Master's Thesis

Predicting diagnoses of patients in the Emergency Room: a multi-label text classification approach

By:
Hanna 't Hart
Student number: 1984691

September 2022

MSc Artificial Intelligence
Graduate School of Natural Sciences
Utrecht University

Submitted in partial fulfilment of the requirements for the degree of M. Sc.

First examiner: Pablo Mosteiro (UU)
Daily supervisor: Marieke van Buchem (LUMC)
Second examiner: Tejaswini Deoskar (UU)

Abstract

Artificial Intelligence (AI) models have big potential in the medical domain because there is so much data available. Previous research shows promising implementations of AI in the medical domain. This thesis aims to build an AI model that predicts diagnoses of patients from the History of Present Illness (HPI) text, to support clinicians in the Emergency Room (ER). The implementation of such a model can be of help to reduce diagnostic errors, and lower the workload of clinicians in the ER. No previous work has tried to make a predictive model for diagnoses, that can support clinicians in the ER, based on Dutch HPI texts that is explainable as well. A multi-label dataset with more than 120,000 HPI texts with corresponding diagnosis labels was used to train a simple baseline model and three deep learning models, including a sequence-to-sequence model and two BERT-based models, to compare to each other. After testing and evaluating the models by F1 score, it was found that none of the models achieve high performance on the whole dataset. Interestingly, the two BERT-based models did achieve high performances when trained on smaller samples of the dataset, consisting of diagnoses that are more difficult for clinicians to distinguish. Therefore, it is proposed to further research this idea of more specific models, to support clinicians to make a decision between more specific diagnoses.

Acknowledgements

First and foremost I want to thank my supervisors Marieke van Buchem and Pablo Mosteiro. I am grateful for all the help Marieke has provided during all meetings, despite the time difference, her interest in the project, and her positive energy. I am grateful for all the help of Pablo with the academic part of the thesis and his professional and formal look on the thesis.

I would also like to show gratitude to the whole AI & WGZ team of the LUMC, which I was a part of for 9 months. The support and weekly meetings were a big help. With special thanks to Laurens Schinkelshoek, whom I could always contact for any question I had.

My research would have been impossible without the expertise of Martijn Bauer. His help and the input he provided on the medical part of the project were of much help.

Furthermore, my sincere thanks to Tejaswini Deoskar, for setting me up with my supervisor, and for being the second reader of my thesis.

The help provided by Iacer Coimbra Alvas Cavalcanti was greatly appreciated as well. His help with the sequence-to-sequence model and his expertise on NLP in the medical domain were of great support during coding. I also want to show my appreciation to Ameen Abu-Hanna, for his input and expertise in the monthly meetings to help me get further and to see things from another perspective.

For all the help with the problems I encountered regarding the GPU and Linux, I would like to thank Michel Villerius.

I am also grateful to Piek Vossen for sharing his knowledge and expertise on the medical BERT model, and for making the model available to use.

Also my sincere thanks to Ivana Brasileiro Reis Pereira, for the support on the literature proposal and the help with stating the research questions.

Furthermore, I would like to offer special thanks to Andrés Mendoza, for all the support that I needed mentally, for helping me come up with solutions when I could only see problems, and for making me believe in myself. And last but not least, thank you Louise Pos for always believing in me and understanding what I am going through.

Contents

1 Introduction

Artificial Intelligence (AI) techniques are developed for the analysis of data and for building predictive models. One of the most promising domains for the application of AI is the medical domain. A hospital has a substantial amount of data available about all patients that enter, and with such a big amount of data at hand, AI can be implemented to process this information and make predictions. Already since the mid-twentieth century, clinical decision support systems have been developed [Miller, 1994]. An example category of AI in the medical domain is medical imaging, where algorithms can analyze images such as computed tomography (CT), magnetic resonance imaging (MRI), and ultrasounds, more accurately than doctors [Lundervold and Lundervold, 2019]. This thesis will not focus on the analysis of images but on the analysis of text. More specifically, on the implementation of an AI prediction model in the Emergency Room (ER) of a hospital, where diagnoses are predicted from Dutch texts.

