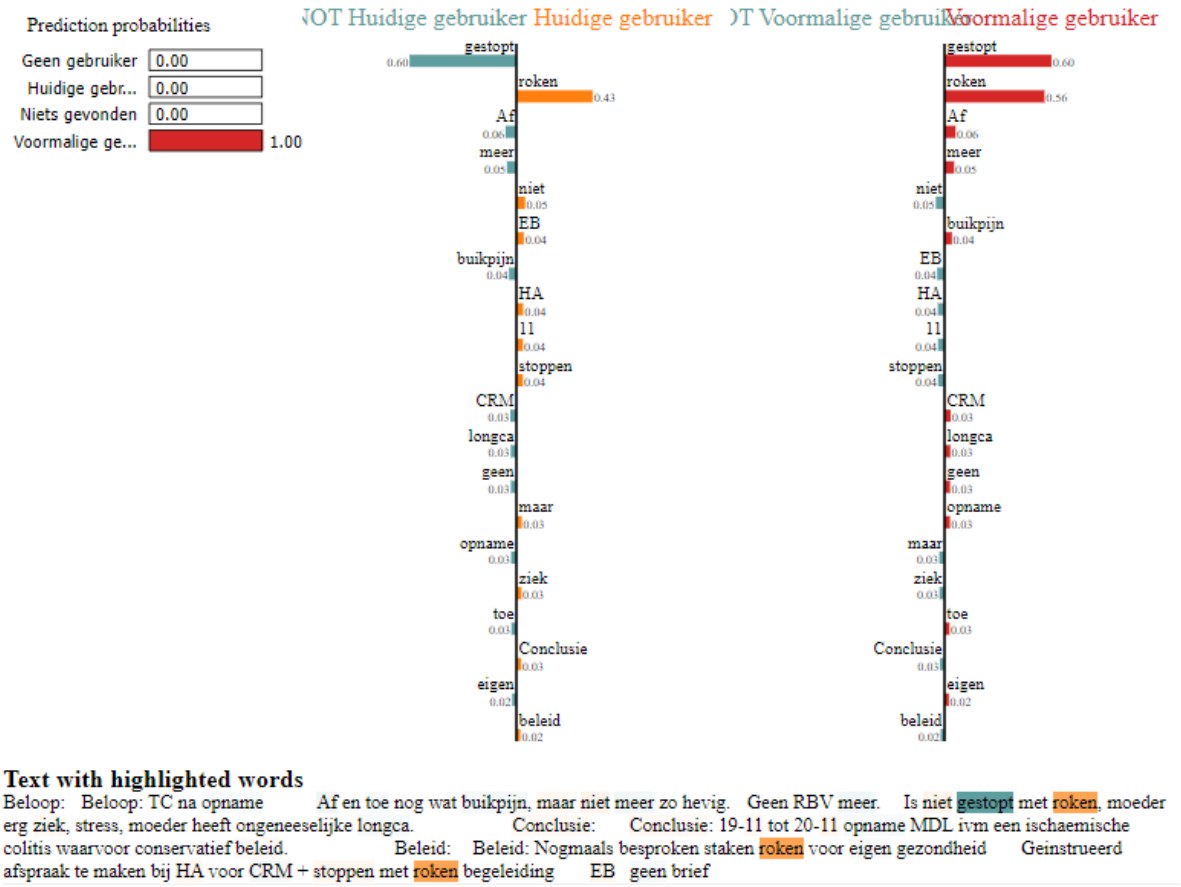


Figure 22: MedRoBERTa.nl LIME Visualization of Text 4 of Experiment 2

7.2.3 Experiment 3: 5 texts misclassified by RobBERT-HAGA and classified correctly by translated ClinicalBERT

In this experiment we compare our best-performing translated BERT model to one of the Dutch BERT models which performed worse. We can then show how translation to English and/or the usage of larger/more topical models is able to improve upon pretraining on top of existing weights of a Dutch BERT model. For text 1, RobBERT views the phrase "Stoppen met roken blijft moeilijk" (Stopping smoking remains difficult) as strong indicator for the text belonging to a current user. This is wanted behaviour, albeit that the model finds a collection of unimportant predictors for the previous user class, which outweigh the current user predictors and causes RobBERT-HAGA to misclassify this text. Text 2 shows a clearer example of how ClinicalBERT is better able to grasp context than RobBERT-HAGA. See figures 23 and 24 below.

RobBERT
Prediction: Geen gebruiker (Non-user)

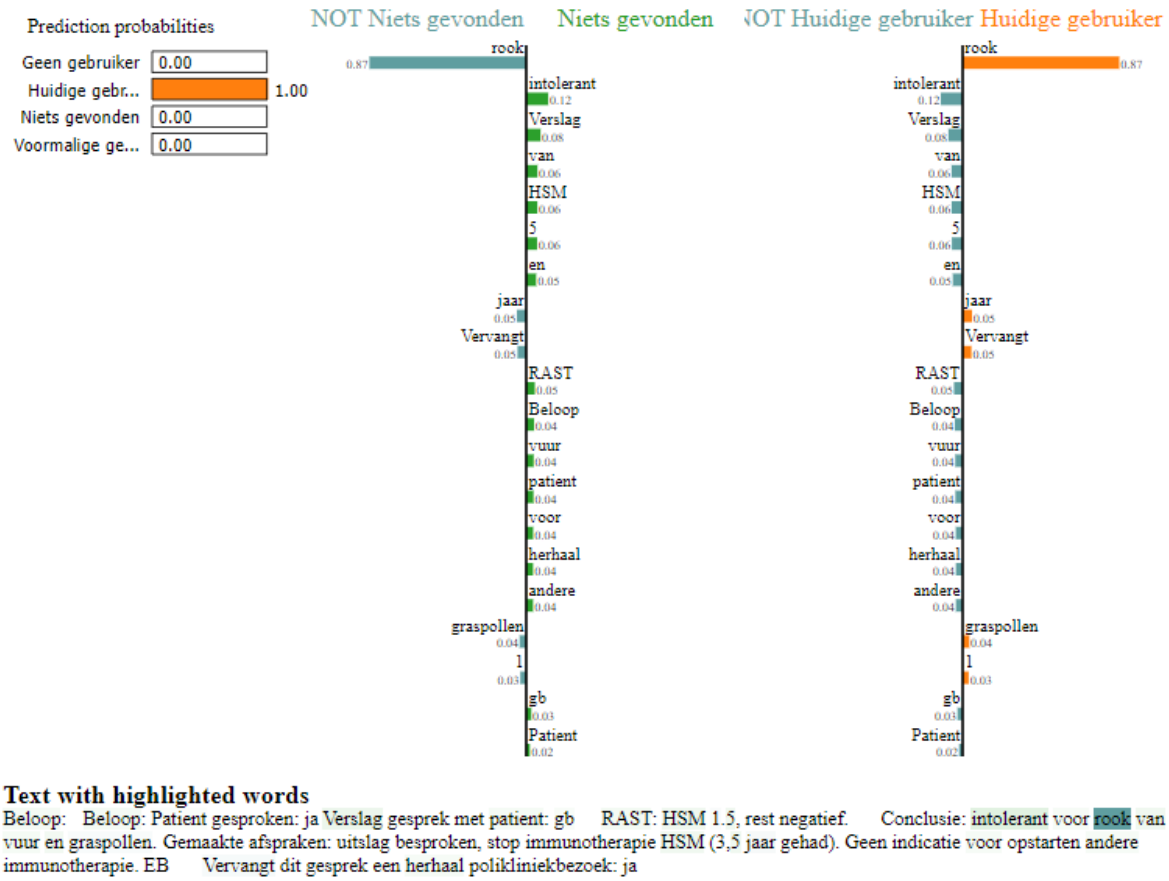


Figure 23: RobBERT LIME Visualization of Text 2 of Experiment 3

Here, RobBERT views the word "rook" (smoke) as indicator for a smoker, even though in the text the word is used to denote smoke that originates from fire, rather than anything related to the act of smoking.

ClinicalBERT

Prediction: Niets gevonden (No information given)

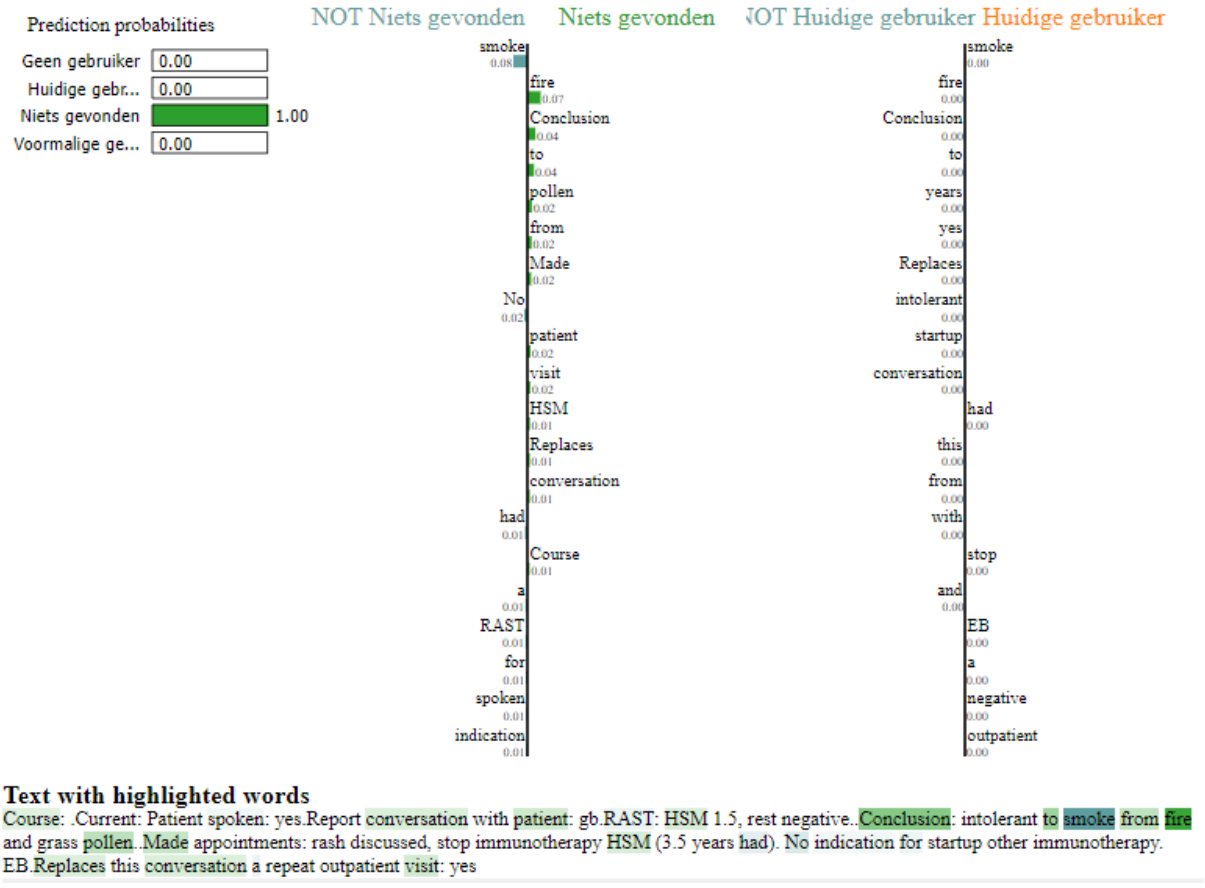


Figure 24: ClinicalBERT LIME Visualization of Text 2 of Experiment 3

ClinicalBERT is able to judge the words surrounding "smoke" correctly to not be about the patient's smoking status and therefore classifies the text correctly. For the other texts, RobBERT-HAGA seems to suffer from the same flaws as HAGALBERT, but on a smaller scale as it performed significantly better. In text 3, RobBERT-HAGA views the words "Bijzonderheden" (Oddities) and "Geen" (None/no) as indicators for a non-smoker, even though these words are not used in a smoking context and nothing is even written about smoking in the entire text. For text 4, the words "stopped smoking" are found and assigned weight towards the previous user class, but observably insignificant words persuade the model into classifying the wrong class. Overall, even though its texts have been translated to English, ClinicalBERT is better able to extract context than RobBERT-HAGA is for the original Dutch texts.

7.2.4 Experiment 4: 5 texts misclassified by CTCue string matching and classified correctly by MedRoBERTa.nl-HAGA

We conducted this experiment out of interest in viewing in which contexts MedRoBERTa.nl-HAGA was able to outperform string matching on the smoking task, as, unlike on the alcohol and drugs tasks, string matching was outmatched. Text 1 showed us that we did not have a string query for the phrase "stoppen met roken" (to stop smoking), leading to string matching assigning the "no information given" class. MedRoBERTa.nl-HAGA is able to recognise this phrasing as indicating a current user, correctly classifying the text as such. In text 2, the sequence "[roken]-roken: nooit" also did not satisfy any of our string matching queries. For text 3 a query does get satisfied, but for the wrong label. The phrasing "smoking - (stoppeed 15

years ago)" matches to the non-user class due to the presence of "smoking -", even though judging from the text the patient is clearly a previous user. Text 5 shows yet another failing of the string matching queries, not recognizing the phrase "roken: 1-2 sig/dg" (smokinng: 1/2 cigarettes a day) belonging to any class, As shown in figure 25, MedRoBERTa.nl-HAGA does find the right context here.

MedRoBERTa.nl
Prediction: Huidige gebruiker (Current user)

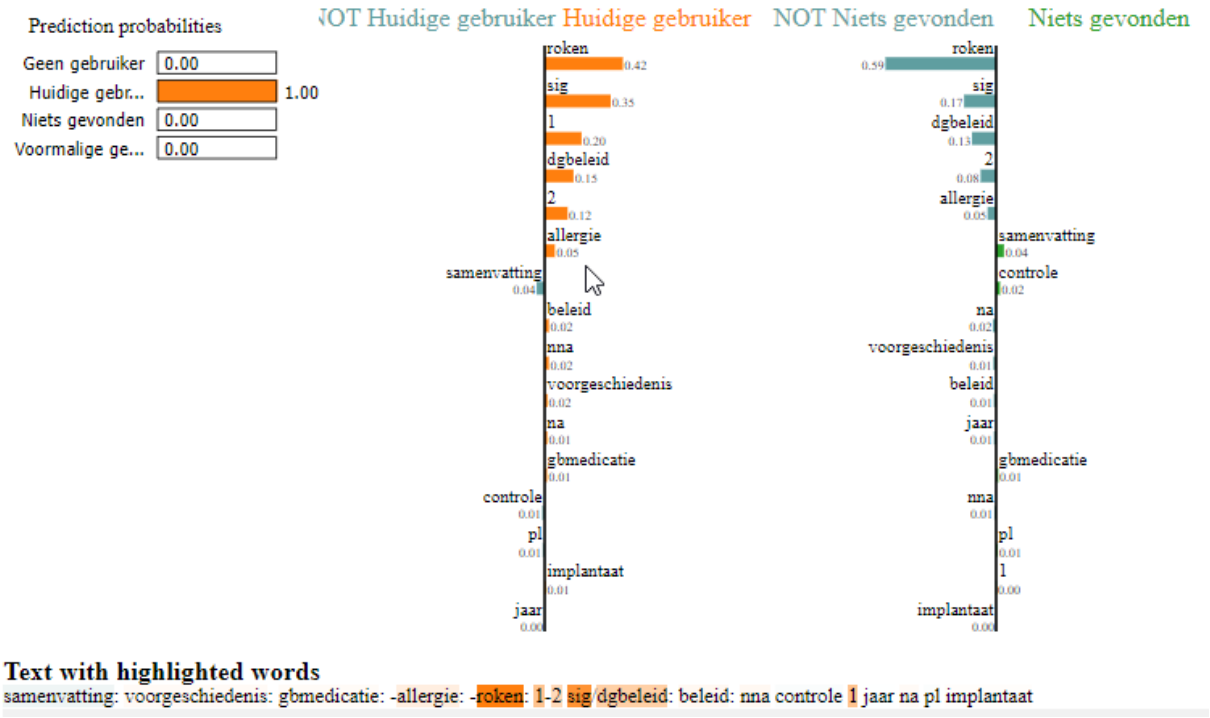


Figure 25: MedRoBERTa.nl LIME Visualization of Text 5 of Experiment 4

7.2.5 Experiment 5: 5 texts misclassified by Stochastic Gradient Descent and classified correctly by MedRoBERTa.nl-HAGA

Similarly to with string matching, we are interesting in to what extent BERT-like models are better able to classify patients on their smoking status than classical machine learning methods are. Note that the LIME classifier could not be used on the sklearn library we used for our Stochastic Gradient Descent approach. We instead opted to use the eli5 library, which has very similar workings as it shows the importances for each feature relative to the given classes. The shortcomings of stochastic gradient descent can be summarized by again misjudging which words are important in a text, assigning too high importances to words that seem irrelevant regarding to smoking status. This is especially evident in text 3, where stochastic gradient descent classified a text containing no information about the patient's smoking status as a non-smoker. We show the predictions of the models on this text below in figures 26 and 27.

Stochastic Gradient Descent
Prediction: Geen gebruiker (Non-user)

Figure 26: Stochastic Gradient Descent LIME Visualization of Text 3 of Experiment 5

The SGD classifier sees multiple words that lead it to classify the patient as a non-user, even though none of these words have anything to do with the act of smoking.

MedRoBERTa.nl
Prediction: Huidige gebruiker (Current user)

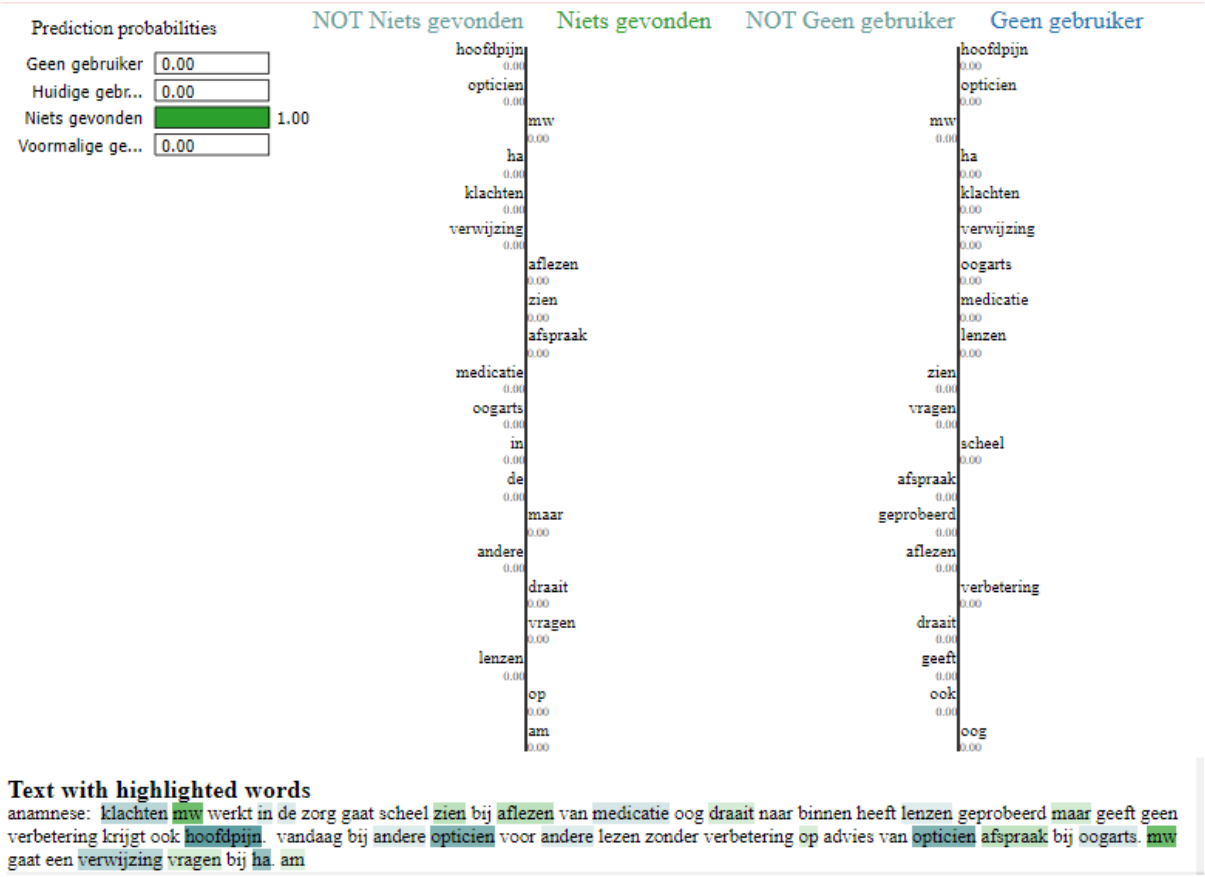


Figure 27: MedRoBERTa.nl LIME Visualization of Text 3 of Experiment 5

Even though there are a lot of indicators for a non-user, the indicators are very weak, leading the model to classify the text correctly. Similar issues are present in each of the 5 texts, showing that deeper models are more able to distinguish between importance between features than classical machine learning methods are, given the context of our experiments.

7.3 t-SNE Embeddings

In this section we show t-SNE embeddings for each of our pretrained models. We again only show these visualizations for the smoking lifestyle classification task. We compare these embeddings, evaluating which model is best able to group texts with the same labels together. We convert each text from the hand-labelled input set to embeddings, and then use t-SNE to position the embeddings in two-dimensional space. We consider a model to be adequate at distinguishing classes when the visualizations shows clumps of texts with the same label, with not too much overlap between the position of the texts of different classes.

7.3.1 HAGALBERT

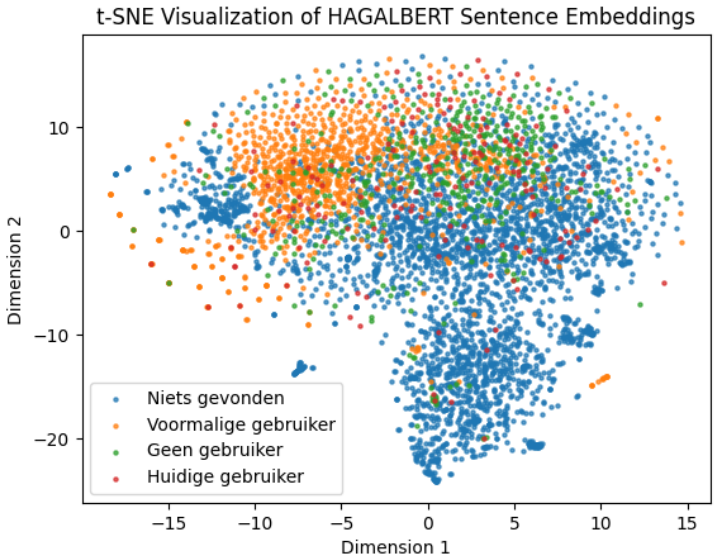


Figure 28: t-SNE visualization of HAGALBERT embeddings of all hand-labeled texts. The classes do not seem to be divided well, as relatively very many instances are placed in close proximity to instances of other classes.

We can see multiple trends within figure 28, there is a large group of "No information given" texts at the bottom of the figure, and a large collection of previous users in the top left. For the other classes, the spread of the embeddings seems random and not many clumps can be identified. This could indicate HAGALBERT is relatively bad at distinguishing between the classes.

7.3.2 RobBERT-HAGA

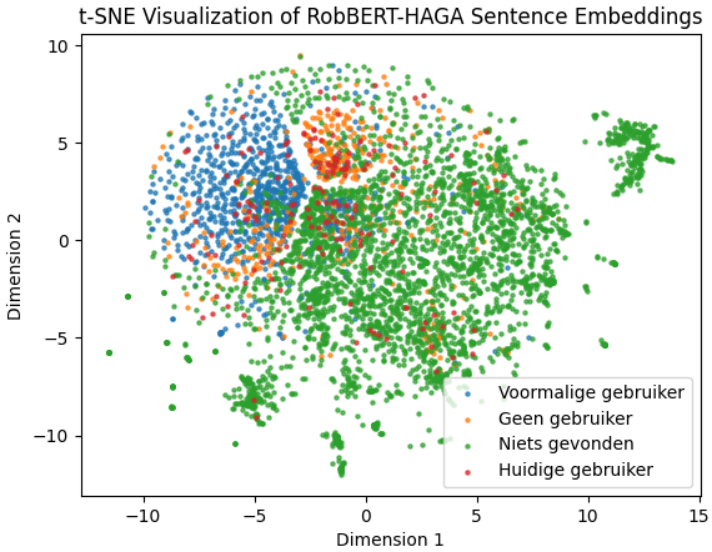


Figure 29: t-SNE visualization of RobBERT-HAGA embeddings of all hand-labeled texts. More separation is present, and clumps can be identified, as opposed to for the HAGALBERT image.

For RobBERT, the distinctions in figure 29 are more clear, and multiple secluded areas with the same label can be identified. Almost all previous users are situated in the top left compared to all over the figure for HAGALBERT, and on the right of the figure a large chunk of "no information given" texts can be found. There is also a large clump of non-users in the top of the middle part. Comparing this figure to HAGALBERT's, RobBERT-HAGA seems to be more able to group same labelled texts together.

7.3.3 belabBERT-HAGA

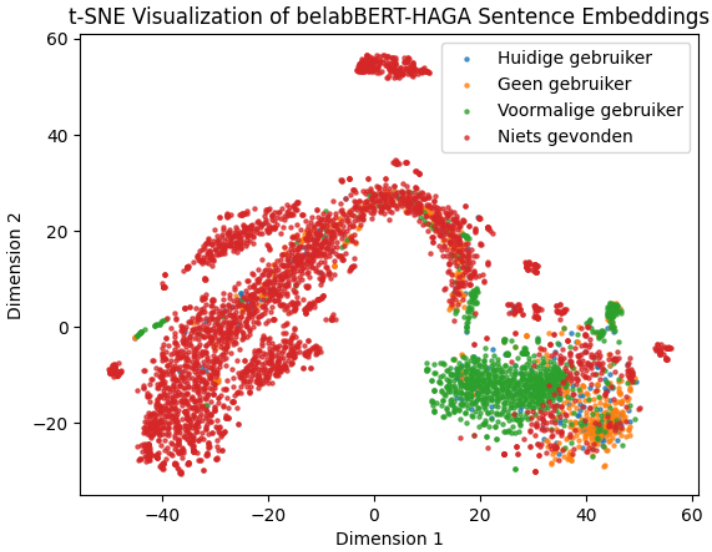


Figure 30: t-SNE visualization of belabBERT-HAGA embeddings of all hand-labeled texts. Some clumps can be identified, but a lot of different classes are present in the left "wave" on the figure.

From figure 30 we can ascertain that belabBERT is worse at distinguishing between the classes than RobBERT-HAGA. It is however adequate at grouping previous users together, which showed in section 7.1.3, as this is the class the model performed best on after finetuning. Non-users appear all over the figure and although some clumps can be identified, their placement still seems too broad to be able to say that the model grasps the classes at all. belabBERT's F1-score of 0.90 on the "no information given" class was the lowest amongst all models that are visualized in this section and their widely spread distribution in this image compared to the other models explains that. The rest of the classes seem grouped together in the bottom right, but are also common among the "red wave" in the left of the image, showing that the model is relatively very poor overall in distinguishing between classes.

7.3.4 MedRoBERTa.nl-HAGA

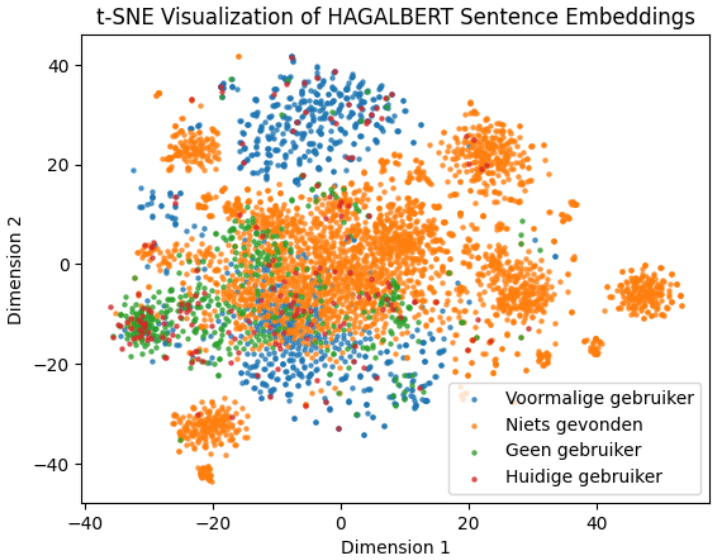


Figure 31: t-SNE visualization of MedRoBERTa.nl-HAGA embeddings of all hand-labeled texts. In comparison, MedRoBERTa.nl-HAGA is much better at distinguishing the classes. It has more clumps and classes are generally more spread out.

