Universiteit Leiden

Master Computer Science

Natural Language Processing for lifestyle recognition in discharge summaries

Name:	Cheyenne Heath
Student ID:	s1647865
Date:	August 14, 2022
Specialisation:	Data Science
1st supervisor:	Marco Spruit
2nd supervisor:	Peter van der Putten
Add. supervisor:	Kalliopi Zervanou
Add. supervisor:	Janet Kist (LUMC)
Add. supervisor:	Jan Pronk (HagaZiekenhuis)

Master's Thesis in Computer Science

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

Abstract

Discharge summaries contain various amounts of information that is currently not always utilized. To make this easier for medical professionals, the information in the discharge summaries must be extracted and added to the corresponding Electronic Health Records (EHRs). During this project, a case study was conducted to extract patients' smoking status from their discharge summary, focusing on current and past smoking patients. To achieve this goal, two different methods are applied and compared. One consists of a string matching approach which consists of patterns found manually in the discharge summaries. The second approach incorporates both sentence embeddings and a neural network. The sentence embeddings are learned by a Dutch BERT model BERTje which we continued pretraining on a part of the discharge summaries. In case of the neural networks, we compared a LSTM and a combination of an LSTM and a CNN. The results of this project showed that the string matching approach had an accuracy of 63%. The results also suggested that the location of the information regarding the smoking status in the sentence is different for past smokers and current smokers. Information regarding past smokers was located farther from the word smoking than information regarding current smokers. Unfortunately, the neural networks were not able to classify the patients well. These approaches tended to classify each patient as a past smoker. Conducting an error analysis on both the networks and the sentence embeddings we found that when trying to distinguish a current smoker from a past smoker the embeddings performed inadequately. This was likely caused by the inability of the BERT model to create embeddings that could capture this information well. Future research might be able to solve this by using more data and finetuning the BERT model on the specific task of recognizing the smoking status.

Acknowledgements

This master thesis is the results of months of hard work and is the final project for graduating the master program Computer Science: Data Science at the Leiden University. With this project I hoped to contribute to the ongoing research on utilizing digital documents such as discharge papers to increase the quality of research at both the LUMC and HagaZiekenhuis.

This research would not be possible without my first supervisor Marco Spruit. The enthusiasm for this project was contagious and I really enjoyed working on it for the past couple of months. I would also like to thank Marco for the helpful feedback and guidance throughout these months.

Also a big thank you to Peter van der Putten who joined at the end of this research as second supervisor. The effort put towards this project on such short notice during the middle of the summer and the feedback on the thesis was really appreciated.

I would also like to thank Janet Kist for the weekly meeting and for the brainstorm sessions we had and Kalliopi Zervanou for the great insights halfway during this project and feedback on the final draft.

This project would not have been feasible without the hard work of Jan Pronk and the other HagaZiekenhuis staff to facilitate the data.

I would also like to thank Luka for the feedback on the ideas and written text and the moral support. Finally, I would like to thank my family and friends for their support and listening ear during this project.

Contents

1 Introduction **1**
- 1.1 Case study: distinguishing smokers from past smokers based on their discharge summaries . . . 2
- 1.2 Research questions . . . 3
- 1.3 Structure . . . 4

2 Related work **5**
- 2.1 SYMBALS . . . 5
  - 2.1.1 Protocol . . . 7
  - 2.1.2 Executing SYMBALS . . . 7
- 2.2 NLP concepts . . . 8
  - 2.2.1 String matching . . . 8
  - 2.2.2 Regular expressions . . . 9
  - 2.2.3 Word and sentence embeddings . . . 9
  - 2.2.4 Spelling . . . 9
  - 2.2.5 Abbreviations . . . 10
  - 2.2.6 Stemming and lemmatization . . . 11
- 2.3 NLP for information extraction . . . 12
- 2.4 Neural approaches . . . 13
  - 2.4.1 Word2Vec . . . 14
  - 2.4.2 BERT . . . 14
  - 2.4.3 RobBERT . . . 15
  - 2.4.4 BERTje . . . 15
  - 2.4.5 mBERT . . . 16
  - 2.4.6 BertNL . . . 16
  - 2.4.7 Neural networks . . . 16
- 2.5 Conclusions . . . 17

3 Research approach **19**
- 3.1 CRISP-DM . . . 19
- 3.2 Evaluation . . . 20

4 Data understanding and preparation **22**
- 4.1 Data gathering . . . 22
- 4.2 Data preparation . . . 23
- 4.3 Exploratory data analysis . . . 24

5 Methods **30**
- 5.1 String matching . . . 30
- 5.2 Pretraining BERT . . . 31
- 5.3 Sentence embeddings . . . 32
- 5.4 LSTM . . . 32

6 Results **35**
6.1 String matching 35
6.2 Neural networks 38
6.3 Qualitative sentence embedding analysis 39

7 Discussion **45**
7.1 Interpretation of results 45
7.2 Limitations 45
7.3 Future research 46

8 Conclusion **48**

List of Figures

1 Flowchart SYMBALS 6
2 Description of BERT pre-training and fine-tuning procedures on different tasks [17] 15
3 Structure of a RNN layer [67] 17
4 Structure of a LSTM layer [67] 17
5 Architecture of a simple CNN network, adapted from [51] 18
6 CRISP-DM Process Diagram 19
7 Distribution of number of sentences per discharge summary . . . 27
8 Top 25 word frequencies with stopwords 28
9 Top 25 word frequencies without stopwords 29
10 Overview of the different NN architectures 34

List of Tables

1 Based on benchmarks [15], [16] 18
2 Confusion matrix example 20
3 Examples of forms answers about smoking status of patient and the desired label 23
4 Example queries for labeling form answers 23
5 Example raw text excerpts from discharge papers to demonstrate the structure 25
6 Data statistics 26
7 Ranges for sentence selection string matching on character level, represents number of characters before “roken” / number of tokens after the words “roken” 31
8 List of queries and their labels for exact string matching. In this order the queries were also performed. All of the queries also require the word “roken” to be present. 35
9 Exact string matching results 36
10 Examples to show the impact of ranges on the prediction label . 37
11 Results of different LSTM architectures 38
12 Results of different LSTM architectures on a smaller input data 39
13 General knowledge test results 40
14 Concept Identity Test results, where PS = Past Smoker and CS = Current Smoker 42
15 Sentences to calculate similarity past smoker/current smoker pair 43
16 Similarity between pairs of past and current smokers 43
17 Paper reference number sorted by topic, result of literature research 58
18 Sentences indicating current smokers used for the similarity scores 58
19 Sentences indicating past smokers used for the similarity scores . 62

1 Introduction

The popularity of extracting information from data is increasing in almost all domains. One particular type of extraction of information has gained even more traction: text mining [24]. While there are many definitions of text mining the most common is that text mining is the process of extracting information from unstructured textual data in the form of patterns, concepts, topics and keywords [58]. Text mining is also a multi-disciplinary field based on other domains such as computational linguistics, statistics, machine learning, data mining and information retrieval [20].

This research will use text mining to extract information about patients, and their lifestyles specifically. Using Natural Language Processing (NLP) for text mining has increased in popularity over the past decade. NLP consists of techniques that utilize the underlying metadata, including content, context, and patterns [19]. Both text mining and NLP can be applied in every domain where text data is digitally stored and available. As this is the case for almost all domains, it is important to keep researching how we can utilize these techniques to make sense of the data and use the data to get more insight into the different domains and tasks.

Both text mining and natural language processing have been used in the medical field for predicting readmission [42], mortality risk [45] and outcomes [6]. The majority of the research that were concerned with utilizing NLP were tasked with identifying Adverse Drug Reactions (ADRs) [55]. This has become a popular research field as it is critical to be able to identify these ADRs because it concerns the health of patients. This research topic was not limited to medical text data that finds its origins at hospitals. By leveraging social media text data, for example data from medical forums [36], it is possible to find ADRs that might not have been reported by the patient to their healthcare provider.

A research domain that is less well known is the extraction of patient information from medical free text. Patient information is a very broad term and can include information such as medical history, familial relations, living situation and genetic test results. Recent research has suggested that while it is a challenging subject, it is possible to extract these types of information [32]. Information about the patient can be very valuable to add to the patients' Electronic Health Records (EHRs). An EHR is a digital chart that contains information about the patient and other notes that the doctors or nurses write down. There are two places where the healthcare professionals can write these notes: in the free text fields or in the diagnostic file. However, to access this diagnostic file, the user has to perform extra steps to search for the right field and fill it in. It is therefore faster, and more common, to write a few keywords and notes in the free text field.

While writing information in the free text field is more convenient for the healthcare professionals, for studies that only use the diagnostic files it means that a lot of information regarding the patient can be missing. While often

information such as age, weight, medical history and other common facts are present in the EHR, lifestyles such as smoking, alcohol consumption, sleep pattern and diets can be written down quickly in the free text field and not in the diagnostic file. It is a desired goal to have a method that searches free text fields and other text data for information regarding the lifestyle, extract it and add it back to the diagnostic file.

Research like described above has often been conducted using the free text fields from the EHR. However, other textual sources such as discharge summaries or other notes that doctors write to each other have yet to be explored exhaustively. Of the studies that are conducted using discharge summaries, the majority is about extracting ADRs [56]. Nevertheless, other studies have shown that is is also possibly to use the discharge summaries to predict readmission to the ICU [42] with relatively good results. This is a promising text source that might contain useful data that might not be written anywhere else. Discharge summaries also have the advantage that they contain a highlight of all the important information of one patient in one document. Discharge summaries often contain a lot of information, from basic personal information such as age and weight to very detailed information about examinations, medication, diagnosis and other notes.

1.1 Case study: distinguishing smokers from past smokers based on their discharge summaries

The Leiden University Medical Center and HagaZiekenhuis in The Hague want to leverage free text data from discharge summaries to extract patient lifestyle information and add to the EHR. The lifestyles that were considered were: smoking, alcohol consumption, drug consumption, daily exercise, professional sport, diet and sleep. To explore the possibilities we designed a small case study. During this case study we investigate to what extent it is possible to recognize current smokers and distinguish them from past smokers using their discharge summary.

Smoking is selected as a case study as to this day it is still one of the leading preventable causes of disease and death. According to the WHO smoking kills 8 million people per year [4]. In the Netherlands it concerns about 20,000 people each year who die because of the consequences of smoking [2]. This makes smoking a very important lifestyle to consider when making a medical decision. To assist doctors making these medical decisions it can be very helpful to keep their EHR up to date with the smoking status of their patient. To do so, using the free text field from other sources such as discharge summaries and extracting this information from it is needed. Although the research aimed at just extracting the smoking status from the discharge summary including never smoking patients, there was not enough data available to do so, thus a smaller study based on recognizing current smokers and past smokers was designed.

This research aims at acquiring a better understanding of what is needed

to develop a method that can distinguish current smokers from past smokers and which methods are the most appropriate. The data we can work with is limited thus it is important that this research provides a good base for future research. We need to consider multiple aspects of such a pipeline, like preprocessing techniques, working with challenges that all free text data has, and additionally challenges that are specific for the medical domain. Besides this, we need to research what kind of techniques are most commonly used and which make sense to use for this task.

The aim of this research is to develop a pipeline research prototype that can identify smoking patients from past smoking patients using the free text data in discharge summaries. The ability to extract lifestyles from these free text fields means that researchers can access this data even if the healthcare professionals do not fill in the designated fields in the diagnostic file. Scientifically, this research aims at providing better understanding of which techniques are best suitable for certain challenges. This will aid further research in using natural language processing for the free text in discharge summaries.

Being able to extract the lifestyles from the discharge summaries also enables the possibility to make the data from these discharge summaries reusable. This can be achieved by making it possible for the extracted lifestyles to end up in the diagnostic file. This way, healthcare workers do not need to read pages of text when a patient returns to the hospital. This can also save time and frustration with both the patient and the medical specialist. The specialist who is asking question about lifestyle indicators do not need to start from scratch, but can use the previous recorded information and pick up from there.

1.2 Research questions

To explore the possibilities of leveraging the free text in discharge papers the following main research question will be answered:

"Which NLP techniques are best suited for recognizing the smoking status of a patient in discharge summaries?"

To answer the main research question, the following sub research questions will be studied and answered during the span of this thesis:

RQ1. What are the challenges in preprocessing and working with free text data from the medical domain?

RQ2. How can the challenges in preprocessing free text data from the medical domain best be addressed?

RQ3. What are the current state of the art NLP techniques for extracting information from text data?

RQ4. How does a simple string matching technique compare to a neural approach for classifying patients based on discharge summaries?

RQ5. To what extent is it possible to determine if a patient used to smoke or smokes currently based on their medical discharge summaries?

Research question 1, 2, and 3 will be answered by performing a literature review and giving an overview of challenges that arise when working with free text data in the medical domain specifically. Different approaches to address these challenges will also be discussed based on the literature review. An overview of the current state of the art techniques will also be given followed by an explanation of the most popular methods. Question 4 will be answered by implementing both techniques on preprocessed data in Python and comparing the results based on accuracy, precision, recall and the F1 score. The final research question will be answered by analysing the output of both models, focusing on which patients are classified correctly and which patients are not. The goal here is to find any patterns that we can learn from.

1.3 Structure

For the remainder of this document, the thesis is structured as follows. After this introduction, the results of the literature review are given in Section 2. Section 3 will cover both the research approach and the evaluation procedure. Section 4 will give an overview of the data as well as the process of data gathering, preprocessing and concludes with an overview of the descriptive statistics of the data. The methods to answer research questions 4 and 5 will be described in Section 5. The results of the models will be described in Section 6. And finally Section 7 and Section 8 will provide a discussion and conclusion to conclude our findings of this stud. In the conclusion each research question will be answered and directions for future work are suggested.

2 Related work

This chapter describes the related work and theoretical background for this thesis. First, the method for executing the literature research is discussed. Then a number of common and relevant NLP concepts are introduced. After this, the findings of current state of the art methods for using NLP in the healthcare domain are described, with a separate section in which methods for Dutch medical text are highlighted. Furthermore an overview of the most common machine learning methods is given as well as an explanation of some popular neural approaches for NLP.

2.1 SYMBALS

The method used for conducting the majority of the literature review for state of the art NLP methods for information extraction in the healthcare domain is SYMBALS [65]. Using the information gained from this literature review the first three research questions should be answered as described in Section 1. SYMBALS is an innovative systematic review methodology, it utilizes both the traditional method of backward snowballing with active learning. Backward snowballing entails that the users uses the reference list of a relevant paper to identify new papers that could be relevant [69].

Active learning is a machine learning method whereby the most relevant data to learn from is chosen by an algorithm. Active learning is very effective as it can achieve greater accuracy with fewer labeled data by choosing its own data to learn from [54]. SYMBALS is used in this research for the literature review because it provides a way to more effectively scan papers without having to read all the abstracts returned from a query.

An overview of the SYMBALS process used for this literature review can be found in Figure 1. The section in the figure that falls within the red dotted line is the core of SYMBALS and starts with a protocol set up in which a few criteria are defined such as the stopping criteria for the active learning and backward snowballing phase but also the criteria on which a paper is selected as relevant or not and the databases to be searched with their matching queries. The database search is the first phase after the protocol setup in which appropriate databases are selected and a query is performed to find all documents that match this query. Next the titles and abstracts are extracted and uploaded in a program that assists in the active learning phase. Based on the recommendation of the author of the paper we selected ASReview [64]. If the stopping criterion is met the process will continue with backward snowballing which is done manually. Once this is done we end up with a list of papers that should contain the relevant information for the research.

However, as the focus of this study shifted after this initial literature review, parts of the final literature review used a different method. For the remaining literature review the queries are executed on Google Scholar to find the appropriate literature to gain the required information. Using the back-

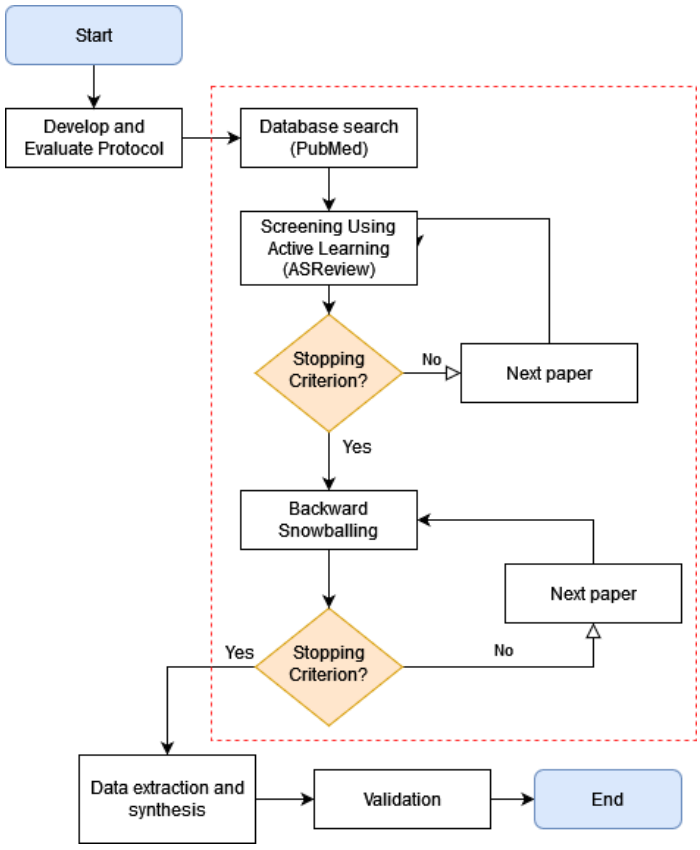


Figure 1: Flowchart SYMBALS

wards snowballing method on relevant papers, additional papers can be found. This approach is more appropriate as we needed a lot of different information about different topics rather than answering one or two specific questions, like we could with the first part of the literature review. We also have a better understanding about the problem and are therefore able to make the queries more specific, resulting in less but more informative hits.