1.1 Problem description

In the ER of a hospital, action by clinicians must be taken rapidly. Patients come in with all different kinds of symptoms or illnesses, and therefore it can be hard for clinicians to diagnose a patient quickly and accurately. It has been found that diagnostic error, where the patient receives a wrong diagnosis, occurs more often in emergency rooms than in the rest of the hospital [Hussain et al., 2019]. Diagnostic errors can result in serious harm to the patient for two reasons [Balogh EP, 2015]. Firstly, when a patient is diagnosed with a wrong illness, the actual illness is not treated. The second reason is that the wrong treatment can also unnecessarily harm the patient. Besides the risk of diagnostic error, the correct diagnosis of patients takes years of medical training, and then still is a time-consuming process. Due to the shortage of clinicians, the demand for experts is often higher than the available supply, which puts a lot of pressure on the clinicians. This can result in a delay in diagnosing a patient.

Clinicians in the hospital thoroughly document patients' clinical encounters in their Electrical Health Records (EHR). These EHRs consist of for example a patients' contact information, insurance and demographics. But also clinical data, such as medical history, allergies, lab resuls and tests ordered. EHRs therefore contain a lot of information that can help diagnosing the patient, such as the initial complaint, the physical examination, and the History of Present Illness (HPI). One hospitalization can generate thousands of data points for an individual, making the EHR a rich source of data for AI models, specifically machine learning prediction models [Li et al., 2020]. Unfortunately, the data in EHRs is hard to use because most of it is written in human language. Structured data is easy to feed to a machine learning model, but text needs to be pre-processed. This is where Natural Language Processing (NLP) comes in. NLP is a form of machine learning which is capable of processing and analysis of free text [Locke et al., 2021]. When used on medical data, it can result in tools for predicting hospital readmission [Huang et al., 2020], classifying radiology reports [Putelli et al., 2020], predicting diagnoses [Blinov et al., 2020], and more. NLP is different for medical data than when it is used on other data because medical text differs considerably from general text. Doctors write the text in the Electronic Health Records down during interviews with a patient or in between seeing patients. Since they are under a lot of time pressure, this often results in non-standard and grammatically incorrect notes. Together with the abbreviations and medical jargon, this type of text is very different from the standard language.

In recent years, the combination of NLP with deep learning and machine learning methods has led to great progress for AI in the medical domain. Powerful AI tools have been widely applied in clinical modeling and gained numerous successes. These models can achieve comparable or even better performance than domain experts can in the diagnosis of certain specific diseases [Rasmy et al., 2021].

1.2 Case description

This thesis aims to create a model with NLP and deep learning tools to predict diagnoses of patients in the ER to support clinicians in making this important decision. The data for this thesis is provided by the Leiden University Medical Center (LUMC). The input data for the model will be Dutch free text, specifically the History of Present Illness (HPI), sometimes called the anamnesis. The HPI is a description of the development of a patients' illness. It includes multiple factors, for instance the context of the pain, the timing of the pain, the duration of the pain, and associated symptoms. The doctor gains this information by asking specific questions to the patient and to the people who know the patient well. This HPI interview is the most important concept upon which doctors base their decision of diagnosing the patient and getting the patient the care they need [Pauker et al., 1976]. In the text of the HPI interview, important entities such as symptoms, complaints, and illness history of the patient can be found. It will be interesting to see how predictive these entities are for the final diagnosis of the patient. If the final diagnosis of the patient can already be known after this interview, no further examinations will be needed, or the clinician can already do more specific tests. Therefore, the main goal of this thesis is to find out if it is possible to create a model that will be a clinical decision support system to assist clinicians in their diagnostic decisions, with the HPI texts only as input.

Up until today, there are a numerous of different models with different techniques that have been developed for deep learning and NLP. Two different kinds of models especially show to be very promising for the task of predicting diagnoses, which are sequence-to-sequence models and BERT-based models. These models are popular because they do not only input loose words to generate an output, but both models use also the context of the words, which makes them very interesting for working with longer texts. A more detailed explanation of these models can be found in chapter 2.

To implement a Natural Language Processing and deep learning model in practice, there is more to it than solely making a working model. To make sure clinicians can trust the system and are willing to work with it, the model cannot be a so-called black box. Deep learning models are known for the fact that the user does not know how the model comes to a certain conclusion. To make sure the user can understand the decision of the model, it needs to be explainable. So besides building a high performing model, the aim is to make the model explainable as well.

As will be discussed in chapter 2, there has been previous research into predicting diseases with deep learning. However, as far as is known, it has not been done before to use NLP and deep learning to make a language model that predicts diagnoses based on the HPI in Dutch free text, which is also explainable and therefore possible to implement in the hospital.

1.3 Research question

The research question this thesis is aiming to answer is:

"How can we make a model to predict diagnoses of patients to support clinicians in the emergency room?"