Figure 31 clearly shows how MedRoBERTa.nl is able to outperform the other pretrained models on the smoking task. Compared to the other figures, the "no information given" texts are way less spread out, and when they are, exist in clumps, showing the model is able to recognise specific types of texts within the "no information given" subset of texts. Furthermore, the non-users also seem way less spread out, showing clumps in the middle to middle-left of the figure. Previous users are somewhat grouped at the top-middle, but also occur often in the bottom-middle. Interestingly, none of the models seem to be able to group current users together well, indicating that there might be a relative lot of ways current smokers are denoted.

8 Discussion

In this section, we provide a detailed discussion of the results, the limitations of our research and we give suggestions for further research.

8.1 Background and Systematic Literature Review

In terms of machine learning methods, we chose the methods to train and test based on a singular paper (Dipnall et al., 2022), as that paper had similar goals and the amount of methods was manageable within the time constraints of our thesis. It could very well be that other types of classical machine learning approaches could have outperformed the ones used in this paper, which is something we leave out for future research.

We decided to perform a systematic literature review into the workings of BERT and the availability of clinical and/or Dutch BERT models as a continuation of our business understanding phase. The goal of this literature review was gathering knowledge of the workings of BERT, finding Dutch BERT models, finding clinical BERT models, altering text so it can be used for BERT and finding methods for explaining BERT models' output.

Within this systematic literature review we found that, in general, models that were pretrained on clinical text data outperform models that were pretrained on general data on clinical tasks. We furthermore found that pretraining a BERT model on top of existing weights tends to outperform models that were merely finetuned on new data. We also found that the RoBERTa architecture outperformed the standard BERT architecture numerous times, prompting us to use this architecture for our own experiments. Another interesting finding was that translation of texts to English outperformed a German BERT model in one paper, which lead us to experimenting with translating our texts to English in order to be able to use English BERT models.

It could be the case that the 85 papers we examined did not give a representative view of the BERT domain, and that in reality assessments that we have not made are closer to the "real" situation. However, judging by the fact that there were a number of source papers added in the backward snowballing phase, we believed we gathered the necessary knowledge of the existing BERT models in literature. Furthermore, we believe our query was extensive enough to cover the existing literature to a high enough standard, although future research could disprove this.

We conducted our literature review using the SYMBALS method, searching databases of scientific papers, screening them and performing backward snowballing. We picked four databases based on our own evaluation on the most important sources for scientific papers. Performing a literature study into the totality of databases that are available for this purpose could have increased the overall quality of the papers we yielded, although we believe the papers we found were of substantial quality for us to answer our research questions. We believe the relevance criteria we proposed for our active learning phase resulted in a reasonable reduction in the amount of papers. Then, our backward snowballing resulted in the addition of 32 papers to this set, totalling 85 papers.

Albeit this amount of papers was on the high end of what we were able to review in this thesis, we believe the extent of the variety amongst the papers gave us substantial practical knowledge and allowed for a smooth continuation of the thesis. We believe the BERT models we found fit well with the requirements, especially MedRoBERTa.nl as it is both a clinical and Dutch BERT model. For the methods we considered for data preprocessing and evaluation we decided to leave out most of them due to time constraints. It would be interesting for future research to test some of these methods, such as applying post-hoc rules to BERT, and visualizing models using UMAP and the BertViz toolkit.

8.2 Feasibility Study

In consultation with HagaZiekenhuis, we created a query to extract the input data for the rest of our thesis. Within this query, we assign labels automatically to each text that we extract. Our first research question concerns examining whether these automatically assigned labels suffice as input for the task of lifestyle classification of Dutch clinical notes. We answer this question with a feasibility study, where we train classical machine learning methods with these labels. Within this feasibility study, we concluded that the labels were not of a standard that adhered to our requirements, partly based on our observation of query parameters having a high feature importance for machine learning models trained on the query labeled data. These models achieved near-perfect performance on a test set which was also query labeled. This judgment of feature importances seems to hold up for smoking status, as we show later that the same machine learning methods are relatively far from perfection on the hand-labeled set. It however does not seem to hold up for alcohol and drug status for the aforementioned reason that our queries achieved similar near-perfection on the hand-labeled test set, apart from the current drugs user class. This could tie in again with the possibility that our string matching query encompasses these lifestyles very well.

Next to our judgment of the feature importances, we performed an edge case study, where we defined an edge case as a text with a high probability for a label that was not assigned to it. There might exist better methods for extracting edge cases which we did not explore. We extracted 100 edge cases per lifestyle, labeled them by hand and found an accuracy of 53%, way lower than the 90% threshold we set for workable labels. In retrospect, it might be that this threshold was too high to judge the edge cases accurately.

Furthermore, the edge cases were pulled from the entire bulk of 148.000 texts, and merely analyzing 100 texts per lifestyle (0,07%) might not have been enough to represent the edge cases well enough. It should be noted that these 100 texts were the 100/148.000 texts with the ultimate highest scores for another class, and it could be that these texts were structural outliers that are almost impossible to classify by any model, regardless of its depth. To counter this claim, it does not seem that the edge case texts are so far removed from "general" texts, judging from the tables we provided for each lifestyle which include exempts from the texts that resemble exempts from the texts of our LIME experiments in section 7.2.

8.3 Manually Labeled Dataset

We included the edge case texts in our manually labeled dataset with the goal of having our deeper BERT-like models adjust to them during finetuning. It is unclear exactly how much in-

fluence the inclusion of these texts had on the performance of the models. For future research, comparing models that were trained with and without these texts could provide clarification on this regard. We hand-labelled a total of 4.700 texts after we ascertained that query labels were not of a high enough standard for answering our research questions. Due to limited annotator availability, just 1.000 of these texts were shared amongst the annotators, while the rest of the texts were solely annotated by the first annotator.

We justified this by showing the inter-rater reliability on these 1.000 texts. This reliability is extraordinarily high. However, because a large portion of the remaining texts were both annotated and reviewed by the same person, bias could be present in this set, as well as possible errors. We believe the chance for this occurring is slim because of the relative easy nature of annotating the texts based on the lifestyle statuses, but it is a risk nonetheless.

Ultimately, the class distribution of the resulting dataset was skewed. For each of the three lifestyles, more than two-thirds of the total texts concerned texts that included no information about that particular lifestyle (the "no information given" class). Furthermore, except for smokers, there were almost no previous users for alcohol and drugs present in the dataset. We believe this is most likely a product of the actual situation, in that there are overall comparably very few patients that are either previous alcohol users or drug users and/or for which this fact is notable enough for a doctor to note during a consult. It seems that if we were to include more previous users of these substances in the dataset that our models would perform way better on these regards.

A future study in which a dataset is used with the same amount of entries for each class could result in higher F1-scores and thus a better model. To what extend this can be achieved remains to be seen, as there appear to be so few previous users for alcohol and drugs. A solution for this could be to group previous users and non-users together, in a similar way to how Heath did in their thesis and the "task 2" experiments we conducted in our query label feasibility study. Another way would be to generate synthetic data. We leave these alterations for future research as well.

8.4 Experimental Setup

Naturally, it could very well also be that these outcomes can be explained by our experimental setup. Our test set "only" includes 940 entries, which were randomly selected from our 4.700 manually labeled dataset. More texts or a more diverse set of texts could very well make the tasks more difficult, showing a need for deeper context-understanding models. The full set of texts, while partly comprised using "edge cases", might have been not diverse enough due to bias appearing somewhere along the creation process. We furthermore did not apply any hyperparameter tuning to any of our BERT-like models to keep within time constraints, which very well could have hampered performance. For this reason, we suggest experimenting with hyperparameter tuning in future research.

8.5 Experiment 1: Pretraining from Scratch

We pretrained a BERT model from scratch using the ALBERT architecture. This model performed comparably way worse than the models we pretrained on top of exisitng weights. We

theorize that this is because of the size of the input set. We had 148.000 clinical texts in total, with a total size of around 2 gigabytes. In comparison, RobBERT was trained on 39 GB of texts and MedRoBERTa.nl on 13 GB of clinical texts. The results suggest that having more in-domain training data produces higher performance, which is a large reason for why we believe HAGALBERT performed so much worse.

It could be that using a different BERT model architecture for our pretraining from scratch approach improves its Macro F1-score on the test set, but we believe that no setup could come close to pretraining on top of other models due to the sheer difference in training size. The ALBERT architecture includes measures like factorized embedding parameterization and cross-layer parameter sharing and the inclusion of both or either one could also have hampered HAGALBERT's performance. Of course, testing this in future research could prove or disprove this. Our LIME and t-SNE visualizations strengthen these points, as HAGALBERT was unable to learn much context compared to MedRoBERTa.nl-HAGA and is less able to distinguish between classes in the embedding space.

8.6 Experiment 2: Pretraining using existing weights

Before we conducted any of our BERT experiments, we held the hypothesis that MedRoBERTa.nl would yield the best results and this came out to be true. As the model parameters are the same among the models and the models were finetuned for the same amount of iterations, it is fair to state that this hypothesis is true within the context of our thesis. Interestingly, belabBERT also got outperformed by the other models, even though it was trained on more general Dutch text data than RobBERT. Both of these models are also RoBERTa-based models. Furthermore, in its respective paper, belabBERT outperformed RobBERT in a direct comparison in a sentiment analysis classification task in belabBERT's source paper.

A possible explanation for the results of this thesis could be that belabBERT was trained on a non-shuffled dataset, but it seems unlikely to us that this produced such a vast difference. Error analysis did not fit in the constraints of this thesis, but could have helped in explaining this phenomenon. The t-SNE visualization of belabBERT did show an interesting trend, suggesting that embeddings were possibly not created correctly due to their abnormal shape in the t-SNE graph compared to the graphs of the other models.

We theorize that the combination of the large amount of input data and the fact that MedRoBERTa.nl is trained on clinical text are the main reasons for MedRoBERTa.nl-HAGA outperforming any other model that was initialized using another model's weights. For future research on Dutch clinical text lifestyle classification, we suggest either directly comparing MedRoBERTa.nl-HAGA to newer/other models or pretraining on top of MedRoBERTa.nl-HAGA using more input data, in a similar manner as we did to train MedRoBERTa.nl to MedRoBERTa.nl-HAGA.

8.7 Experiment 3: Translating Dutch input data for English clinical domain BERT models

We translated only the 4.700 hand-labeled texts to English for finetuning BioBERT and ClinicalBERT. We did this as we were unable to translate the entire bulk of texts within reasonable

time. We used the *opus-mt-nl-en* model for this, because of its low cost and availability within HuggingFace. We believe there might be better (neural) translation models available, and a more sophisticated model might improve performance in future research. The translated finetuned models performed relatively really well, outperforming every other model apart from MedRoBERTa.nl-HAGA on the smoking task. ClinicalBERT was even able to outperform MedRoBERTa.nl-HAGA on the alcohol task.

There are multiple possible explanations for this relatively high performance. For one, it could be that pretraining using our bulk of texts curtailed the other models' classification abilities and that models like RobBERT and MedRoBERTa.nl would perform better on the test set if we had not pretrained them further but rather finetuned them directly like we did for ClinicalBERT and BioBERT. Another reason could again be the difference in domain-specific training data. ClinicalBERT in particular was trained on over 2 million clinical notes, which is comparable to MedRoBERTa.nl and contains much more clinical data than RobBERT and belabBERT. This could explain why ClinicalBERT performs similarly to MedRoBERTa.nl and outperforms RobBERT and belabBERT on these tasks. Of course this does not take into account that MedRoBERTa.nl was first further pretrained on our 148.000 clinical notes from HagaZiekenhuis.

Within our LIME experiments, we showed 5 texts to which ClinicalBERT assigned the right labels and RobBERT did not. We found here that ClinicalBERT was better than RobBERT at grasping the context of the texts, by having less and more reliable predictors per class. For example, for one text ClinicalBERT recognized that "smoke from fire" did not refer to the act of smoking, whereas RobBERT did view the phrase as such. RobBERT also seemed to have more trouble than ClinicalBERT viewing the phrase "stopped smoking" as indicator for a previous user.

We believe for these reasons that the high performance of finetuning on translated texts is very promising, and we very strongly recommend to explore this option within other experiments where BERT models are applied. For example, by finding a way to reduce the time needed to translate a text, bigger datasets can be translated within reasonable time and can then serve as input, possibly improving performance even more. We especially recommend taking this approach for Dutch BERT research as, as far as we could find, translating Dutch texts to English and then using English BERT models is a scientific novelty.

8.8 Comparing Results among Models

Within the context of the experiments we performed, which were tailored to the data at hand, as well as HagaZiekenhuis' requirements, we found varying results between classifying smoking status, alcohol status and drug use status among patients. On smoking status, string matching was outperformed by our application of deep BERT models. Here, our model that was pretrained on top of the MedRoBERTa.nl model and was finetuned on our hand-labeled set performed the best, by achieving a Macro F1-score of 0.93. For alcohol and drug use however, we found that the string matching queries we created were able to achieve near perfection on our uniform test set.

There are multiple possible explanations for this. For one, it could be that the queries we

used for alcohol and drug status are extensive enough to encompass almost all cases. It could also be explained by the fact that alcohol and drug status are often noted in a standardized way, which we encountered during manually labelling our dataset. This would make it relatively straightforward for a query to grasp the entire context of a patient's use of these substances. It could also be that alcohol and drug status are simply less important to note than smoking status, for example it could be the case that a patient that smokes has more or more widespread health risks than a user of alcohol or drugs. This could then lead doctors to note smoking status more often and more elaborately.

In our LIME experiment 5 we compared our string matching method to MedRoBERTa.nl on five texts. This experiment provides useful insights in how the queries could be improved and it shows partly why string matching was outperformed by MedRoBERTa.nl. We can ascertain from these texts that very minor alterations could be made to the queries in order to to improve performance. For example, by adding "stoppen met roken" to the query for previous users, 2 of the 5 shown texts would be classified correctly by string matching alone. It appears to be the case that the task of classifying smoking status is potentially just as solvable as the alcohol and drug statuses, given more query conditions are added.

In this thesis we did not have time to perform an extensive error analysis, but this could aid in researching whether a similar performance to alcohol and drugs can be achieved for queries on smoking status. The precisions and recalls for these classes for string matching show that the method is struggling in particular with current users and non-users, so more accurate queries for these classes could improve the F1-score significantly.

8.9 Recommendations HagaZiekenhuis

Ultimately, we propose the MedRoBERTa.nl-HAGA model to HagaZiekenhuis for extracting patients' smoking statuses. This model performed the best on this task in terms of Macro F1-score. While this model is highly accurate, we show in LIME experiment 2 that there are certain situations in which the model is unable to identify the correct status. It seems that the model has trouble completely understanding the context of the word "niet" (not) in the texts. While this word should negate the other predictors in the near proximity it instead gets interpreted as part of another class, such as in the example "is niet gestopt met roken" (has not stopped smoking). These are factors that need to be reckoned with after implementing the model.

However, out of all of the models, MedRoBERTa.nl-HAGA clearly showed it is able to understand and interpret the context smoking status in clinical texts best. This is especially evident in LIME experiments 1, 4 and 5, where MedRoBERTa.nl-HAGA was shown to only assign weight to relevant features and can interpret these features better than the models it was compared to. Furthermore, in our t-SNE visualization we show that MedRoBERTa.nl-HAGA is much more able to distinguish between classes than HAGALBERT, RobBERT-HAGA and belabBERT-HAGA, as texts of the same class are closer to each other in comparison and more secluded clumps are present.

For extracting alcohol use status and drugs use status we suggest refining the string matching query used in this thesis, as this method yielded the highest Macro F1-score on these tasks.

The queries could already be implemented for these purposes, as they already perform almost flawlessly on classifying both alcohol and drugs usage, with the exception for classifying current drug users, for which there is still significant performance to be gained. For these reasons we specifically performing tests with texts that concern current drugs users and refine the queries using the findings that are made.

8.10 Recommendations Future Research

All in all, for future research, we recommend experimenting with different class distributions in the input dataset. Furthermore, for the input dataset we recommend testing whether the inclusion of the edge cases texts had a positive influence or not. For classical machine learning approaches, we advise to test more methods as this could possibly increase performance. A literature review of model parameters can also increase performance when a grid search is applied rather than the random search that was applied in our experiments. We furthermore recommend experimenting with different model setups and architectures for BERT pretraining from scratch, as its performance was significantly worse than almost all of the other models that were tested. The inclusion of more or more balanced input data could also positively influence its performance. For all BERT models, the application of hyperparameter tuning could increase performance in future research.

For translation, we suggest comparing the neural translation model we used to more sophisticated neural translation models and finding a way to translate large datasets within reasonable time, as translation showed great promise in the limited form it was implemented in this thesis. Lastly, an extensive error analysis for the smoking task could aid in deciding whether to use deep BERT models or string matching, as there exists a possibility that the addition of certain terms to the string matching queries could lead to string matching achieving similar performance on the smoking task as it did on the alcohol and drug usage tasks.

9 Conclusion

In this thesis, we attempted to answer our main research question. In order to be able to answer this question we first answered four subquestions, which are summarized in sections 9.1 to 9.5. In order to answer our research questions we conducted a case study with HagaZiekenhuis, from which we received a collection of clinical notes. Each of these texts is labeled using string queries. The collection of these queries serves as our string matching model.

We conducted a literature review into the field of text classification and classical machine learning approaches. We conducted a separate systematic literature review into BERT models as a continuation of our business understanding phase, with the overall goal of gathering knowledge of the workings of BERT, Dutch BERT models and clinical BERT models. Overall, we decided to create and evaluate several different BERT-like models on the task-at-hand, these being a pretrained model from scratch, several models that were trained using existing Dutch models' weights and using English clinical BERT models.

For our data extraction process, multiple queries were created in consultation with HagaZiekenhuis, resulting in each text being labeled on the basis of the respective patient's smoking, alcohol and drug usage status.

9.1 Subquestion 1

This question is: "To what extent do labels obtained from string matching queries suffice as input for the task of lifestyle classification of clinical notes?". In order to test the quality of these labels, we trained classical machine learning methods on the query labels. These methods were able to almost flawlessly predict labels on our test sets. We found however that the most important features from those methods almost completely entailed the query parameters we used to label the texts, meaning the models were not learning new information. In order to test the usefulness of the labels we extracted edge cases texts and found that only 53% of the labels of these texts were correct, prompting us to disregard the query labels and move towards creating a hand-labeled input dataset.

For this dataset, three annotators were involved in annotating a total of 4.700 texts on the basis of smoking, alcohol and drug use status. From these texts, a training set of size 3.760 and a test set of size 940 were selected randomly. On this test set, we evaluated our string matching, classical machine learning and BERT-like approaches.

9.2 Subquestion 2

Subquestion 2 is: "To what extent can the task of lifestyle classification of clinical notes be solved using string matching?". On the classification of smoking status, we found that the string matching method we applied was able to achieve a Macro F1-score of 0.84. On the classification of alcohol use, the Macro F1-score was 0.98 and on drug use the Macro F1-score was 0.91.

9.3 Subquestion 3

Subquestion 3 is: "To what extent can the task of lifestyle classification of clinical notes be solved using classical machine learning methods?". Our Stochastic Gradient Descent approach achieved a Macro F1-score of 0.85 on smoking, 0.71 on alcohol and 0.60 on drugs.

9.4 Subquestion 4

Subquestion 4 entails: "To what extent can the task of lifestyle classification of clinical notes be solved using BERT-like models?" On the classification of smoking, we found that deeper BERT models outperform both string matching and classical machine learning methods. Our best-performing BERT model on each lifestyle task, MedRoBERTa.nl-HAGA, was able to achieve a Macro F1-score of 0.93 on smoking. Close second was our ClinicalBERT model that was solely finetuned on translated data, which achieved a Macro F1-score of 0.92. On alcohol, MedRoBERTa.nl-HAGA achieved a Macro F1-score of 0.79 and on drugs this score was 0.77.

9.5 Main Research Question

The goal of this thesis was to answer our main research question. The main research question is: "To what extent can the most appropriate BERT language model improve upon classical machine learning approaches and string matching on the task of classifying patient lifestyle statuses in free text Dutch clinical notes?". Within the context of this thesis we found our best-performing BERT model MedRoBERTa.nl-HAGA to outperform string matching and classical machine learning on our smoking classification task, achieving a Macro F1-score of 0.93, compared to 0.84 of string matching and 0.85 of classical machine learning. On our alcohol classification task MedRoBERTa.nl-HAGA was outperformed however by string matching, achieving a Macro F1-score of 0.79, while string matching achieved 0.98. It did however outperform classical machine learning, which achieved a score of 0.71. For the drugs task the same phenomenon occurred, with MedRoBERTa.nl-HAGA achieving a Macro F1-score of 0.77, string matching 0.91 and classical machine learning 0.60.

9.6 Contributions and Suggestions

In this thesis, we created and evaluated models for three lifestyle classification tasks. Within our HagaZiekenhuis case study, we suggest the MedRoBERTa.nl-HAGA model to them for extracting smoking status and the string matching queries for extracting alcohol and drugs statuses, as these were the models that yielded the highest performance. For future research, these models can be further analysed, as well as used as baselines for their respective tasks. We furthermore show the applicability of translation to Dutch BERT tasks, as these models yielded performance that was very close to MedRoBERTa.nl-HAGA, even though these models were only finetuned on a small subset of our data.

References

Abid, M. A., Ullah, S., Siddique, M. A., Mushtaq, M. F., Aljedaani, W., & Rustam, F. (2022, May). Spam SMS filtering based on text features and supervised machine learning techniques. *Multimedia Tools and Applications*, *81*(28), 39853–39871. Retrieved from https://doi.org/10.1007/s11042-022-12991-0 doi: 10.1007/s11042-022-12991-0

Adhikari, A., Ram, A., Tang, R., & Lin, J. (2019, August). DocBERT: BERT for Document Classification. Retrieved 2023-01-30, from http://arxiv.org/abs/1904.08398 (Publisher: arXiv)

Agarwal, S., & Yu, H. (2009, September). Automatically classifying sentences in full-text biomedical articles into introduction, methods, results and discussion. *Bioinformatics*, *25*(23), 3174–3180. Retrieved from https://doi.org/10.1093/bioinformatics/btp548 doi: 10.1093/bioinformatics/btp548

Agnikula Kshatriya, B. S., Sagheb, E., Wi, C.-I., Yoon, J., Seol, H. Y., Juhn, Y., & Sohn, S. (2021, November). Identification of asthma control factor in clinical notes using a hybrid deep learning model. *BMC medical informatics and decision making*, *21*(Suppl 7), 272. doi: 10.1186/s12911-021-01633-4

Akuma, S., Lubem, T., & Adom, I. T. (2022, September). Comparing bag of words and TF-IDF with different models for hate speech detection from live tweets. *International Journal of Information Technology*, *14*(7), 3629–3635. Retrieved from https://doi.org/10.1007/s41870-022-01096-4 doi: 10.1007/s41870-022-01096-4

Allada, Y. Wang, V. Jindal, M. Babee, H. R. Tizhoosh, & M. Crowley. (2021, November). Analysis of Language Embeddings for Classification of Unstructured Pathology Reports. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (pp. 2378–2381). doi: 10.1109/EMBC46164.2021.9630347

Almazaydeh, L., Abuhelaleh, M., Tawil, A. A., & Elleithy, K. (2023, April). Clinical text classification with word representation features and machine learning algorithms. *International Journal of Online and Biomedical Engineering (iJOE)*, *19*(04), 65–76. Retrieved from https://doi.org/10.3991/ijoe.v19i04.36099 doi: 10.3991/ijoe.v19i04.36099

Amin, S., Neumann, G., Dunfield, K., Vechkaeva, A., Chapman, K. A., & Wixted, M. K. (2019, September). MLT-DFKI at CLEF ehealth 2019: Multi-label classification of ICD-10 codes with BERT. , *2380*. Retrieved from https://ceur-ws.org/Vol-2380/paper_67.pdf

B. Yang, Y. Yang, Q. Li, D. Lin, Y. Li, J. Zheng, & Y. Cai. (2023, August). Classification of Medical Image Notes for Image Labeling by Using MinBERT. *Tsinghua Science and Technology*, *28*(4), 613–627. doi: 10.26599/TST.2022.9010012

Balasundaram, S., & Kapil. (2010, December). On lagrangian support vector regression. *Expert Systems with Applications*, *37*(12), 8784–8792. Retrieved from https://doi.org/10.1016/j.eswa.2010.06.028 doi: 10.1016/j.eswa.2010.06.028

Banerjee, I., Davis, M. A., Vey, B. L., Mazaheri, S., Khan, F., Zavaletta, V., ... Patel, B. (2022, November). Natural Language Processing Model for Identifying Critical Findings-A Multi-Institutional Study. *Journal of digital imaging*. doi: 10.1007/s10278-022-00712-w

Biau, G. (2012, April). Analysis of a random forests model. *J. Mach. Learn. Res.*, *13*(null), 1063–1095.

Boateng, E. Y., & Abaye, D. A. (2019, November). A review of the logistic regression model with emphasis on medical research. *Journal of Data Analysis and Information Processing*, *07*(04), 190–207. Retrieved from https://doi.org/10.4236/jdaip.2019.74012 doi: 10.4236/jdaip.2019.74012

C. Mao, L. Yao, & Y. Luo. (2020, December). A Pre-trained Clinical Language Model for Acute Kidney Injury. In *2020 IEEE International Conference on Healthcare Informatics (ICHI)* (pp. 1–2). doi: 10.1109/ICHI48887.2020.9374312

Chaichulee, S., Promchai, C., Kaewkomon, T., Kongkamol, C., Ingviya, T., & Sangsupawanich, P. (2022, August). Multi-label classification of symptom terms from free-text bilingual adverse drug reaction reports using natural language processing. *PloS one*, *17*(8), e0270595. doi: 10.1371/journal.pone.0270595

Chen, P.-F., Chen, L., Lin, Y.-K., Li, G.-H., Lai, F., Lu, C.-W., ... Lin, T.-Y. (2022, May). Predicting Postoperative Mortality With Deep Neural Networks and Natural Language Processing: Model Development and Validation. *JMIR medical informatics*, *10*(5), e38241. doi: 10.2196/38241

Chowdhury, G. (2003, January). Natural language processing. *Annual Review of Information Science and Technology*, *37*(1), 51–89. doi: 10.1002/aris.1440370103

Coutinho, I., & Martins, B. (2022, December). Transformer-based models for ICD-10 coding of death certificates with Portuguese text. *Journal of biomedical informatics*, *136*, 104232. doi: 10.1016/j.jbi.2022.104232

D. Pan, X. Zheng, W. Liu, M. Li, M. Ma, Y. Zhou, ... P. Wang (2020, May). Multi-label Classification for Clinical Text with Feature-level Attention. In *2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)* (pp. 186–191). doi: 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00042

de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., Noord, van, G., & Nissim, M. (2019, December). Bertje: A dutch bert model. *ArXiv*.