2.1.1 Protocol

This literature review using SYMBALS aims to find the state of the art methods for information extraction using NLP methods in the medical domain. In order to decide on the methods used during this case study, it is necessary to get a good understanding about the state of the art methods. To find relevant papers the PubMed [3] database is used to perform the following query on title and abstract: NLP OR Natural Language Processing AND EHR or Electronic Health Records AND "Free Text". The following criteria are used to conclude if a paper is relevant after executing the query in the database and extracting the title and abstracts:

- Uses a NLP method to extract information or classify data
- Uses method on free text, not structured text
- Is not a systematic review

To carry out the systematic review, the software ASReview [64] is used. The stopping criterion of the active learning phase is determined by formula 1, where R is an estimate for the number of relevant papers that can be found, N equals the total number of papers, r is the number of relevant papers and i is the number of irrelevant papers. An additional stopping criterion is introduced which prevents us from continuing if we found 10% of the total number of papers of consecutive irrelevant papers. We look at 10% of the number of included papers to use during the backwards snowballing step in the same order that ASReview presented them.

$$R \approx N \times \frac{r}{r+i} \tag{1}$$

2.1.2 Executing SYMBALS

Applying the query to the PubMed database yielded 370 papers that matched the query. According to the SYMBALS description this means that for the next step, screening using active learning, we need to first manually review 10% of the papers to determine what our stopping criterion will be. After reviewing 37 papers, we found 4 relevant papers and 33 irrelevant papers that did not meet the criteria we set up in the protocol. This means we can apply formula 1, to determine our stopping criterion for the active learning phase.

This resulted in an approximated R of 38. We want to find 95% of all relevant papers thus we need to continue with the active learning phase until we find 36 relevant papers, or if we evaluate 37 (10% of N) consecutive papers as irrelevant.

We stopped when we found 36 relevant papers. In total we reviewed 106 papers during the active learning phase. The next step is to perform backward snowballing. A list of the order in which ASReview showed the papers was kept manually. This order was later used for the backward snowballing step. This step resulted in an additional 3 papers to include in our list. After finishing the SYMBALS procedure we found 39 papers that contained relevant information for the literature research of which the results are discussed below.

2.2 NLP concepts

This section highlights some of the most used NLP concepts, and concepts that are potentially relevant for the preprocessing of the data used in this thesis. The selection of these concepts is made based upon similar research and preliminary insights in the Dutch discharge notes. First, two common approaches to NLP that are not machine learning are discussed. Then the approach of using word and sentence embeddings are discussed and finally an overview is given of common preprocessing techniques that are used when using text data.

2.2.1 String matching

String or pattern matching is a very popular approach to NLP problems. In its most simple form string matching is performed by finding an exact match of a certain string in the text [38]. The advantage of string matching algorithms compared to machine learning algorithms is that no labeled training data is needed. A number of rules are applied manually to create the string matching algorithm.

String or pattern matching has various applications within different fields. For example, it can be used for text, image, signal, and speech processing [23]. String matching algorithms can be divided in two categories: exact and approximate algorithms. Where exact string matching approaches are the simplest of the two: the exact string needs to be present in the target text, approximate algorithm require more computation and are more complex. Approximate algorithms find sub strings that closely match the original string by using one of similarity measures available. These approaches can be very useful in situations where spelling errors are an important challenge. Approximate algorithms can work with spelling errors as instead of searching for an exact match, the most similar match is found.

The string matching approach might be very useful for the case study in this thesis. To determine whether a person smokes of used to smoke there might be clear strings present in the text that could distinguish the two classes from each other. The downside to this would be that the strings to match on need

to be constructed in a well motivated way. This requires a deep understanding of the data.

2.2.2 Regular expressions

Regular expressions are a specific method for exact string matching. Regular expressions use a specialized syntax which can be used to specify a pattern [61]. This approach can be used to create complex rules. A pattern consist of special tokens, also called metacharacters, as well as regular letters and numbers. To create the rules the regular letters and numbers are combined with the metacharacters.

Regular expressions might not always be the best choice. For simple string matching it might be unnecessary to create a regular expression rule, since that could be too complex. Another reason why regular expressions might not be the optimal choice is when working with large amounts of data. Regular expressions are almost always slower than string operations [61]. The main reason to use regular expressions would be when you need complex rules or when all the information that needs to be found follows the same pattern.

2.2.3 Word and sentence embeddings

In order to use text data as input for many algorithms, it first has to be transformed to some sort of numerical representation. Whilst there are many ways to do so, like word frequency matrices, tf-idf weighted representations and N-grams, they often produce large, high dimensional and sparse matrices. In addition to this, they are unable to take word order or syntactic and semantic similarities of the words into account [28]. These disadvantages are not as common when using embeddings. Embeddings are dense, latent representation of the text, often learnt by an unsupervised machine learning algorithm. There are multiple approaches to learning these latent embeddings, and most of them use neural networks as a base. Some examples of approaches to train the word embeddings are Word2Vec [39], glove [46], fastText [8], ELMo [47] and BERT [18], of which a few will be discussed in Section 2.4.

To train word embeddings, often large databases are needed. However, there currently exists a lot of pretrained embeddings that can be used to transform text data to dense representations. This can be useful if there is not enough data available to train your own embeddings.

2.2.4 Spelling

Spelling mistakes are almost unavoidable when working with unstructured free text data and are one of the most common challenges in the NLP domain. A lot of the approaches used in NLP are based on similarity between words, which can be measured using a number of different metrics. In order to achieve the best result for this task, it can be essential that terms are all in the correct spelling. However, unstructured text, and especially something like clinical

notes, are expected to contain several spelling and grammar errors because they are written by humans. We might expect that something like a discharge paper contains less spelling mistakes than a clinical note as clinical notes are for personal use and therefore it matters less if there are spelling errors included, while discharge papers are shared between other doctors of medical staff and tend to be more formal. Earlier, in Section 2.2.1, we saw that approximate or fuzzy string matching can be a solution for dealing with spelling errors because they do not require an exact match by using a similarity measure.

Research on correcting spelling errors and other ways to address this challenge is studied both inside and outside the medical domain [62]. The challenge of spelling errors is not new and has been researched for decades. In 2001 it was stated that the correction of spelling in medical records is a critical issue because they found that the rate of spelling errors in medical records was 10% higher than in other texts such as newspapers [50].

2.2.5 Abbreviations

When working with unstructured text, abbreviations are the next challenge. Abbreviations are different than spelling errors as they actually are meant to be written in their format. If treated like spelling errors, the abbreviations will likely get the wrong context.

Here the first hurdle when working with abbreviations is presented, they need to be distinguished from spelling errors. An obvious approach would be to build a database for domain specific abbreviations [71]. There are databases online which contain a large number of domain specific abbreviations, like `abbreviations.com`, however these are not complete and can contain multiple entries for the same abbreviation. For example, the abbreviation "PY" which is often used in combination with smoking means in that context "pack year" and describes a measure that is used for the amount that a person smokes for a long period. The database contains 26 definitions but none of them describes "pack year". In addition to this it is not uncommon that people create their own abbreviations to make writing down information easier and faster.

Therefore an important step in working with abbreviations is recognizing them. However, this task proves to be challenging. During a study executed by Wu et al. [70] they concluded that existing NLP systems achieved sub optimal performance in abbreviation identification. They achieved F-scores ranging from 0.165 to 0.601 which were lower compared to the gold standard of expert identification of abbreviations. More recent research has found a slight improvement by utilizing a deep neural network and achieving an accuracy of 0.719 for detecting and normalizing abbreviations in scientific [76].

In the medical field a research executed by Jaber et al. proved to be very powerful for disambiguation of clinical abbreviations by making use of one-fits-all classifier with deep contextualized representation from pretrained language model like BERT [27]. They achieved an accuracy of 99.13% using MS_BERT which is a BERT model pre-trained on notes from neurological examination

for Multiple Sclerosis (MS) patients. They achieved this score by continuing to train the MS_BERT model on the specific disambiguation task and applied a neural classifier on the hidden state of all the tokens. The neural approach they used was a feedforward layer, activation ReLu and another feedforward layer.

Disambiguation of the abbreviations might not always be necessary. In some cases abbreviations contain important information about the structure of the text, or an approach is used which can process abbreviations like word or sentence embeddings. If abbreviations are included in the text on which methods to create sentence embeddings are trained, chances are that they can place the abbreviation in the correct context, meaning the same context as their full form [26].

2.2.6 Stemming and lemmatization

Earlier, in Section 2.2.3, we already described that working with text data can lead to very high dimensional and sparse representations. Besides using word embeddings to learn a dense representation of the text, there are other methods that can be utilized to create less sparse data representation without using latent features. These methods are stemming and lemmatization and both aim to normalize words to a common base. From the two methods, stemming is the simpler variant. Stemming works by removing prefixes and suffixes of words to find the common base or root. For example, the words "medications" and "medical" are brought back to a common root "medic" by removing the suffixes of the words. Which also shows where the downside of stemming lies. Not only can it lead to non existing words, for example if we stem "studies" is will be stemmed to its root "studi", but it can also stem to a word that might have a different meaning. In the first example both words have a different meaning. However, it is still a popular method because it is able to capture the general meaning of the different words without using much calculations or a external dictionary.

The most popular stemming algorithm for the English language is the Porter algorithm [66] which works by removing common suffixes from words. It is a very simple but efficient algorithm. A Dutch version of the Porter algorithm [31] is also available which has a similar performance to the English version.

An other approach to minimizing the dimensionality of text data is by using lemmatization. Unlike stemming, lemmatization uses a dictionary and produces the lemma of the word. For example, words like "saw" and "are" will be brought back to their roots "see" and "be". Lemmatization tries to bring words back to their dictionary entry. The definitions of words that are brought back to the same lemma are often more similar than the definitions of the words that are brought back to the same root when using stemming [29].

Whether to use lemmatization or stemming depends partly on the problem and the type of data. Often lemmatization is preferred because it uses a dic-

tionary and therefore might provide more accurate results. However, it is more computationally intense because it requires a dictionary look up. Stemming is faster because it does not require the dictionary lookup and sometimes the difference between using lemmatization and stemming is only marginally different [7]. The decision to use stemming or lemmatization therefore depends on a lot of factors. For lemmatization a dictionary is needed, which might not always be available and be complete. Although some research shows that the difference between lemmatization and stemming might only be marginally different, this really depends on the problem, the morphology of the language and the importance of the information loss during stemming.

2.3 NLP for information extraction

Working with free unstructured text data brings some challenges to the table compared to working with structured data [59]. These challenges also translate to the medical domain. Where structured text data, such as filled in patient forms, document data in contained fields, free text data such as discharge letters contain information in an unstructured manner [5]. Because of it's unstructured nature, performing a simple search on keywords can result in low recall [10]. Using this free text data is very valuable as the structured text fields are not always filled out completely. It often happens that a certain ADR or other information about a patient is present in the unstructured data but not in the structured fields [25]. Leveraging the unstructured data can therefor improve the treatment op the patients [53].

The free unstructured data in the medical domain also has it's own unique challenges. This is partly due to the nature of the domain but also because there exist only limited standards or guidelines for creating discharge letters [30]. While there is a general consensus about what should be in a discharge letter, there are no clear guidelines for how it should be documented in the letter [68]. Having a standard or guidelines for creating discharge letters would greatly improve the quality of the discharge letters [63]. In the medical field there are multiple ways to describe the same clinical concepts across different disciplines, hospitals and even differ from medical worker to medical worker [35]. The same can be said for acronyms and abbreviations which do not occur in for example a dictionary as was stated in Section 2.2.5.

Copious amounts of different methods to extract patient information from free text fields and discharge summaries have been proposed over the past decade. Often, standard NLP techniques such as regular expressions can get reasonable results in identifying information such as outcomes [6], and adverse drug reactions [21] in the EHRs.

Not only are deep learning techniques used for identifying ADRs in EHRs, they also perform extremely well for predicting tasks such as predicting the risk of mortality for patients with acute myocardial infarction [45] where researchers were able to get an accuracy of 92.89%. Using deep learning techniques for other prediction tasks also outperformed the more simple pattern matching

methods. An approach using word embeddings as input for a RNN architecture yielded a f measure of 0.755 compared to a string matching baseline of 0.65 when labeling adverse drug reactions in Twitter posts [13]. Another research showed great performance increase by using a CNN architecture over a string matching baseline [40].

While there has been substantial research covering the extracting of ADR from medical free text fields, extracting information about patients and specifically about lifestyles has not been covered as rigorously [52]. Extracting lifestyles from these unstructured text fields using NLP techniques might be feasible [75], however this study was done on a small dataset and might therefore not be as accurate. Extracting familial relations of patients has also been a relative new field that yields decent results, obtaining F1 scores of 0.869 and 0.791 in the training and test sets, respectively [57]. These results were obtained by using a deep learning approach, and performed much better than a rule based approach. Other information, such as temporal structures of clinical events have been successfully extracted by using a hybrid approach based on rules and syntactical analysis [22]. Another goal of using EHR information, could be to asses the risk of inpatient violence [**?**]. By using a SVM on the output of an embedding algorithm, paragraph2vec [34], which is similar to word2vec on a paragraph level instead of word level.

One particular lifestyle, smoking, has been covered the most, compared to other lifestyles such as sleep, exercise and drug use. One research conducted in 2017 [44] showed that extracting a patients' smoking status from dental records was possible, by comparing all the entries for the i2b2 (Informatics for Integrating Biology & the Bedside) smoking challenge [1]. All results were an improvement, but the approach by Clark et al [12] based on a support vector machines performed well for classifying current smokers. A different approach by Cohen et al [14] performed well for classifying past smokers using a combination of word level rules and a support vector machines. Another paper [49] which looked at extracting smoking status from EHR by using a NLP tool based on a SVM that uses words a bag of words [74], showed that for current smokers, a sensitivity (which answers: of all the current smokers, how many did we predict correctly?) of 92% and specificity (which answers: of all the not current smokers, how many did we predict as not current smokers?) of 86%, and for ever smoking patients, the NLP-based algorithm achieved a sensitivity of 94% and specificity of 73%.

2.4 Neural approaches

This section gives an overview of popular neural approaches for NLP problems which were mentioned in section 2.3, first two different methods to get dense representations of text data called embeddings will be discussed. Furthermore an overview of different Neural Network methods to classify text data using the dense representation is given.

2.4.1 Word2Vec

Word2Vec is a word level algorithm that is used to learn word associations from a large corpus of text [39]. These word associations are represented as word vectors. The Word2Vec algorithm uses a neural network to learn these vector representations. The algorithm can also utilize either of two different model architectures: continuous bag-of-words (CBOW) or continuous skip-grams. The skip-gram model predicts the surrounding context words given the target word [39] while the continuous bag-of-words model works by calculating the conditional probability of the target word given the surrounding context words. Thus the skip-gram does the exact opposite of the CBOW model [72].

2.4.2 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a language representation model which is designed to pre-train deep bidirectional representations from unlabeled text [17]. It is able to learn the bidirectional representations by taking both the left an right context in all the layers of the corpus into account an jointly conditioning them. A pretrained BERT model can also be fine-tuned by utilizing one extra output layer as a result. As opposed to Word2Vec the input that BERT requires a sentence or a sentence pair, thus making this algorithm work on a sentence level as opposed to a word level. BERT is pretrained on two specific tasks: masked language modelling (MLM) and next sentence prediction (NSP).

In the case of the MLM task, some of the tokens are masked at random and the model tries to predict the original words that are masked based on the context. This enables the model to train a deep bidirectional representation by fusing the left and right context of the masked token. The next task is next sentence prediction. This is an important step because it enables the model to learn a relationship between two sentences, which for example downstream tasks such as Question Answering are based on. To achieve this, sentence pairs are needed of which 50% is an actual pair AB where sentence B follows A and the other 50% are pairs where sentence B does not follow sentence A.

To fine-tune the BERT model labelled data of the appropriate task is needed to fine tune its' parameters. The Figure 2 shows both the pre-training task and the fine-tuning of the BERT model. The pre-training consists of both MLM and NSP while the fine-tuning part is trained on different tasks such as question and answer pairs, named entity recognition and multi-genre natural language inference.

For this research we are working with Dutch text data and thus it is important to explore which possibilities for Dutch data there currently are. The rest of this section will therefore go over some BERT variations that have been pre-trained on Dutch text data.