To find the answer to this question, the following sub-questions will need to be answered:

1. From a clinical perspective in the emergency room, what is required of a diagnoses prediction model to be practically applicable?
2. Under the criteria of sub-question 1, what metrics can we use for the evaluation of the model?
3. Should we use an intrinsically explainable model, or is it necessary to add an explainability model to our trained model?
4. Can we use a sequence-to-sequence model for this task?
5. Can we use a BERT-based model for this task? And is the pre-trained clinical Dutch BERT-based model MedRoberta.nl generalizable to other datasets, in particular, to the dataset used in this research?

A more extensive explanation of the specific models will be given in section 2.

1.4 Thesis outline

The remaining sections of this thesis are structured as follows: section 2 of this thesis contains explanations of the key concepts and a discussion of previous related work. In section 3, the dataset used for this research is explained in more detail. Section 4 describes the methods and the models used. The experimental setup of the experiments can be found in section 5. Section 6 shows the results of these experiments, and finally, the discussion can be found in section 7, consisting of the analysis of the results, the conclusions of the research questions, the limitations of the research and the further research that is proposed.

2 Literature Research

This thesis is about the implementation of a language model in the emergency room of a hospital. It focuses on how to use Natural Language Processing on medical data, multi-label classification, deep learning, explainability, and model evaluation. In this section, these and other key concepts will be explained in more detail and related work will be discussed.

2.1 Natural Language Processing

Notes in Electronic Health Records (EHRs) contain loads of interesting clinical information. The EHR describes symptoms, reasons for diagnoses, radiology results, daily activity, patient history, physical examination results, HPI interviews, lab tests, and more [Huang et al., 2020]. One patient can be associated with hundreds of notes within one stay in the hospital, which can lead up to thousands of data points for one individual, all together in a health record [Li et al., 2020]. A doctor in the emergency room needs to make decisions under time constraints, and having to read a great number of clinical notes will add up to this doctor's workload and pressure.

Natural Language Processing (NLP) is the analysis of human language, carried out by computers. NLP is used to give the ability to computers to handle text as a complex syntactic piece of data [Locke et al., 2021]. NLP can be used in the medical domain to process and analyze the text elements of EHRs. The majority of current EHRs consist of free text. This is appealing to the users because of the flexibility of expression; every clinician can write down what they think is necessary and in any way that they want to write it. This can come in handy during an interview with the patient, when the doctor should pay as much attention to the patient as possible, instead of entering the data. During conversations (such as the HPI interview), the documentation is recorded in typed form. This is time effective but results in non-standard and incomplete notes. It also results in a different language compared to the general language [Verkijk et al., 2021]. Firstly because the clinical notes contain common usages of medical terms, such as illnesses, medicines, treatments, symptoms, and their abbreviations. This type of language is much less common in general text data like Wikipedia texts, books, or news articles. Secondly, the sentences in clinical notes tend to be grammatically incorrect and shorter compared to the standard language. To be time-efficient, it is easier for clinicians to simplify sentences and leave out function words. Building a model that learns useful representations of clinical text can therefore be a challenge [Chapman et al., 2011]. This complicates the task of diagnosis prediction based on EHRs since they contain an extensive amount of unstructured, poorly organized data that is less manageable for the analysis using NLP-based methods [Blinov et al., 2020].

2.2 Machine learning classification

Within machine learning classification, tasks differ in the type of classification. The most used and simplest type of classification is binary classification. Binary classification is the task of classifying the target value into either one of two groups. A test to see whether a patient has a specific disease or not is a binary classification (with either yes or no as the prediction). As will be discussed below, during the last decades machine learning and binary classification have become very popular and widely adopted in the medical field. Diagnosis identification is one of the biggest motivations for using machine learning in this domain. An example of a classification task that has been widely studied is cancer detection. In 2010, Microsoft created "InnerEye", a tool that can detect brain tumours more rapidly than humans can [Smiti, 2020]. In recent years, machine learning for diagnosis identification has become more popular and other tools have been created for specific diseases. Himes et al. (2009) created a predictive model for Chronic Obstructive Pulmonary Disease (COPD) in asthma patients, using electronic health records. With their model, they could classify asthma patients that are at risk of getting COPD. Maarseveen et al. (2020) also used electronic health records, but to identify patients with Rheumatoid Arthritis. Other machine learning implementations specifically focus on predicting mortality [Lin et al., 2019], sepsis [Goh et al., 2021] or readmission [Huang et al., 2020].