De Bruyne, L., De Clercq, O., & Hoste, V. (2021a, April). Emotional RobBERT and insensitive BERTje: Combining transformers and affect lexica for Dutch emotion detection. , 257–263. Retrieved from https://aclanthology.org/2021.wassa-1.27

De Bruyne, L., De Clercq, O., & Hoste, V. (2021b, December). Prospects for dutch emotion detection: Insights from the new emotionl dataset. *Computational Linguistics in the Netherlands Journal*, *11*, 231–255. Retrieved from https://clinjournal.org/clinj/article/view/138

Delobelle, P., Winters, T., & Berendt, B. (2020, November). RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the association for computational linguistics: Emnlp 2020* (pp. 3255–3265). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.findings-emnlp.292 doi: 10.18653/v1/2020.findings-emnlp.292

Delobelle, P., Winters, T., & Berendt, B. (2022, February). RobBERTje: a Distilled Dutch BERT Model. *Computational Linguistics in the Netherlands Journal 2021*. Retrieved 2023-02-11, from https://arxiv.org/abs/2204.13511 doi: 10.48550/ARXIV.2204.13511

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of

deep bidirectional transformers for language understanding. , 4171–4186. Retrieved from https://aclanthology.org/N19-1423 doi: 10.18653/v1/N19-1423

de Wynter, A., & Perry, D. J. (2020, November). Optimal Subarchitecture Extraction For BERT. Retrieved 2023-02-14, from http://arxiv.org/abs/2010.10499 (Publisher: arXiv)

Dipnall, J. F., Lu, J., Gabbe, B. J., Cosic, F., Edwards, E., Page, R., & Du, L. (2022, August). Comparison of state-of-the-art machine and deep learning algorithms to classify proximal humeral fractures using radiology text. *European journal of radiology*, *153*, 110366. doi: 10.1016/j.ejrad.2022.110366

Dixit, S., Mao, W., McDade, K. K., Schäferhoff, M., Ogbuoji, O., & Yamey, G. (2022, November). Tracking financing for global common goods for health: A machine learning approach using natural language processing techniques. *Frontiers in Public Health*, *10*. Retrieved from https://doi.org/10.3389/fpubh.2022.1031147 doi: 10.3389/fpubh.2022.1031147

Farhadian, M., Shokouhi, P., & Torkzaban, P. (2020, July). A decision support system based on support vector machine for diagnosis of periodontal disease. *BMC Research Notes*, *13*(1). Retrieved from https://doi.org/10.1186/s13104-020-05180-5 doi: 10.1186/s13104-020-05180-5

Feller, D. J., IV, O. J. B. D. W., Zucker, J., Yin, M. T., Gordon, P., & Elhadad, N. (2020, January). Detecting social and behavioral determinants of health with structured and free-text clinical data. *Applied Clinical Informatics*, *11*(01), 172–181. Retrieved from https://doi.org/10.1055/s-0040-1702214 doi: 10.1055/s-0040-1702214

Fink, M. A., Kades, K., Bischoff, A., Moll, M., Schnell, M., Küchler, M., ... Kleesiek, J. (2022, September). Deep Learning-based Assessment of Oncologic Outcomes from Natural Language Processing of Structured Radiology Reports. *Radiology. Artificial intelligence*, *4*(5), e220055. doi: 10.1148/ryai.220055

Fu, S., Thorsteinsdottir, B., Zhang, X., Lopes, G. S., Pagali, S. R., LeBrasseur, N. K., ... Sohn, S. (2022, March). A hybrid model to identify fall occurrence from electronic health records. *International journal of medical informatics*, *162*, 104736. doi: 10.1016/j.ijmedinf.2022.104736

Gaye, B., Zhang, D., & Wulamu, A. (2021, September). Sentiment classification for employees reviews using regression vector- stochastic gradient descent classifier (RV-SGDC). *PeerJ Computer Science*, *7*, e712. Retrieved from https://doi.org/10.7717/peerj-cs.712 doi: 10.7717/peerj-cs.712

Gomollón, F., Gisbert, J. P., Guerra, I., Plaza, R., Villarroya, R. P., Almazán, L. M., ... Montoto, C. (2021, December). Clinical characteristics and prognostic factors for crohn's disease relapses using natural language processing and machine learning: a pilot study. *European Journal of Gastroenterology &amp Hepatology*, *34*(4), 389–397. Retrieved from https://doi.org/10.1097/meg.0000000000002317 doi: 10.1097/meg.0000000000002317

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., ... Poon, H. (2022, January). Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ACM Transactions on Computing for Healthcare*, *3*(1), 1–23. Retrieved 2023-02-14, from https://dl.acm.org/doi/10.1145/3458754 doi: 10.1145/3458754

Han, S., Zhang, R. F., Shi, L., Richie, R., Liu, H., Tseng, A., ... Tsui, F. R. (2022, March). Classifying social determinants of health from unstructured electronic health records using deep learning-based natural language processing. *Journal of biomedical informatics*,

127, 103984. doi: 10.1016/j.jbi.2021.103984

Hassan, S. U., Ahamed, J., & Ahmad, K. (2022, April). Analytics of machine learning-based algorithms for text classification. *Sustainable Operations and Computers*, *3*, 238–248. Retrieved from https://doi.org/10.1016/j.susoc.2022.03.001 doi: 10.1016/j.susoc.2022.03.001

Heath, C. (2022). *Natural language processing for lifestyle recognition in discharge summaries.*

Hu, D., Zhang, H., Li, S., Wang, Y., Wu, N., & Lu, X. (2021, July). Automatic Extraction of Lung Cancer Staging Information From Computed Tomography Reports: Deep Learning Approach. *JMIR medical informatics*, *9*(7), e27955. doi: 10.2196/27955

Hu, W., & Wang, S. Y. (2022, March). Predicting Glaucoma Progression Requiring Surgery Using Clinical Free-Text Notes and Transfer Learning With Transformers. *Translational vision science & technology*, *11*(3), 37. doi: 10.1167/tvst.11.3.37

Huang, K., Altosaar, J., & Ranganath, R. (2020, November). ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. Retrieved 2023-01-30, from http://arxiv.org/abs/1904.05342 (Publisher: arXiv)

Huang, S., Cai, N., Pacheco, P. P., Narrandes, S., Wang, Y., & Xu, W. (2018, January). Applications of support vector machine (SVM) learning in cancer genomics. *Cancer Genomics &amp Proteomics*, *15*(1). Retrieved from https://doi.org/10.21873/cgp.20063 doi: 10.21873/cgp.20063

IBM. (n.d.). *What are naïve bayes classifiers?* Retrieved from https://www.ibm.com/topics/naive-bayes

Islam, M. Z., Liu, J., Li, J., Liu, L., & Kang, W. (2019, November). A semantics aware random forest for text classification. In *Proceedings of the 28th ACM international conference on information and knowledge management.* ACM. Retrieved from https://doi.org/10.1145/3357384.3357891 doi: 10.1145/3357384.3357891

Jaiswal, A., Tang, L., Ghosh, M., Rousseau, J. F., Peng, Y., & Ding, Y. (2021, December). RadBERT-CL: Factually-Aware Contrastive Learning For Radiology Report Classification. *Proceedings of machine learning research*, *158*, 196–208.

Jalal, N., Mehmood, A., Choi, G. S., & Ashraf, I. (2022, June). A novel improved random forest for text classification using feature ranking and optimal number of trees. *Journal of King Saud University - Computer and Information Sciences*, *34*(6), 2733–2742. Retrieved from https://doi.org/10.1016/j.jksuci.2022.03.012 doi: 10.1016/j.jksuci.2022.03.012

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., ... Liu, Q. (2020, November). TinyBERT: Distilling BERT for Natural Language Understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 4163–4174). Online: Association for Computational Linguistics. Retrieved 2023-02-14, from https://www.aclweb.org/anthology/2020.findings-emnlp.372 doi: 10.18653/v1/2020.findings-emnlp.372

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., & Levy, O. (2020, March). SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, *8*, 64–77. Retrieved from https://aclanthology.org/2020.tacl-1.5 doi: 10.1162/tacl$_{a0}$0300

Kibriya, A. M., Frank, E., Pfahringer, B., & Holmes, G. (2004, December). Multinomial naive bayes for text categorization revisited. In G. I. Webb & X. Yu (Eds.), *Ai 2004: Advances in artificial intelligence* (pp. 488–499). Berlin, Heidelberg: Springer Berlin Heidelberg.

Kormilitzin, A., Vaci, N., Liu, Q., & Nevado-Holgado, A. (2021, August).

Med7: A transferable clinical natural language processing model for electronic health records. *Artificial Intelligence in Medicine*, *118*, 102086. Retrieved from https://doi.org/10.1016/j.artmed.2021.102086 doi: 10.1016/j.artmed.2021.102086

Krippendorff, K. (1980). *Content analysis: An introduction to its methodology*. Sage.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019, December). ALBERT: A lite BERT for self-supervised learning of language representations. Retrieved from https://openreview.net/forum?id=H1eA7AEtvS

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020, February). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics (Oxford, England)*, *36*(4), 1234–1240. doi: 10.1093/bioinformatics/btz682

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... Zettlemoyer, L. (2020, July). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. , 7871–7880. Retrieved from https://aclanthology.org/2020.acl-main.703 doi: 10.18653/v1/2020.acl-main.703

Lin, C., Bethard, S., Dligach, D., Sadeque, F., Savova, G., & Miller, T. A. (2020, April). Does BERT need domain adaptation for clinical negation detection? *Journal of the American Medical Informatics Association : JAMIA*, *27*(4), 584–591. doi: 10.1093/jamia/ocaa001

Lin, C., Miller, T., Dligach, D., Bethard, S., & Savova, G. (2021, June). EntityBERT: Entity-centric Masking Strategy for Model Pretraining for the Clinical Domain. In *Proceedings of the 20th Workshop on Biomedical Language Processing* (pp. 191–201). Online: Association for Computational Linguistics. Retrieved 2023-01-31, from https://www.aclweb.org/anthology/2021.bionlp-1.21 doi: 10.18653/v1/2021.bionlp-1.21

Lin, C.-H., Hsu, K.-C., Liang, C.-K., Lee, T.-H., Liou, C.-W., Lee, J.-D., ... Fann, Y. C. (2021, November). A disease-specific language representation model for cerebrovascular disease research. *Computer methods and programs in biomedicine*, *211*, 106446. doi: 10.1016/j.cmpb.2021.106446

Lipton, Z. C., Elkan, C., & Naryanaswamy, B. (2014, September). Optimal thresholding of classifiers to maximize f1 measure. , 225–239.

Liu, H., Zhang, Z., Xu, Y., Wang, N., Huang, Y., Yang, Z., ... Chen, H. (2021, January). Use of BERT (Bidirectional Encoder Representations from Transformers)-Based Deep Learning Method for Extracting Evidences in Chinese Radiology Reports: Development of a Computer-Aided Liver Cancer Diagnosis Framework. *Journal of medical Internet research*, *23*(1), e19689. doi: 10.2196/19689

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019, July). RoBERTa: A Robustly Optimized BERT Pretraining Approach. Retrieved 2023-01-30, from http://arxiv.org/abs/1907.11692 (Publisher: arXiv)

Louppe, G. (2014, October). *Understanding random forests: From theory to practice.* Retrieved from http://rgdoi.net/10.13140/2.1.1570.5928 doi: 10.13140/2.1.1570.5928

Lu, H., Ehwerhemuepha, L., & Rakovski, C. (2022, July). A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance. *BMC Medical Research Methodology*, *22*(1). Retrieved from https://doi.org/10.1186/s12874-022-01665-y doi: 10.1186/s12874-022-01665-y

Matwin, S., & Sazonova, V. (2012, September). Direct comparison between support vector machine and multinomial naive bayes algorithms for medical abstract classification. *Journal of the American Medical Informatics Association*, *19*(5), 917–917. Retrieved from https://doi.org/10.1136/amiajnl-2012-001072 doi: 10.1136/amiajnl-2012-001072

McHugh, M. L. (2012, October). Interrater reliability: the kappa statistic. *Biochem Med (Zagreb)*, *22*(3), 276–282.

Michalopoulos, G., Wang, Y., Kaka, H., Chen, H., & Wong, A. (2021, June). UmlsBERT: Clinical domain knowledge augmentation of contextual embeddings using the Unified Medical Language System Metathesaurus. , 1744–1753. Retrieved from https://aclanthology.org/2021.naacl-main.139 doi: 10.18653/v1/2021.naacl-main.139

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021, April). *Deep learning–based text classification* (Vol. 54) (No. 3). Association for Computing Machinery (ACM). Retrieved from https://doi.org/10.1145/3439726 doi: 10.1145/3439726

Mitchell, J. R., Szepietowski, P., Howard, R., Reisman, P., Jones, J. D., Lewis, P., ... Rollison, D. E. (2022, March). A Question-and-Answer System to Extract Data From Free-Text Oncological Pathology Reports (CancerBERT Network): Development Study. *Journal of medical Internet research*, *24*(3), e27210. doi: 10.2196/27210

Mu, Y., Tizhoosh, H. R., Tayebi, R. M., Ross, C., Sur, M., Leber, B., & Campbell, C. J. V. (2021, July). A BERT model generates diagnostically relevant semantic embeddings from pathology synopses with active learning. *Communications medicine*, *1*, 11. doi: 10.1038/s43856-021-00008-0

Mutlu, G., & Acı, Ç. İ. (2022, October). SVM-SMO-SGD: A hybrid-parallel support vector machine algorithm using sequential minimal optimization with stochastic gradient descent. *Parallel Computing*, *113*, 102955. Retrieved from https://doi.org/10.1016/j.parco.2022.102955 doi: 10.1016/j.parco.2022.102955

Nanda, G., Vallmuur, K., & Lehto, M. (2018, January). Improving autocoding performance of rare categories in injury classification: Is more training data or filtering the solution? *Accident Analysis &amp Prevention*, *110*, 115–127. Retrieved from https://doi.org/10.1016/j.aap.2017.10.020 doi: 10.1016/j.aap.2017.10.020

Naseem, U., Dunn, A. G., Khushi, M., & Kim, J. (2022, December). Benchmarking for biomedical natural language processing tasks with a domain specific ALBERT. *BMC Bioinformatics*, *23*(1), 144. Retrieved 2023-01-31, from https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-022-04688-w doi: 10.1186/s12859-022-04688-w

Olthof, A. W., van Ooijen, P. M., & Cornelissen, L. J. (2022, October). The natural language processing of radiology requests and reports of chest imaging: Comparing five transformer models' multilabel classification and a proof-of-concept study. *Health Informatics Journal*, *28*(4), 14604582221131198. doi: 10.1177/14604582221131198

Palmer, E. L., Hassanpour, S., Higgins, J., Doherty, J. A., & Onega, T. (2019, July). Building a tobacco user registry by extracting multiple smoking behaviors from clinical notes. *BMC Medical Informatics and Decision Making*, *19*(1). Retrieved from https://doi.org/10.1186/s12911-019-0863-3 doi: 10.1186/s12911-019-0863-3

Park, H.-A. (2013, April). An introduction to logistic regression: From ba-

sic concepts to interpretation with particular attention to nursing domain. *Journal of Korean Academy of Nursing*, *43*(2), 154. Retrieved from https://doi.org/10.4040/jkan.2013.43.2.154 doi: 10.4040/jkan.2013.43.2.154

Park, S., Bong, J.-W., Park, I., Lee, H., Choi, J., Park, P., ... Kang, S. (2022, November). ConBERT: A Concatenation of Bidirectional Transformers for Standardization of Operative Reports from Electronic Medical Records. *Applied Sciences*, *12*(21), 11250. Retrieved 2023-02-11, from https://www.mdpi.com/2076-3417/12/21/11250 doi: 10.3390/app122111250

Qader, W. A., Ameen, M. M., & Ahmed, B. I. (2019, June). An overview of bag of wordsimportance, implementation, applications, and challenges. In *2019 international engineering conference (IEC).* IEEE. Retrieved from https://doi.org/10.1109/iec47844.2019.8950616 doi: 10.1109/iec47844.2019.8950616

Ramachandran, G. K., Lybarger, K., Liu, Y., Mahajan, D., Liang, J. J., Tsou, C.-H., ... Uzuner, (2023, February). Extracting medication changes in clinical narratives using pre-trained language models. *Journal of Biomedical Informatics*, 104302. doi: 10.1016/j.jbi.2023.104302

Rasmy, L., Xiang, Y., Xie, Z., Tao, C., & Zhi, D. (2021, May). Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *npj Digital Medicine*, *4*(1), 86. Retrieved 2023-02-14, from https://www.nature.com/articles/s41746-021-00455-y doi: 10.1038/s41746-021-00455-y

Reuver, M. (2020, November). *Finding the smoke signal: Smoking status classification with a weakly supervised paradigm in sparsely labelled dutch free text in electronic medical records.*

Richter-Pechanski, P., Geis, N. A., Kiriakou, C., Schwab, D. M., & Dieterich, C. (2021, December). Automatic extraction of 12 cardiovascular concepts from German discharge letters using pre-trained language models. *Digital health*, *7*, 20552076211057662. doi: 10.1177/20552076211057662

Rokach, L., & Maimon, O. (2005). Decision trees. In *Data mining and knowledge discovery handbook* (pp. 165–192). Springer-Verlag. Retrieved from https://doi.org/10.1007/0-387-25465-x$_9$ doi: 10.1007/0-387-25465-x$_9$

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019, October). Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, *abs/1910.01108*. Retrieved from http://arxiv.org/abs/1910.01108

Sarica, S., & Luo, J. (2021, August). Stopwords in technical language processing. *PLOS ONE*, *16*(8), e0254937. Retrieved from https://doi.org/10.1371/journal.pone.0254937 doi: 10.1371/journal.pone.0254937

Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020, March). A comparative analysis of logistic regression, random forest and KNN models for the text classification. *Augmented Human Research*, *5*(1). Retrieved from https://doi.org/10.1007/s41133-020-00032-0 doi: 10.1007/s41133-020-00032-0

Sharma, D. K., Chatterjee, M., Kaur, G., & Vavilala, S. (2022). 3 - deep learning applications for disease diagnosis. In D. Gupta, U. Kose, A. Khanna, & V. E. Balas (Eds.), *Deep learning for medical applications with unique data* (p. 31-51). Academic Press. Retrieved from

https://www.sciencedirect.com/science/article/pii/B9780128241455000058 doi: https://doi.org/10.1016/B978-0-12-824145-5.00005-8

Spruit, M., Verkleij, S., de Schepper, K., & Scheepers, F. (2022, February). Exploring Language Markers of Mental Health in Psychiatric Stories. *Applied Sciences*, *12*(4), 2179. Retrieved 2023-02-11, from https://www.mdpi.com/2076-3417/12/4/2179 doi: 10.3390/app12042179

Teitelbaum, J. (2022). *Lectures on machine learning.*

Tian, Y., Zhang, Y., & Zhang, H. (2023, January). Recent advances in stochastic gradient descent in deep learning. *Mathematics*, *11*(3), 682. Retrieved from https://doi.org/10.3390/math11030682 doi: 10.3390/math11030682

Tiedemann, J., & Thottingal, S. (2020, November). OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd annual conference of the european association for machine translation* (pp. 479–480). Lisboa, Portugal: European Association for Machine Translation. Retrieved from https://aclanthology.org/2020.eamt-1.61

Uman, L. S. (2011, February). Systematic reviews and meta-analyses. *J. Can. Acad. Child Adolesc. Psychiatry*, *20*(1), 57–59.

van Es, B., Reteig, L. C., Tan, S. C., Schraagen, M., Hemker, M. M., Arends, S. R. S., ... Haitjema, S. (2023, January). Negation detection in dutch clinical texts: an evaluation of rule-based and machine learning methods. *BMC Bioinformatics*, *24*(1). Retrieved from https://doi.org/10.1186/s12859-022-05130-x doi: 10.1186/s12859-022-05130-x

van Haastrecht, M., Sarhan, I., Ozkan, B. Y., Brinkhuis, M., & Spruit, M. (2021, May). SYMBALS: A systematic review methodology blending active learning and snowballing. *Frontiers in Research Metrics and Analytics*, *6*. Retrieved from https://doi.org/10.3389/frma.2021.685591 doi: 10.3389/frma.2021.685591

Van Olmen, J., Van Nooten, J., Philips, H., Sollie, A., & Daelemans, W. (2022, April). Predicting COVID-19 Symptoms From Free Text in Medical Records Using Artificial Intelligence: Feasibility Study. *JMIR Medical Informatics*, *10*(4), e37771. Retrieved 2023-02-11, from https://medinform.jmir.org/2022/4/e37771 doi: 10.2196/37771

Verkijk, S., & Vossen, P. (2021, December). Medroberta.nl: A language model for dutch electronic health records. *Computational Linguistics in the Netherlands Journal*, *11*, 141–159. Retrieved from https://clinjournal.org/clinj/article/view/132

Wang, H., Xiong, J., Yao, Z., Lin, M., & Ren, J. (2017, 12). Research survey on support vector machine. EAI. doi: 10.4108/eai.13-7-2017.2270596

Wang, Z., Shao, Y.-H., Bai, L., Li, C.-N., Liu, L.-M., & Deng, N.-Y. (2018, September). Insensitive stochastic gradient twin support vector machines for large scale problems. *Information Sciences*, *462*, 114–131. Retrieved from https://doi.org/10.1016/j.ins.2018.06.007 doi: 10.1016/j.ins.2018.06.007

Wirth, R., & Hipp, J. (2000, January). Crisp-dm: Towards a standard process model for data mining..

Wong, Z. S.-Y., & Akiyama, M. (2013). Statistical text classifier to detect specific type of medical incidents. *Studies in health technology and informatics*, *192*, 1053. doi: 10.3233/978-1-61499-289-9-1053

Wouts, J. V. (2020, July). Text-based classification of interviews for mental health – juxtaposing the state of the art. Retrieved 2023-02-14, from http://arxiv.org/abs/2008.01543 (Publisher: arXiv)

Xie, K., Gallagher, R. S., Conrad, E. C., Garrick, C. O., Baldassano, S. N., Bernabei, J. M.,

... Roth, D. (2022, April). Extracting seizure frequency from epilepsy clinic notes: a machine reading approach to natural language processing. *Journal of the American Medical Informatics Association : JAMIA*, *29*(5), 873–881. doi: 10.1093/jamia/ocac018

Yao, L., Jin, Z., Mao, C., Zhang, Y., & Luo, Y. (2019, December). Traditional Chinese medicine clinical records classification with BERT and domain specific corpora. *Journal of the American Medical Informatics Association*, *26*(12), 1632–1636. Retrieved 2023-01-31, from https://academic.oup.com/jamia/article/26/12/1632/5573314 doi: 10.1093/jamia/ocz164

Yao, L., Mao, C., & Luo, Y. (2019, April). Clinical text classification with rule-based features and knowledge-guided convolutional neural networks. *BMC Medical Informatics and Decision Making*, *19*(S3). Retrieved from https://doi.org/10.1186/s12911-019-0781-4 doi: 10.1186/s12911-019-0781-4

Yu, S., Su, J., & Luo, D. (2019, November). Improving BERT-based text classification with auxiliary sentence and domain knowledge. *IEEE Access*, *7*, 176600–176612. Retrieved from https://doi.org/10.1109/access.2019.2953990 doi: 10.1109/access.2019.2953990

Yu, Z., Yang, X., Sweeting, G. L., Ma, Y., Stolte, S. E., Fang, R., & Wu, Y. (2022, September). Identify diabetic retinopathy-related clinical concepts and their attributes using transformer-based natural language processing methods. *BMC medical informatics and decision making*, *22*(Suppl 3), 255. doi: 10.1186/s12911-022-01996-2

Zhan, K., Peng, W., Xiong, Y., Fu, H., Chen, Q., Wang, X., & Tang, B. (2021, April). Novel Graph-Based Model With Biaffine Attention for Family History Extraction From Clinical Text: Modeling Study. *JMIR medical informatics*, *9*(4), e23587. doi: 10.2196/23587

Zhao, S. S., Hong, C., Cai, T., Xu, C., Huang, J., Ermann, J., ... Liao, K. P. (2019, September). Incorporating natural language processing to improve classification of axial spondyloarthritis using electronic health records. *Rheumatology*, *59*(5), 1059–1065. Retrieved from https://doi.org/10.1093/rheumatology/kez375 doi: 10.1093/rheumatology/kez375

Zheng, C., sum Lee, M., Bansal, N., Go, A. S., Chen, C., Harrison, T. N., ... An, J. (2023, March). Identification of recurrent atrial fibrillation using natural language processing applied to electronic health records. *European Heart Journal - Quality of Care and Clinical Outcomes*. Retrieved from https://doi.org/10.1093/ehjqcco/qcad021 doi: 10.1093/ehjqcco/qcad021

Zhou, X., Wang, Y., Sohn, S., Therneau, T. M., Liu, H., & Knopman, D. S. (2019, October). Automatic extraction and assessment of lifestyle exposures for alzheimer's disease using natural language processing. *International Journal of Medical Informatics*, *130*, 103943. Retrieved from https://doi.org/10.1016/j.ijmedinf.2019.08.003 doi: 10.1016/j.ijmedinf.2019.08.003

A CTCue String Matching Query

In this section we provide the queries that were used to label the clinical texts from HagaZiekenhuis within CTCue. These queries also serve as our method of string matching, which we compared to our classical machine learning and BERT-like approaches in section 7.

A.1 Smoking

For our smoking, alcohol and drugs queries multiple conditions were used which were checked in order. In this section we lay out the conditions for every subclass of the smoking lifestyle in this order. Note that if one condition fails the next one is checked and so on. This means that the final class serves as the class to which texts are assigned to when all of the other conditions fail. As the texts are in Dutch, naturally the queries are as well. For this reason, we provide the Dutch keywords within quotation marks and their English translation in parentheses.

1. **Previous user**
   Report contains either:
   - "Rookt niet meer" (Does not smoke anymore)
   - "Gestopt met roken" (Stopped smoking)
2. **Current user**
   Report contains either:
   - "Rookt +" (Smokes +)
   - "Roken +" (Smoking +)
   - "Roker +" (Smoker +)
   - "Rookster +" (Smoker +)
   - "Rookt: ja" (Smokes: yes)
   - "Roker: ja" (Smoker: yes)
   - "Rookster: ja" (Smoker: yes)
   - "Rookt soms" (Smokes sometimes)
   - "Rookt: soms" (Smokes: sometimes)
   - "Roken + +" (Smoking + +)
3. **Non-user**
   Report contains either:
   - "Rookt -" (Smokes -)
   - "Roken -" (Smoking -)
   - "Roker -" (Smoker -)
   - "Rookster -" (Smoker -)
   - "Rookt: nee" (Smokes: no)
   - "Roker: nee" (Smoker: no)

- "Rookster: nee" (Smoker: no)
- "Rookt nooit" (Smokes never)
- "Rookt: nooit" (Smokes: never)
- "Roken - -" (Smoking - -)

4. **No information given**

A.2 Alcohol

For alcohol, the same principles apply as for smoking.

1. **Current user**
   Report contains either:
   - "Alcohol +"
   - "Alcohol: ja" (Alcohol: yes)
   - "Drinkt alcohol" (Drinks alcohol)
   - "Alcohol af en toe" (Alcohol from time to time)
   - "Alcohol per week" (Alcohol per week)
   - "Alcohol week"
   - "Invloed van alcohol" (Influence of alcohol)
2. **Non-user**
   Report contains either:
   - "Alcohol -"
   - "Alcohol: nee" (Alcohol: no)
   - "Drinkt geen alcohol" (Does not drink alcohol)
   - "Geen alcohol" (No alcohol)
   - "Alcohol: geheel niet" (Alcohol: not at all)
   - "Alcohol: geen" (Alcohol: none)
3. **No information given**

A.3 Drugs

For drugs, again, the same principles apply.

1. **Current user**
   Report contains either:
   - "Drugs +"
   - "Drugs: ja" (Drugs: yes)
   - "Gebruikt drugs" (Uses drugs)

2. **Non-user**
   Report contains either:
   - "Drugs -"
   - "Drugs: nee" (Alcohol: no)
   - "Drugs niet" (Drugs not)
   - "Geen drugs" (No drugs)
3. **No information given**

B Full Lime Experiments

B.1 Experiment 1

5 texts misclassified by HAGALBERT and classified correctly by MedRoBERTa.nl.

B.1.1 Text 1

Translated Text
"Anamnesis: Painful, cold soles of the feet. Especially in the cold months. Is now known to have hip complaints. Doesn't know if those originated from back issues. Tract history: Tract history: DM - chol - Smoking - Hypertension. Intervention for spinal stenosis. Physical examination: adp and atp +/+ now normal colour. Additional examination: Index: pressure of 1.2 bdz. Conclusion: Conclusion: Foot complaints related to the phenomenon of Raynaud. Possible in case of neurogenic dysregulation in case of possible recurrent spinal canal stenosis. No indication for PAOD Policy: Policy: Explanation. Wait and see analysis of the hip and, if necessary, start neurology traject."

True Label
Geen gebruiker (Non-user)

HAGALBERT
Prediction: Huidige gebruiker (Current user)

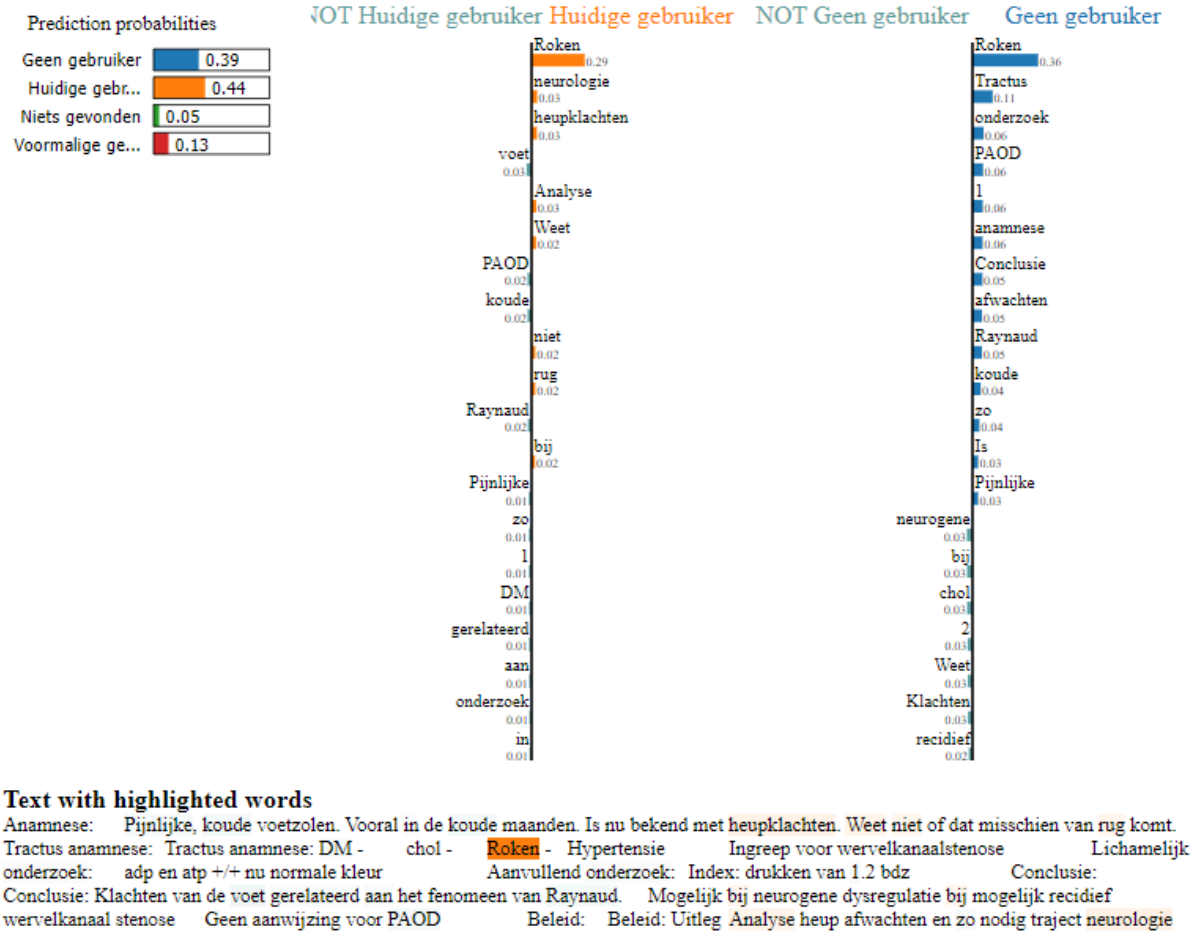


Figure 32: HAGALBERT LIME Visualization of Text 1 of Experiment 1

Judging from this image, HAGALBERT views the word "Roken" (smoking) as a strong indicator for both the current and non-user classes. The model ultimately decides to classify the text as a current user due to the term being more strongly related to current users. This is indicated by the colouring of the word in the text at the bottom.