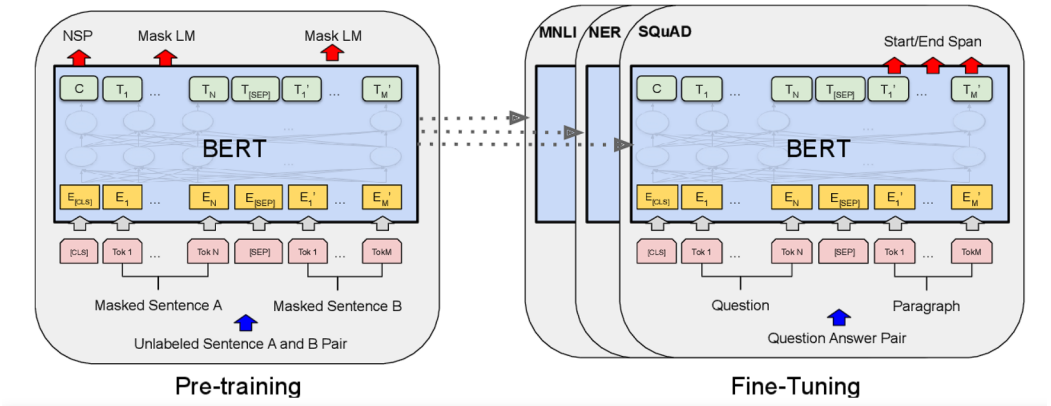


Figure 2: Description of BERT pre-training and fine-tuning procedures on different tasks [17]

2.4.3 RobBERT

RobBERT [16] is a state of the art Dutch BERT model based on RoBERTa [37] where the main chances lie in the training of the model. RoBERTa trains longer on a bigger corpus of text with bigger batches. RobBERT is pretrained using the RoBERTa training regime which consist of NSP and MLM. RobBERT is trained on the Dutch section of the OSCAR corpus, which is a large multilingual corpus. It contains 6.6 billion words divided over 126 million lines of text. This corpus is much larger (39 GB) than most Dutch corpus used by other BERT models. The second version of RobBERT changed the RoBERTa's native tokenizer to a Dutch tokenizer which was constructed using the same OSCAR corpus.

2.4.4 BERTje

BERTje [15] is another Dutch pretrained BERT model. BERTje is based on the standard BERT model and made modifications in the pretraining data generation procedure for both MLM and NSP. Instead of training on NSP, BERTje is trained using the SOP objective. SOP means that the second sentence in each training example is either the next or the previous sentence. Instead of the MLM objective where single word pieces are randomly masked, word pieces that belong to the same word are masked. They masked 15% of the tokens using this strategy. This strategy was applied because they found that some suffixes of words are too easy to predict [33]. BERTje is pretained on a collection of data including: a collection of contemporary and historical fiction novels, TwNC [43] (Dutch new corpus), SoNaR-5000 [41] (a multi-genre reference corpus), web news and Wikipedia. The data combined covers 12.1 GB.

2.4.5 mBERT

mBERT stands for multilingual BERT and is trained on Dutch wikipedia pages by the same authors as the original BERT model [18].

2.4.6 BertNL

Finally we have a fourth Dutch BERT model option, BERTNL [9]. BERTNL is trained on the SoNaR-500 corpus [41].

2.4.7 Neural networks

In this section two neural network approaches will be highlighted: RNN and CNN. Both types of networks are popular in the medical NLP domain as found in Section 2.3. This section aims to explain both network approaches and motivate which is more appropriate for this case study.

RNN and LSTM RNNs are popular because of their ability to capture the context of an input. This means that it can create a better understanding how a certain word is related to their surrounding words in a sentence when applied to the text domain [72]. RNNs are also capable of handling input of various lengths by allowing the hidden layers to loop back to themselves.

The problem with RNNs can be that they can lead to the vanishing and exploding gradient problem [67]. This problem occurs when training a RNN that is too deep and results in a too low gradient which means it is harder to train the weight. This has a domino effect on all sequential weights and prevents the RNN from learning of long-term memory. The Long Short Term Memory (LSTM) method was then introduced which eliminated this problem [67]. This makes the LSTM much more suitable for text data and explains the popularity in NLP problems [67].

Figures 3 and 4 show the difference between a standard RNN structure which contains a single layer and the LSTM structure which contains interacting layers. There is an input *x0,x1,...,xt* and an ouput *h0,h1,...,ht*. Figure 3 shows the structure of the RNN layer, which consist of an input, an output, and an activation function. Figure 3 shows that the input of the previous hidden layer is used as input for the next hidden layer. Figure 4 shows that there is more going on in the layer that just one activation function. It uses a control mechanism to solve both the long-term dependence and vanishing gradient problem of the RNN. Three different "gate" structures are added, the forget gate layer, the input gate layer, and the output gate layer which it can use so selectively manipulate the information.

CNN Another popular approach for solving NLP tasks with neural networks is to make use of Convolutional Neural Networks (CNN). CNNs involve a series of filters of different shapes and sizes which reduce the original input matrix to a lower dimension [73]. This can be very helpful for text data as it applies a

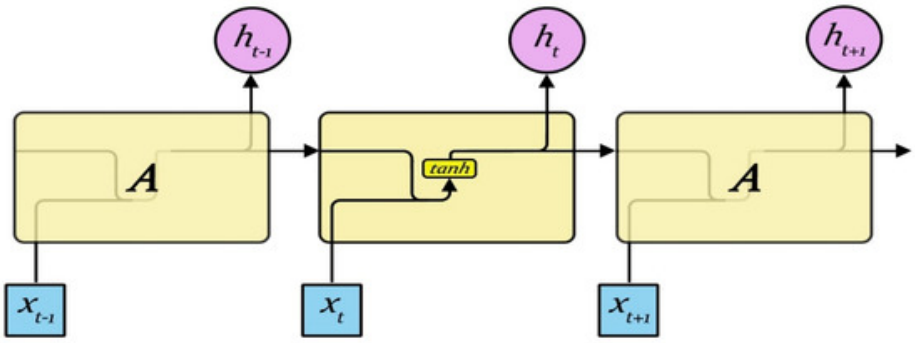


Figure 3: Structure of a RNN layer [67]

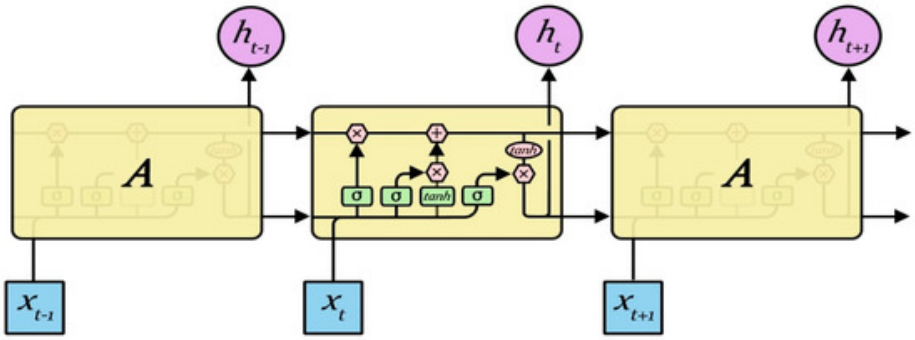


Figure 4: Structure of a LSTM layer [67]

form of dimensionality reduction. Figure 5 shows the architecture of a simple CNN network. There are different kinds of layers: the convolution layer, the max-pooling layer and the fully connected layer. The convolution layer extracts the high level features from the input by moving over the input with a set stride. The max-pooling layer reduces the spatial size of the feature extracted in the convolution layer by applying dimensionality reduction and returning the maximum value of the feature and finally we have the fully connected layer which is used to perform the classification on the flattened output of the final pooling layer.

2.5 Conclusions

Based on the literature research the following conclusion can be made. First of all there are multiple challenges that are present within working with free text in the medical domain. While most of these challenges are common for all text domains, abbreviations provide an extra challenge in the medical domain as there are no complete dictionaries. Based on discussion with the medical staff of the LUMC we decided against spelling correction and abbreviation disambiguation as they argued, based on experience, that it would be very challenging to distinguish spelling errors from abbreviations in discharge summaries.

Table 1 shows results of a benchmark done on all dutch BERT models. It

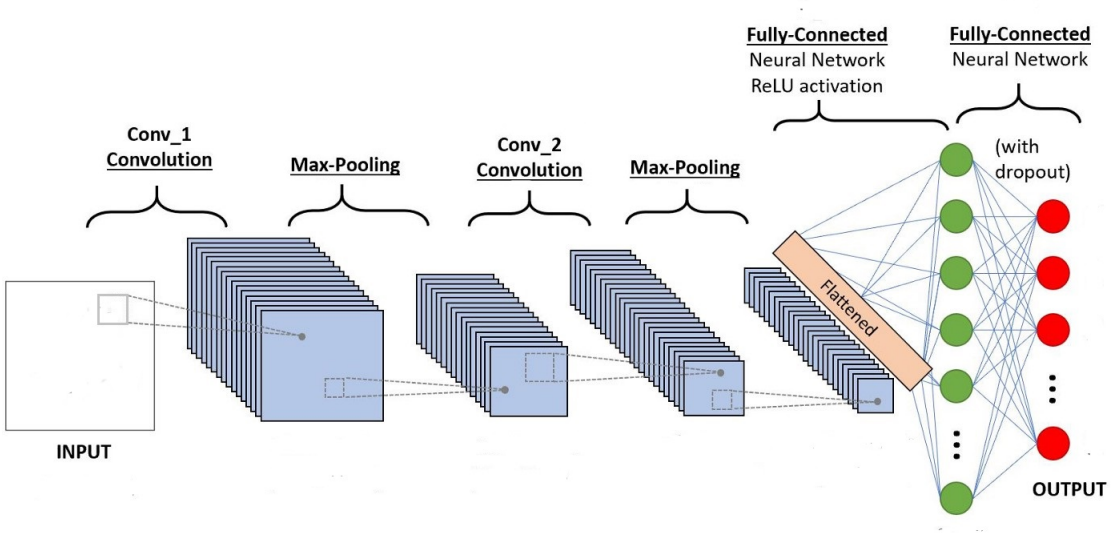


Figure 5: Architecture of a simple CNN network, adapted from [51]

	Part of Speech Tagging (%)	**Named Entity Recognition** (%)
RobBERT	96.40	89.08
BERTje	**96.48**	**90.24**
mBERT	96.20	88.61
BERT-NL	96.10	85.05

Table 1: Based on benchmarks [15], [16]

shows that BERTje scores the highest in both Part of Speech Tagging and Named Enitity Recognition. This resulted in BERTje being the BERT model that will be used in this research.

Both LSTM and CNN networks are commonly used in the research that we found. However, there was a bit more support for using LSTM as opposed to CNN. This might be the case as LSTMs have been around longer than CNNs thus there might be more research done with LSTMs. However, even in recent research, LSTMs are still often used. LSTMs will be the main structure that is used as the neural approach during this research. In addition to this we will recreate the network used by Zeghdaoui et al [73] which combines a LSTM and CNN network.

3 Research approach

This section explains which research methods are used during this thesis. The first subsection will discuss the research framework that is used during the entire research: CRISP-DM. The sections after this explain the approach to answering the fourth and fifth research questions, as the approach to answering the first three research questions is explained in Section 2.1. This section will be concluded with an explanation of the evaluation procedure.

3.1 CRISP-DM

The CRISP-DM method will be followed as a guideline for developing the models used in this research. This indicates that not every step will be necessarily executed. CRISP-DM (cross-industry standard process for data mining) was developed in 2000 to be a standard process model for data mining [11]. The other advantage that CRISP-DM has is that it is independent of both the industry sector and the technology used. The process model is divided into six phases which are depicted in figure 6.

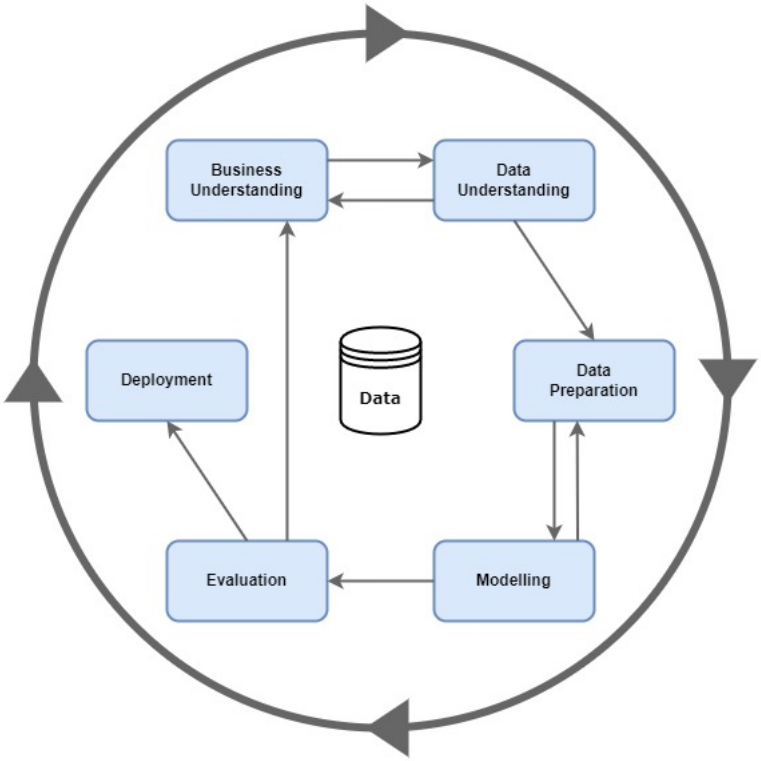


Figure 6: CRISP-DM Process Diagram

The business understanding phase is the first phase and consists of understanding the project objectives and requirements from the business perspective. In the case of this research, this will consist of understanding the gap in the existing research and the needs of the hospital. The data understanding phase

is next and starts with the data collection. This phase also includes exploring the data to get a better understanding of the data. The data understanding phase is also used to identify data quality problems, such as spelling errors, size, and label distribution, and to discover first insights about the data that can help form covers all activities to prepare the data for input and construct the final dataset. This also includes attribute selection and learning the data. The modeling phase is used to select and construct various models that are suitable for the research, which also includes fine-tuning and parameter selection. Finetuning and parameter selection is also a point where the data might need to be altered for different models, thus stepping back between the data preparation phase and modeling phase is included in the CRISP-DM approach. And finally, we have the evaluation and deployment phases where the model is evaluated and reviewed, and finally deployed to the customer. The previously mentioned deployment phase does not fall within the scope of this project and might be done at a later stage.

By following the CRISP-DM method the first few phases of data understanding and data exploration will be described in Section 4. The process of data exploring and data preparation will be done using Python and the following packages: Pandas, Numpy and Wordcloud. The implementation of the models is explained in more depth in Section 5 and is completely done using Python on secured servers of the LUMC hospital.

3.2 Evaluation

Both models will be evaluated on precision, recall, accuracy and the F1 score. All these metrics are based on true positives, true negatives, false positives and false negatives. An example is given in Table 2 how these are calculated.

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

Table 2: Confusion matrix example

The following equations show how the metrics used in the evaluation of the models are calculated:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

The accuracy gives an overview of the percentage of discharge letters that were correctly classified. The precision shows the proportion of positive classifications that were correct while the recall shows the proportion of actual positives that were identified correctly. Both the precision and recall will be given for both current smoking patients and past smoking patients.

During this research, we have to consider what measure we are more interested in. While precision is better if false negatives are not too much of a problem, recall is better if they are. Considering this research, we have concluded that recall is more important than precision because wrongly classifying a patient could hypothetically have a big impact on the advice a doctor or other medical staff would give.

Furthermore, for both models a error analysis will be carried out. This can be of benefit for future research and gives us insight into why the models might classify certain discharge papers wrongly.

4 Data understanding and preparation

This section will describe the data used for this project according to the second phase of the CRISP-DM process. First, the process of data gathering will be discussed, after that the preprocessing procedures are explained and finally an overview of the data using descriptive statistics will be given.

4.1 Data gathering

The data that was made available by the HagaZiekenhuis in The Hague consisted of a big chunk of discharge papers from different departments within the hospital. The range of the data was set to be from 2015 to 2020. The data was gathered by using CTCue which is a platform that enables healthcare facilities to easily access their data through a self-serve platform. Considering it was a challenging task to retrieve the right documents through CTCue, a keyword had to be specified to string match on. The string that was chosen was the word "roken" which means smoking in Dutch. The string was matched on the field which represented the discharge summary.

However, there were no labels available and thus an approach had to be taken to label the data. As we want to apply string matching to the discharge letters it would be an option to manually review the discharge letters to find the labels. However, this approach would be too time-consuming for the scope of the project. Another way would be to find patterns in the text and apply string matching to determine the labels, but this would eliminate the baseline. To retrieve the labels for this dataset we looked at other data that was available. Luckily the Haga hospital also had an extensive record of filled in information forms about patients. The answers to the forms that contained information about whether the patient used to smoke or not were used to extract the labels about the discharge letters.

The answers, however, were not in a clear label format, but rather in short sentences explaining what the smoking history of the patient is. Some examples are given in Table 3. The examples show that there is not one clear pattern to assign labels on. The approach to find all the patterns and their corresponding labels was to manually review at least 200 for answers. After manually reviewing them and writing down the patterns and the assigned labels, we continued searching through the answers forms until no new patterns are noticed for at least 100 consecutive forms. A Python script was written to show the forms in random order. In total 386 answers forms were manually reviewed.