Besides binary classification, other types of classification are multi-class and multi-label. Multi-class classification is the task of classifying the target value into one out of three or more groups. Exactly one group can be assigned to the target value and not more. An example of multi-class classification is the task of classifying a patient's hospital readmission stay into one of three classes: zero days, less than thirty days, or more than thirty days. In this case, the readmission can only fall into one class, and not more.

Patients can get more than one diagnosis per encounter. This means that for this project, the model needs to be able to predict multiple diagnoses when necessary. To do this, a multi-label classification model will be needed. In multi-class classification, the classes are mutually exclusive, which means for this project that every patient can only get assigned one diagnosis by the model. However, diagnoses do not necessarily exclude each other. As will be shown in section 3, some patients in the data got diagnosed with multiple diagnoses at once. In multi-label classification, machine learning algorithms can predict multiple non-exclusive classes. This is more interesting for this project since the model will show all the diagnosis categories that are relevant for a specific patient.

Binary classification has been used widely in predictions in the medical domain, but multi-label classification rarely, even though multi-label classification models have shown great promise. Zhou et al. (2021) defined a multi-label classification task for the diagnosis of diabetes complications. Their model had very high performance and showed that with a multi-label classification approach they can predict multiple factors at once [Zhou et al., 2021]. Another example of multi-label classification for disease prediction is the model “GroupNet” [Chen et al., 2019]. Physical examination records of patients were used here to predict the probability of different chronic diseases. The multi-label classification prediction - based only on structured data - achieved great performance.

2.3 Deep Learning

Deep learning is a subset of machine learning, based on artificial neural networks to mimic the structure and function of the human brain. Typically, deep learning algorithms make use of a large number of hidden layers which results in a complex structure for defining inputs to outputs, where machine learning algorithms use structured data to make predictions and define specific features from the input data and structure these into tables. Deep learning models can process unstructured data like text and images, because it automates feature extraction. Then, via gradient descent and backpropagation, deep learning models adjust themselves, allowing them to make predictions of new input and correct for errors. This complicated structure of deep learning algorithms makes them able to output more complex predictions [Education, 2020]. Deep learning techniques have achieved great success in for example computer vision tasks, including image classification, image detection and image segmentation [Maier et al., 2019]. Compared to results obtained with standard machine learning methods, deep learning approaches achieve better performances. However, training deep neural networks does require more computational resources and time [Hahn and Oleynik, 2020]. For natural language processing, there is a need for a lot of data to train the model on, which can take a great amount of time. Luckily, there are language models that have already been pre-trained on huge corpora including Wikipedia, books, and news articles.

Deep learning supports many of the natural language processing tools that are popular in the medical domain. Deep neural networks are designed for classification, they can identify linguistic and grammatical elements and map them to one another [Bresnick, 2018]. With this ability, deep learning models can understand complex semantic meanings. But as stated in section 2.1, the free-text in Electronic Health Records consists of messy, incomplete, and inconsistent notes. Therefore, deep learning models currently still struggle with the task of identifying clinically relevant elements and finding relationships between them. Other challenges include the quantity and the quality of EHR data, biased data, and the interpretability of the model [Xiao et al., 2018]. Nevertheless, deep learning is very promising and recently it has been implemented widely in medical applications [Chen, 2020]. Because of the success of deep learning in image classification, image detection and image segmentation, there has been a lot of research on tools for clinical imaging. For example, Liu et al. (2014) used brain MRIs to diagnose Alzheimer’s disease

[Liu et al., 2014]. The model needed less prior knowledge and a smaller dataset than machine learning methods for the same task and still achieved a greater performance. Another example of clinical imaging is the model of Esteva et al. (2017). They created a convolutional neural network that can classify skin cancer lesions, using only images as inputs [Esteva et al., 2017]. Besides clinical imaging, there has also been an application of deep learning techniques for clinical Natural Language Processing problems. For example, the classification of radiology free-text reports based on pulmonary embolism findings [Chen et al., 2018]. This convolutional neural network can classify the reports highly accurate. Another example is the de-identification of patient data in Electronic Health Records ([Dernoncourt et al., 2017]; [Ekbal et al., 2016]; [Liu et al., 2017]). Without manual feature engineering, these deep learning models achieved great performance. The data can then be shared or published, while still protecting the confidentiality of patients. Every day, more deep learning and NLP techniques are used to create tools in the medical domain. State-of-the-art models achieve very high performances, which makes these methods so promising for the prediction of diagnoses.