MedRoBERTa.nl-HAGA

Prediction: Geen gebruiker (Non-user)

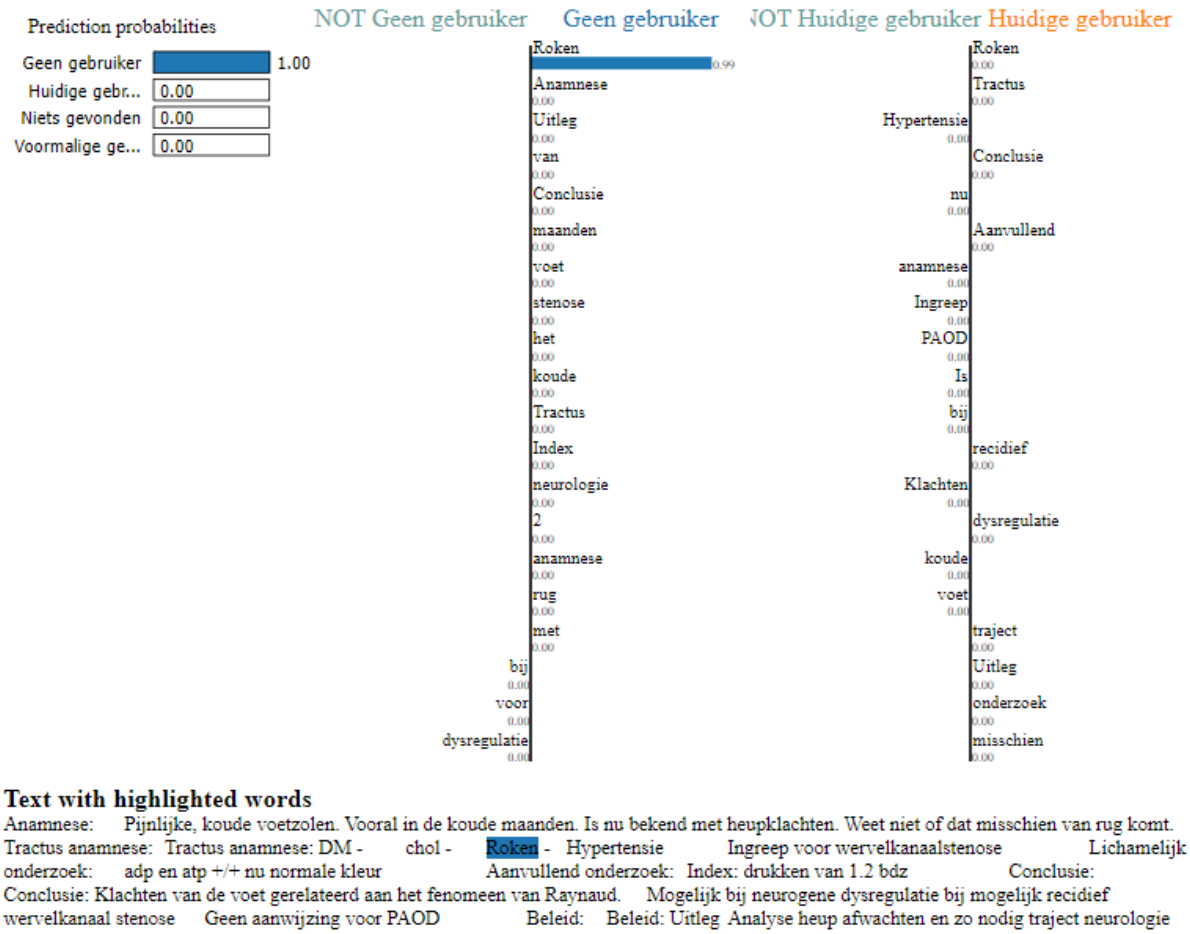


Figure 33: MedRoBERTa.nl-HAGA LIME Visualization of Text 1 of Experiment 1

For MedRoBERTa.nl there is way less disparity in its evaluation, being completely sure of its correct prediction of the non-user class. This indicates the model is better able to grasp context than HAGALBERT, by correctly viewing the term "Roken" in this context.

B.1.2 Text 2

Translated Text

"Summary: Course of radiology: 2014 Increased spondyloarthrosis and disc disease. 2016 flare after discontinuation. Case history: Talked about smoking, advice to stop. Sleeps well, also functions well in daily life. Adalimumab still per 4w. Physical examination: Hand osteoarthritis. Conclusion: Conclusion: AS in remission under adalimuma.b Policy: Policy: Co 1j (combi) + TC. Stop smoking. Try adalimumab per 5w."

True Label

Huidige gebruiker (Current user)

HAGALBERT

Prediction: Geen gebruiker (Non-user)

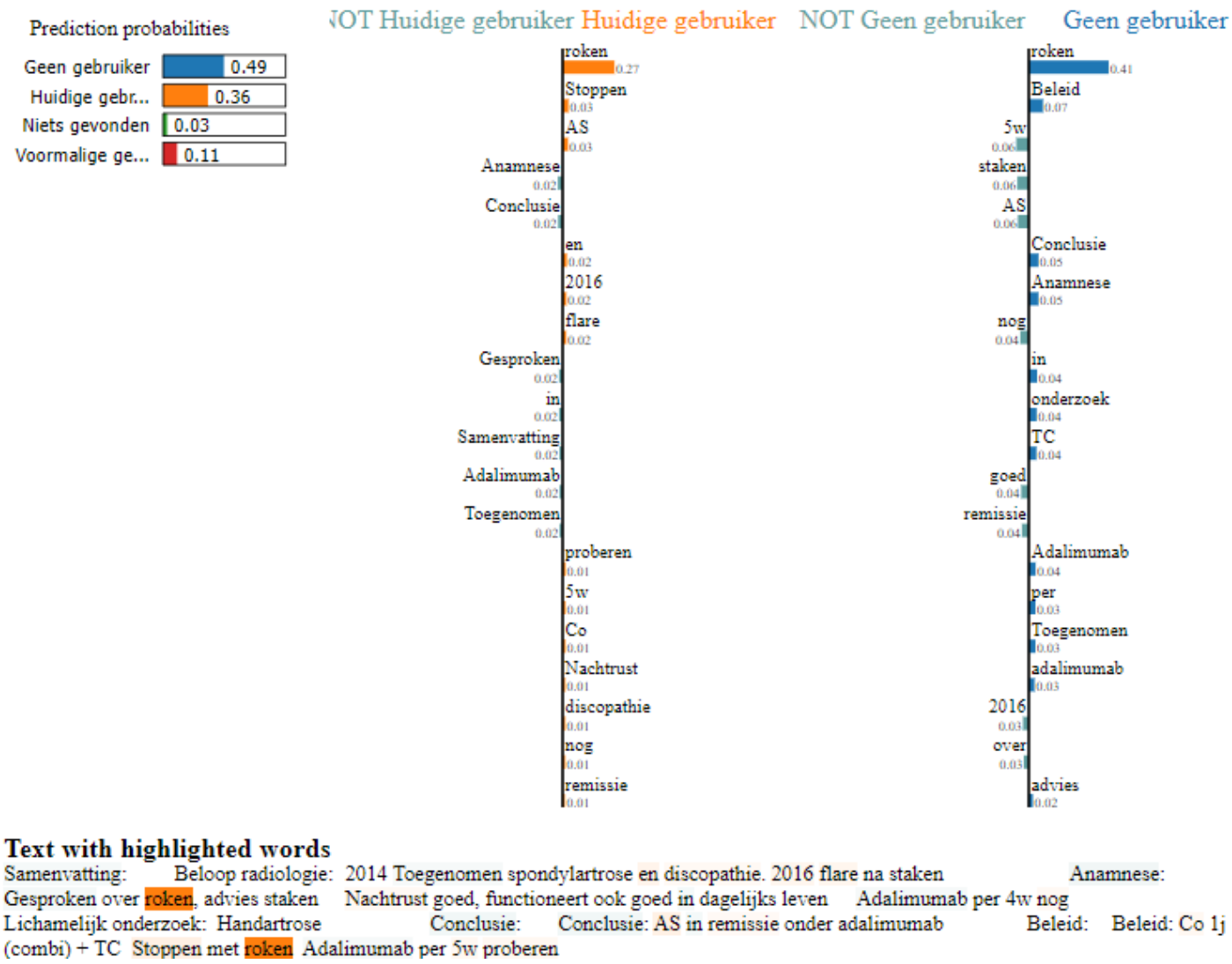


Figure 34: HAGALBERT LIME Visualization of Text 2 of Experiment 1

As was the case with text 1, HAGALBERT views the word "roken" as a strong indicator for both of the presented classes. However in this case, the model predicts the patient to be a non-user. Within the visualization this can be ascribed to other words in the text that push the model in that particular direction, such as "Beleid" (policy), "Conclusie" (conclusion) and "Anamnese" (anamnesis).

MedRoBERTa.nl-HAGA

Prediction: Huidige gebruiker (Current user)

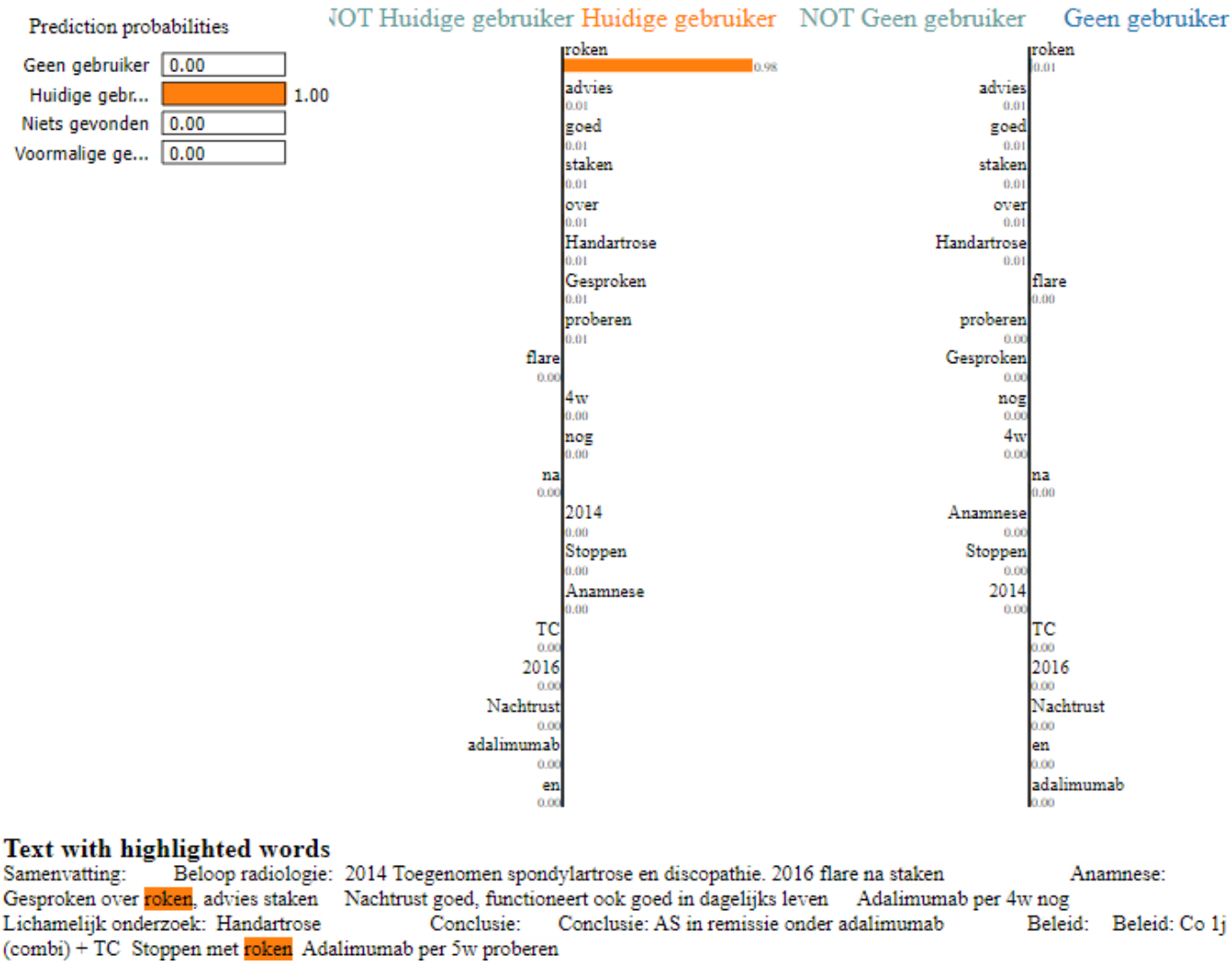


Figure 35: MedRoBERTa.nl LIME Visualization of Text 2 of Experiment 1

And again for MedRoBERTa.nl, the model is able to grasp the context of the word "roken" correctly, resulting in way less disparity and a correct prediction.

B.1.3 Text 3

Translated Text

"Summary: Course: Course: It is going well, Sputum is not too bad. Still would like to quit smoking and has seen a group session brochure in the HAGAstoppen. Doesn't feel like doing anything, daughter also thinks she has depression. This is all happening since Corona infection early December. Conclusion: Conclusion: No exacerbations, even despite Corona infection, but deconditioning, age, persistent smoking. Policy: Policy: Stop smoking medication by registering via lung nurse (Sinefuma was not possible). Other actions: - Appointment in 6 months"

True Label

Huidige gebruiker (Current user)

HAGALBERT

Prediction: Geen gebruiker (Non-user)

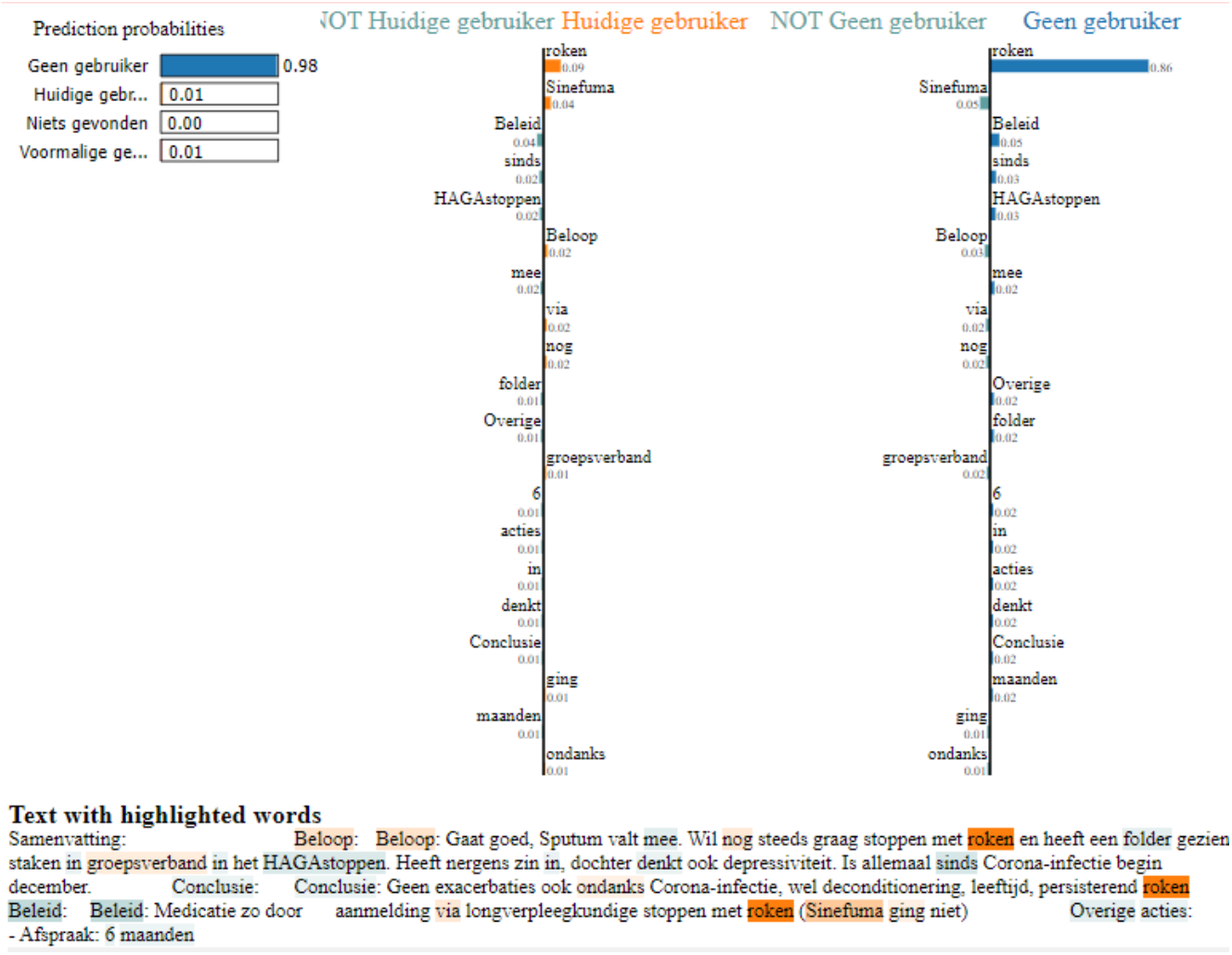


Figure 36: HAGALBERT LIME Visualization of Text 3 of Experiment 1

Interestingly, HAGALBERT seems to detect Sinefuma as a predictor for a current user, which is the true label. Sinefuma is a program that helps people to stop smoking, so this is wanted behaviour. Nevertheless, just as for text 1, the model views the presence of the words "Beleid" and "Conclusie" enough to strongly indicate the patient is a non-user, outweighing the words for current user.

MedRoBERTa.nl-HAGA
Prediction: Huidige gebruiker (Current user)

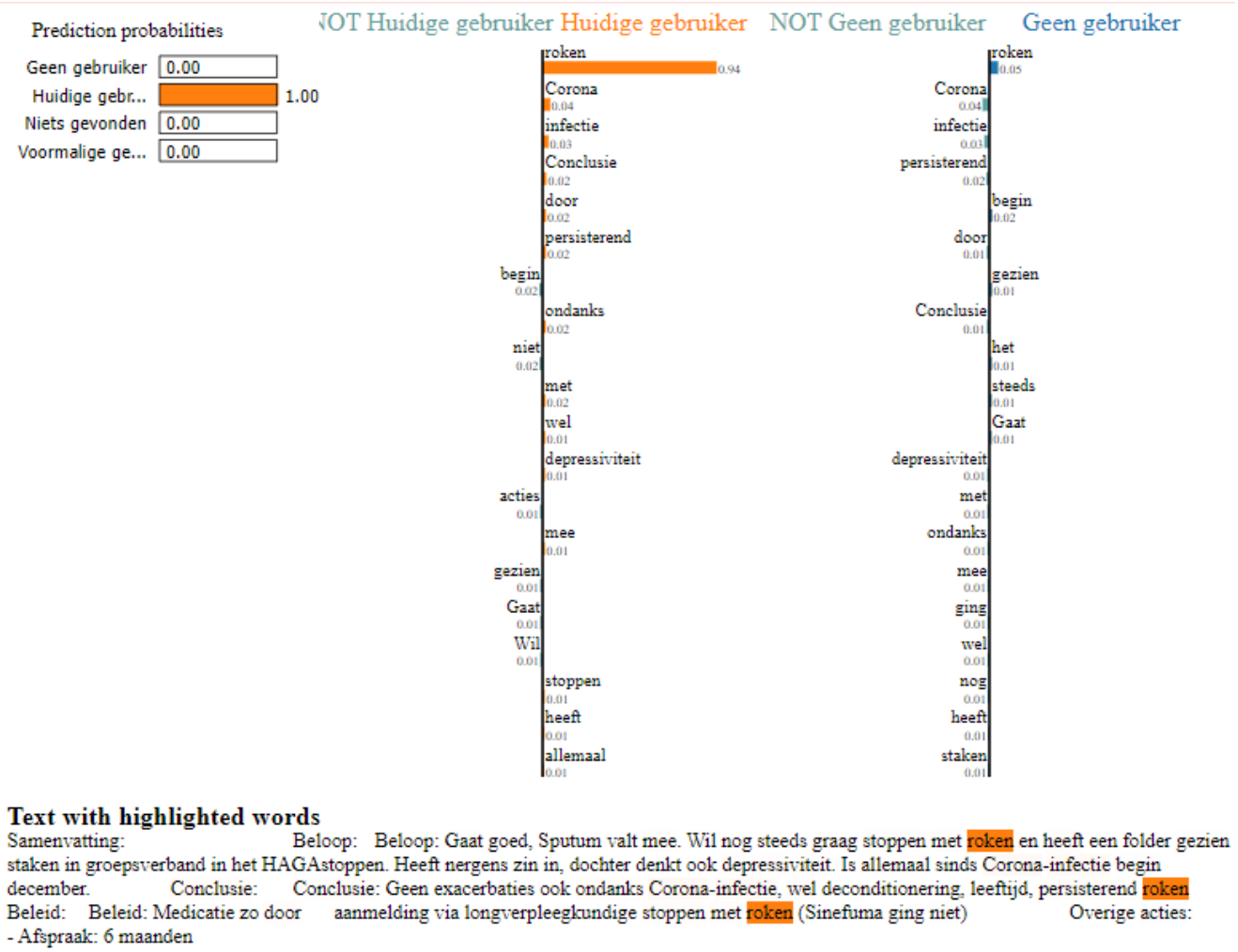


Figure 37: MedRoBERTa.nl LIME Visualization of Text 3 of Experiment 1

This model is again able to judge the context of the text perfectly, recognizing the patient has not stopped smoking yet and should therefore be classified as a current user.

B.1.4 Text 4

Translated Text

"Reason for coming / Referral: Reason for referral: Eyelids. Anamnesis: History Blank. Medication None. Anamnesis Since 0.5 years progressive headache and fatigue after onset depend on upper eyelid li. Can also see a little less here, especially when driving a car, has to lift the eyebrows completely. Can't read well at night because of fatigue. Doesn't notice any difference between left and right. Intoxications: Smoking stopped 20 years ago, alcohol, drugs. Allergies: None known. Physical examination: Eyebrows bdz at the level of the orbital rim. Dermatochalasis right mild, left to the eyelashes. Levator function bdz intact, good function. No ptosis. Conclusion: Conclusion: Mild dermatochalasis. Policy: Policy: Unfortunately, the severity of the dematochalasis does not meet the conditions that the health insurer sets for reimbursement for a blepharoplasty. The patient was informed of the possibilities of self-payment. The patient will think about this."

True Label

Voormalige gebruiker (Previous user)

HAGALBERT
Prediction: Geen gebruiker (Non-user)

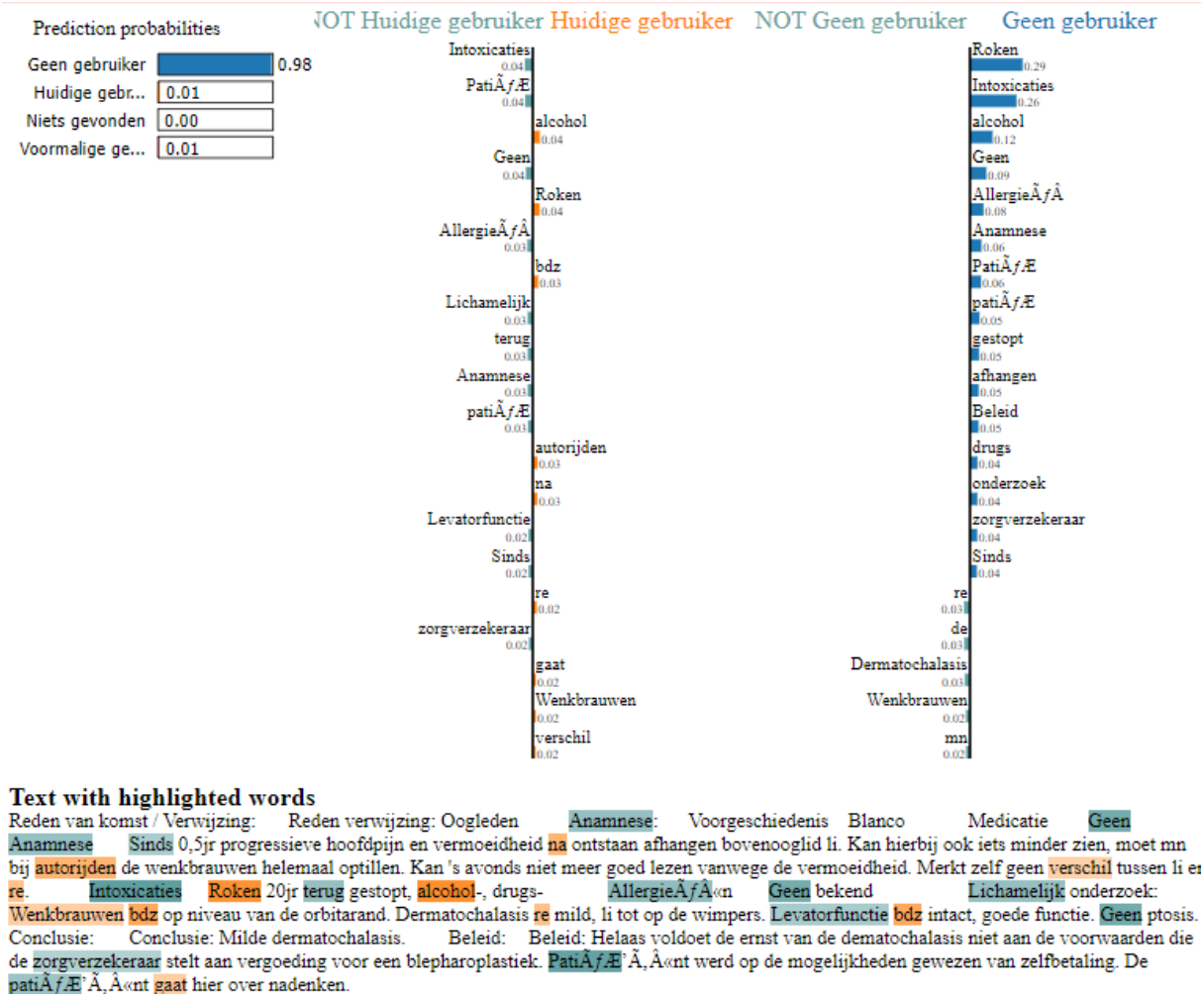
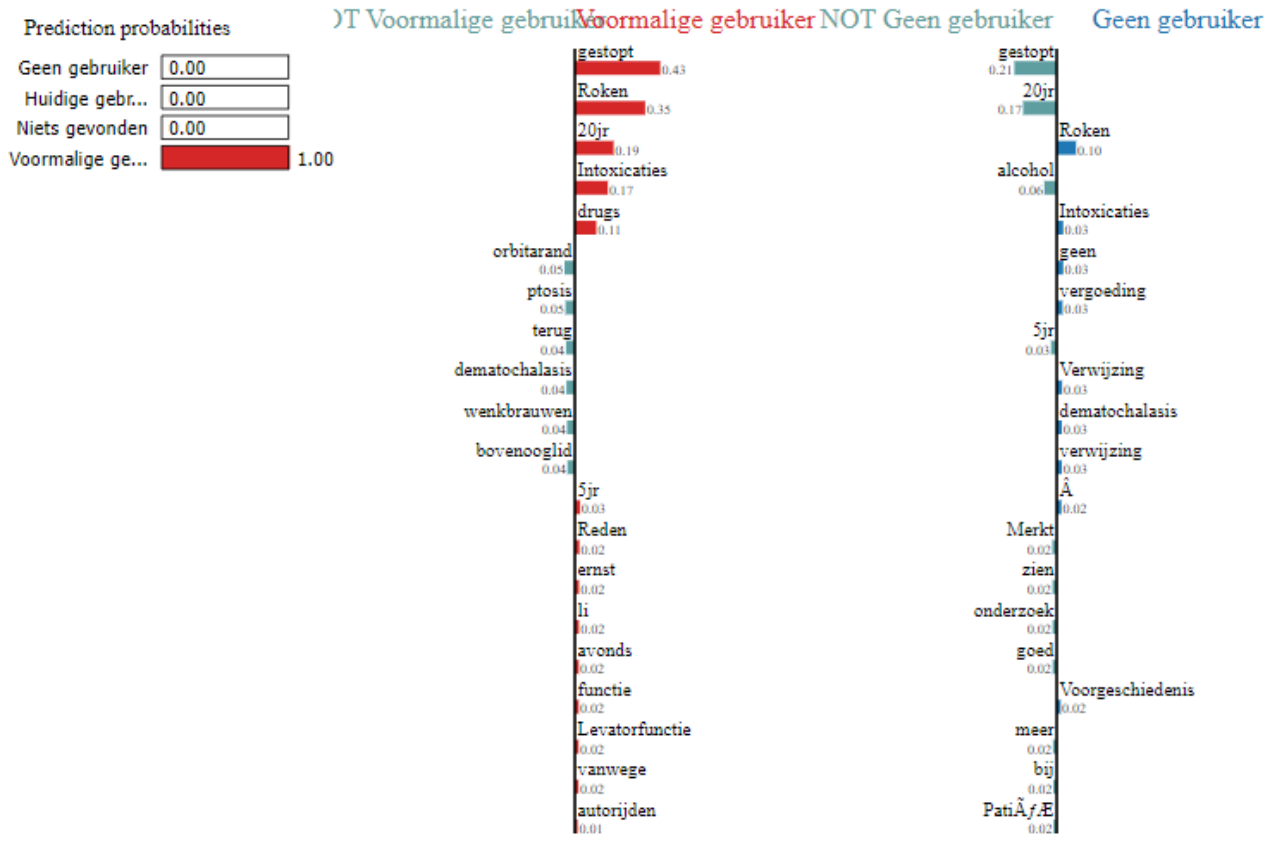


Figure 38: HAGALBERT LIME Visualization of Text 4 of Experiment 1

As established before, the presence of the Dutch word for anamnesis is regarded by HAGALBERT to be a strong indicator for the non-user class. We can ascertain from this image that the words "Intoxicaties" (intoxications), "Geen" (no/none), "Patient" and "Allergieën" (allerguies) further contribute to a non-user prediction. The reason the spelling of the two last words is different in the image is that the model parses the "ë" symbol in that particular way. It is interesting that the word "Intoxicaties" is seen as strong indicator for non-user as the word often denotes that the patient's smoking, alcohol and drug use statuses are stated next, regardless of the patient is an active user of those substances or not.