A few examples of the rules that were found and applied to the form answer data can be found in Table 4. To verify that these labels were accurate enough we compared the results from applying the automatically assigned labels to a small sub-sample of 200 manually reviewed form answers, which were again chosen at random. This resulted in an accuracy score of 85%. In an ideal situation, we would manually review all the data, unfortunately, there was not enough time to manually label all the data.

Form answer	Label
al lang roken, half pakje tot pakje per dag	Smokes
gestopt sinds opname	Used to smoke
Eerst 20 per dag, nu ongeveer 5 per dag	Smokes
in de jaren 70 gestopt	Used to smoke
circa 10/dag	Smokes
tussendoor 10 jaar gestopt en daarna weer begonnen.	Smokes
gestopt sinds voorjaar 2019 , rookte van 15e tot 75e 5sig per dag	Used to smoke

Table 3: Examples of forms answers about smoking status of patient and the desired label

Query	Label
(“pack” OR “py” OR “pd” OR “pakje”) AND NOT (“voorheen” OR “gestopt” OR “roken”)	Current smoker
(“af en toe” OR “sporadisch” OR “ per ”) AND NOT (“voorheen” OR “gestopt” OR “roken”	Current smoker
“gestopt” AND “was”	Current smoker
“stoppen”	Current smoker
“minder” OR “voorheen meer”	Current smoker
“rookt niet meer” OR “voorheen” OR “sinds opname”	Past smoker
“gestopt” AND NOT “was”	Past smoker
(“pack” OR “py” OR “pd” OR “pakje”) AND (“voorheen” OR “gestopt” OR “roken”) AND NOT “was”	Past smoker
“rookte” AND NOT “stoppen”	Past smoker
“gerookt” AND NOT “niet”	Past smoker

Table 4: Example queries for labeling form answers

Furthermore, the discharge summaries contained sensitive information about the patients. Therefore names of patients, doctors, and other medical staff were removed from the text and replaced by generic terms before delivering the data. This process was very time-consuming as it had to be done manually by staff from the HagaZiekenhuis.

4.2 Data preparation

Both discharge summaries and form answers that are now labeled are linked on pseudo patient numbers. These numbers are connected in a different file. Unfortunately, we did not have information for all patients. After matching the form answer labels to the discharge paper we ended up with the final dataset.

To prepare the data, a few preprocessing steps were taken. First, the sentences in the letters were not in a clear format, see Table 5 for some examples. The examples show that there is no set punctuation structure like dots to mark

the end of a sentence. A mixture of dots enters and forward slashes are used to mark the end of the sentences. Therefore sentences would be split on enters first. This resulted in sentences that did not follow the traditional sentence format, but, as can be seen in the examples, the written text in a discharge letter does not require the traditional format when describing intoxications, patient history, or summarization of medication. Next, we decided to split a sentence on a dot if the sentence is followed by a space. This was done to prevent splitting sentences where numbers are presented from splitting in the middle of a number, for example "ABG: pH 7.47, pCO2 4.6, bic 26, BE 1.1, pO2 10.8, O2sat 97%", which would otherwise be split in four sentences.

The next step in preprocessing was replacing all the uppercase letters with lowercase letters using the lower() function in Python. This was applied to reduce the number of unique tokens and minimize the number of rules needed for string matching. The final step in the preprocessing process is removing all empty sentences. The empty sentences are removed because they are unnecessary and do not provide any information. We decided against stopword removal from the text data as for the methods used during this research these stopwords might be necessary to the context of the sentences. Therefore, we do end up with a larger corpus of words, but this might increase the understanding of the context by the BERT model. Furthermore, removing stopwords during research by Qiao et al. [48] had no improved effect. Additionally, they found that stopwords gained as much attention from the model as non-stopwords. This confirmed that keeping stopwords in the summaries would presumably not influence the performance of the models negatively.

4.3 Exploratory data analysis

This section describes some descriptive statistics about the data. These statistics were calculated using Python. The analysis used the raw data without the data preparation. To get a better understanding of the data, several statistics about the data were calculated and are described in Table 6. The following observations can be made from Table 6. In total, we have 6560 discharge summaries available of which 3861 summaries from patients who are currently (at the time of writing the summary) smoking and 2699 who used to smoke. This is a 58/42 distribution which is fairly balanced.

From Table 6 we can also see that we have the discharge summaries of 480 patients, which means we have an average of 14 summaries per patient at different moments in time. This can be explained by the fact that a doctor might write multiple discharge summaries or that different departments write multiple summaries per patient. Of the 480 patients, 254 are smoking at the time of writing the summary, and 226 are past smokers.

Next we will have a closer look at the data on sentence and token level. In total, when splitting the letters like described in Section 4.2, we have 221,349 sentences which means an average of 34 sentences per discharge summary. The complete distribution can be found in Figure 7. We can see that the

Discharge letter excerpt	Label
In verleden analyse A'hove vanwege benauwdheid, 'pufjes' gekregen (klinkt als bronchiale hyperreactiviteit). Kan niet duidelijk aangeven of klachten hetzelfde waren als toen. Is om onduidelijke redenen gestopt met inhalatiemedicatie. Geen astma als kind Intoxicaties: Roken: niet meer, Alcohol: -. Sociale anamnese: werkt als gevangenisbewaarder, sinds 7 weken ziektewet vanwege nekklachten Familieanamnese: HVZ- DM- HC- HT- Astma/COPD- Tante COPD bij fors roken.	Past Smoker
Voorgeschiedenis: Hypertensie, vitamine B12 deficiëntie, migraine Medicatie: metoprolol tablet 50mg ranitidine bruistablet 300mgm macrogol 300 Vitamine b Allergieën: Lactose Intoxicaties: Roken+. Alcohol+, Drugs-	Current Smoker
A: (woont in [locatie] tezamen met begeleider op de poli) Gaat heel erg goed, geen hoesten. Geen relevante dyspneu. Dit jaar geen exacerbaties gehad. Roken: 15 shag per dag. stoppen met roken niet goed haalbaar	Current Smoker
Geen benauwdheidsklachten meer. Geen hoestklachten. Geen exacerbatie gehad. is aan het minderen met roken, rookt nu 1 shaq/half uur! (voorheen continu)	Current Smoker
[leeftijd]-jarige patient met een cardiale voorgeschiedenis presenteert zich met per toeval ontdekte infiltratieve afwijkingen op cardiale beeldvorming - bilateraal infiltratieve afwijkingen met enige groundglass DD infectieus, (COVID?), dd cardiaal - volledig klachtenvrij, geen pulmonale klachten - blanco pulmonale voorgeschiedenis - gestopt met roken - supranormale longfunctie - Bij herhaling van beeldvorming geringe afname van groundglass, echter geen normalisatie. Atypisch beeld - Serologie SARS CoV-2 negatief	Past Smoker

Table 5: Example raw text excerpts from discharge papers to demonstrate the structure

Total number of discharge letters	6560
Average number of discharge letters per patient	13.67
Currently smoking patients discharge letters	3861
Past smoking patients discharge letters	2699
Total number of patients	480
Currently smoking patients	254
Past smoking patients	226
Number of sentences	221,349
Average number of sentences per discharge letter	33.74
Number of tokens/words	2,651,460
Number of unique tokens/words	115,645
Average token/word length	6.01
Average unique token/word length	9.42
Average sentence length	85.14

Table 6: Data statistics

majority of the discharge summaries is below 50 sentences, however there is still a good proportion that contain more than 50 sentences per discharge summary. This indicates that a noticeable portion of the discharge summaries has more information per summary.

For the token level analysis, we can see in Table 6 that there are 2,651,460 tokens in total included in the dataset of which 115,645 unique tokens. On average we see that a token is 9 characters long. To get a deeper understanding of the token distribution we created two histograms depicting the frequency. In Figure 8 shows the top 25 most occurring words when we include stopwords such as: 'en', 'de', 'met', 'van', 'in', 'bij', 'op', 'per', 'een', 'geen' and 'niet'. However, because this does not give a good overview of the non-stop words we also created Figure 9 which shows the top 25 frequent words which are not stopwords. This shows more informative and context-related words. We removed the numbers for both frequencies, as numbers did not give us any new information and a lot of medication descriptions contained numbers. We for example see that words that are related to medicines such as "oraal", "tablet", "stuk", and "mg" occur quite often. This can indicate that discharge summaries often include a description of the medication a patient takes.

Other words we can see that are used often are "dag", which is the most occurring word in this dataset, but also "patiënt", "klachten", "pain" and "onderzoek" which all make sense to occur in a discharge summary, indicating that patients, their pain, and symptoms are discussed as well as the research that has been done.

Another notable pattern is that some words are followed by a ":" indicating that a list or summary is about to follow. This can be seen with the words "conclusie", "anamnese" and "beleid". These words however did also occur without the ":" but less often. With the word "conclusie" this was 2623, "anamnese" 2991 and "beleid" 2264. Because the data analysis done above was carried

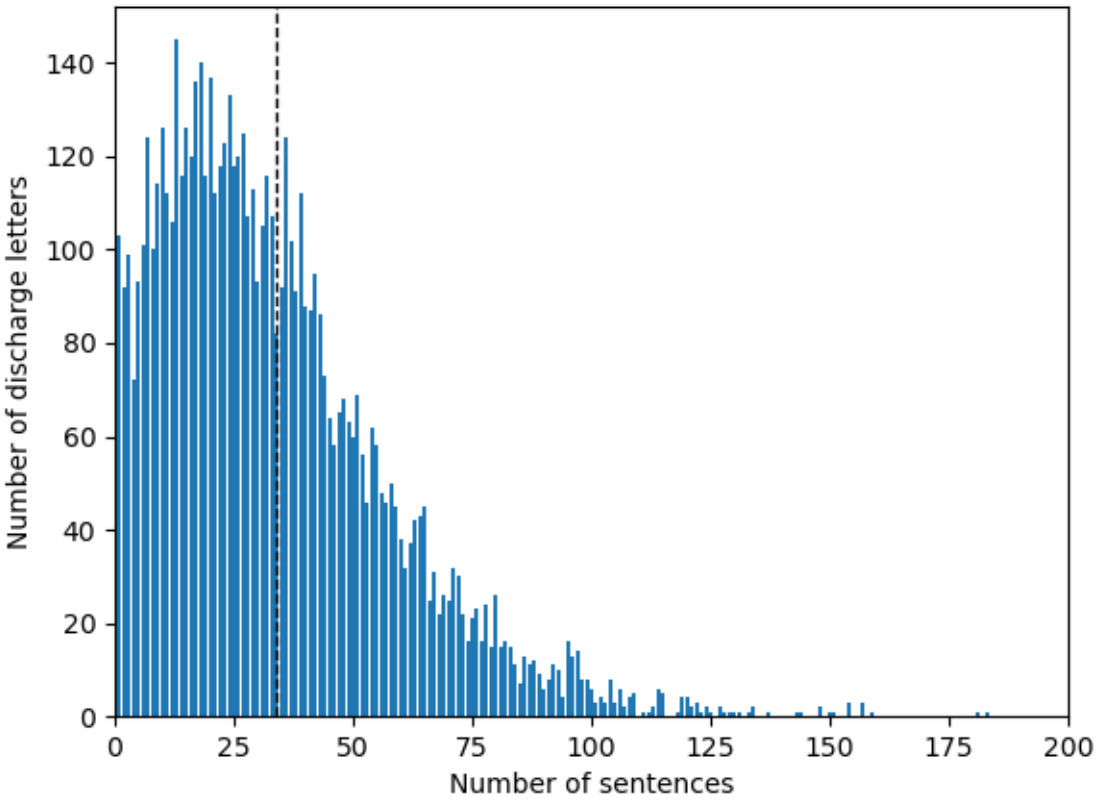


Figure 7: Distribution of number of sentences per discharge summary

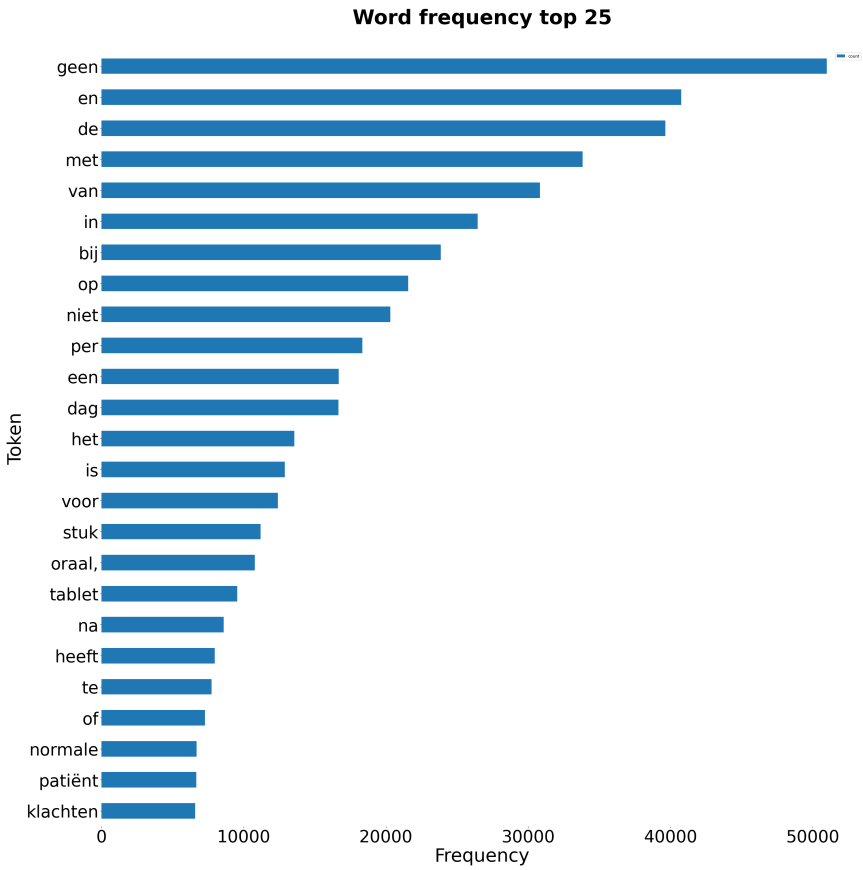


Figure 8: Top 25 word frequencies with stopwords

out on the raw data, it shows the importance of removing non-alphanumeric characters.

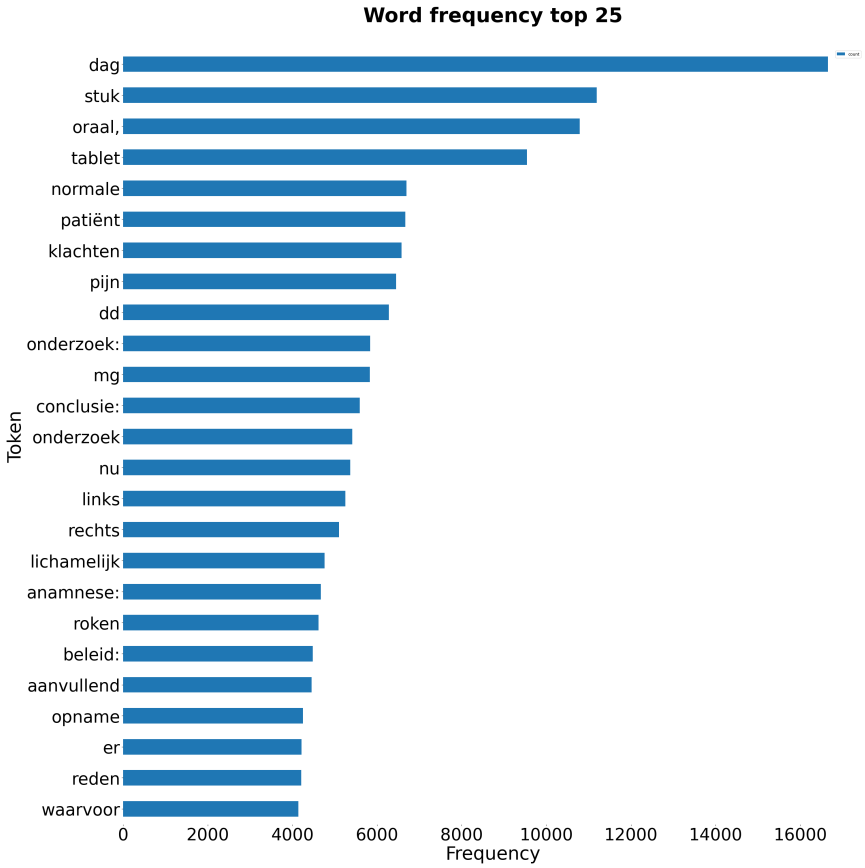


Figure 9: Top 25 word frequencies without stopwords

5 Methods

This section will go over the methods that are used to answer the research questions. Section 5.1 will cover the method that is to be used to create the string matching model. Sections 5.2 and 5.3 will explain how the selected BERT model is pretrained and how the sentence embeddings are extracted. Section 5.4 discusses which neural network structures are chosen and how they are implemented.

5.1 String matching

As described in Section 2.2.1 string matching is the process of searching for exact or approximate strings in the source text. In case of this research, we have chosen to apply exact string matching because literature research showed that this method works very well for structured term matching. Since we only have a limited amount of terms we want to match on, this can best be executed by exact string matching.