2.3.1 Sequence-to-sequence models

A type of deep learning models that is regularly used for text is the sequence-to-sequence model, often called seq2seq. Seq2seq models have been used for numerous language tasks such as machine translation, speech recognition, and question answering. As the name indicates, seq2seq models are used to convert sequences of one domain to sequences in another domain. For example in machine translation, it can convert an English sequence to a French sequence. In this project, both the input data (HPI texts) and the output data (diagnosis label) consist of a sequence of text, therefore this task can also be done by a seq2seq model. The seq2seq model consists of two parts: an encoder and a decoder. The encoder turns the input into an internal hidden vector, which contains an input item plus its context. The decoder then takes this vector and turns it into the desired output, considering the context. These two different parts are two separate models, commonly recurrent neural networks, combined into one big model.

Seq2seq models have been used in the medical domain before. One example is the model of Zhang-James et al. (2021), which accurately predicts new cases and deaths of Covid-19, up to 30 days in the future. The input sequences came from past Covid-19 cases and deaths, infection numbers, and resident mobility. The output sequences are future Covid-19 cases. This example shows that a seq2seq model can perform well on medical data in a modern situation. Another example is the seq2seq model of Liu et al. (2018). Their model predicts mortality and discharge status within the next 24 hours from structured medical data. The seq2seq model of Lee et al. (2019) shows that seq2seq models also perform great on medical data. The model can generate medical text, in the form of a summary of clinical notes. These examples show promising results for the use of seq2seq models in the medical domain. As far as we know, there has not yet been a seq2seq model for the prediction of diagnoses, therefore it remains to be observed if this model will work well for this specific task too.

2.3.2 BERT-based models

A deep learning model that is used primarily in the field of NLP is the transformer model. Transformers were introduced in 2017 by Google [Vaswani et al., 2017]. Before this time, the most state-of-the-art NLP systems relied on deep learning models called Recurrent Neural Networks (RNNs). The biggest problem of RNNs is that they struggle to handle long sequences of text. An RNN-based model would forget the first part of the text by the time it got to the end. Another problem with RNNs is that they are hard to train. You cannot train them on too much data, because they process words sequentially, and cannot run in parallel. Transformer models can parallelize, and therefore it is possible to train very big models. An extensive explanation of transformer models can be found in the paper of Vaswani et al. (2017). The most important part to understand is called attention. Attention is a mechanism that allows a language model to pay attention to every word in the sequence before deciding the output, instead of deciding output from only one word. The attention mechanism will take the important and relevant words of the input and assigns a higher weight to these words, which increases the accuracy of the prediction of the output. Transformer models contain a specific type of attention mechanism, called self-attention. Self-attention allows a network to understand a word in its context, relating different positions of a sequence to compute the output. The Transformer model is the first model relying entirely on this self-attention mechanism.

In 2018, Google introduced a transformer-based model called Bidirectional Encoder Representations from Transformers (BERT). As explained by Devlin et al. (2019), the makers of BERT, the model is designed to pre-train deep bidirectional representations from unlabeled words by using both the left and the right context in all layers [Devlin et al., 2019]. BERT is fully based on the transformer model by Vaswani et al. (2017) but uses bidirectional self-attention instead of constrained self-attention. The biggest limitation of earlier language models is that they are unidirectional, which means that every token can only give attention to previous tokens (in a left-to-right architecture). As a solution for this limitation, BERT uses a Masked Language Model (MLM). The MLM masks some words of the input randomly, aiming to predict the original word based on its context. This happens in the pre-training phase of the model. By adding one small layer to the core model, BERT can be used for a wide variety of language tasks. This is called fine-tuning. It is possible to take a pre-trained BERT model and fine-tune it for the task you want it to do, for example, sentence classification, question answering, or named entity recognition. This pre-training/fine-tuning framework has been proven to be very effective in NLP.