MedRoBERTa.nl-HAGA
Prediction: Voormalige gebruiker (Previous user)



Text with highlighted words

Reden van komst / Verwijzing: Reden verwijzing: Oogleden Anamnese: Voorgeschiedenis Blanco Medicatie Geen Anamnese Sinds 0,5jr progressieve hoofdpijn en vermoeidheid na ontstaan afhangen bovenooglid li. Kan hierbij ook iets minder zien, moet mn bij autorijden de wenkbrauwen helemaal optillen. Kan 's avonds niet meer goed lezen vanwege de vermoeidheid. Merkt zelf geen verschil tussen li en re. Intoxicaties Roken 20jr terug gestopt, alcohol-, drugs- AllergieÃƒÂ«n Geen bekend Lichamelijk onderzoek: Wenkbrauwen bdz op niveau van de orbitarand. Dermatochalasis re mild, li tot op de wimpers. Levatorfunctie bdz intact, goede functie. Geen ptosis. Conclusie: Conclusie: Milde dermatochalasis. Beleid: Beleid: Helaas voldoet de ernst van de dematochalasis niet aan de voorwaarden die de zorgverzekeraar stelt aan vergoeding voor een blepharoplastiek. PatiÃƒÆ'Ã‚Â«nt werd op de mogelijkheden gewezen van zelfbetaling. De patiÃƒÆ'Ã‚Â«nt gaat hier over nadenken.

Figure 39: MedRoBERTa.nl LIME Visualization of Text 4 of Experiment 1

The words that were deemed important by HAGALBERT are largely ignored by MedRoBERTa.nl and with good cause, as the model is able to predict the correct outcome with 100% certainty. Contrarily to HAGALBERT, this model views the word "Intoxicaties" as a strong indicator for the previous user class rather than the non-user class.

B.1.5 Text 5

Translated Text

"Summary: History: pleiomorphic adenoma removed left parotid gland. Medication: none. Allergy: -. Smoking: +. Policy: Policy: provisionally withholds treatment, now starts chemotherapy, will contact again in due course"

True Label

Huidige gebruiker (Current user)

HAGALBERT

Prediction: Geen gebruiker (Non-user)

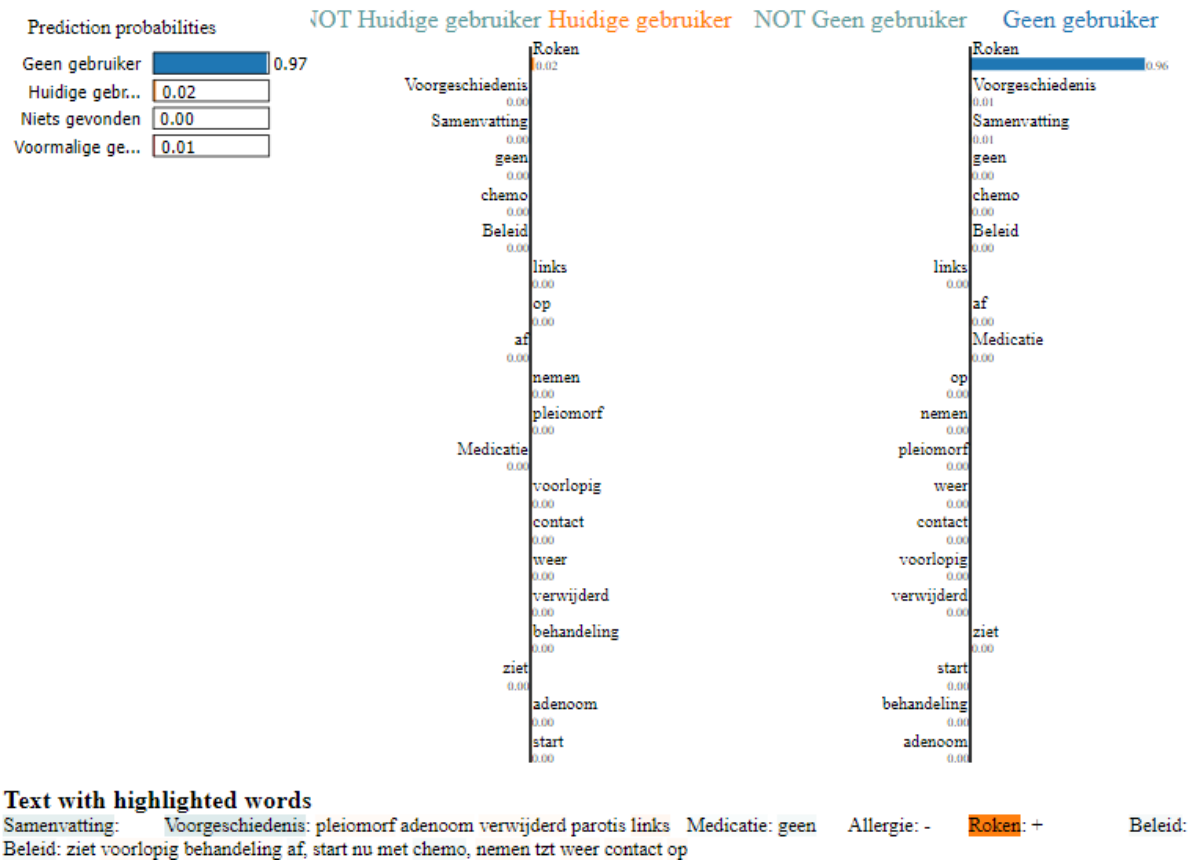


Figure 40: HAGALBERT LIME Visualization of Text 5 of Experiment 1

HAGALBERT is unable to recognise the "+" symbol as indicator for a current user and rather views the appearance of the word "roken" as indicator for non-user in this context, which is unwanted behaviour.

MedRoBERTa.nl-HAGA

Prediction: Huidige gebruiker (Current user)

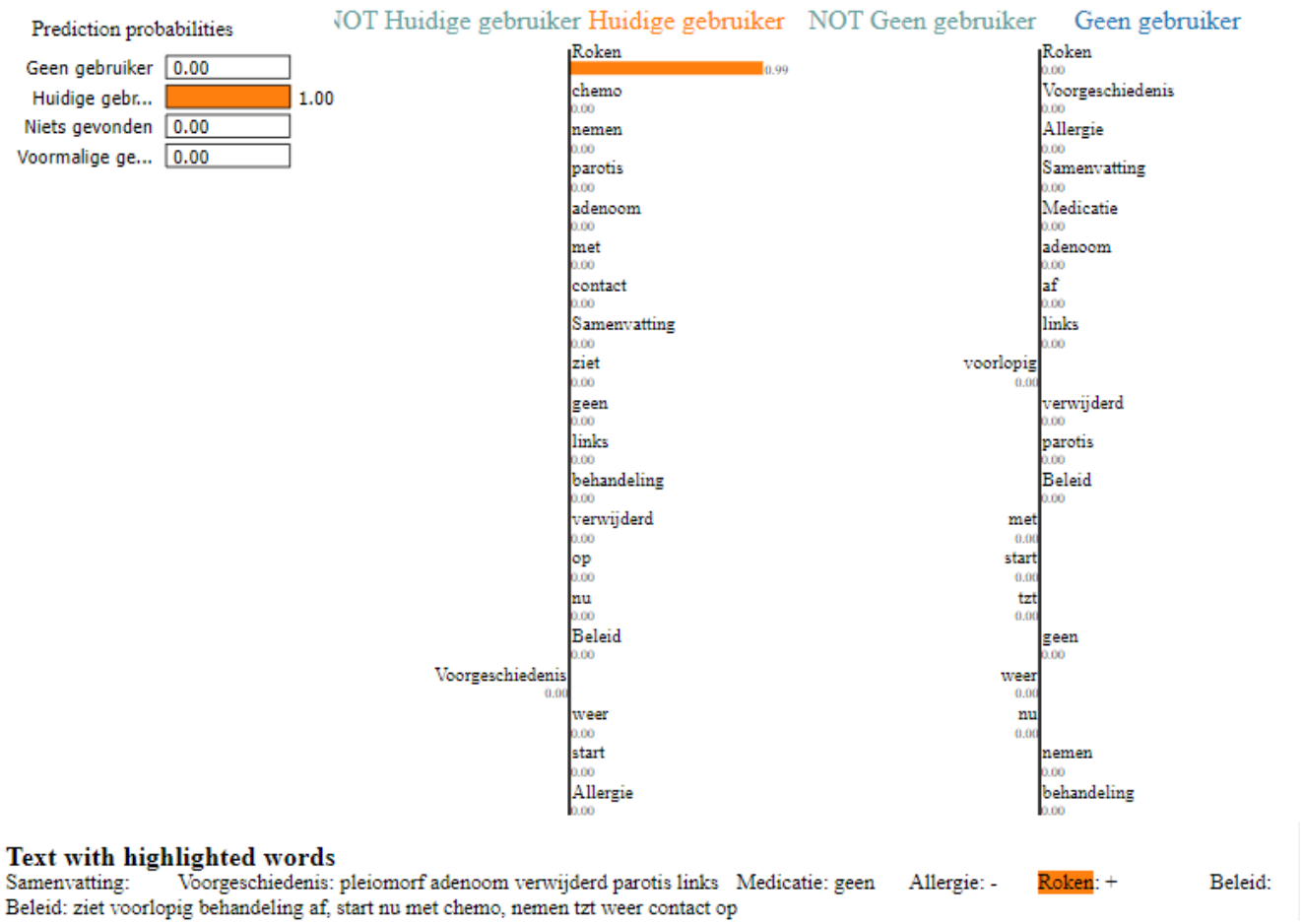


Figure 41: MedRoBERTa.nl LIME Visualization of Text 5 of Experiment 1

For MedRoBERTa.nl the context of the text is again understood perfectly, classifying the text correctly with full certainty.

B.2 Experiment 2

5 texts misclassified by MedRoBERTa.nl

B.2.1 Text 1

Translated Text
Anamnesis: Prolonged cough where sometimes white mucus comes up, complaints started after a cold. Not had these periods before. GP has prescribed a course. A bit short of breath while on prednison. Asthma does not run in the family. Has not smoked since November after starting again on 2nd. Medication: has had foster, doesn't help. It might relieve though. Covid: 3x plus booster. No admission. No surgeries.

True label
Voormalige gebruiker (Previous user)

MedRoBERTa.nl
Prediction: Geen gebruiker (Non-user)

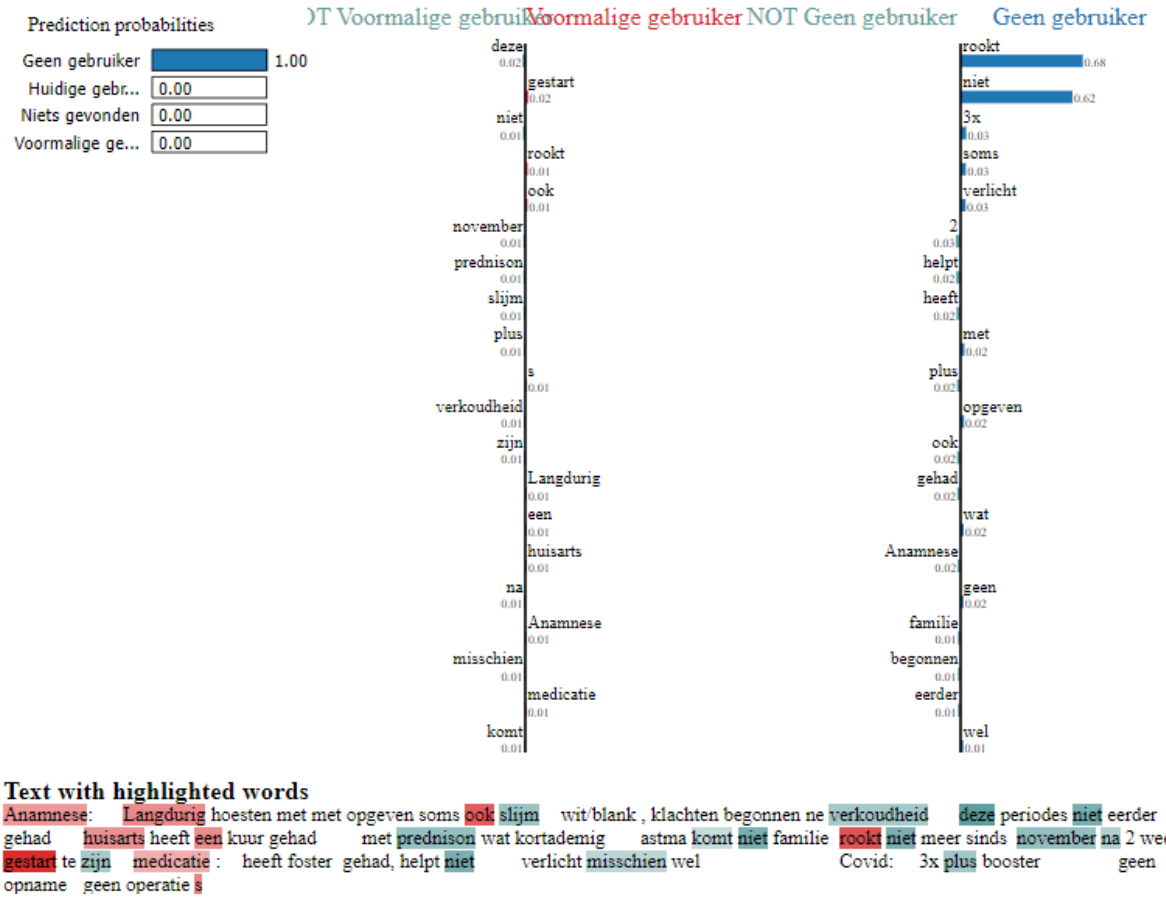


Figure 42: MedRoBERTa.nl LIME Visualization of Text 1 of Experiment 2

Here, MedRoBERTa.nl fails by not recognizing the sequence "rookt niet meer sinds..." (does not smoke anymore since...) as an indication of a previous user, it instead detects the sequence "rookt niet" (does not smoke) and classifies the patient wrongly in the non-user category. In defence of the model, this text in particular contains some unconventional and questionable spelling and grammar compared to other texts, which could have misguided the model as well.

B.2.2 Text 2

Translated Text
Course: Course: Still smokes, but a little less. Gets bruises very easily. Explanation given that these occur due to prednisone use on thin skin and also clopidogrel. Agreed that if quitting smoking fails to contact Sinefuma. The patient can either contact them or we can do it for them.

True label
Huidige gebruiker (Current user)

MedRoBERTa.nl
Prediction: Geen gebruiker (Non-user)

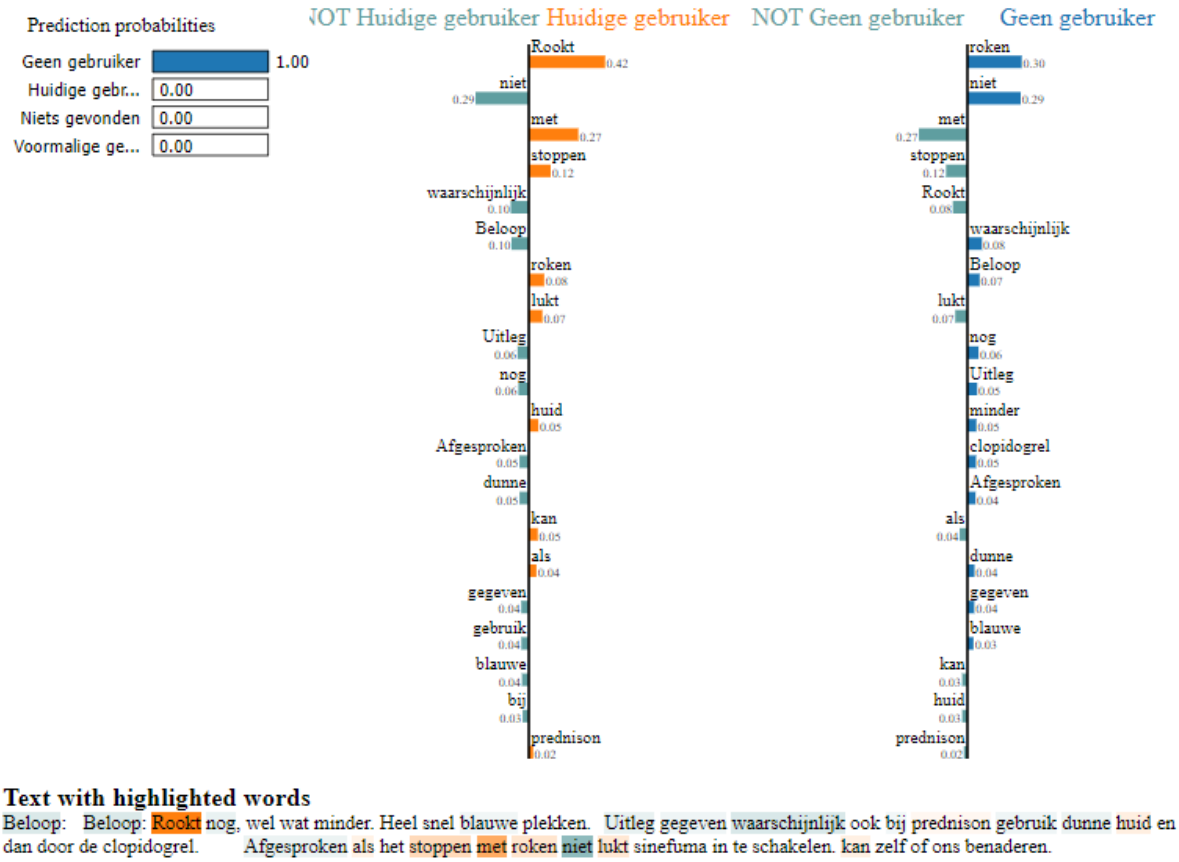


Figure 43: MedRoBERTa.nl LIME Visualization of Text 2 of Experiment 2

We can ascertain from this visualization that MedRoBERTa.nl misclassified this text by not recognizing the sequence "als het stoppen met roken niet lukt..." (if the patient is unable to stop smoking) as indicator for a current smoker. The model rather viewed the exempt "roken niet" from that subtext as indicator for a non-user, which shows in the scores given to the words. The words "stoppen met roken" (to stop smoking) were not viewed as significant enough to sway the model's judgement in another direction.

B.2.3 Text 3

Translated Text

Reason for coming / Referral: Reason for referral: Reason for coming: through MDL doctor Bhalla. Is a bit shaky, with hands especially, in the supine position almost counting cents. Shaking hands stiffly too. At gastroenterologist because of constipation, dd neurogenic cause in M. Parkinson or by chemoradiation? Case history: History: emphysema. cholecystectomy. 2011 CL. BINET B. 2012 (8) 6x FC-R, complete remission, hypogamma in CLL. 2013 (9) IVIG started due to pansinusitis. 2019 (1) neck, mediastium and abdominal lymphadenopathy, BINET C, no TP53, no del 17P, but unmutated IGHV. 2019 (4-3) Rituximab and start venetocla.x 2019 (1-4) start 400mg and Rituximab 2. 2019 (29-4) 3rd R. 2019 (27/5) 4th R. 2019 (22/7) 6th R. 2019 (12) tapering off IVIG to every 6 weeks. 2020 (4) IVIG on hold due to Corona. 2020 (7) start of zitromax due to low IgG. 2020 (9) complete remission CLL on CT. 2021 (1) analysis MDL due to abdominal pain, no explanation. 2021 (3) end venetoclax, wait and see CLL. 2021 (5) analysis gyn and pain team: blockade n iioingualis due to pain complaints. Medication: movicolone, azithromycin, amitriptyline.

True label
Niets gevonden (No information given)

MedRoBERTa.nl
Prediction: Geen gebruiker (Non-user)

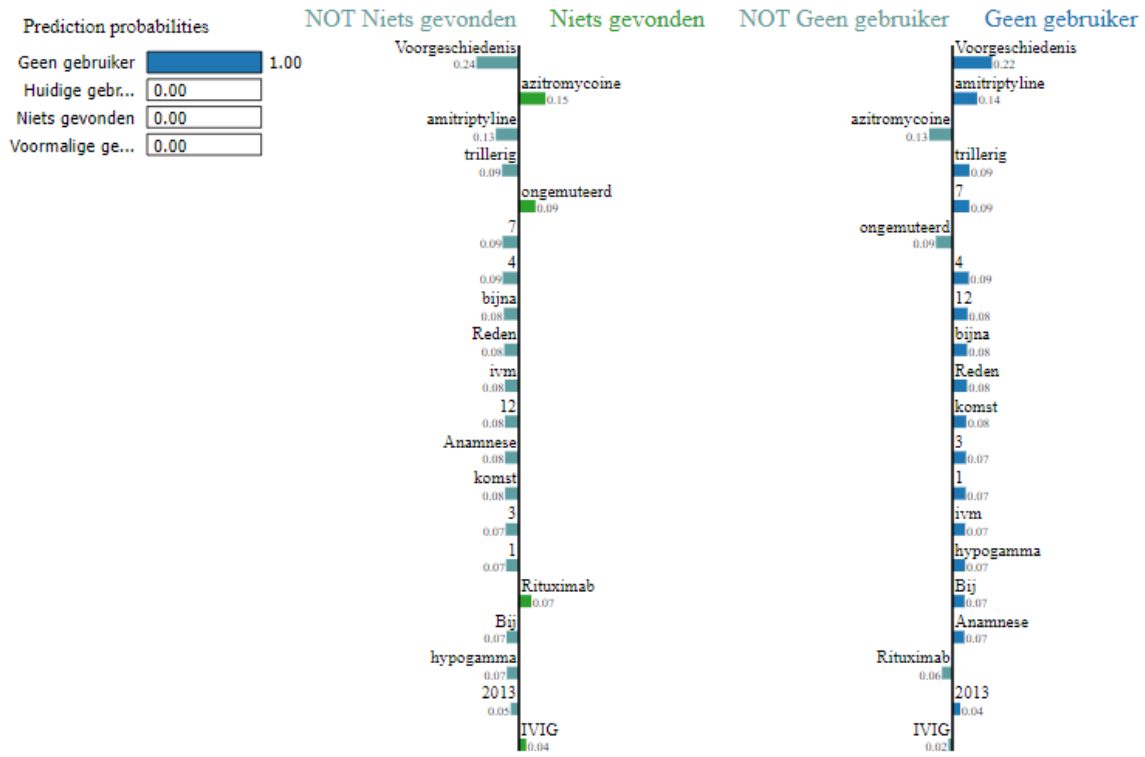


Figure 44: MedRoBERTa.nl LIME Visualization of Text 3 of Experiment 2

This text includes no statements related to smoking whatsoever. The model finds multiple indicators for non-users however, with the word "voorgeschiedenis" (previous history) as strongest indicator. We hypothesize that this word in particular is a strong indicator as we found that when a previous history is described in a clinical text it often includes the patient's smoking status, although this is obviously not the case here. Other indicators like "amitriptyline", an antidepressant, seem less obvious. It could be that other texts from the dataset include patients that use this medicine and that are non-smokers.

B.2.4 Text 4

Translated Text
Course: Course: TC after admission. Occasionally some stomach pain, but no longer so severe. No RBV anymore. Has not stopped smoking, mother very ill, is stressed, mother has incurable

lung carcinoma. Conclusion: Conclusion: 19-11 to 20-11 admission MDL due to ischemic colitis for which conservative management is in place. Policy: Policy: Again discussed quitting smoking for their own health. Instructed to make an appointment with HA for CRM + smoking cessation counseling EB. No letter.

True label
Huidige gebruiker (Current user)

MedRoBERTa.nl
Prediction: Voormalige gebruiker (Previous user)

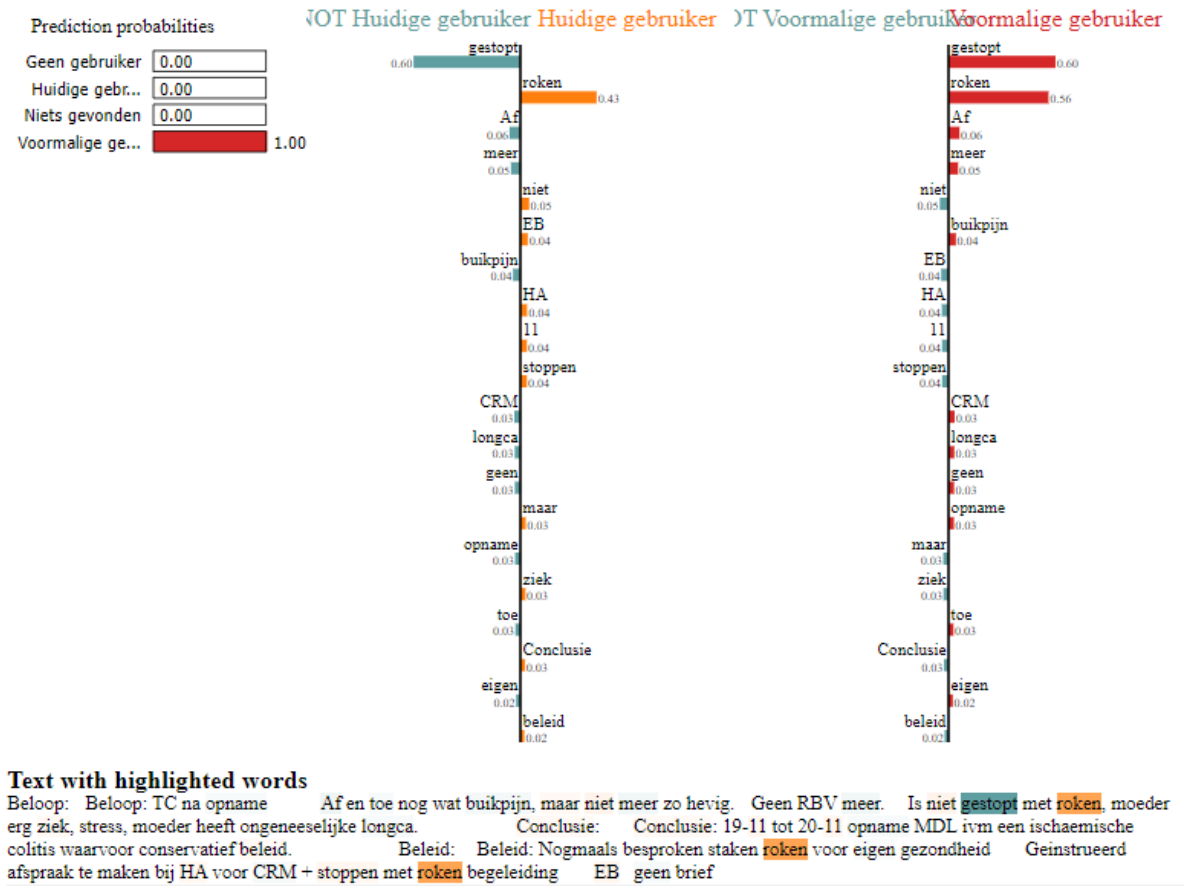


Figure 45: MedRoBERTa.nl LIME Visualization of Text 4 of Experiment 2

For this text, the model does not recognise the words "is niet" (has not) as negator for the sequence "gestopt with roken" (stopped smoking), hence it fails in classifying the text correctly.