Before the implementation step was executed, we first had to search the data for patterns to match. One obvious first step was to find the word "roken" as this term was used to select the discharge letters from CTCue. After this was done, 10% (650) of the discharge letters were randomly selected to manually review. At this stage the data was not yet split into a train and test set as we wanted to include as much data as possible. However, this does make train/test set leakage an issue. This means it has to be kept in mind that the results of the string matching approach might a more optimistic estimation than a concrete result. The manual review procedure was executed by looking at the content of the letter one by one centered on the word "roken", we kept track of the range in which a pattern was visible and what the pattern was. If a pattern was already recorded but with a different range, the largest of the two was eventually used for the string matching procedure. The manual reviewing could stop after not noticing any new patterns for 100 consecutive letters. In total 484 letters were manually reviewed to find the patterns. A pattern is included in the final list of queries if it occurs more than (10% of 484 = 4,84) 5 times in the reviewed summaries. In doing so we prevent that the collection of queries are too specific and might be biased. The discharge summaries that were manually expected were then removed from the dataset on which the queries were applied. This resulted in a test set of 5910 discharge summaries.

The string matching procedure was coded in Python. The first step was to import the preprocessed CSV data file into a pandas dataframe. Next the word "roken" had to be found in the text, the index of the first letter of the word is saved in a variable. Next, we apply the queries to different range sizes to find the best performing range. The range that will be tried is based on the ranges that were found while manually reviewing the letters. The ranges varied from just 1 token to almost 100 tokens. To prevent trying each queries unique range we decided to round up every range to either 0 or 5, thus 6 becomes 10, and

3 becomes 5. The final ranges that will be tried can be found in Table 7.

Range	Range
10/10	10/20
20/20	40/20
40/40	20/40
60/60	10/40
80/80	20/80
100/100	20/100

Table 7: Ranges for sentence selection string matching on character level, represents number of characters before "roken" / number of tokens after the words "roken"

The queries will be applied to the range by first selecting the range as a new string and then using an if statement to see if the pattern is present within the string.

5.2 Pretraining BERT

The next method that was applied involved sentence embeddings created by using a BERT model. The BERT model had to be a Dutch pre-trained model since we are using Dutch data. The BERT model that was selected was BERTje as it showed the best performance for the majority of the tasks that it was evaluated on, compared to other Dutch models. Since we did not have enough labeled data to fine-tune the BERTje model on our task we opted to use the discharge letters that we didn't include in the final dataset as input for extra pretraining of the BERTje model to increase the domain-specific knowledge of the model. The pretraining was done by using Next Sentence Prediction and Masked Language Modelling to train BERTje on this data.

In order to execute both of the tasks, the data had to be prepared. The data was split in half for the NSP task. One half of the summaries was used to generate sequential sentence pairs. In case of the other half, we chose two random sentences that were not sequential. A label was assigned to both types pairs, 1 if they were a pair of predecessors and successors and 0 if they were not. The MLM task required to mask 15% of tokens of the tokenized input with a new token. Using a masking array where 15% of the entries were masked as long as certain tokens (CLS (1), SEP (2), and PAD (0)) were not masked. These tokens are necessary for indicating the start, end, and padding of the sentence. After the preparation of the input, the model is set in training mode and trained for 2 epochs as recommended by the authors of BERTje.

The pretraining of the BERT model was done on a server from the LUMC which contained 4 CPU cores and 16 GB RAM. In total, the pretraining process took 50 hours to finish.

5.3 Sentence embeddings

As stated in Section 2.2.3 we first need to transform the text data into a numerical input to give as input for the neural network that we have chosen. To do this we used the pretrained BERTje model as described in the section above. Before the pre-trained BERT model can be used to extract the sentence embedding, the input data first has to be prepared. Each sentence needs an [CLS] and [SEP] token, thus these were added at the appropriate location. After this is applied, each sentence is split into tokens. If a sentence has more than 512 words, it is sliced so that just 512 words remain. These tokens are then mapped to their vocabulary indices. Then the input is fed to the BERT model to retrieve the outputs. The outputs contains an abundant amount of information about the model, however, we only want to extract the hidden states. The hidden states are then reshaped to a tensor which is grouped by tokens. After each sentence is represented by a tensor vector (grouped by token) we can obtain the sentence embeddings multiple ways.

We ended up with a ragged NumPy array as each letter contained a different number of sentences. To remove the raggedness from the NumPy arrays we applied padding to the array by padding each shortest discharge summary to match the longest discharge summary. However, doing this for the longest letter resulted in a dataset that was too big and couldn't be handled by the server. To accommodate this we looked at a distribution of the document length in several sentences (times the size of the sentence embedding), which can be found in Figure 7. We can see that there are a few outliers that were extremely long compared to the mean which is 33 sentences per discharge summary. A sentence embedding has a size of 752 (the number of hidden states per sentence), thus the average embedding size is 24,816 (752 * 33.74). These outliers were inspected manually and we found that they contained very long lists of medication at the end and some even attached a previous letter at the end. Thus we decided to slice the letters to match a length of 94,000 (125 * the embedding length of 752 per sentence) instead of padding all letters to the maximum length which is 205,824. The shorter embeddings were padded to a maximum size of 94,000 as well.

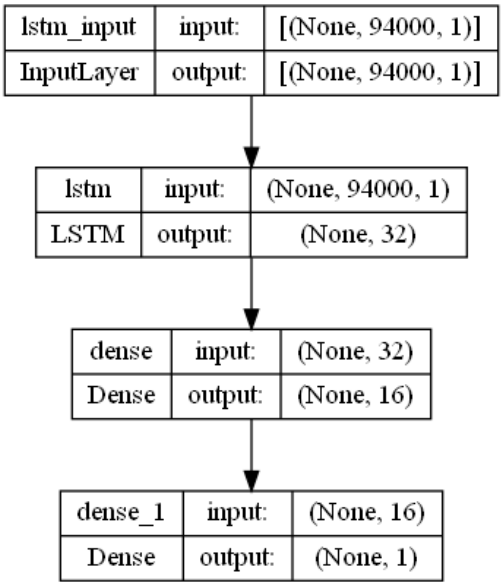
5.4 LSTM

Finally, to compare a deep learning model to our baseline string matching model, we selected the LSTM architecture with some variations and an architecture that combines a CNN and LSTM layer as explained in Section 2.

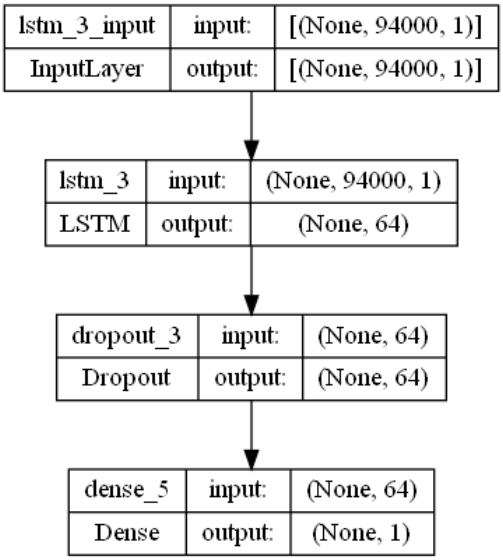
Four variants of the LSTM model were selected, of which the summary can be found in Figure 10a, 10b, 10c and 10d. The models are ordered in increasing complexity. The first model contains one LSTM layer, followed by two dense layers. The next two models introduce the dropout layer to see if that has a positive effect on training. The final model is based on research done by Zeghdaoui et al. [73] and combines a convolutional layer with an LSTM layer to achieve the best accuracy.

A second experiment with the Neural Networks was done by reducing the input text to just a small section of the text. The approach for selecting the range of this text is similar to the approach for selecting the range for the string matching approach. The word "roken" was found in the text and multiple ranges were selected. However, since training a neural network takes more time a limited range was tried. The ranges that were tried are 20/20 40/40 and 100/100. These parts of the text were treated as one sentence and their embeddings were retrieved with the same method as described above. Since we treated them as one sentence, no padding or slicing was necessary.

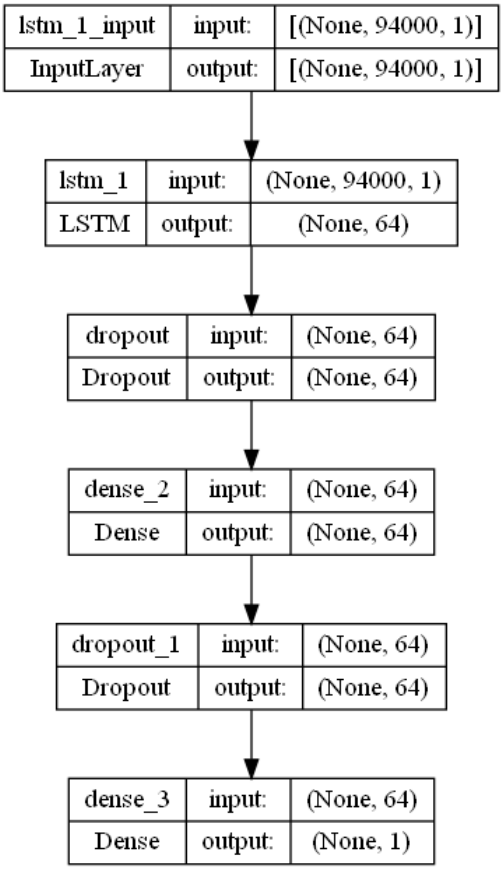
The combined runtime of all the Neural Networks that were each trained for 10 epochs on the LUMC server was 75 hours. All the programming was executed using Python version 3.9.5.



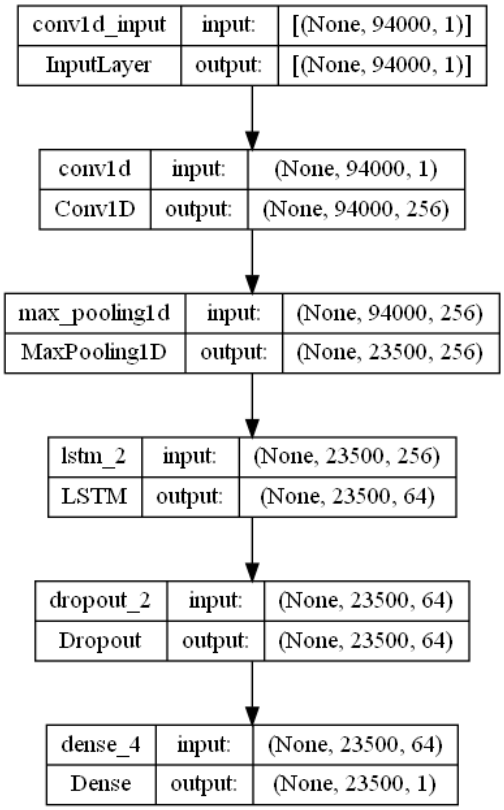
(a) Model 1: Simple LSTM



(b) Model 2: LSTM with additional dropout layer



(c) Model 3: LSTM with multiple dropout layers



(d) Model 4: CNN + LSTM combination

Figure 10: Overview of the different NN architectures

6 Results

This section shows the results of both the string matching approach as well as the neural network approach. A qualitative analysis of the sentence embedding is also performed. Both types of approaches need a manual review of the wrongly classified discharge summaries. This is included to see if there are any patterns to be discovered which can aid in future research.

6.1 String matching

The first approach we performed was the string matching approach. The method is applied as described in Section 5. The unique patterns that were discovered after inspecting the discharge summaries manually can be found in Table 8. The queries described in this table also include the work "roken" to be present in the string, as described in Section 5. Most of the rules include words that specifically indicate the status such as "stoppen", "gestaakt" and "verminderen" but also words that indicate a time span are often included such as "jaar", "per dag" and "tot".

Query	Label
"roken+" OR "roken +"	current smoker
"door" AND "stoppen"	past smoker
"gevolg" AND "stoppen"	past smoker
"gestopt" OR "gestaakt" AND NOT ("niet" OR "was")	past smoker
"tot" AND "jaar"	past smoker
"roken-" OR "roken -"	past smoker
"stoppen" OR "staken" OR "verminderen" OR "persisterend "	current smoker
"per dag" OR "packyears" OR "/dag" OR "pakje" OR "pack"	current smoker
"pd" OR "py"	current smoker

Table 8: List of queries and their labels for exact string matching. In this order the queries were also performed. All of the queries also require the word "roken" to be present.

These queries were able to cover 5381 discharge summaries of the 6560 in total. This means that 1179 were not able to be labeled. We discovered that of these 1179 it was either not clear if the patient smoked or not, was a sentence that only included "roken" or contained patterns that were not as frequent as the queries described above. In total we identified around 20 patterns that occurred more than 5 times in the 484 letters that were inspected. Other queries were not included since they occurred not frequent enough.

Range	Accuracy	Precision Current Smoker	Recall Current Smoker	F1-score Current Smoker	Precision Past Smoker	Recall Past Smoker	F1-score Past Smoker
10/10	0.60	0.60	**0.91**	**0.72**	**0.59**	0.08	0.14
20/20	0.61	0.63	0.78	0.70	0.55	0.30	0.39
40/40	**0.63**	0.66	0.72	0.69	0.55	0.30	0.39
60/60	**0.63**	0.66	0.71	0.69	0.57	0.45	**0.50**
80/80	**0.63**	0.66	0.71	0.69	0.56	**0.46**	**0.50**
100/100	**0.63**	**0.67**	0.70	0.69	0.56	**0.46**	**0.50**
10/20	0.61	0.62	0.86	**0.72**	**0.59**	0.18	0.28
40/20	0.61	0.63	0.78	0.70	0.56	0.30	0.39
20/40	**0.63**	0.66	0.72	0.69	0.56	0.42	0.48
10/40	**0.63**	0.64	0.80	0.71	**0.59**	0.31	0.41
20/80	**0.63**	**0.67**	0.70	0.69	0.56	0.45	**0.50**
60/80	**0.63**	**0.67**	0.70	0.69	0.56	0.45	**0.50**

Table 9: Exact string matching results

Table 9 shows the accuracy, precision, recall, and f1 scores on the different ranges. There are some noticeable patterns in these results. Accuracy scores are 0.63 for almost all scores except the 10/10, 20/20, 10/20, and 40/20 ranges. These results show that the most information might be gained at the end range and not the beginning range. The precision score for current smokers is better in the wider ranges. The recall on the other hand is the highest for the shortest range 10/10. As a result of the high recall on the shorter ranges, the f1 score is the highest for the 10/10 and 10/20 ranges. The past smokers class has much lower scores overall, indicating that the queries might be better at detecting current smokers than past smokers. The precision for the past smoker category is the highest for the smaller ranges, while the recall is higher for the larger ranges. On average the f1 score is lower for the past smoker category and higher for the larger ranges.

To illustrate the impact of the ranges, Table 10 shows some examples where the size of the range make the difference between certain labels. The first 4 rows show 2 different sentences with 2 different ranges. In these examples the more characters after the word "roken" caused that the prediction assigned the wrong label. However the last 4 rows show the opposite where the increase in characters after the word "roken" made sure that the predicted label was correct.

Range	Label	Prediction	Text
20/20	past smoker	current smoker	or te laten dringen, roken ook nagenoe
20/40	past smoker	past smoker	or te laten dringen, roken ook nagenoeg gestopt. gemaakte a
20/20	current smoker	current smoker	erder herseninfarct. roken migraine b
20/40	current smoker	past smoker	erder herseninfarct. roken migraine behandeling gestopt d
80/80	past smoker	past smoker	. actinische keratose. glaucoom ods cardiale risicofactoren: familie anamnese - roken reeds 25 jaar gestaakt hypertensie + hypercholesterolemie + diabetes mell
20/20	past smoker	current smoker	familie anamnese - roken reeds 25 jaar g
20/40	past smoker	past smoker	familie anamnese - roken reeds 25 jaar gestaakt hypertensie

Table 10: Examples to show the impact of ranges on the prediction label

6.2 Neural networks

The four models that are tested for this research are described in Section 5.

	Train Accuracy	**Test Accuracy**	**Test Recall Past Smoker**	**Test Precision Past Smoker**	**Test F1 Past Smoker**
Model 1	0.54	0.37	**1.00**	0.38	0.55
Model 2	0.58	0.37	**1.00**	0.38	0.55
Model 3	0.59	0.38	0.90	0.36	0.53
Model 4	**0.61**	**0.45**	0.95	**0.40**	**0.56**

	Test Recall Current Smoker	**Test Precision Current Smoker**	**Test F1 Current Smoker**
Model 1	0.00	0.00	0.00
Model 2	0.00	0.00	0.00
Model 3	0.10	0.06	0.08
Model 4	**0.12**	**0.10**	**0.11**

Table 11: Results of different LSTM architectures

Unfortunately, the results show that the neural networks are not performing as well as expected. Looking at the recall and precision values for both current and past smokers suggests that most of the discharge summaries are being classified as past smokers. We find that while the training accuracy for model 4 came close to the best test accuracy of the string matching results, the test accuracy for all models showed that all four models underperformed compared to the string matching approach. It is noticeable that the recall for the past smoker is extremely high while the precision is much lower. Both precision and recall scores for current smokers were all 0 or slightly above 0.

To get a better understanding of why the neural networks behave this way, error analysis was needed. The first step was reducing the input data. As we are using concatenated sentence embedding, which in itself is already quite long, the number of embeddings per summary is very large. To reduce it we will get a range of several words and treat this range as one sentence.