Because of their great performance, BERT-based models are now one of the most popular models to handle textual input. The models have been widely applied to various domains, such as the financial domain, biomedical domain, and recently also the medical domain. The reason BERT-based models are easy to implement in a specific domain is that they can be trained on any corpus you would like. General BERT models are trained on data like Wikipedia pages, books, and news articles. To make a clinical BERT model, it can be pre-trained on clinical data instead of general data. This approach has resulted in multiple clinical BERT models, such as ClinicalBERT [Huang et al., 2020], Med-BERT [Rasmy et al., 2021], and BEHRT [Li et al., 2020]. ClinicalBERT was pre-trained on a dataset called MIMIC-III, which is a large, open database of clinical data of patients from a hospital in Boston [Johnson et al., 2016]. ClinicalBERT was then fine-tuned on the task of predicting a patient's hospital readmission. The results of this show that a BERT model that is pre-trained onto domain-specific data instead of general data results in greater performance. Med-BERT was pre-trained on a structured dataset containing data from over 28 million patients. Med-BERT was then fine-tuned on several prediction tasks. BEHRT was pre-trained on data from a network of more than 600 general practitioners based in the United Kingdom. After pre-training, BERHT was fine-tuned on the task of predicting a patient's chance of a specific disease in the future. This resulted in very high accuracy in predicting a patient's disease for the next visit, for the next six months, and for the next twelve months. To our knowledge, there has been only one BERT-based model in the medical domain that was fine-tuned specifically for the task of diagnosis prediction of patients visiting the hospital. That is RuEHR-BERT, a Russian BERT variant [Blinov et al., 2020]. RuEHR-BERT was pre-trained on Russian datasets, including the HPI and symptoms of doctor's visits from over a million patients. One thing missing in this BERT model is that it does not show any explanations of how the model came to a certain conclusion. More about this issue can be read in section 2.4.

The existing clinical BERT-based models all show very promising results for using this language model in the medical domain. However, these models are all pre-trained on data in non-Dutch language. This means that they can only be used on data in that specific language. Pre-trained Dutch BERT models are needed when the model will be given Dutch input. Dutch BERT models do exist, for example, BERTje [Vries et al., 2019] and RobBERT [Delobelle et al., 2020]). Both of these models are trained on Dutch Wikipedia, news, and web data. Up until this day, there has been only one BERT language model that is trained on Dutch medical data: MedRoBERTa.nl, a BERT model trained on Dutch Electronic Health Records [Verkijk et al., 2021].

2.3.3 MedRoBERTa.nl

MedRoBERTa.nl is the first Dutch BERT-based model that is pre-trained on clinical notes. The data the model is trained on consists of nearly 10 million documents, including HPIs, notes from doctors and nurses, letters from the hospital to the patient, and the emergency room report. All documents come from two hospitals in The Netherlands, namely the Amsterdam Medical Centre (AMC) and the Free University Medical Centre (VUMC).

The model was created mainly for a project called A-PROOF [Kim, 2021]. The goal of this project is to create a model that can predict the functioning level of a Covid-19 patient in different domains. An example domain is "Walking", with a functioning level from zero to five where zero means that the patient has no ability of walking and five means the patient walks perfectly. The model was fine-tuned for the domain classification task of this A-PROOF project. Verkijk (2021) also anonymized the language model. Therefore, it could be published and used publicly, without putting the safety of the patients at risk.

Since the model from Verkijk et al. is the first and only BERT-based model that is pre-trained on Dutch medical notes, it can be interesting to see if it is generalizable to other datasets. If this would be the case, it is scientifically relevant to know that BERT-based models can be pre-trained and then used on similar data, without having to pre-train it on this new data. The MedRoBERTa.nl model has been trained on quite similar data as the data described in section 3, which means that the model might be useful to use on this data too. Since the fine-tuning task is different from the task of this thesis, the model will still need to be fine-tuned for the multi-label classification task of predicting diagnoses of patients arriving in the emergency room.

2.4 Explainability

Explainability, also referred to as interpretability, is the concept that a machine learning model can be explained in such a way that a human being can understand it. According to Adadi and Berrada (2018), models are explainable when their inner workings can be understood by humans. Complex machine learning algorithms, especially deep learning algorithms, are so complicated that a human cannot understand what is going on inside the model. These models are called a "black box". You give it input, something happens, and it gives back output. But it is hard to understand what the model did with this input to come to this certain conclusion. Deep learning has become the preferred modeling approach because of its high accuracy for various tasks. However, deep learning algorithms are also very opaque, which means that it is more challenging to interpret what is going on internally. Recent research emphasizes more and more that artificial intelligence systems should be able to support explanation and understanding, rather than just solving a problem [Holzinger et al., 2019]. Therefore explainability has become just as important as accurate results.