B.2.5 Text 5

Translated Text
Intoxications: Smoking - Smoking: yes. Alcohol- Alcohol: no. Drugs- Drugs: no.

True label
Huidige gebruiker (Current user)

MedRoBERTa.nl
Prediction: Geen gebruiker (Non-user)

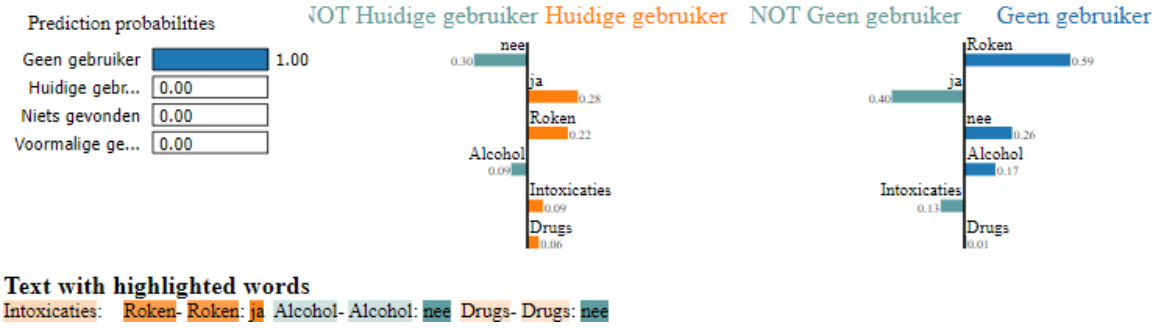


Figure 46: MedRoBERTa.nl LIME Visualization of Text 5 of Experiment 2

For this text, it seems the model views the minus sign as indicator that the patient is a non-user, but the minus sign is merely used to denote the smoking status is stated next. The presence of the word "nee" (no) in close proximity also seems to misguide the model in its classification of the text.

B.3 Experiment 3

5 texts misclassified by RobBERT and classified correctly by translated ClinicalBERT.

B.3.1 Text 1

True Label
Huidige gebruiker (Current user)

RobBERT
Prediction: Voormalige gebruiker (Previous user)

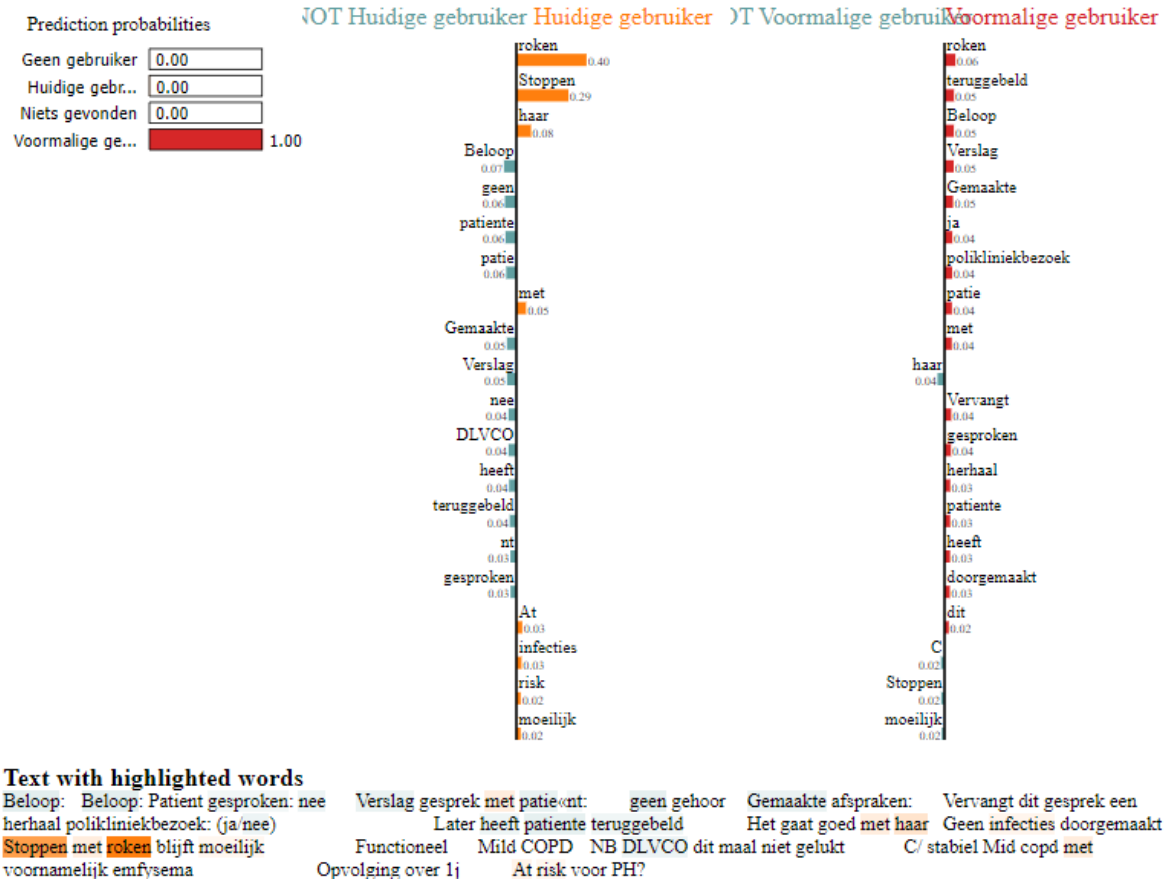


Figure 47: RobBERT LIME Visualization of Text 1 of Experiment 3

Interestingly, RobBERT seems to misclassify the text based on a predefined set of words that it links to previous users, as can be inferred from the word scores. Even though "stoppen met roken" (to stop smoking) is viewed as strong indicator for a current user, this sequence of words present in the text causes the model to classify the patient as a previous smoker.

ClinicalBERT

Prediction: Huidige gebruiker (Current user)

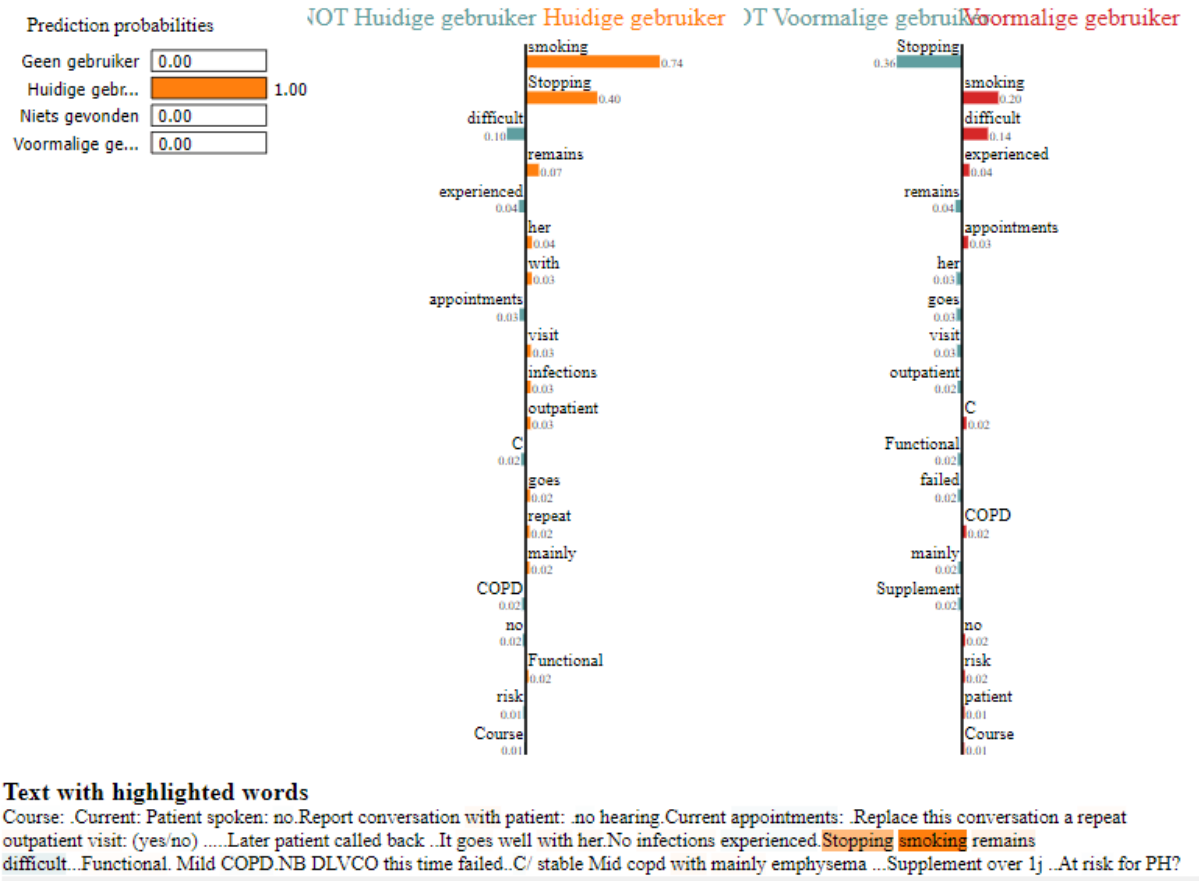


Figure 48: ClinicalBERT LIME Visualization of Text 1 of Experiment 3

ClinicalBERT is able to link the words "Stopping smoking" to previous users, correctly classifying the text.

B.3.2 Text 2

True Label
Niets gevonden (No information given)

RobBERT
Prediction: Huidige gebruiker (Current user)

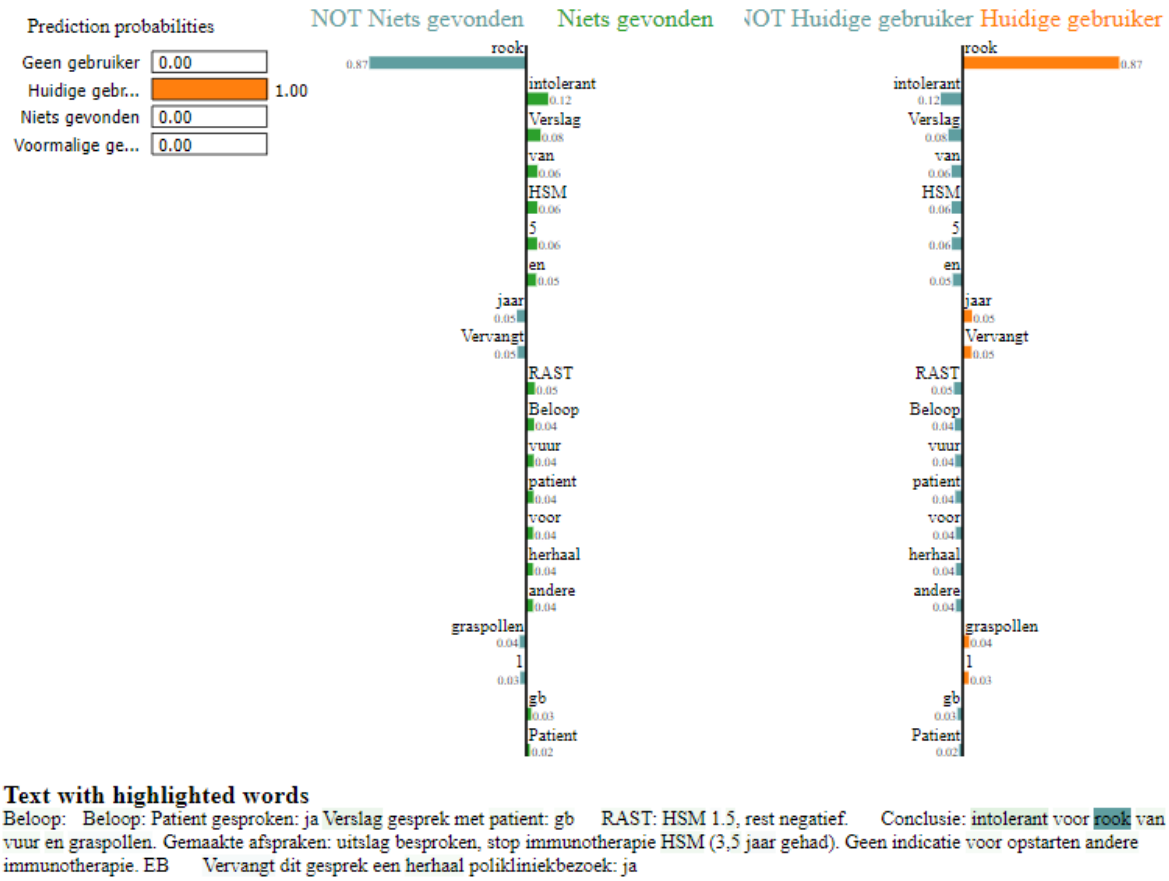


Figure 49: RobBERT LIME Visualization of Text 2 of Experiment 3

Here, RobBERT views the word "rook" (smoke) as indicator for a smoker, even though in the text the word is used to denote smoke that originates from fire, rather than anything related to the act of smoking.

ClinicalBERT

Prediction: Niets gevonden (No information given)

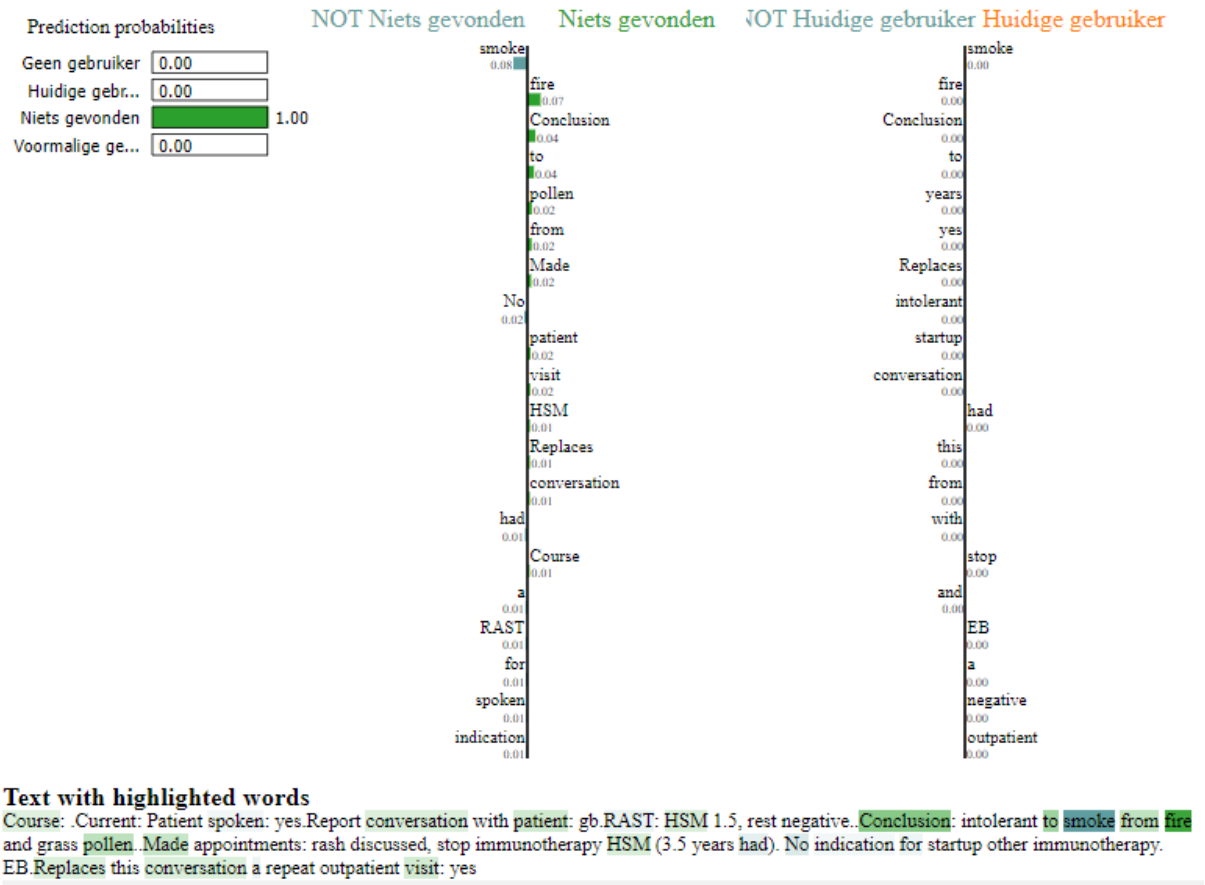


Figure 50: ClinicalBERT LIME Visualization of Text 2 of Experiment 3

ClinicalBERT is able to judge the words surrounding "smoke" correctly to not be about the patient's smoking status and therefore classifies the text correctly.

B.3.3 Text 3

True Label
Niets gevonden (No information given)

RobBERT
Prediction: Geen gebruiker (Non-user)

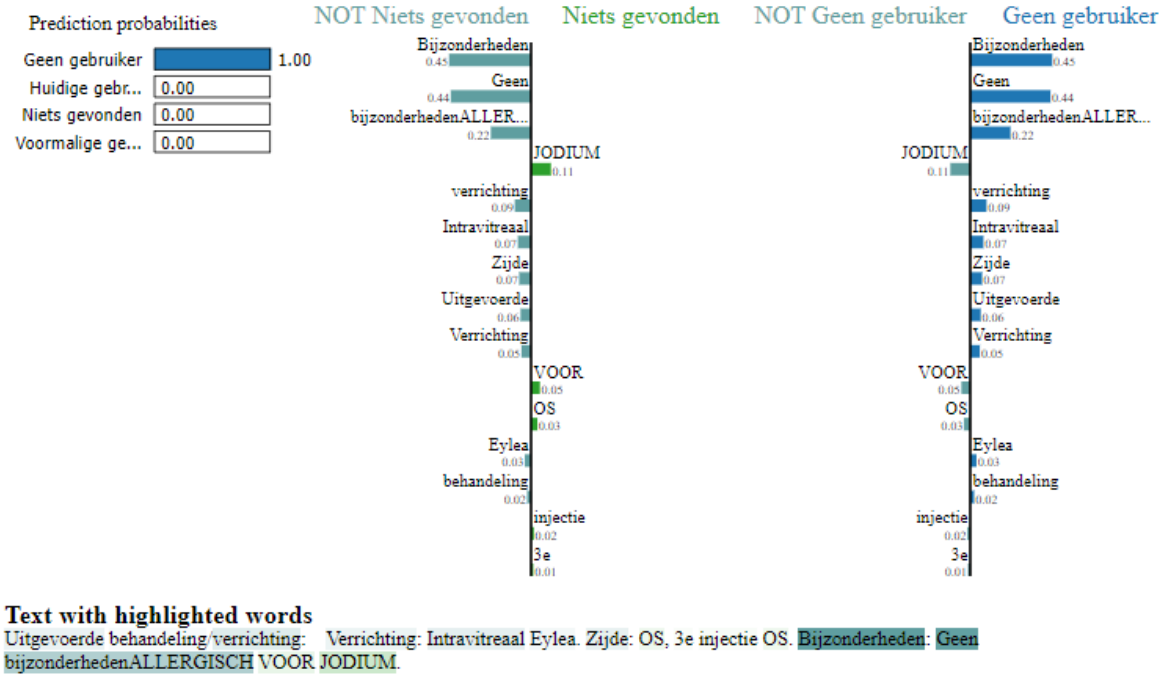


Figure 51: RobBERT LIME Visualization of Text 3 of Experiment 3

Here, RobBERT views the sequence "Bijzonderheden: Geen bijzonderheden" (Particularities: No particularities) as strong indicator for the non-user class. It could be that this particular sequence shows up in other texts that do concern non-smokers, similarly to the sequence that is viewed as indicator by RobBERT in text 1 of this experiment.

ClinicalBERT

Prediction: Niets gevonden (No information given)

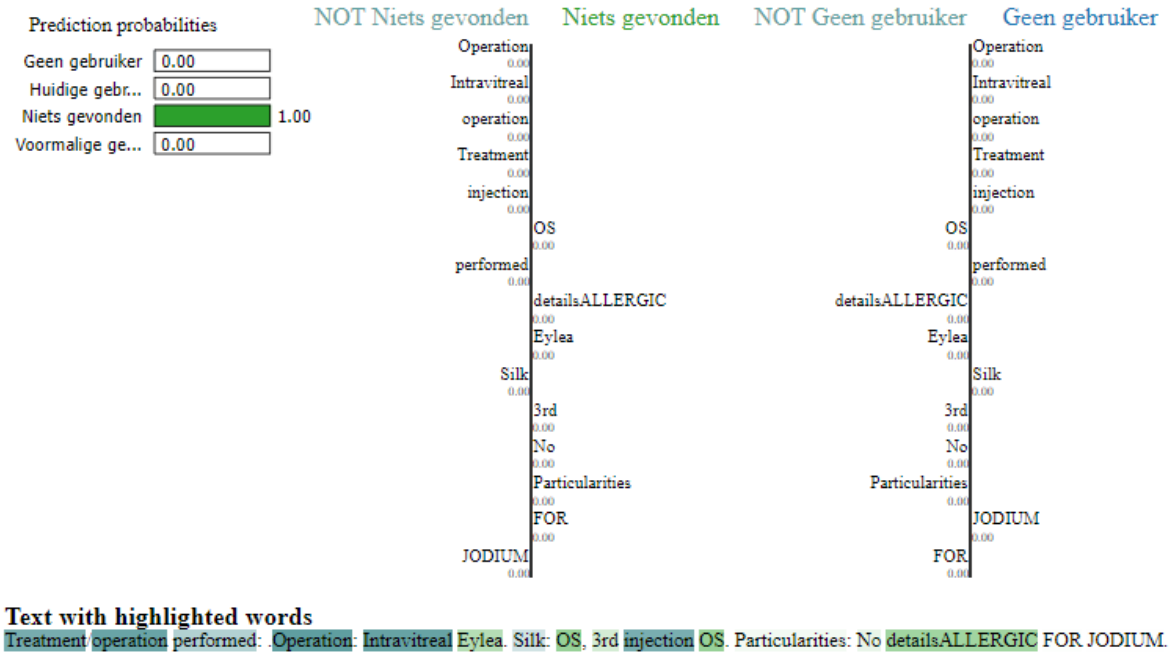


Figure 52: ClinicalBERT LIME Visualization of Text 3 of Experiment 3

ClinicalBERT finds a lot of indicators for the non-user class but assigns them such low impor-

tance that the correct no information given class is predicted.

B.3.4 Text 4

True Label
Voormalige gebruiker (Previous user)

RobBERT
Prediction: Huidige gebruiker (Current user)

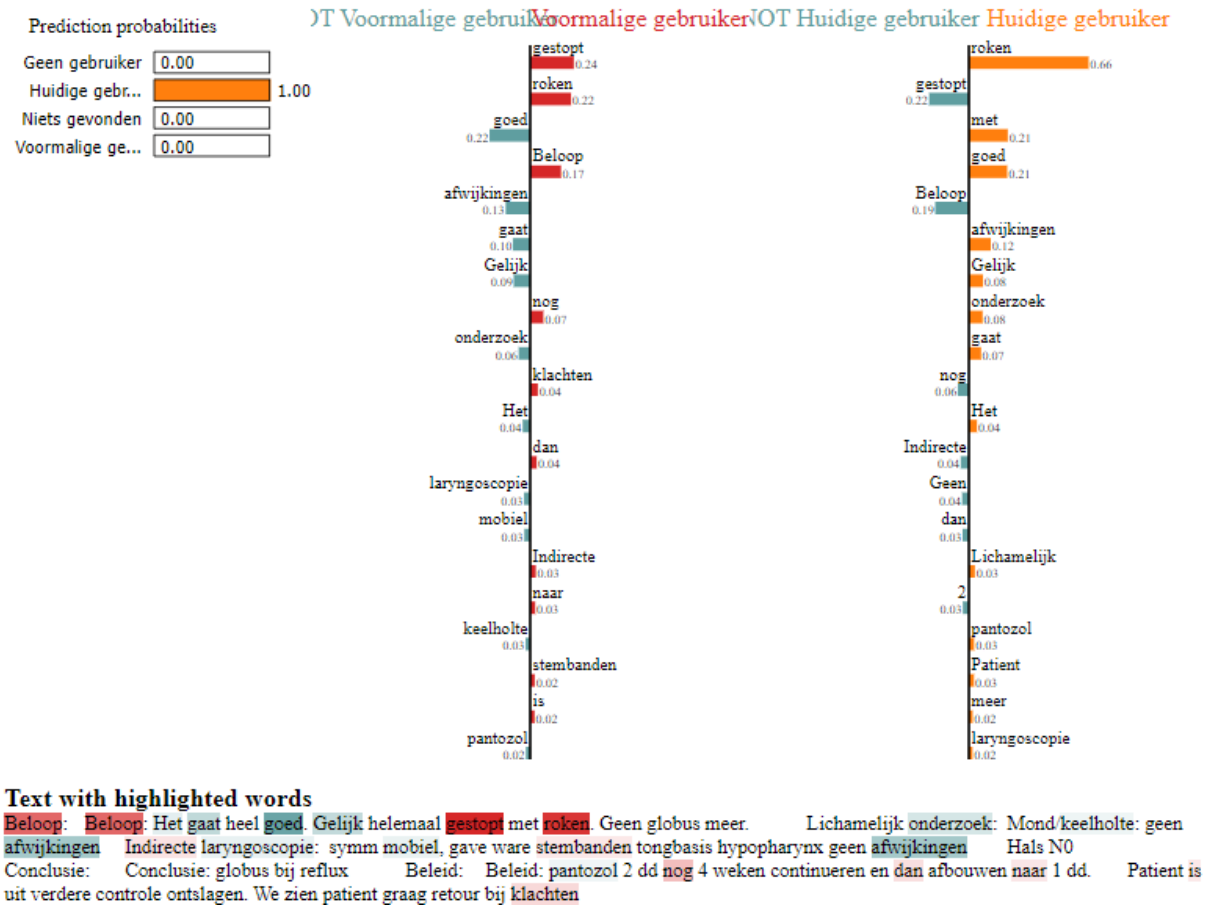


Figure 53: RobBERT LIME Visualization of Text 4 of Experiment 3

Interestingly, RobBERT is unable to classify the text on the basis of the words "gestopt met roken" (stopped smoking), rather it recognizes these words as indicators for previous users but sees more indicators for the current user class.

ClinicalBERT
Prediction: Voormalige gebruiker (Previous user)

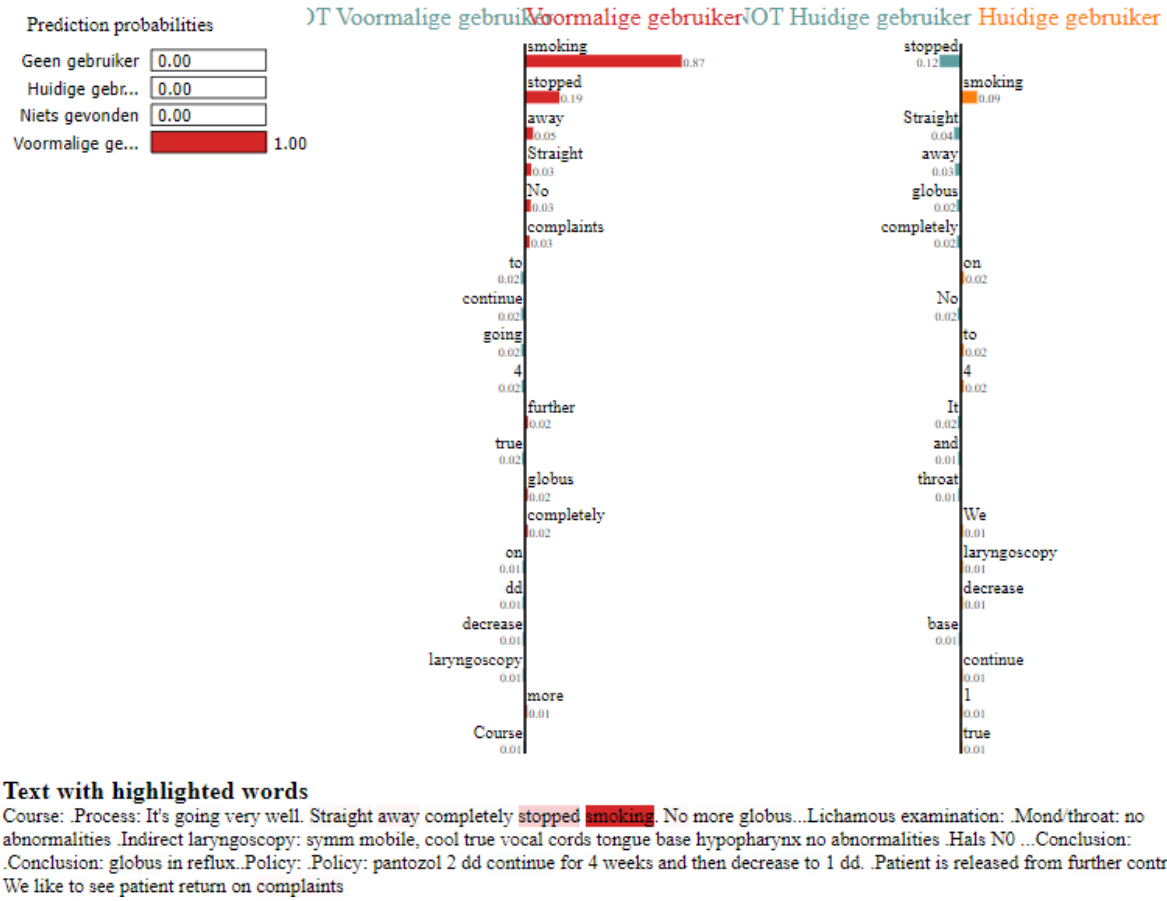


Figure 54: ClinicalBERT LIME Visualization of Text 4 of Experiment 3

Contrary to RobBERT, ClinicalBERT does find the correct indicators for a previous smoker and classifies the text as such.