To see if the input data is the problem we will take the same approach as used for the string matching approach. The word "roken" is searched in the text and a range of 20 characters before and 20 characters after this word. Then we create sentence embeddings in the same way as described before, by taking the average of the word embeddings. These embeddings are then used as input for the same neural networks. The results can be found in Table 12. This yielded slightly better results but still under performed compared to the string matching approach. Using a smaller input dataset is therefore not helping the performance of the neural approaches.

	Train Accuracy	Test Accuracy	Test Recall Past Smoker	Test Precision Past Smoker	Test F1 Past Smoker
Model 1	0.58	0.37	**1.00**	0.38	0.55
Model 2	0.61	0.39	**0.95**	0.42	0.56
Model 3	0.58	0.37	0.90	0.38	0.53
Model 4	**0.63**	**0.47**	0.95	**0.45**	**0.61**

	Test Recall Current Smoker	Test Precision Current Smoker	Test F1 Current Smoker
Model 1	0.00	0.00	0.00
Model 2	0.04	0.02	0.02
Model 3	0.06	0.11	0.08
Model 4	**0.10**	**0.14**	**0.12**

Table 12: Results of different LSTM architectures on a smaller input data

6.3 Qualitative sentence embedding analysis

Since both approaches to the neural network did not perform well, a qualitative analysis of the sentence embeddings was executed to discover if they are able to capture the context of the sentences in the dataset well. Two tests were applied to the dataset as described by Tawfik et al [60]: a general knowledge and concept identity test. The general knowledge test attempts to investigate the robustness of the embeddings to reflect common sense. This is achieved by removing stop words and another non-important tokens, followed by calculating the cosine similarity of both the original sentence and the modified sentence. The higher the similarity the better the embedding captures the important information of the sentence. 10 sentences were selected from the dataset of which 5 have smoking in them and 5 don't. The results can be found in Table 13.

As we can see for all sentences the similarity score is above 0.94, this indicates that the model can capture the important information of the sentence well and that the removal of stop words might not be a necessary step.

Original Sentence	Modified Sentence	Similarity
intoxicaties: sinds aug gestopt met roken	intoxicaties: sinds aug gestopt roken	0.99
refluxklachten aanhoudend, mede bij roken en alcohol. nu alleen nog klachten door het roken en bij veel koffie en stress	refluxklachten aanhoudend, roken alcohol. alleen klachten roken veel koffie stress.	0.95
wil niet praten over stoppen met roken, wil iets hebben hiervoor.	wil niet praten stoppen roken, wil iets hebben hiervoor.	0.98
tevens was er eenmalig sprake van rood rectaal bloedverlies, geduid als anorectaal bij obstipatie. klinisch knapte patiënt op.	was eenmalig sprake rood rectaal bloedverlies, geduid anorectaal obstipatie. klinisch knapte patiënt op.	0.99
ventrale zijde bovenbenen naar zowel laterodorsale zijde onderbenen met ook mediale zijde	ventrale zijde bovenbenen zowel laterodorsale zijde onderbenen mediale zijde	0.98
op dat moment geen andere neurologische uitvalsverschijnselen en op de monitor geen ritmestoornis	moment geen andere neurologische uitvalsverschijnselenmonitor geen ritmestoornis	0.95
beloop bovengenoemde patiënte werd op 09-12-2021 opgenomen in verband met passagère woordvindstoornissen, meest waarschijnlijk door tia in het acm-stroomgebied links.	beloop bovengenoemde patiënte werd 09-12-2021 opgenomen verband passagère woordvindstoornissen, meest waarschijnlijk tia acm-stroomgebied links.	0.98
was zelf verbaasd dat het haar niet zo interesseerde, wel vermoeiende dag gehad waardoor mogelijk te verklaren	was zelf verbaasd haar niet zo interesseerde, wel vermoeiende dag gehad waardoor mogelijk verklaren.	0.98
roken gestopt 26 jaar geleden, daarvoor 35 jaar 1 pakje per dag	roken gestopt 26 jaar geleden, daarvoor 35 jaar 1 pakje dag	0.99
intoxicaties de patiënt rookte en is gestopt met roken	intoxicaties patiënt rookte is gestopt roken	0.95

Table 13: General knowledge test results

The second test is the concept identity test and tries to measure to what extent the model is able to encode the meaning of for example abbreviations in the sentence. To do so we again select 10 different sentences with abbreviations and modify them in such a way that the abbreviations are expanded to their original meaning. We then compare the output for both sentences from our best performing neural network to see if this might influence the performance. The results can be observed in Table 14.

The results show that expanding the abbreviations did not have a huge impact, only for one of the 10 sentences we tried, the classification changes to the true label.

Original Sentence	Modified Sentence	Original Label	Modified Label	True Label Label
opmerkingen mbt roken: kleine sigaren .	opmerkingen met betrekking tot roken: kleine sigaren .	PS	PS	CS
intoxicaties: roken 8-10 kleine sigaren/dag, alcohol 2 eh/dag, drugs geen.	intoxicaties: roken 8-10 kleine sigaren per dag, alcohol 2 eenheden per dag, drugs geen.	PS	PS	CS
intoxicaties roken+ (3 pakjes/week, voorheen 1 pakje/dag)	intoxicaties roken+ (3 pakjes per week, voorheen 1 pakje per dag)	PS	PS	CS
intoxicaties: roken 40 py, alcohol nooit meer, geen drugs	intoxicaties: roken 40 packyears, alcohol nooit meer, geen drugs	**PS**	**CS**	**CS**
intox: roken: 50 py, 25 jaar gestopt.	intox: roken: 50 packyears, 25 jaar gestopt.	PS	PS	PS
roken: >20jr geleden gestopt.	roken: meer dan 20jaar geleden gestopt.	PS	PS	PS
patient rookte 20 pd	patient rookte 20 packdays	PS	PS	PS
roken 3-4 sig/dag, voorheen tot 2 pakjes per dag, 52 packyears, nu ook e-smoker.	roken 3 tot 4 sig per dag, voorheen tot 2 pakjes per dag, 52 packyears, nu ook e-smoker.	PS	PS	CS
sinds een week gestopt met roken na 52 py	sinds een week gestopt met roken na 52 packyears	PS	PS	PS
roken: 60jr geleden begonnen	roken: 60 jaar geleden begonnen	CS	CS	CS

Table 14: Concept Identity Test results, where PS = Past Smoker and CS = Current Smoker

A final test to check the quality of the embeddings is to see what the similarity scores are for sentences pairs that are labeled as past smoker and current smoker. Again, 10 random sentences are selected, 5 labeled as past smokers and 5 labeled as current smokers which are given in Table 15. The cosine similarity measure is used to see how similar the sentences are. The

results are described in Table 16.

Sentence ID	Sentence	Label
1	roken gestopt sinds 32 jaar, diabetes mellitus, bmi 26	Past Smoker
2	Intoxicaties: roken- alk niet meer, daarvoor weinig	Past Smoker
3	verslag gesprek met patiënt: gaat beter. minder dyspnoisch, waarschijnlijk toch door stoppen met roken.	Past Smoker
4	rookt niet meer	Past Smoker
5	roken–	Past Smoker
6	roken++	Current Smoker
7	met haar werd nog eens nadrukkelijk gesproken over het staken van het roken	Current Smoker
8	intoxicatie: roken: ja	Current Smoker
9	stoppen met roken nog niet gelukt, het is wel minder maar nog niet geheel gestopt.	Current Smoker
10	roken: 1 pakje/dag, alcohol: enkele per jaar	Current Smoker

Table 15: Sentences to calculate similarity past smoker/current smoker pair

Sentence Pair: Past Smoker/ Current Smoker	Similarity	Sentence Pair: Past Smoker/ Past Smoker	Similarity
1 & 10	0.89	1 & 3	0.91
2 & 8	0.91	1 & 2	0.91
3 & 7	0.85	4 & 5	0.78
4 & 9	0.79		
5 & 6	0.89		
Sentence Pair: Current Smoker/ Current Smoker	**Similarity**		
1 & 2	0.86		
3 & 4	0.87		
2 & 5	0.82		

Table 16: Similarity between pairs of past and current smokers

The results show that the BERT model, that was pretrained on the medical data, produces embeddings that are very similar if compared to the results of past and current smokers. The lowest similarity is 0.79 and is scored between sentence 4 and 9, while the highest similarity is 0.91 and is scored between pair 2 8. However, we see that if we match past smokers with past smokers and current smokers with current smoker we achieve very similar results.

To verify that this is not a coincidence, the same test was done on a larger scale. 100 sentences from past smoking patients were selected and 100 sentences from current smoking patients were selected. The sentences were selected at random and a full overview of the sentences can be found in the appendix. Instead on manually making pairs, each sentence was compared to all the other sentences in the other group. Thus for comparing the similarity of past smokers with current smoker, past smokers with past smokers and current smokers with current smokers, there were 10,000 combinations for each group. The similarities were averaged and the following results were found: for past smokers and current smokers pair an average similarity of 0.89 was found, for past smokers and past smokers an average of 0.94 was found and for current smokers and current smokers an average similarity of 0.88 was found. This is similar to the the results showed in Table 16, although the average score for the larger test set were slightly higher than the smaller test set. It is still noticeable that the combination of past smokers and past smokers is higher than the past smoker and current smokers pairs and also higher than the current smokers and current smokers pairs. This reinforces the suggestion that the observed results in Table 16 are not a coincidence.

7 Discussion

This section contains a discussion of the results, limitations of the research, and suggestions for future research.

7.1 Interpretation of results

The results from both approaches show that using a string matching approach yields better results than using the neural approach. This suggests that using the selected neural network architectures might not be best suited for this data and this problem. The string matching approach showed that for classifying a patient as a current smoker a smaller range of characters is more suited while classifying someone as a past smoker, a larger range of characters is more suited. Often it was noticeable that words indicating that a person is currently smoking are closer to the word "roken" in the sentence than words indicating that a person used to smoke.

Furthermore, the sentence embedding did not seem to capture the differences between current smokers and past smokers very well. The qualitative sentence embedding showed that sentence pairs containing one sentence indicating a current smoker and one indicating a past smoker still yielded high similarity scores. The similarities were slightly higher for a sentence pair that contained two sentences indicating past smokers than for the mixed sentence pairs. But the embeddings were not able to capture the similarity between two sentences indicating a current smoker as well as a past smoker. This can be one of the causes that the neural approaches tended to classify all discharge summaries as past smokers even though the distribution of the labels was relatively balanced.

Removing stop words did not seem to affect the similarity between the original sentences and modified sentences by a lot, indicating that the embeddings do capture the important information quite well without paying too much attention to the stop words. Expanding the abbreviations had barely an effect on the number of correctly classified instances. Of the ten examples that were tried only one example changed in the classification label. These results showed that expanding the abbreviations did not contribute to a better classification using the neural network.

7.2 Limitations

This research had several limitations and they are all linked to the data and the process of data gathering and data labeling. Due to the nature of the data, it might be that different discharge summaries of the same patient contradict each other. While a patient might say to one doctor that they don't smoke they can tell another doctor that they used to smoke. This causes the different summaries to give different information regarding the smoking status of the same patient. Because we have labels per patient, and not per discharge summary,

it might be that a discharge summary mentions that a patient doesn't smoke but did get the label that they are currently smoking since the information that the label is based on did indicate this.

Another limitation based on the nature of this data is that even if the patient is honest, there might be a time difference between discharge summaries. Some discharge summaries might be taken months apart indicating that the first time the summary was written the patient still smoked, while in the later summary the patient quit smoking. It then depends on when the form on which the label is based was filled in and which label both summaries get.

This also brings us to the process of labeling data. Because we had a very large set of data, due to time limitations it was not possible to review each summary by hand. To solve this we acquired answers to the forms that were filled in which contained very condensed sentences indicating if the person smokes or used to smoke, as seen in Table 3. These labels were then used to label the patient and their discharge summaries.

A final limitation based on the data is the fact that the case study we presented in this research is based on a very nuanced difference between meanings. We would have preferred to have labeled data of nonsmoking patients so that the performance of classifying between nonsmoking, past smoking, and current smoking patients could be compared. However, this was not possible since there were no labels regarding nonsmoking patients.

7.3 Future research

This research provided a base for future research and in this section some suggestions for future research are given. First of all, for future research, we suggest using a dataset that is manually labeled for each discharge summary as this will increase the accuracy of the labels. We also suggest this labeling to be done by medical professionals as they have the domain knowledge to make better sense of the content of the discharge summaries. Getting the labels from multiple professionals also allows us to compute Cohen's kappa, which is a statistic that presents the inter-rater agreement, and therefore get an even better insight into the accuracy of the labels.

Next we would suggest getting a dataset that can be used for finetuning the BERT model as opposed to pre-training the BERT model. This allows the BERT model to get a better understanding of the task without needing the computational time and resources that are needed to pretrain the BERT model. During this research, we choose to pretrain BERT on an unlabeled dataset since the labeled dataset we acquired was already quite small. Finetuning the BERT models also provides BERT with a preview of the specific task as opposed to just using NSP and MLM which are more general tasks.

The exact string matching approach could be improved by adding more queries and rules. However, it is important to keep a stopping criterion in mind because otherwise the approach might be too specific to the dataset and might not be more widely or generally applicable anymore. To see if a neural

approach can be suitable for this task more research is needed. This can be done by either trying more diverse neural architectures or a combination of the string matching approach and neural architecture. Previous research on different tasks showed that this was promising. However, this was not implemented in this research since the neural network did not perform correctly. A combination of the string matching method and neural network could prove to be a powerful combination if a network architecture that performs well is found.

8 Conclusion

This section concludes this research and will answer all the research questions that were provided in Section 1. First, all the sub questions will be answered and finally, the main research question will be answered.

RQ1 What are the challenges in preprocessing and working with free text data from the medical domain?

This question was answered in Section 2.2 by performing a literature research. The main challenges in preprocessing free text in the medical domain are similar to working with free text in general. These challenges include spelling errors, abbreviations, and sparseness of the data. However, unlike regular text, the medical domain uses a lot of abbreviations to make it easier to write down notes. The abbreviations, however, are not always the same between hospitals, domains, or even doctors. This introduces an extra layer of difficulty since there are not any lists of these abbreviations and if there are they are not complete and might contain different meanings behind the abbreviations. This makes it also harder to distinguish abbreviations from spelling errors.

RQ2 How can the challenges in preprocessing free text data from the medical domain best be addressed?

Dealing with spelling errors can be done by using approximate string matching, also called fuzzy matching. This works by calculating a distance metric to see which word is most likely to be misspelled. Abbreviations are a bit more difficult to work with. As we first need to know if a word is an abbreviation or a spelling error. To do this multiple approaches are available, from creating a dictionary to recognizing them using NLP approaches. The next step for addressing the challenge of abbreviations is the disambiguation of the abbreviation. Again, a dictionary can be used, but recent research has also proven to be successful in using BERT models to disambiguate abbreviations.

To address the sparseness of the text data two techniques are commonly used: stemming and lemmatization. Both these techniques aim to bring words back to a simpler and more basic form, therefore reducing the number of words in the vocabulary. The difference between the two is that stemming reduces sparseness by removing the prefixes and suffixes from the word, resulting in basic forms that might not always exist. While lemmatization brings words back to their lemma or their dictionary form. Because both forms operate in different ways, depending on multiple factors such as the nature of the text, the presence of a dictionary and the importance of correctly spelled words, a well considered motivation needs to be made in order to choose one over the other.

RQ3 What are the current state of the art NLP techniques for extracting information from text data?

The current state-of-the-art techniques for extracting information from text data differ from task to task. Regular expression gets most of the time reasonable, if not the best, results. Proving that sometimes a simple technique is best suited for a task such as identifying outcomes and adverse drug reactions. However, the introduction of BERT research within the domain of information extraction from medical free text has seen an increase in research suggesting that BERT-like models might be even better at identifying ADRs but also extracting familial relations. Deep learning techniques also have improved a lot of the results achieved by classic machine learning techniques such as Support Vector Machines and Random Forests. However, neural networks have often the disadvantage that they are hard to interpret which is not always helpful for research. A more recent movement within research showed the combination of using word embeddings and neural networks to achieve a state of the art results.

RQ4 How does a string matching technique compare to a neural approach for classifying patients based on discharge summaries?

The results of both the string matching approach and the neural approach showed that the string matching technique outperformed the neural approach.The string matching was carried out on different sets of ranges, where a range is defined as a number of characters before the word "roken" and a number of characters after the word "roken". The string matching approach scored an accuracy of 63% on all ranges except the ranges 10/10, 20/20, 10/20, and 40/20. This indicated that most information is won after the word "roken". The combination of the LSTM and CNN network resulted in an accuracy of 45% on the test set. It was noticeable that the recall was very high for the past smoker category indicating that the network tended to assign the label past smoker to all discharge summaries. However, this might not be due to the neural approach itself. It can also be caused by the embeddings that are used as input for the neural approach, since they were not able to capture the differences between a past smoker and a current smoker adequately.

RQ5 To what extent is it possible to determine if a patient used to smoke or smokes currently based on their medical discharge summaries?