Healthcare workers often have problems with trusting a complex model, because some of these models are impossible to understand for a human, having millions of different parameters. For clinicians to accept and trust the models, they must be able to understand why and how the model came to a certain conclusion. Only then they are willing to use the output of such a system [Stiglic et al., 2020], since trusting a wrong decision of a model will have a high impact in the medical domain. In this project, if the doctor assumes the model is giving the right diagnosis without knowing how it came to this conclusion, the diagnosis can be wrong and in the introduction is discussed that this can be of high risk for a patient. From interviews with clinicians, Tonekaboni et. al. (2019) found out that clinicians view explainability as "a means of justifying their clinical decision-making in the context of the model's prediction". During these interviews, all clinicians expressed the need to understand the models' relevant features that align with what they know of the medical practice. Clinicians found models with lower accuracy still acceptable when it was clear to them why the model performed this way. This underlines that the need for explainability in healthcare is just as high as the need for highly accurate models, for the model to be of support for a clinician.

There is a close relationship between the performance of a model and its explainability. Often the best performing methods are the hardest to understand, and the ones providing the clearest explanations are less accurate [Bologna and Hayashi, 2017]. Recently, deep learning has been successfully applied in modeling EHR data for prediction, but the gain in accuracy is at cost of an output of the model that is hard to understand. Choi et al. (2016) discuss that there have been several attempts at interpreting neural networks, but these methods are not sufficiently developed for application in healthcare. Choi et al. therefore used a modeling strategy RETAIN. RETAIN is an attention model that explains the prediction results while keeping the prediction accuracy high. RETAIN relies on the attention mechanism, also found in BERT models. RETAIN was tested on a large EHR dataset and can achieve high performance in accuracy and also offer an intuitive explanation of the result.

Markus et al. (2021) discuss that explainability is costly for a model's performance, and it is therefore only needed in certain situations. Namely, when the harm of misclassification is very high and when we need to work on user trust, satisfaction and acceptance of the model for the use in practice. Both of these situations apply to a prediction model in the healthcare domain, because a wrong prediction can be costly for a patient's wellbeing or life, and because there is not yet much acceptance and trust from the clinicians in AI models.

Explainability relying on the self-attention mechanism also found in BERT models is one way to make the model explainable. This is called intrinsic explainability, which is built into the model. There also exist explainability models that you can add to the existing model.

2.4.1 Self-attention mechanism

An interesting part of the BERT model architecture is the self-attention mechanism. This mechanism gives the ability to find relationships among concepts. In a way, it can be used as an explainability mechanism as well. By showing the features that the model paid the most attention to, it can be interpreted as the explainability of the output. There exist tools to visualize this attention of the model, such as the tool of Vig (2019).

2.4.2 LIME

An external model that can be used for explainability is LIME. LIME can be applied to any machine learning model, attempting to understand it by changing the input of data samples and seeing how the predictions change. LIME tweaks a single data feature value and then observes the impact of this tweaking on the output. This can then answer the question of what features were most important for the model to come to a specific prediction. The output of LIME is a list of explanations, showing the contribution of each feature.

2.5 Evaluation

A crucial step in making a classification model is evaluation. Without the right evaluation, you cannot say whether the model performs well. One of the most used methods for classification evaluation is the accuracy of the model [Novakovi et al., 2017]. A lot of articles write about whether the model is accurate or not. However, accuracy only measures a small part of the performance of the model. For example, if the model always predicts the biggest label in an imbalanced dataset, the accuracy will still be high. This high accuracy is useless because a model that predicts the same diagnosis for every patient that comes into the ER is not helpful. Therefore, a suitable evaluation method is needed for this prediction model.

Multi-label classification requires a different evaluation than the methods used in other classification methods. In binary and multi-class classification methods, the most common evaluation criteria is a standard set of metrics that include accuracy, precision, recall, and the F1 measure. The F1 measure here is important since it is calculated with both the precision and the recall equally, it is simply the harmonic mean of the two. This measure is the most used one with imbalanced data. But in multi-label classification, the model predicts a set of labels, and therefore this set is not either correct or incorrect, but can also be partially correct [Sorower, 2010]. This cannot be captured by these aforementioned evaluation metrics, where a prediction is either correct or incorrect. Therefore, the evaluation of multi-label classification is a bit more challenging.

In previous literature, two evaluation metrics have been suggested for multi-label classification: the macro F1 measure and the micro F1 measure. Label-based measures first evaluate each label separately and then averages these results over all labels. In macro averaging, the F1 score gets computed on individual class labels first, and then averaged over all classes. In micro averaging, the F1 scores are computed globally over all instances and all class labels. Macro averaging weighs each class equally and is therefore not influenced by the number of instances of each class. On the other hand, micro averaging is influenced by classes that have more instances than others, because it equally weighs the data points instead of the classes. This means that the micro average score will still be high when the classifier only performs well in the majority of classes.