B.3.5 Text 5

True Label
Huidige gebruiker (Current user)

RobBERT
Prediction: Geen gebruiker (Non-user)

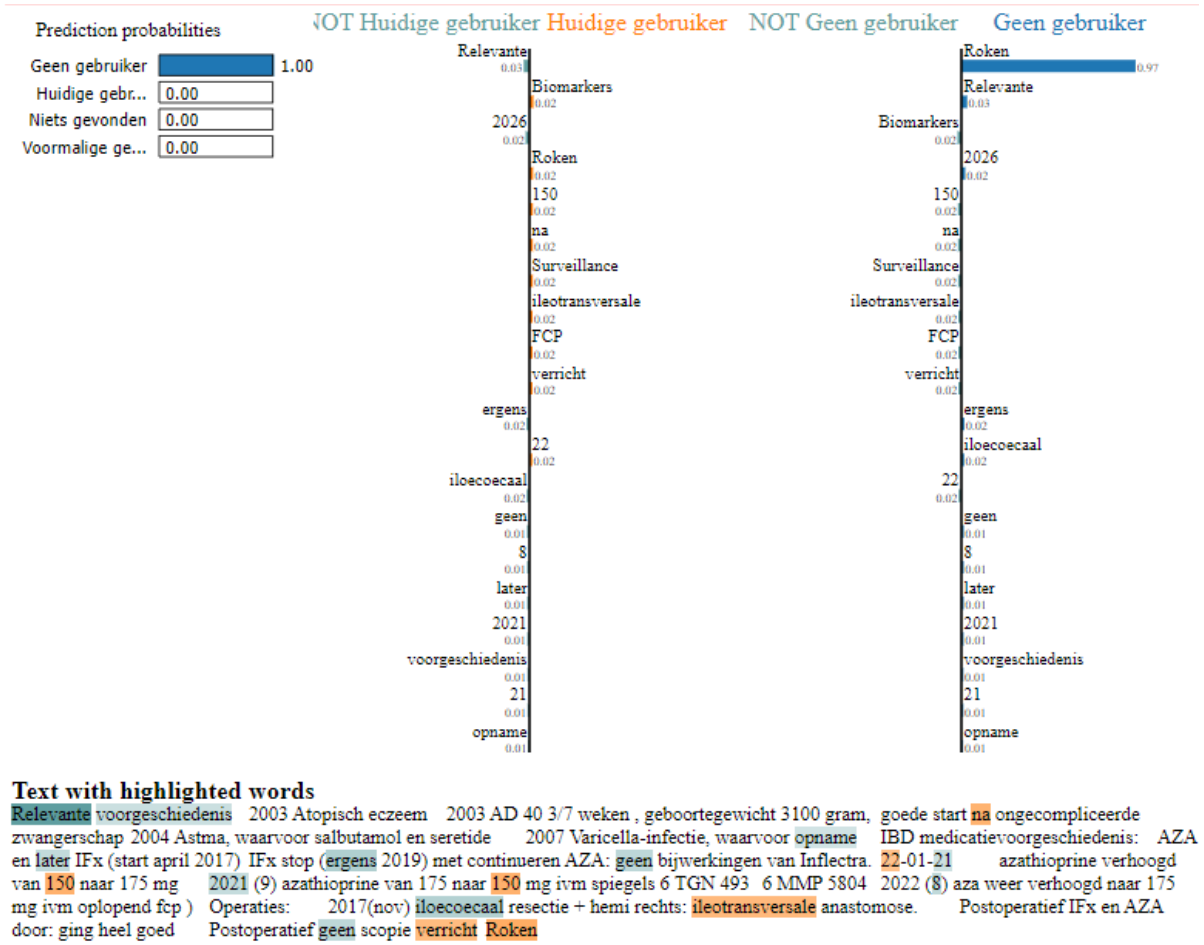


Figure 55: RobBERT LIME Visualization of Text 5 of Experiment 3

Here, the classification is almost solely based on the presence of the word "Roken" (Smoking) in the text. From a human perspective, the word does not seem to have any surrounding words that denote context, nor are there any other words in the text that denote smoking status. Our annotators viewed the patient as a current user, while RobBERT views the sole presence of this word as indicator for a non-user.

ClinicalBERT

Prediction: Huidige gebruiker (Current user)

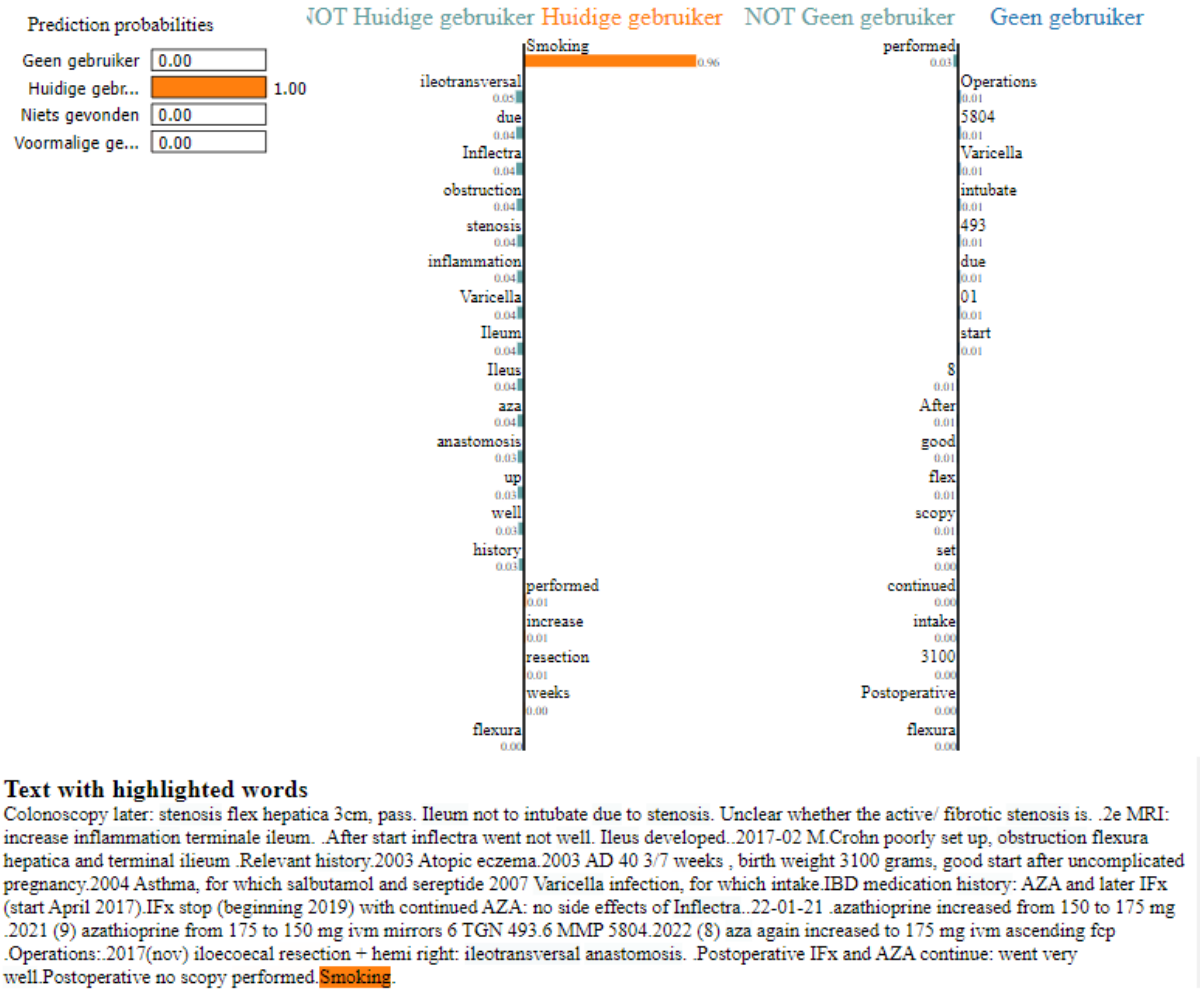


Figure 56: ClinicalBERT LIME Visualization of Text 5 of Experiment 3

ClinicalBERT does view the word "Smoking" as indicator for a current user, correctly classifying the text.

B.4 Experiment 4

5 texts misclassified by CTCue string matching and classified correctly by MedRoBERTa.nl.

B.4.1 Text 1

Translated text
Course: Course: regular check-up after acdf c4-5 and hnp l4-5 in emergency setting dated 12-8-2022. It's going very well, no more complaints. Very satisfied. Physical examination: wounds healed nicely. Consideration / differential diagnosis: Consideration / differential diagnosis: excellent recovery from surgery. Really emphasizes to quit smoking for prevention of future problems. No further restrictions. No follow-up appointments made.

True Label
Huidige gebruiker (Current user)

String matching

Prediction: Niets gevonden (No information given)

MedRoBERTa.nl
Prediction: Huidige gebruiker (Current user)

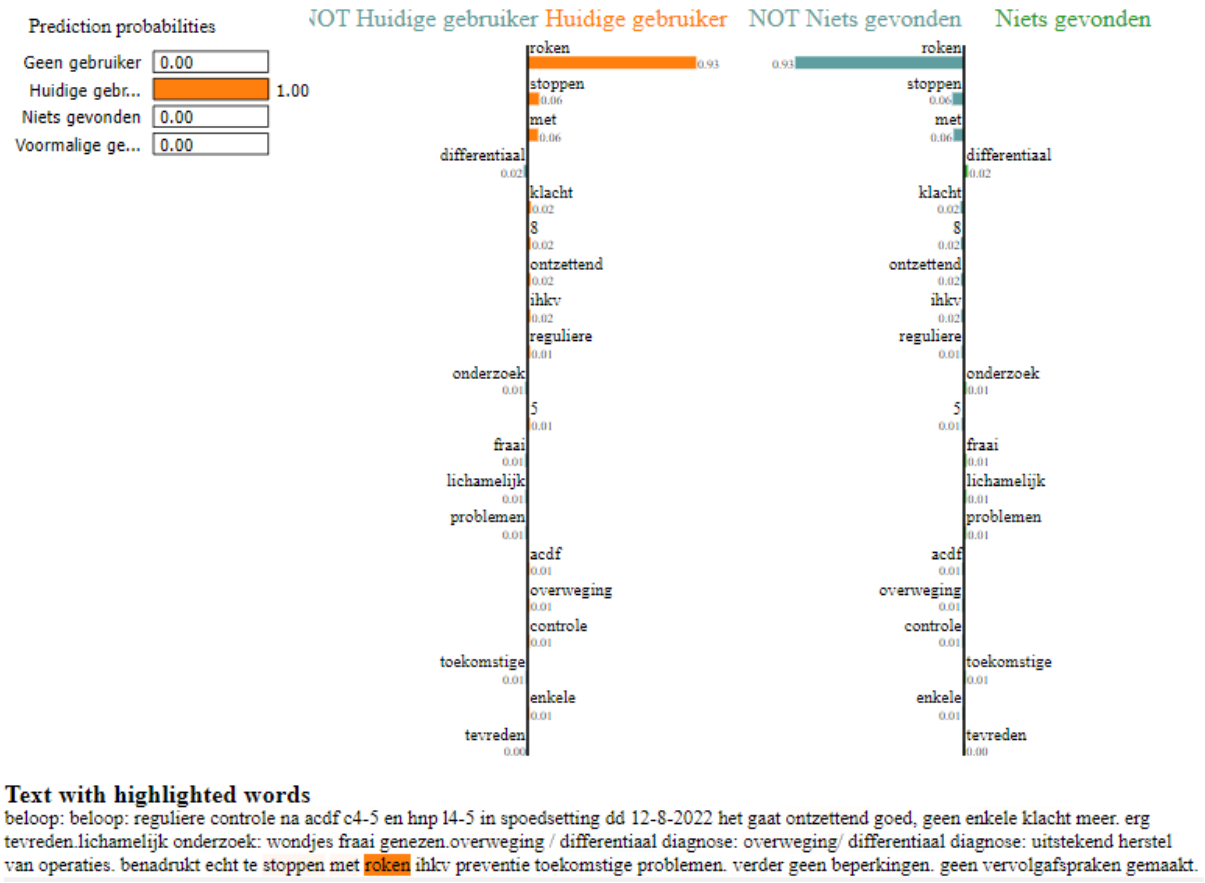


Figure 57: MedRoBERTa.nl LIME Visualization of Text 1 of Experiment 4

Our string matching queries do not include any case for linking "stoppen met roken" (to stop smoking) with any label, meaning the rest class is picked. MedRoBERTa.nl did learn this link and is thus able to classify this text correctly.

B.4.2 Text 2

Translated text
Anamnesis and special examination. [intoxications][smoking]-smoking: never[alcohol]-alcohol: yes-type of alcohol: beer-amount of alcohol: between 2 and 6 units per day[drugs]-drugs: no[preoperative][patient identifiers] age at surgery: 53.

True Label
Geen gebruiker (Non-user)

String matching
Prediction: Niets gevonden (No information given)

MedRoBERTa.nl
Prediction: Geen gebruiker (Non-user)

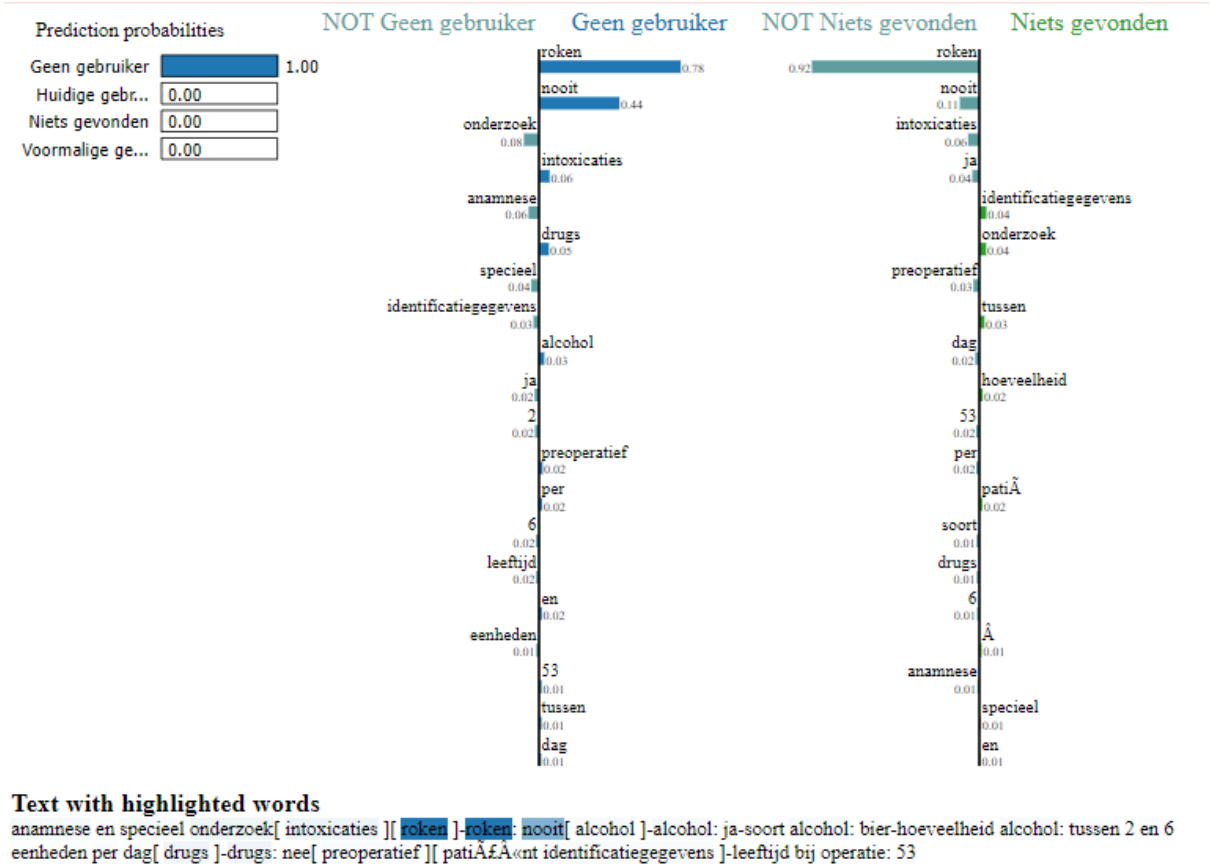


Figure 58: MedRoBERTa.nl LIME Visualization of Text 2 of Experiment 4

Our string matching queries do not include any case for linking "roken: nooit" (smoking: never) with any label, meaning the rest class is picked. MedRoBERTa.nl did learn this link and is thus able to classify this text correctly. Addition of "roken: nooit" to the queries would be a minimal effort and would result in a correct string matching classification.

B.4.3 Text 3

Translated text
Social: lives with daughter after hip fracture, walks with walker. Family history: no dvt or le in family (as far as he knows). Intoxications: smoking - (stopped 15 years ago, 20 yrs), alcohol -, drugs -

True Label
Voormalig gebruiker (Previous user)

String matching
Prediction: Geen gebruiker (Non-user)

MedRoBERTa.nl
Prediction: Voormalig gebruiker (Previous user)

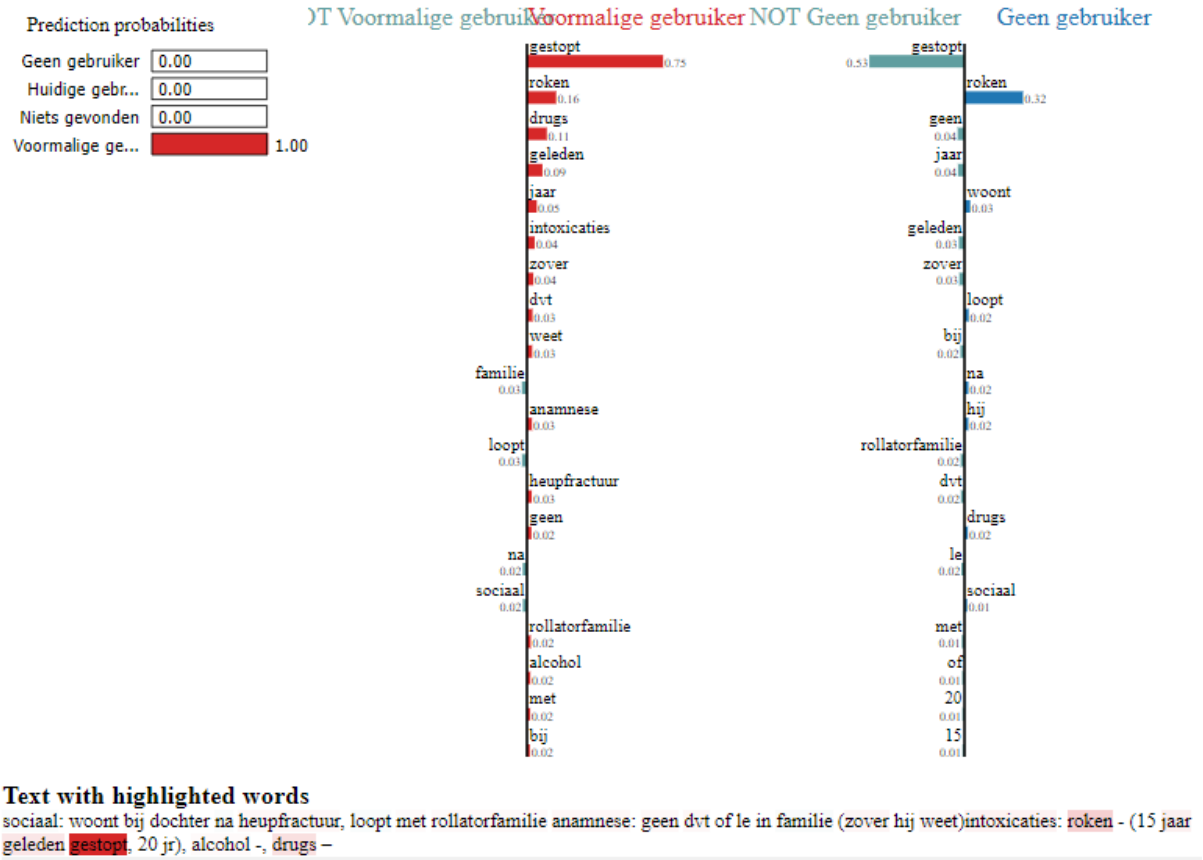


Figure 59: MedRoBERTa.nl LIME Visualization of Text 3 of Experiment 4

Our string matching queries match the sequence "roken -" (smoking -) to the non-user class, even though the text in parentheses denotes that the patient has in fact stopped smoking and should therefore be classified as a previous user. MedRoBERTa.nl does recognise this, showing it understands the context of the text.

B.4.4 Text 4

Translated text
Conclusion: c/ consultation less life, good fetal condition here. b/- return 1st line patient and partner would like to plan an introduction, especially in view of wider growth and wish not to pass ad 41 wks, scheduled for next Monday 08:00, hopefully for amniotomy, possibly for miso/foley, this also with discussed the set - given the Christmas rush, explicitly discussed that there is a chance that the introduction can be postponed if there are patients with more urgency / indication - clear call instructions - feedback to vk tara by telephone.

True Label
Niets gevonden (No information given)

String matching
Prediction: Geen gebruiker (Non-user)

MedRoBERTa.nl
Prediction: Niets gevonden (No information given)

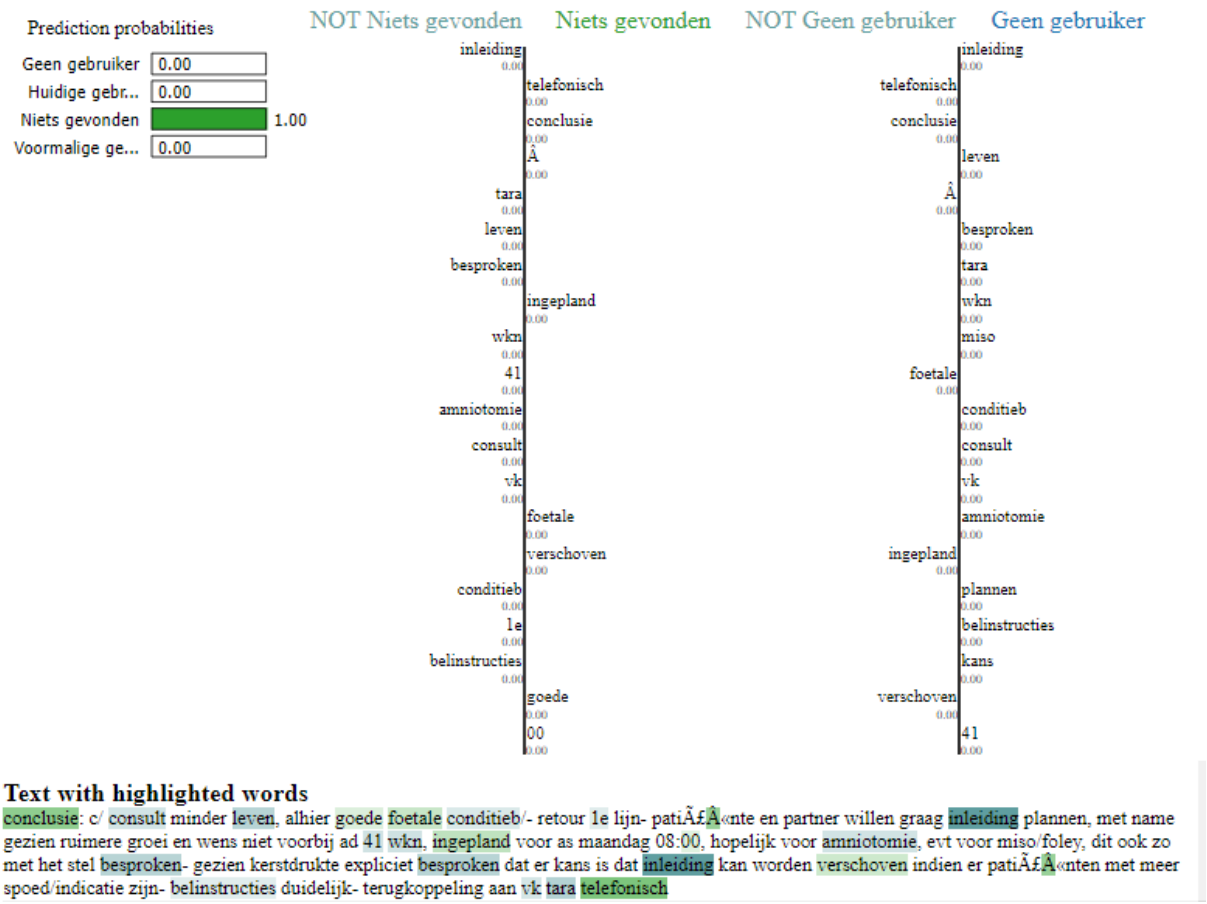


Figure 60: MedRoBERTa.nl LIME Visualization of Text 4 of Experiment 4

Our string matching queries match the sequence "roken -" (smoking -) to the non-user class, even though the sequence is part of the larger word "besproken" (discussed) and the minus sign is used as a delimiter, meaning there is nothing stated here about smoking status at all. As there is nothing in the text that denotes smoking status MedRoBERTa.nl correctly classifies the text as no information given.

B.4.5 Text 5

Translated text
Summary: history: gbmedication: -allergy: -smoking: 1-2 sig/dgpolicy: policy: nna control 1 year after pl implant.

True Label
Huidige gebruiker (Current user)

String matching
Prediction: Niets gevonden (No information given)

MedRoBERTa.nl
Prediction: Huidige gebruiker (Current user)

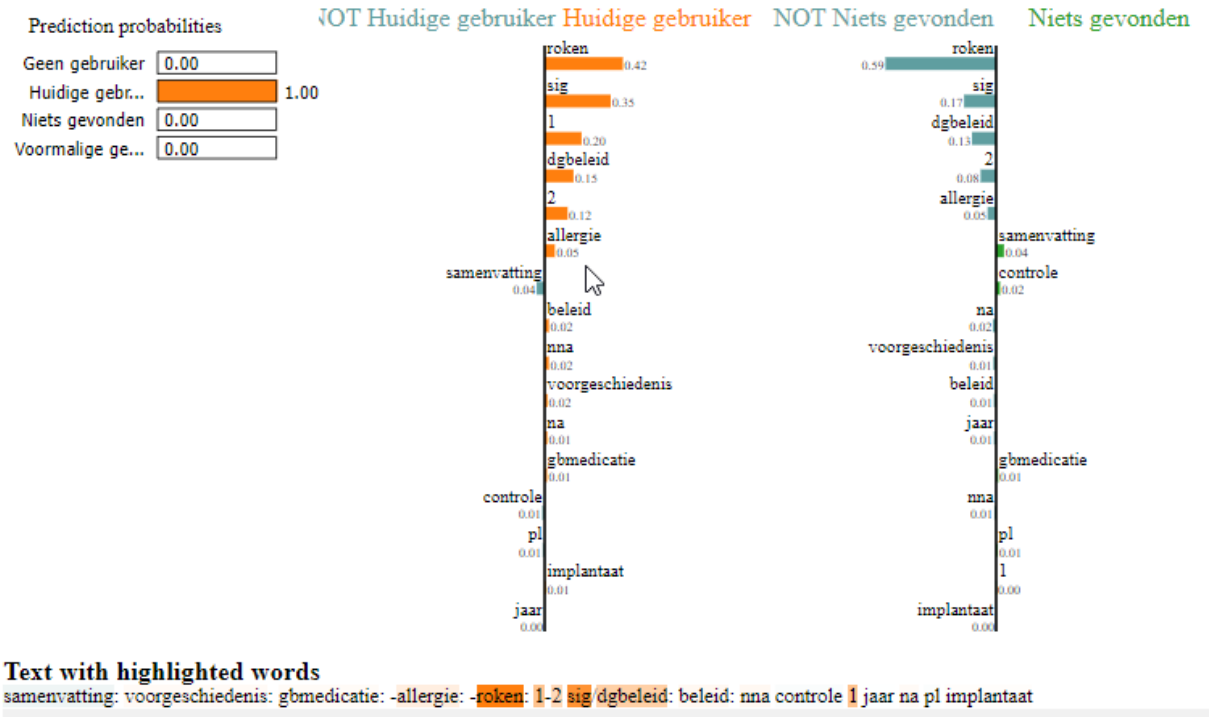


Figure 61: MedRoBERTa.nl LIME Visualization of Text 5 of Experiment 4

We did not include anything resembling "roken: 1-2 sig" (smoking: 1-2 cigarettes) in our queries, meaning the "no information given" class is assigned. MedRoBERTa.nl is able to grasp the context of this statement and is therefore able to classify the text correctly.