Based on this research it is hard to quantify to what extent we can classify if a patient used to smoke or smokes currently based on their medical discharge summary. However, we can see based on the results from the string matching approach that we were able to classify 63% of the summaries correctly. If we also take into account the limitations of this study there still are some potential methods to be discovered in future research. While this research did not provide the best solution to this task, the groundwork was laid discovering the challenges and providing a baseline using a string matching approach.

Which NLP techniques are best suited for recognizing the smoking status of a patient in discharge summaries?

The goal of this research was to find out which NLP techniques were best suited for recognizing the smoking status of a patient in discharge summaries. To study this we created a case study in which we tried to recognize and differentiate between past smokers and current smokers. Based on the literature study that was performed two different approaches were chosen: a string matching approach as a baseline and a more advanced neural network approach for which we selected different LSTM architectures and a combination of the LSTM and CNN structure. The string matching approach was chosen since it often performed reasonably well in related research. To improve the embedding we pretrained the BERTje model on part of our data set improving the accuracy of the information that the embeddings were able to capture. If we had more labeled data, finetuning on this specific task would be preferred.

Next, we found that unfortunately the neural approach did not perform as well as the string matching approach and tended to classify each instance as a past smoker. Performing a qualitative analysis on the sentence embeddings we found that the embeddings were still not able to capture the difference between past and current smokers very well. The embeddings did capture the past smoker's information better than the current smoker's information which might indicate why the results of the neural approach were more in favor of the past smoker. While this research was not able to give a clear indication which NLP techniques are best suited for recognizing the smoking status of patients in discharge summaries, it did reveal some additional challenges working with discharge summaries. In the future, it might be beneficial to explore combinations of string matching and neural network architectures.

This case study, although giving some beneficial insights, is not representative of all possible lifestyle factors. Because of the challenges we faced during this research, we were not able to create a pipeline draft that would be generally applicable to different lifestyles.

References

[1] Informatics for integrating biology the bedside. https://www.i2b2.org/, 2022.

[2] Nationale drug monitor editie 2022. https://www.nationaledrugmonitor.nl/tabak-sterfte/, 2022.

[3] Pubmed. https://pubmed.ncbi.nlm.nih.gov/, 2022.

[4] Who factsheet tobacco. https://www.who.int/news-room/fact-sheets/detail/tobacco, 2022.

[5] K. Adnan, R. Akbar, S. W. Khor, and A. B. A. Ali. Role and challenges of unstructured big data in healthcare. *Data Management, Analytics and Innovation*, pages 301–323, 2020.

[6] K. Barbour, D. C. Hesdorffer, N. Tian, E. G. Yozawitz, P. E. McGoldrick, S. Wolf, T. L. McDonough, A. Nelson, T. Loddenkemper, N. Basma, et al. Automated detection of sudden unexpected death in epilepsy risk factors in electronic medical records using natural language processing. *Epilepsia*, 60(6):1209–1220, 2019.

[7] I. Boban, A. Doko, and S. Gotovac. Sentence retrieval using stemming and lemmatization with different length of the queries. *Advances in Science, Technology and Engineering Systems*, 5(3):349–354, 2020.

[8] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

[9] A. Brandsen, A. Dirkson, S. Verberne, M. Sappelli, D. Manh Chu, and K. Stoutjesdijk. Bert-nl a set of language models pre-trained on the dutch sonar corpus. 2019.

[10] H. Cao, P. Stetson, and G. Hripcsak. Assessing explicit error reporting in the narrative electronic medical record using keyword searching. *Journal of Biomedical Informatics*, 36(1):99–105, 2003. Patient Safety.

[11] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, R. Wirth, et al. Crisp-dm 1.0: Step-by-step data mining guide. *SPSS inc*, 9(13):1–73, 2000.

[12] C. Clark, K. Good, L. Jezierny, M. Macpherson, B. Wilson, and U. Chajewska. Identifying smokers with a medical extraction system. *Journal of the American Medical Informatics Association*, 15(1):36–39, 2008.

[13] A. Cocos, A. G. Fiks, and A. J. Masino. Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in twitter posts. *Journal of the American Medical Informatics Association*, 24(4):813–821, 2017.

[14] A. M. Cohen. Five-way smoking status classification using text hot-spot identification and error-correcting output codes. *Journal of the American Medical Informatics Association*, 15(1):32–35, 2008.

[15] W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim. Bertje: A dutch BERT model. *arXiv preprint arXiv:1912.09582*, 2019.

[16] P. Delobelle, T. Winters, and B. Berendt. Robbert: a dutch roberta-based language model, 2020.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[19] C. Dreisbach, T. A. Koleck, P. E. Bourne, and S. Bakken. A systematic review of natural language processing and text mining of symptoms from electronic patient-authored text data. *International journal of medical informatics*, 125:37–46, 2019.

[20] W. Fan, L. Wallace, S. Rich, and Z. Zhang. Tapping the power of text mining. *Communications of the ACM*, 49:76–82, 09 2006.

[21] C. Foreman, W. Smith, G. Caughey, and S. Shakib. Categorization of adverse drug reactions in electronic health records. *Pharmacology Research Perspectives*, 8, 04 2020.

[22] A. Funkner, D. Zhurman, and S. Kovalchuk. *Extraction of Temporal Structures for Clinical Events in Unlabeled Free-Text Electronic Health Records in Russian*, volume 287. 11 2021.

[23] S. I. Hakak, A. Kamsin, P. Shivakumara, G. A. Gilkar, W. Z. Khan, and M. Imran. Exact string matching algorithms: Survey, issues, and future research directions. *IEEE access*, 7:69614–69637, 2019.

[24] H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, and M. R. Yeganegi. Text mining in big data analytics. *Big Data and Cognitive Computing*, 4(1):1, 2020.

[25] T. Hernandez-Boussard, S. Tamang, D. Blayney, J. Brooks, and N. Shah. New paradigms for patient-centered outcomes research in electronic medical records: an example of detecting urinary incontinence following prostatectomy. *eGEMs*, 4(3), 2016.

[26] A. Jaber and P. Martínez. Disambiguating clinical abbreviations using pre-trained word embeddings. In *HEALTHINF*, pages 501–508, 2021.

[27] A. Jaber and P. Martínez. Disambiguating clinical abbreviations using a one-fits-all classifier based on deep learning techniques. *Methods of Information in Medicine*, 2022.

[28] K. S. Kalyan and S. Sangeetha. Secnlp: A survey of embeddings in clinical natural language processing. *Journal of biomedical informatics*, 101:103323, 2020.

[29] D. Khyani, B. Siddhartha, N. Niveditha, and B. Divya. An interpretation of lemmatization and stemming in natural language processing. *Journal of University of Shanghai for Science and Technology*, 2021.

[30] B. J. King, A. L. Gilmore-Bykovskyi, R. A. Roiland, B. E. Polnaszek, B. J. Bowers, and A. J. H. Kind. The consequences of poor communication during transitions from hospital to skilled nursing facility: A qualitative study. *Journal of the American Geriatrics Society*, 61(7):1095–1102, 2013.

[31] W. Kraaij and R. Pohlmann. Porter's stemming algorithm for dutch. *Informatiewetenschap*, pages 167–180, 1994.

[32] M. Kushima, R. Matsuo, T. Ogawa, K. Araki, Y. Hasegawa, S. Nozue, E. Okazaki, and H. Koga. Development of patient information extraction method by sequence labeling using electronic medical records. In *2020 IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL)*, pages 105–110, 2020.

[33] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. Albert: A lite BERT for self-supervised learning of language representations, 2019.

[34] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[35] R. Leaman, R. Khare, and Z. Lu. Challenges in clinical natural language processing for automated disorder normalization. *Journal of Biomedical Informatics*, 57:28–37, 2015.

[36] R. Leaman, L. Wojtulewicz, R. Sullivan, A. Skariah, J. Yang, and G. Gonzalez. Towards internet-age pharmacovigilance: extracting adverse drug

reactions from user posts to health-related social networks. In *Proceedings of the 2010 workshop on biomedical natural language processing*, pages 117–125, 2010.

[37] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert.

[38] C. Lovis and R. H. Baud. Fast Exact String Pattern-matching Algorithms Adapted to the Characteristics of the Medical Language. *Journal of the American Medical Informatics Association*, 7(4):378–391, 07 2000.

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[40] Z. Min. Drugs reviews sentiment analysis using weakly supervised model. In *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pages 332–336. IEEE, 2019.

[41] N. Oostdijk, M. Reynaert, V. Hoste, and I. Schuurman. The construction of a 500-million-word reference corpus of contemporary written dutch. In *Essential speech and language technology for Dutch*, pages 219–247. Springer, Berlin, Heidelberg, 2013.

[42] N. Orangi-Fard, A. Akhbardeh, and H. Sagreiya. Predictive model for icu readmission based on discharge summaries using machine learning and natural language processing. *Informatics*, 9(1), 2022.

[43] R. Ordelman, F. de Jong, A. Van Hessen, and H. Hondorp. Twnc: a multifaceted dutch news corpus. *ELRA Newsletter*, 12(3/4):4–7, 2007.

[44] J. Patel, Z. Siddiqui, A. Krishnan, and T. Thyvalikakath. Identifying patients' smoking status from electronic dental records data. *Studies in Health Technology and Informatics*, 245:1281–1281, 2017.

[45] S. N. Payrovnaziri, L. Barrett, D. Bis, J. Bian, and Z. He. Enhancing prediction models for one-year mortality in patients with acute myocardial infarction and post myocardial infarction syndrome, 04 2019.

[46] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[47] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[48] Y. Qiao, C. Xiong, Z. Liu, and Z. Liu. Understanding the behaviors of BERT in ranking. *arXiv preprint arXiv:1904.07531*, 2019.

[49] S. Regan, J. B. Meigs, S. K. Grinspoon, and V. A. Triant. Determinants of smoking and quitting in hiv-infected individuals. *PLoS One*, 11(4):e0153103, 2016.

[50] P. Ruch, R. H. Baud, A. Geiddbühler, C. Lovis, A.-M. Rassinoux, and A. Riviere. Looking back or looking all around: comparing two spell checking strategies for documents edition in an electronic patient record. In *Proceedings of the AMIA Symposium*, page 568. American Medical Informatics Association, 2001.

[51] S. Saha. A comprehensive guide to convolutional neural networks, 2018.

[52] D. Saleheen, W. Zhao, R. Young, and C. P. N. et al. Loss of cardioprotective effects at the ¡i¿adamts7¡/i¿ locus as a result of gene-smoking interactions. *Circulation*, 135(24):2336–2353, 2017.

[53] G. Schmajuk and J. Yazdany. Leveraging the electronic health record to improve quality and safety in rheumatology. *Rheumatology International*, 37:1603–1610, 2017.

[54] B. Settles. Active learning literature survey. 2009.

[55] Z. Shen and M. Spruit. Automatic extraction of adverse drug reactions from summary of product characteristics. *Applied Sciences*, 11(6):2663, 2021.

[56] K. R. Siegersma, M. Evers, S. H. Bots, F. Groepenhoff, Y. Appelman, L. Hofstra, I. I. Tulevski, G. A. Somsen, H. M. den Ruijter, M. Spruit, et al. Development of a pipeline for adverse drug reaction identification in clinical notes: Word embedding models and string matching. *JMIR Medical Informatics*, 10(1):e31063, 2022.

[57] J. F. Silva, J. R. Almeida, and S. Matos. Extraction of family history information from clinical notes: deep learning and heuristics approach. *JMIR medical informatics*, 8(12):e22898, 2020.

[58] R. Talib, M. K. Hanif, S. Ayesha, and F. Fatima. Text mining: techniques, applications and issues. *International Journal of Advanced Computer Science and Applications*, 7(11), 2016.

[59] M. Tanwar, R. Duggal, and S. K. Khatri. Unravelling unstructured data: A wealth of information in big data. In *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)*, pages 1–6. IEEE, 2015.

[60] N. S. Tawfik and M. R. Spruit. Evaluating sentence representations for biomedical text: Methods and experimental results. *Journal of biomedical informatics*, 104:103396, 2020.

[61] K. Thompson. Programming techniques: Regular expression search algorithm. *Communications of the ACM*, 11(6):419–422, 1968.

[62] H. D. Tolentino, M. D. Matters, W. Walop, B. Law, W. Tong, F. Liu, P. Fontelo, K. Kohl, and D. C. Payne. A umls-based spell checker for natural language processing in vaccine safety. *BMC medical informatics and decision making*, 7(1):1–13, 2007.

[63] M. Unnewehr, B. Schaaf, R. Marev, J. Fitch, and H. Friederichs. Optimizing the quality of hospital discharge summaries–a systematic review and practical tools. *Postgraduate Medicine*, 127(6):630–639, 2015.

[64] R. van de Schoot, J. de Bruin, R. Schram, P. Zahedi, J. de Boer, F. Weijdema, B. Kramer, M. Huijts, M. Hoogerwerf, G. Ferdinands, et al. An open source machine learning framework for efficient and transparent systematic reviews. *Nature Machine Intelligence*, 3(2):125–133, 2021.

[65] M. van Haastrecht, I. Sarhan, B. Yigit Ozkan, M. Brinkhuis, and M. Spruit. Symbals: A systematic review methodology blending active learning and snowballing. *Frontiers in Research Metrics and Analytics*, 6, 2021.

[66] C. J. Van Rijsbergen, S. E. Robertson, and M. F. Porter. *New models in probabilistic information retrieval*, volume 5587. British Library Research and Development Department London, 1980.

[67] D. Wei, B. Wang, G. Lin, D. Liu, Z. Dong, H. Liu, and Y. Liu. Research on unstructured text data mining and fault classification based on rnn-lstm with malfunction inspection report. *Energies*, 10(3), 2017.

[68] J. Wimsett, A. Harper, and P. Jones. Review article: Components of a good quality discharge summary: A systematic review. *Emergency Medicine Australasia*, 26(5):430–438, 2014.

[69] C. Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, pages 1–10, 2014.

[70] Y. Wu, J. C. Denny, S. T. Rosenbloom, R. A. Miller, D. A. Giuse, and H. Xu. A comparative study of current clinical natural language processing systems on handling abbreviations in discharge summaries. In *AMIA annual symposium proceedings*, volume 2012, page 997. American Medical Informatics Association, 2012.

[71] H. Xu, P. D. Stetson, and C. Friedman. A study of abbreviations in clinical notes. In *AMIA annual symposium proceedings*, volume 2007, page 821. American Medical Informatics Association, 2007.

[72] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing, 2017.

[73] M. W. Zeghdaoui, O. Boussaid, F. Bentayeb, and F. Joly. Medical-based text classification using fasttext features and cnn-lstm model. In *International Conference on Database and Expert Systems Applications*, pages 155–167. Springer, 2021.

[74] Q. T. Zeng, S. Goryachev, S. Weiss, M. Sordo, S. N. Murphy, and R. Lazarus. Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system. *BMC medical informatics and decision making*, 6(1):1–9, 2006.

[75] X. Zhou, Y. Wang, S. Sohn, T. M. Therneau, H. Liu, and D. S. Knopman. Automatic extraction and assessment of lifestyle exposures for alzheimers disease using natural language processing. *International Journal of Medical Informatics*, 130:103943, 2019.

[76] L. Zilio, H. Saadany, P. Sharma, D. Kanojia, and C. Orasan. Plod: An abbreviation detection dataset for scientific documents. *arXiv preprint arXiv:2204.12061*, 2022.