B.5 Experiment 5

5 texts misclassified by Stochastic Gradient Descent and classified correctly by MedRoBERTa.nl. Note that the LIME classifier could not be used on the *sklearn* library we used for our Stochastic Gradient Descent approach. We instead opted to use the *eli5* library, which has very similar workings as it shows the importances for each feature relative to the given classes. Within these visualizations, the text is stated under each class' evaluation. When a word is labeled red this means that the word contributes negatively to the model while making an evaluation for the respective class. Similarly, green text means the word contributes positively.

B.5.1 Text 1

Translated text
Course: course: patient spoken: no. Report conversation with patient: no response. Appointments made: . Does this conversation replace a repeat outpatient visit: (yes/no): . The patient later called back. She is doing well, no infections, giving up smoking remains difficult. Functional. Mild copd nb dlvco not successful this time, c/ stable mid copd with mainly emphysema. Follow-up in 1y . At risk for ph?

True Label
Huidige gebruiker (Current user)

Stochastic Gradient Descent
Prediction: Niets gevonden (No information given)

Figure 62: Stochastic Gradient Descent LIME Visualization of Text 1 of Experiment 5

Here, even though the terms "stoppen met roken" (to stop smoking) contribute to the correct class, other terms negate this effect. These terms are not related to smoking and seem to be linked to other texts where no smoking status was found.

MedRoBERTa.nl

Prediction: Huidige gebruiker (Current user)

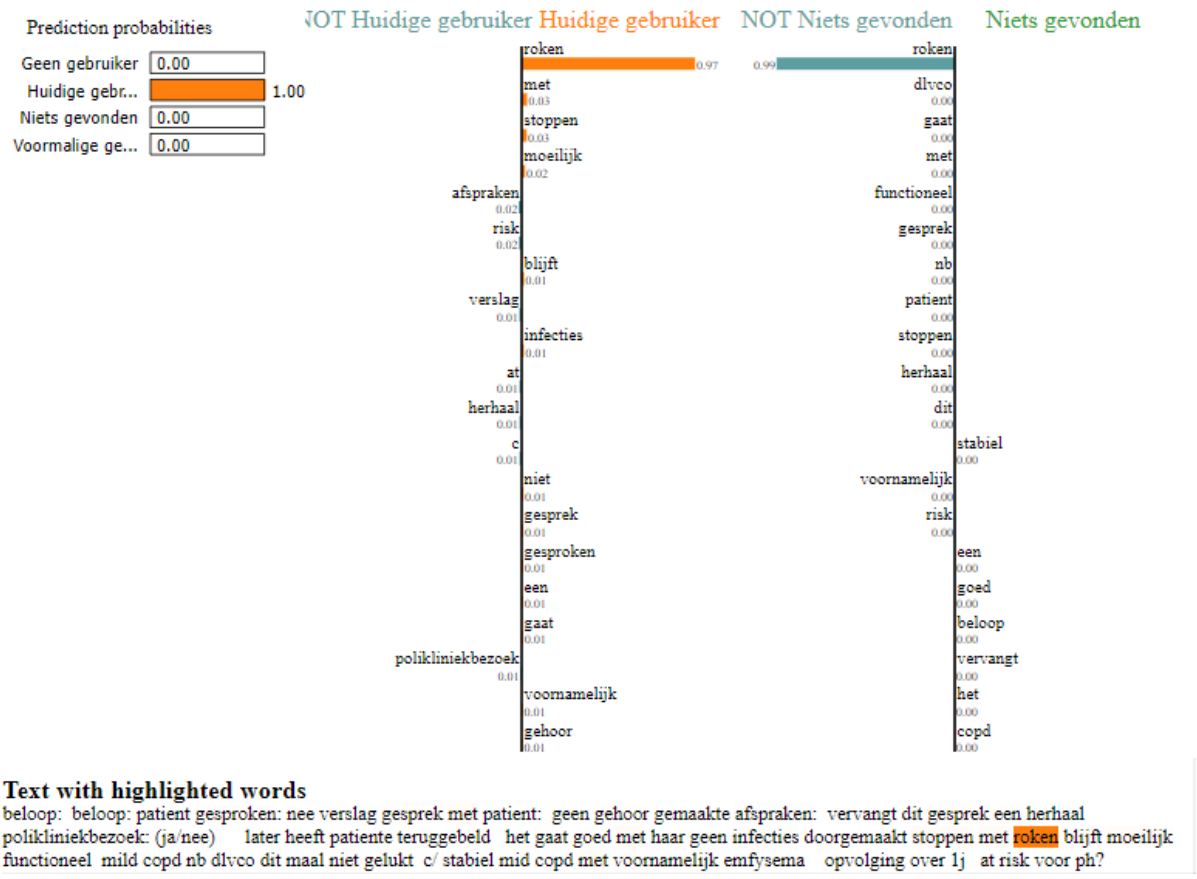


Figure 63: MedRoBERTa.nl LIME Visualization of Text 1 of Experiment 5

Here, the mere presence of the word "roken" (smoking) cause the model to completely disregard the "no information found" class. The words that were highlighted for Stochastic Gradient Descent did not have effect on MedRoBERTa.nl's evaluation at all.

B.5.2 Text 2

Translated text
Course: course: patient spoken: no, no response. Report conversation with patient: . Agreements: . Does this conversation replace a repeat outpatient visit: (yes/no): . Indication after CT: see there. Picture of SAD, in my opinion, not picture of asthma: but smoking related, after all, has little effect on medication and rather no abnormal histamine test. Reschedule? Try again. Called again at 3:56 pm, and was reached. Explained situation. Primarily to stop smoking.

True Label
Huidige gebruiker (Current user)

Stochastic Gradient Descent
Prediction: Niets gevonden (No information given)

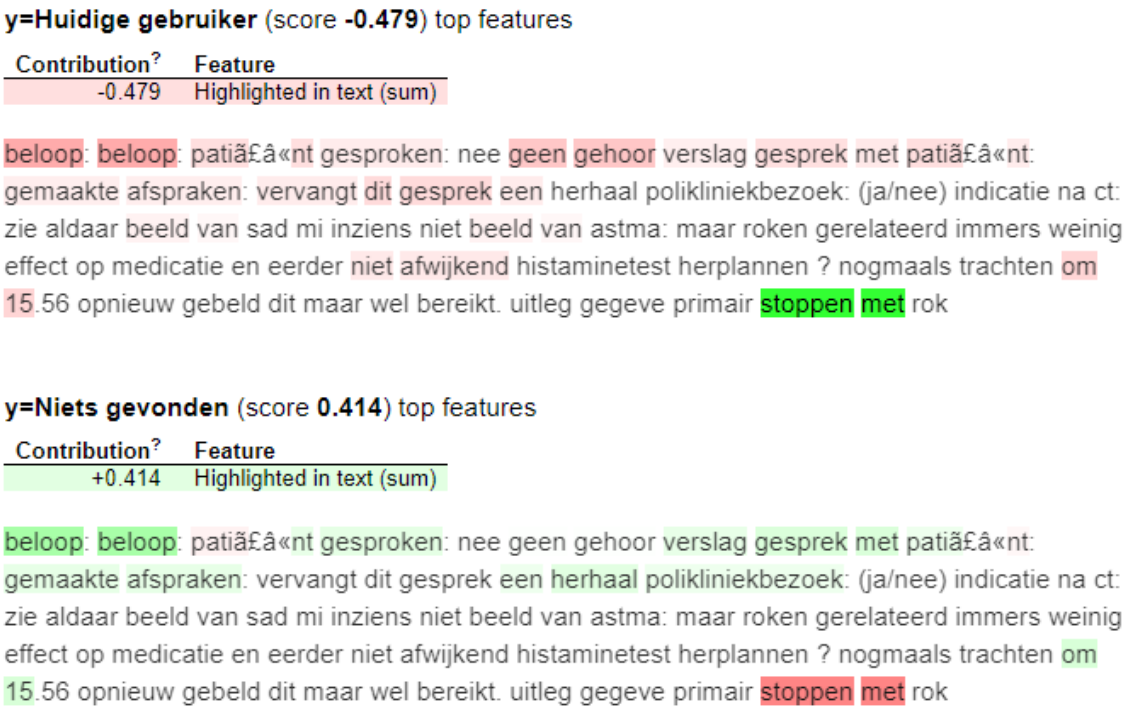


Figure 64: Stochastic Gradient Descent LIME Visualization of Text 2 of Experiment 5

Same as with the first text, the "stoppen met" (to stop with) is not considered enough for a current user evaluation, due to the amount of insignificant words that are contributing to a no information given class.

MedRoBERTa.nl
Prediction: Huidige gebruiker (Current user)

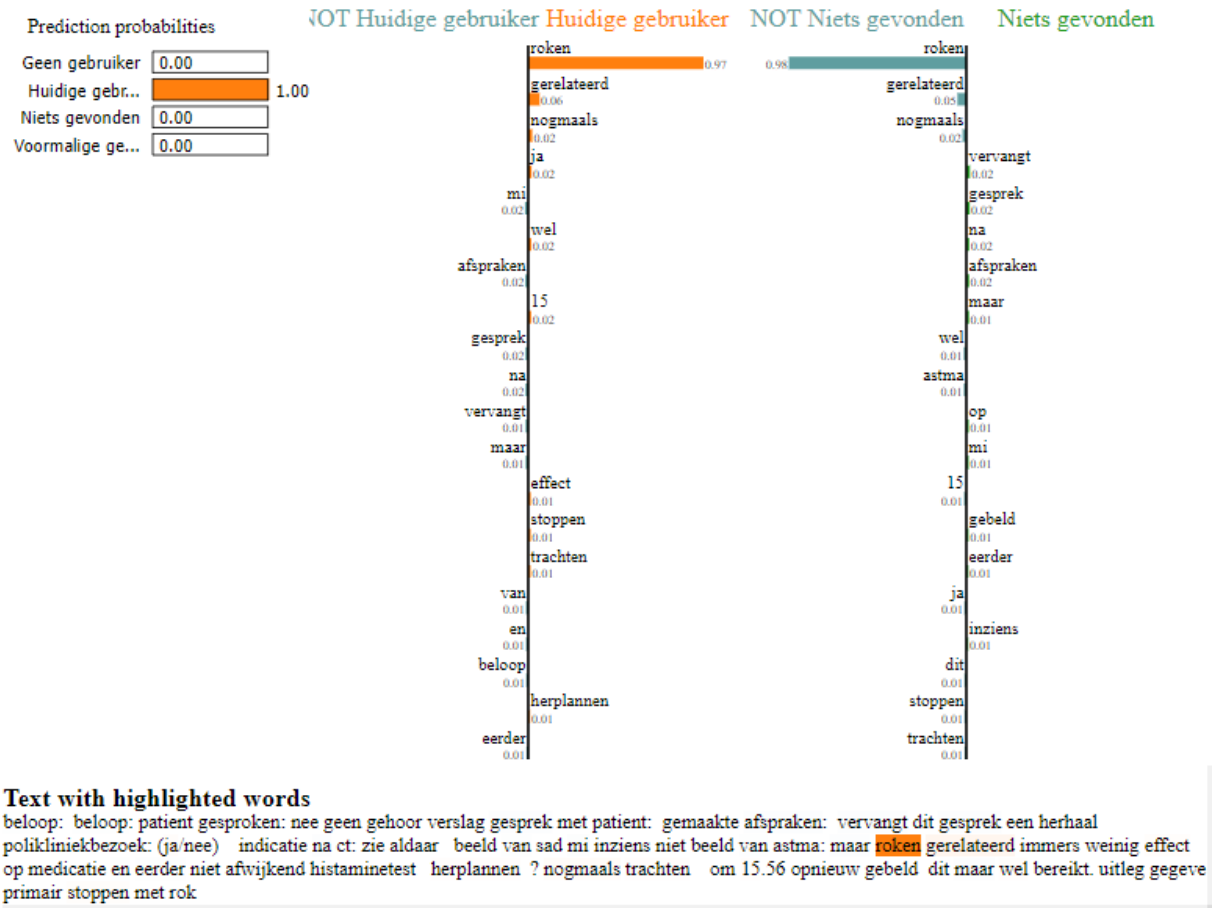


Figure 65: MedRoBERTa.nl LIME Visualization of Text 2 of Experiment 5

Again, the mere presence of the word "roken" (smoking) cause the model to disregard the "no information found" class and the highlighted words from SGD do not matter here.

B.5.3 Text 3

Translated text
Anamnesis: complaints: madam works in healthcare, squints when reading medication, eye turns inward, has tried contact lenses but this does not improve the situation, also gets a headache. Went to another optician today for other contact lenses, without improvement. On the advice of an optician, an appointment with an ophthalmologist was made. Madam is going to ask for a referral from ha. Am

True Label
Niets gevonden (No information given)

Stochastic Gradient Descent
Prediction: Geen gebruiker (Non-user)

Figure 66: Stochastic Gradient Descent LIME Visualization of Text 3 of Experiment 5

The SGD classifier sees multiple words that lead it to classify the patient as a non-user, even though none of these words have anything to do with the act of smoking.

MedRoBERTa.nl
Prediction: Huidige gebruiker (Current user)

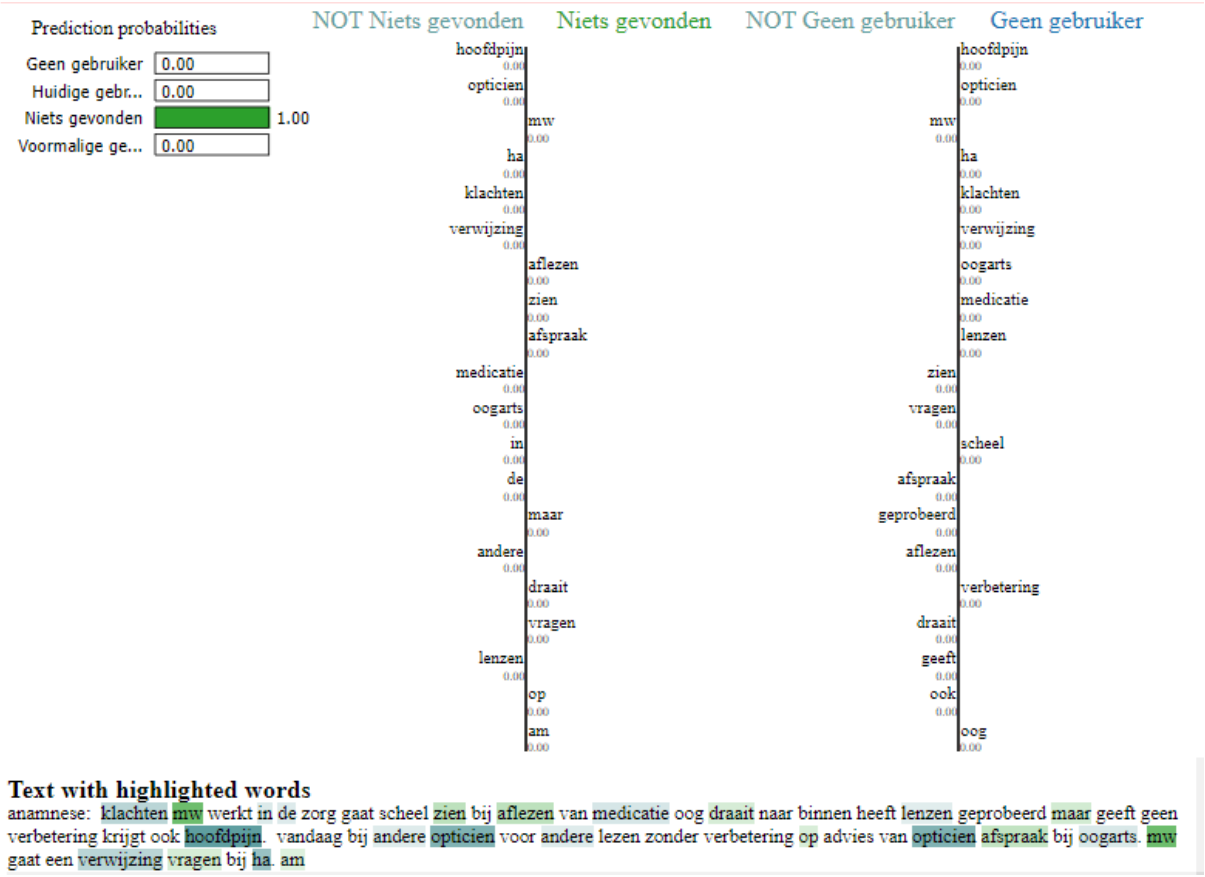


Figure 67: MedRoBERTa.nl LIME Visualization of Text 3 of Experiment 5

Even though there are a lot of indicators for a non-user, the indicators are very weak, leading the model to classify the text correctly.

B.5.4 Text 4

Translated text
Course: Previous appointment canceled follow-up on diab su. Scheduled again for tomorrow. In my opinion not necessary. Partus uncomplicated. HbA1c 40 (14/01). Patient called, no questions, no complaints, good curves. If it is not necessary to come, it also saves her a trip to the hospital. Advice: HbA1c. Annual check-up at the GP + early OGTT in the next pregnancy. Patient adequately informed. Appointment canceled for tomorrow.

True Label
Niets gevonden (No information given)

Stochastic Gradient Descent
Prediction: Geen gebruiker (Non-user)

Figure 68: Stochastic Gradient Descent LIME Visualization of Text 4 of Experiment 5

The SGD classifier sees multiple words that lead it to classify the patient as a current user, even though none of these words have anything to do with the act of smoking.

MedRoBERTa.nl
Prediction: Huidige gebruiker (Current user)

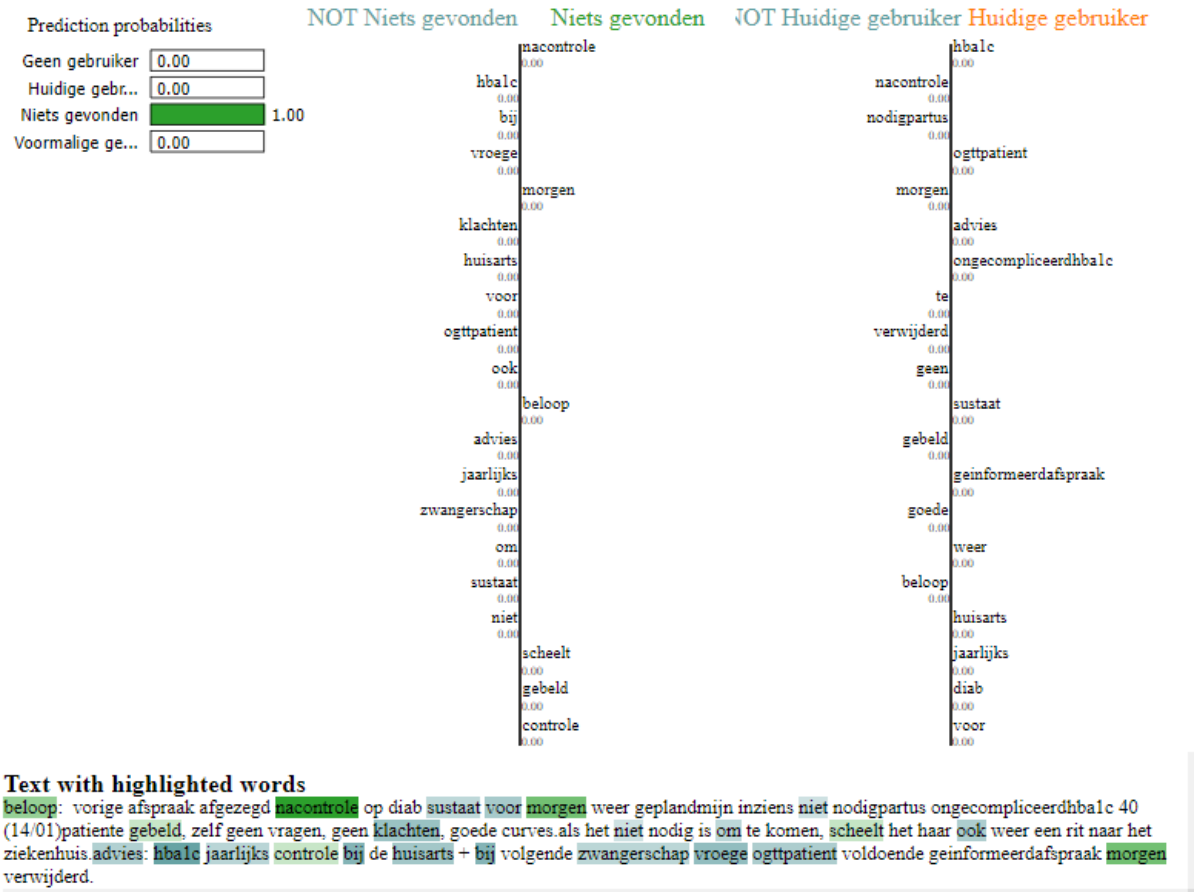


Figure 69: MedRoBERTa.nl LIME Visualization of Text 4 of Experiment 5

MedRoBERTa.nl correctly finds no indicators for a current user.

B.5.5 Text 5

Translated text
Reason for coming / referral: Reason for referral: eyelids. Case history: blank history. Medication: none. Case history: since 0.5 years progressive headache and fatigue after onset depend on the left upper eyelid. Can also see a little less, especially when driving a car, has to lift the eyebrows completely. Can't read well at night because of fatigue. Doesn't notice any difference between left and right. Intoxications: stopped smoking 20 years ago, alcohol -, drugs -. Allergies: none known. Physical examination: eyebrows both at the level of the orbital rim. Dermatochalasis right mild, left up to the eyelashes. Levator function both sides intact, good function. No ptosis. Conclusion: Conclusion: mild dermatochalasis. Policy: Policy: unfortunately, the severity of the dematochalasis does not meet the conditions that the health insurer sets for reimbursement for a blepharoplasty. The patient was informed of the possibilities of self-payment. The patient will think about this.

True Label
Voormalig gebruiker (Previous user)

Stochastic Gradient Descent
Prediction: Geen gebruiker (Non-user)

y=Geen gebruiker (score **0.105**) top features

Contribution[?]	Feature
+0.105	Highlighted in text (sum)

reden van komst / verwijzing: reden verwijzing: oogleden anamnese: voorgeschiedenis blanco medicatie geen anamnese sinds 0,5jr progressieve hoofdpijn en vermoeidheid na ontstaan afhangen bovenooglid li. kan hierbij ook iets minder zien, moet mn bij autorijden de wenkbrauwen helemaal optillen. kan 's avonds niet meer goed lezen vanwege de vermoeidheid. merkt zelf geen verschil tussen li en re. intoxicaties roken 20jr terug gestopt, alcohol-, drugs- allergieen geen bekend lichamelijk onderzoek: wenkbrauwen bdz op niveau van de orbitarand. dermatochalasis re mild, li tot op de wimpers. levatorfunctie bdz intact, goede functie. geen ptosis. conclusie: conclusie: milde dermatochalasis. beleid: beleid: helaas voldoet de ernst van de dematochalasis niet aan de voorwaarden die de zorgverzekeraar stelt aan vergoeding voor een blepharoplastiek. patiã£â«nt werd op de mogelijkheden gewezen van zelfbetaling. de patient gaat hier over nadenken.

y=Voormalige gebruiker (score **-1.456**) top features

Contribution[?]	Feature
-1.456	Highlighted in text (sum)

reden van komst / verwijzing: reden verwijzing: oogleden anamnese: voorgeschiedenis blanco medicatie geen anamnese sinds 0,5jr progressieve hoofdpijn en vermoeidheid na ontstaan afhangen bovenooglid li. kan hierbij ook iets minder zien, moet mn bij autorijden de wenkbrauwen helemaal optillen. kan 's avonds niet meer goed lezen vanwege de vermoeidheid. merkt zelf geen verschil tussen li en re. intoxicaties roken 20jr terug gestopt, alcohol-, drugs- allergieen geen bekend lichamelijk onderzoek: wenkbrauwen bdz op niveau van de orbitarand. dermatochalasis re mild, li tot op de wimpers. levatorfunctie bdz intact, goede functie. geen ptosis. conclusie: conclusie: milde dermatochalasis. beleid: beleid: helaas voldoet de ernst van de dematochalasis niet aan de voorwaarden die de zorgverzekeraar stelt aan vergoeding voor een blepharoplastiek. patiã£â«nt werd op de mogelijkheden gewezen van zelfbetaling. de patient gaat hier over nadenken.

Figure 70: Stochastic Gradient Descent LIME Visualization of Text 5 of Experiment 5

The SGD classifier views the presence of "alcohol-" and "drugs-" to indicate that the patient is also a non-smoker. This is likely due to other texts containing this format of text about patients that do not use any of the listed substances.

MedRoBERTa.nl
Prediction: Huidige gebruiker (Current user)

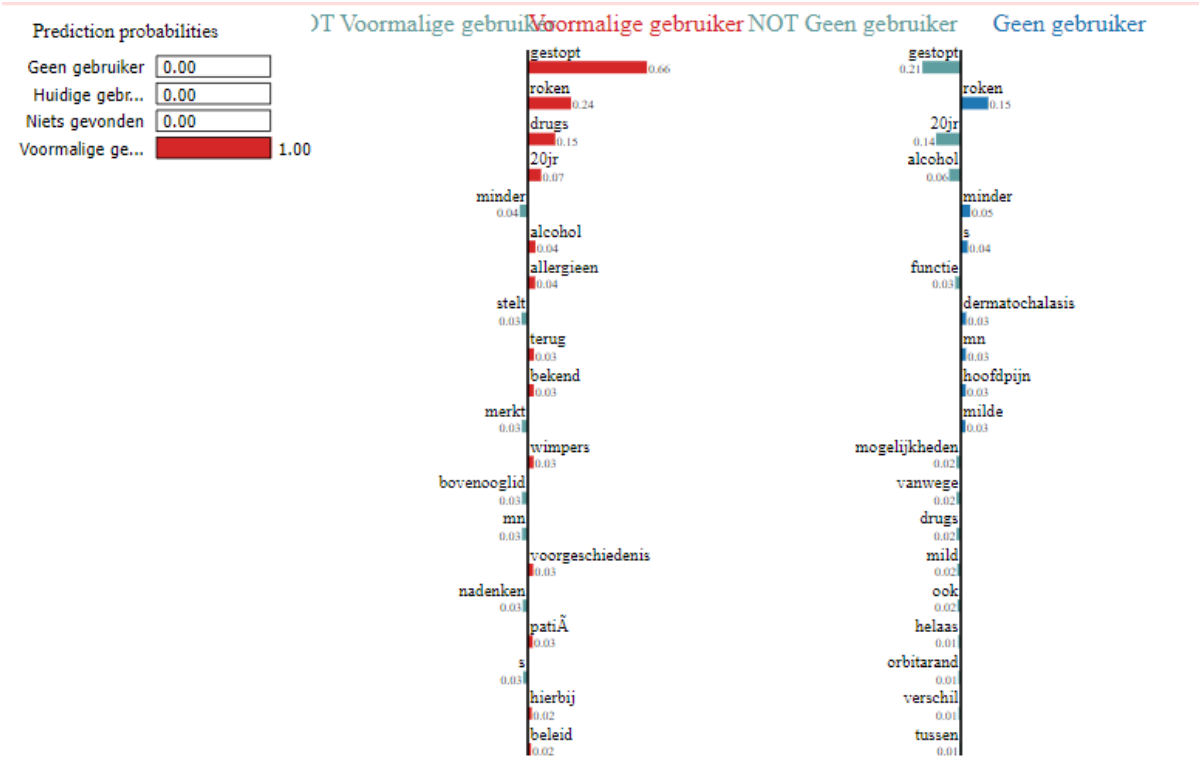


Figure 71: MedRoBERTa.nl LIME Visualization of Text 5 of Experiment 5

MedRoBERTa.nl correctly sees the word "gestopt" (stopped) as strong indicator for a previous user and classifies the text correctly.