Appendix

Table 17: Paper reference number sorted by topic, result of literature research

Method	Reference Number
RegEx	[6], [21], [22], [38], [13], [71], [53]
LSTM	[63], [69]
CNN	[38], [69]
RNN	[13], [63]
BERT	[13], [27]
Word2Vec	[43]
DNN	[43], [53]
SVM	[12], [14], [47]

Table 18: Sentences indicating current smokers used for the similarity scores

Sentence
risicofactoren roken, alcohol, hypertensie, diabetes
intoxicatie: alcohol- roken: half pakje per dag, 20py drugs allergien: - roken: ja: 2018 daginvulling: werk: internationaal chauffeur geweest.”
intoxicaties de patiŽnt rookt sinds 1965, (20 sigaretten per dag). aantal packyears: 55. opmerkingen mbt roken: is aan het minderen. de patiŽnt drinkt alcohol, minder dan 2 eenheden per dag.
intoxicaties: rookt 50 jaar meer dan 20 per dag, drinkt geen alcohol nooit gedaan.
intoxicaties: roken + (1 pakje per dag), alcohol -, drugs -. sociaal: huismoeder
rf: roken, medicatie/ prednison 5 mg atorvastatine 20 mg (gestopt, 4 weken nu)
patient werd gezien op onze tia-dagbehandeling i.v.m. een tia in de linkerhemisfeer dd 29-12-2007 bij dm, roken en hyperlipidemie.
beleid: laba/lama/ics pijnstilling ophogen vod 7-1 icc transfer 6-1 stop roken poli 6 weken met longfunctie
cardiale risicofactoren: roken + (45 pack years, afgelopen 4 jaar e-sigaret), alcohol -, ht -, fam +
intoxicaties: roken+, alcohol - lichamelijk onderzoek niet zieke of pijnlijke indruk. zit rechtop.
cardiale risicofactoren roken +, obesitas +, dm in zwangerschap, ht+, hc -, fa ++ allergie: bisoprolol, amoxicilline
dyspneu met totale respiratoire insufficientie bij verdenking exacerbatie copd obv persisterende roken.
intoxicaties: roken was gestopt maar in januari weer begonnen, rookt er 10-15/dag. daarvoor vanaf 17e gerook
intoxicaties: roken+ (e-sigaret), alc-, drugs- sociale anamnese: woont zelfstandig, geen hulp thuis

intoxicaties: roken 5/dag, alcohol sporadisch 10/jaar, geen drugs familie: negatief voor hart- en vaatziekten.
risicofactoren: roken + 10sig/dag, dm -, ht+, hc-, fa -, hindoestaans +. obesitas + intoxicaties: roken + alcohol -
diagnose: dus doorgemaakte pe unprovoked daarbaast non obstructive emfysema bij persisterend roken geen ph geconstateerd doorgaan ,met apixaban ivm boezemfibrilleren
probeert te minderen met roken.
risicofactoren roken: ja, tot een jaar geleden sigaretten, nu vapen hypertensie: nee hypercholesterolemie: nee
intoxicaties roken: roken ++ alcohol: af en toe drugs: nee sociaal ligt in een scheiding.
risicofactoren: roken + (20 sig per dag), fa+ (broers en zussen <60 jaar hartinfarct)
geen frequente exacerbaties geen bijwerkingen apixaban roken niet gestaakt
afgesproken techniek: aa - pols, tensie bij opname - nuchtertijden besproken - av vernevelen ivm roken (pakje per dag sinds 10 jaar)
triggers: persisterende allergeen expositie, persisterend roken.
risicofactoren: roken: 38 packyears thuismedicatie: tranxene, diazepam, omeprazol
polyneuropathie: ja (simms 2) nefropathie: ja macrovasculair lijden: ja roken: ja fa: moeder dm type 2 overgewicht: ja hypertensie: ja hypercholesterolemie: ja
intoxicaties: roken: vanaf 11e tot heden, ongeveer 20 py, man rookt ook nog sociaal: getrouwd
geen long c.q luchtweg klachten familie anamnese negatief voor longziekten roken sinds jaren met blootstelling 50 pakjaren
werkdiagnose: exacerbatie copd bij persisterend roken dan wel beginnende (virale) luchtweginfectie
intoxicaties: - roken: gestart: 1978 (1 sigaret per dag) -alcohol: gestopt met drinken
exacerbatie copd met enige eosinofilie geluxeerd door mogelijk expositie allergenen en persisterend roken risicofactoren roken"
toch ook hoesten, komt door het roken.
hypercholesterolemie(-), gebrek aan lichaamsbeweging (-), overgewicht (-), stress (+), roken (+), roken partner (-), alcohol (-).
beleid: afvallen stoppen met roken rugligging vermijden (evt sleep position trainer) mandibulair repositieapparaat (mra)
blijft geagiteerd contact met parnassia jhd dr. chen, tot voor huidige opname clozapine 350 mg en rokend, geen geagiteerd of psychotisch toestandsbeeld.
intoxicatie: roken: 2 sigaren per dag. 20 jaar lang alcohol 1 wiskey per dag. 17 jaar lang drugs: nooit
intoxicaties: roken e-sigaret (tot 7 jaar geleden sigaretten gerookt), alcohol sporadisch.
roken: ja fa: nee overgewicht: ja hypertensie: ja hypercholesterolemie: ja
intox/ roken +-50py shag, alcohol 2-3eh/dg, drugs- geen
conclusie exacerbatie copd, meest waarschijnlijk op basis van pneumonie, dd persisterend roken.

conclusie: stoppen met roken en alcohol drinken.
fa: tante en oma trombosebeen. vader hvz. ht-, hc-, dm-, roken+
cardiale risicofactoren roken ja 25 sig/dag hypertensie nee hypercholesterolemie nee diabetes mellitus nee
persisterend roken.
diagnose: copd met bijkomend emfyseem (niet evidente bulleuze type op ct) over jaren progressief bij roken chronische bronchitiden staat niet op voorgrond
conclusie: ap 1/4 na cabg niet gestopt met roken, rookt 3 pakjes per dag c 1j tenzij klachten
leefstijl: roken, voedingsgewoonte aangepast, is bij dietiste geweest voor advies.
polyneuropathie: nee nefropathie: micro-albuminurie macrovasculair lijden: nee roken: ja fa: vader dm type 2 overgewicht: ja hypertensie: nee hypercholesterolemie: ja
op de vraag of de patient blijvend wil stoppen met roken is door de patient geantwoord: ja, graag op korte termijn stoppen met roken.
intoxicaties: -. roken+. patient rookt sinds 30 jaar, maximaal 2-3 pakjes per dag.
intox: roken+ , alcohol - drugs -
patient woont in verpleeghuis vanwege cognitie,as dabigatran roken 60jr.
cf: roken+, alcohol-, ht+, hc+, dm-, fa+ vaders en 4 broers met hvz a/ vanmiddag om 16-16:30u
intoxicaties: rookt 1 pakje sigaretten per dag.
polyneuropathie: nee nefropathie: micro-albuminurie macrovasculair lijden: nee roken: ja fa: vader dm type 2 overgewicht: ja hypertensie: nee
beleid stop roken, 1 keer per week zoete olie ads.
intoxicaties: 1 e alcohol per dag, roken+ lichamelijk onderzoek: algemeen: niet ziek en pijnlijk bij beweging extremiteiten: knie links
gepland eerder contact bij toename klachten, pijnklachten of sensibiliteitsafname stoppen met roken
cardiovasculaire risicofactoren roken:+, hypertensie:+, hypercholesterolemie:+, diabetes mellitus:-, obesitas:-
intoxicaties roken: 20 per dag. medicatie mdl omeprazol 2 x 40 mg allergien geen allergien
als kind geen astma in rust geen klachten hoesten:+ niet opvallend, sputum+ roken: op 15 jarige leeftijd begonnen pakje per dag , 55 py is aan het revalideren
intoxicaties: alcoholgebruik: 2-3 keer per week, 1-2 eh roken: niet , is 19-11-2019 gestopt, dit gaat heel goed! drugsgebruik: nooit
roken: 40j, nu 2-3per dag, vroeger veel meer.
risicofactoren: roken +, hypertensie +, diabetes +, hypercholesterolemie +, familieanamnese +, van hindoestaanse afkomst
rf/ dm-, ht+, hc-, roken+, fam gb
normale flow aa. vertebrales en a.basilaris wv cea links roken ja lichaamsbeweging voldoende voeding gezond
conclusie: verbetering bloedwaardes, persisterende neutrocytose en trombocytose passend bij roken.

intoxicaties: roken: ja, alcohol: sporadisch.
beloop: tc pt gesproken lab koude aggl neg stoppen met roken heeft wel iets geholpen 24-4 naar is_doctor0 operatie rug wacht op uitslag
is niet gestopt met roken.
intoxicatie: roken+
patiente met hoog risico coronairlijden bij fors langdurig roken, 3 jaar geleden geen coronairlijden aangetoond
intoxicaties: roken: rookt 20 py, geen alcohol gedronken
voorgeschiedenis: 2013 reversed veneuze femoro-truncale bypass rechts persisterend roken copd alcohol abuses
emfyseem icm small airway disease als voorstadium copd, met achteruitgang bij persisterend roke
echter blijven roken.
intoxicaties: roken: roken50 packyears.
intoxicaties: roken + alcohol -
snurken zonder osas 2009 leukoplakie larynx, tijdens follow-up na smr verdwenen 2010 copd gold i bij roken 2010 functionele buikklachten 2015 seh ivm dyspnoe
intoxicaties: roken + (20jr gerookt), alcohol 4eh/dg, drugs -. sociale anamnese: woont zelfstandig met echtgenote
intoxicaties: roken: 15 sig/dag , alcohol: geen , drugs: geen allergien: geen bekend familie anamnese: geen epilepsie
rf roken: 1 pak shag/d dm - ht - fam: moeder op 65 mi alc: 6 eh/d soc: verhuizer
intox: roken: tot 13 mei gestopt en rookt nu af en toe weer. alc: geen. gyna: 3x bevallingen vaginaal.
roken+++ (al vanaf 14e) woont inmiddels weer thuis kan en doet alles zelf. medicatie: atorvastatine
rookt ongeveer 10 sigaretten per dag. is maar 1 week in gulden huis geweest, werd er gek van daar en ging roken als een ketter.
leefstijladvies besproken, belang stoppen met roken besproken en info sinefuma gegeven.
intoxicaties: roken pakje per dag, alcohol: -, drugs: - huidige medicatie: lyrica 2x 75 mg
roken: ja fa: nee overgewicht: ja hypertensie: ja hypercholesterolemie: ja
cardiale risicofactoren: roken + (45 pack years, afgelopen 4 jaar e-sigaret), alcohol -, ht -, fam +
intoxicaties: roken sinds 15 jaar, dagelijks 2-3 glaasjes rose.
intoxicaties: roken: sinds 14e jaar meer dan een pakje sigaretten per dag allergien: geen
aandacht risicofactoren (incl. rr, roken) - nu geen verdere controle, zo nodig revisie.
beloop: weer begonnen met roken. verder gaat het redelijk.
mindere longfunctie in combinatie met enige hyperinflatie wd secundair aan kleine luchtwegziekte bij roken

Table 19: Sentences indicating past smokers used for the similarity scores

Sentence
intox, roken 55py (6m gestopt), alcohol- (vroeger problematisch)
intox: gestopt met roken
daarnaast werd patiŽnt sterk geadviseerd te stoppen met roken
intoxicaties roken: gedurende 45 jaar 1 pakje shag per dag alcohol: - drugs: -
met mogelijk subtiele tree-in-bud verdichtinkjes (bijvoorbeeld links apicaal), meest waarschijnlijk roken gerelateerd
advies: absoluut niet roken, nicotine pleister voorgeschreven.
intoxicaties: roken 15sig/dag alcohol- allergien -
samenvattend lijkt er in de longfunctie inderdaad sprake van astma waarbij het persisterend roken de klachten zullen beÔnvloeden.
cardiovasculaire risicofactoren: roken+, ht-, hc-, fam-, dm-
tens, fontaine ii 2012 artrose, tendinitis beide polsen (ha)? 2012 prematuur perifeer vaatlijden bij roken en familiale belasting, pijn linkerflank
intoxicaties: roken sinds 60 jaar, 10 tot 20 sigaretten per dag, in verleden 50 sigaretten per dag.
intoxicaties roken: 14 maanden geleden gestopt, voorheen zware roker.
beleid dringend advies stop roken!
risicofactoren roken, hypertensie en familieanamnese 2016 (4) vagale collaps
intoxicaties roken: nee alcohol: nee drugs: nee
risicofactoren roken: ja, 50 jaar 20 shag per dag.
cardiovasculaire risicofactoren: cva intoxicaties: roken 1.5 jaar geleden gestopt na 50 pakjaren, geen overmatig alcoholgebruik, geen drugsgebruik.
roken (+, 10-15/dag, gestopt na het infarct), roken partner (-), alcohol (sporadisch 1 eh/dag).
gestopt met roken na opname.
intoxicaties roken - roken gestopt in 2017
intoxicaties de patient rookte sinds 1980, (10 shag per dag) en is gestopt met roken.
intoxicaties: - roken: gestopt, voorheen 5 sigaretten per dag - alcohol: 1 wijn/dag
intoxicaties roken: recent gestopt (35-40 shag per dag sinds 15 jarige leeftijd)
intoxicaties de patient rookte sinds 1966, (2 sigaretten per dag) en is gestopt met roken.
intoxicaties: in 1988 gestopt met roken (10py), 4eh koffie per dag, geen alcohol
intoxicaties: roken-, alcohol-. sociaal: woont met partner.
intoxicaties roken: - alcohol: - drugs: -
intoxicaties roken: sinds 2 jaar gestopt, 50 py alcohol: vanaf 2006 nooit meer. drugs: nooit sociale anamnese s
intoxicaties: roken 50pack-years, nu gestopt. lichamelijk onderzoek: linker oor: intact trommelvlies met lucht
intoxicaties: roken: 40 jaar 5-10 sigaretten per dag, inmiddels 7 jaar gestopt. alcohol: 2-3 eh per dag. drugs: g

intoxicaties: roken: gestopt alcohol: drugs: allergieŽn: geen bekend sociaal: woont met man samen performance
stoppen roken lukt nog niet
roken: nee, gestopt sinds: 10 jaar geleden
risicofactoren: gestopt met roken, familiair belast allergieen: geen bekend
intoxicaties: roken: 40/dag, alcohol: nee, drugs: nee. allergieen: geen bekend sociaal: woont met echtgenote.
-roken: gestopt -rookt(e) sinds (jaartal): 1978 -soort tabak: sigaretten -aantal sigaretten: 10
roken: 5 jaar geleden rookstop. 10-15 igaretten per dag.
roken: nee fa overgewicht: nee hypertensie: nee hypercholesterolemie: nee
risicofactoren: ht+, dm ii+, hc-, roken: 2-3/week, voorheen: shag 4-5 per dagn
roken: nee fa: nee overgewicht: nee hypertensie: nee hypercholesterolemie: nee
intoxicaties: roken in verleden, 45 jaar geleden gestopt allergieen: geen
intoxicaties de patiŽnt rookte sinds 1962, (25 sigaretten per dag) en is gestopt met roken sinds 2018. aantal packyears: 70. opmerkingen mbt roken: 2 jaar gestopt.
risicofactoren: eerder infarct, roken in verleden en hypercholesterolemie.
intox/ caffeine: 4e/dag alcohol: 1e/wk roken: tot 1978 1 pakje per 2 dagen
intoxicaties roken: nee, alcohol bij gelegenheid
roken: neen alcoholgebruik: geen
conclusie: achteruitgaande longfunctie bij persisterend roken, onderliggend acos.
intoxicatie gestopt met roken, ongeveer 12 packyears
intoxicaties, roken gestopt, 40-50 sig. per dag van 22-58 jaar
intoxicaties: roken heel lang geleden gestopt, maximaal 10-15 jaar gerookt
rookt weer maar gaat stoppen roken
cardiovasculair risicoprofiel: - roken: + (rookte op stressgerelateerde momenten, nu sinds jaren gestopt) - hypertensie: -.
rf: roken: ja alcohol: 4 to 5 bierjes 1 x per week drugs: geen htn: ja dm: ja
intoxicaties de patiŽnt rookte sinds 1978, (40 shag per dag) en is gestopt met roken sinds 2021
intoxicaties: roken: niet. allergie: huisstofmijt.
drie maanden later is patiŽnt gestopt met roken, waarna hij moeilijker kon ademen en klachten kreeg van pijn op de borst.
intoxicaties: roken 23 jaar geleden gestopt daarvoor 2 pakjes per dag.
sinds 14m ook gestopt met roken.
intoxicaties: roken: 3/4 jaar geleden gestopt. 55 jaar gerookt, pakje per dag
vasculaire risicofactoren: leeftijd, belaste familie, enigszins roken in verleden
intoxicaties: roken: gestopt sinds 11-12-2019 (7y), alcohol: -, drugs: - allergieen: hooikoorts
intoxicaties: geen alcohol, drugs of roken
cardiale risicofactoren: roken+, dm+, hc+, ht+, fa-
intoxicaties: roken: 40 jaar 5-10 sigaretten per dag, inmiddels 7 jaar gestopt.

intoxicaties: roken: 4sig/dag gedurende >30jaar,alcohol: bijna dagelijks 1l whiskey
intoxicaties: roken + wiet, geen alcohol gebruik/misbruik
intoxicaties de patiŽnt rookte sinds 1995, (40 sigaretten per dag) en is gestopt met roken sinds 2019.
roken 16 jr stop
intoxicaties: roken: pakje sig/dag gedurende 50jaar
intoxicaties: roken -, alcohol -, drugs-.
intoxicaties de patiŽnt rookte en is gestopt met roken.
roken +.
intoxicaties: roken gestopt, voorheen 3-4 per dag
intox: roken gestopt
intoxicaties: roken: 3 weken geleden gestopt, 1 pakje shag per dag 35py
cvrf: eerder herseninfarct, ht, hc, eerder roken, fa+
roken: weer begonnen, hield stoppen gezien de stress niet vol
intoxicaties: roken gestopt, voorheen 6-7 sig/dag
risicofactoren hypercholesterolemie, vroeger roken, alcoholgebruik, adipositas
intox/ roken -, alcohol 3 eh/wk, drugs - fam/ moeder borstkanker op 80-jarige leeftijd.
intoxicaties de patiŽnt rookte en is gestopt met roken.
crf: familiaal: (vader acs 40jr), roken-, dm+, ht+,hc+ intox: roken: 40py
intoxicatie roken: lang geleden gestopt.
intoxicaties: roken -, alcohol -
intoxicaties: roken gestaakt
intoxicaties: roken: gestopt alcohol: drugs:
intoxicaties: is in 2000 het roken gestaakt
intoxicaties: alcohol: borrel 1-2x/week roken: in verleden drugs: geen.
intoxicatie: roken: niet alcohol: niet sociaal: gepensioneerd, voorheen in de tuin werkzaam
roken: nee, gestaakt
intox roken: vroeger, 37py.
intoxicaties: roken 50jr gerookt, nu gestopt. alcohol -, drugs -
conclusie vroeger rokende man met hypertensie en recente status na cva, rope score 4
roken gestopt.