2.6 Summary

This section has given an overview of the key concepts needed for this project and how they are applied to healthcare. Related machine learning and deep learning models in the medical domain were shown, even models that predict diagnoses. A deep learning language model that is pre-trained on Dutch medical data has been discussed, as well as another specific language model. It was found that the explainability of a model is just as important as the performance of the model. This also answers the first sub-question. For a model to be helpful to clinicians in predicting diagnoses of patients in the ER needs to have two specific characteristics. First, it needs to have a high performance, for it to be usable. Second, it needs to be explainable, for it to be trusted.

Even though there does exist some research on the topic of multi-label classification models to predict diagnoses for patients with deep learning, to our knowledge there has not been such a model that can be used for Dutch medical data, that has a high performance, and that is also explainable. These elements are needed if you want to implement such a model in the hospital, or more specifically, in the emergency room. The rest of this thesis will be focusing on implementing these methods, to make a model that can support doctors in the emergency room. A model that can take the HPI in free-text as input and can give probabilities back for each diagnosis category, how probable it is that this patient can be diagnosed with that diagnosis, together with an explanation of what features made the model come to this decision. Once it has come to a model that can predict this accurately and explainable for Dutch data, it might be interesting to look at how such a model can be implemented in a Dutch hospital. Previous literature has tried these two tasks separately, but not yet together.

3 Dataset

The data that is used for this project comes from patients who had an encounter in the emergency room of Leiden University Medical Center between January 2012 and December 2021. The EHRs of the patients consist of a lot of data. The feature that is used in this project is the text of the History Of Present Illness interview, to find out to which degree it is possible to predict a diagnosis from this interview. As stated in section 2, while writing down the information in a patient's EHR, a clinician can be in a rush, and therefore write in ungrammatical and short sentences. Sometimes, information is left out. And besides this, the diagnosis data of the EHRs can also be hard to use since there can be found multiple diagnoses in an EHR of a patient for one encounter. This means that the right diagnosis label(s) first manually needs to be matched to the encounter before it is possible to use it in a deep learning model. Matching the right diagnosis with the right encounter can become a hard and time-consuming task. Therefore, it is decided to use a different type of data for this thesis, instead of the EHRs. The documents used are the letters that have been written by the clinician of the emergency room and sent to the general practitioner (GP) of the patient.

3.1 GP letters

These letters, from now on called GP letters, are the most important documents for the project. The clinician writes this letter to inform the GP of the patient about the small moment of care that happened in the emergency room since the GP has the overall responsibility for a patient's care. The clinician in the ER provides urgent care, and other problems should then be picked up by the GP. As a result, the clinician at the ER needs to send a letter via email to the GP within one working day of the discharge of the patient. There are some specific requirements for what this letter needs to contain, which can be found in table 1.

Dutch heading	**English explanation**
Mandatory fields	
Kerngegevens	Information of the patient
Reden van komst	Reason for coming in
Conclusie	Diagnosis/Conclusion
Beleid	What has been done in the ER
Voorgeschiedenis	Medical history
Medicatie	Medication
Anamnese	History of Present Illness
Lichamelijk onderzoek	Physical examination
Allergieeën	Allergies
Optional fields	
Aanvullende anamnese	Additional History of Present Illness
Sociale anamnese	Social History of Present Illness
Familie anamnese	Family History of Present Illness
Aanvullend onderzoek	Additional examination

Table 1: Information headings of GP letters with the English explanation

Most of this information is the same information as can be found in the patient's EHR, except that often the information in the GP letter is much more extensive. This is because any clinician can write what they want and how they want it in an EHR. Some clinicians can be very short or vague in writing things down, which makes it hard to collect enough information about the patient. In the GP letter, on the other hand, clinicians should always write down as much as they know to give the GP as much information as needed. An example of this difference can be found in table 2. Note that the data has been changed slightly, to make sure of the privacy of the patient. In this example, it can be seen that the information in the EHR is much less complete than the information in the GP letter.

Data from GP letter	Data from EHR
The above patient was readmitted after a recent discharge due to dizziness. For more than a year already, episodes of fever, abdominal pain attacks and clinically elevated inflammatory parameters have been reported. Imaging showed a large liver, but no other explanation for the long-term complaints. An infiltrate was seen pulmonary and antibiotic treatment was initiated with oral doxycycline. The patient was re-admitted with the above complaints, whereby dizziness was in the foreground at that time, but would also have been playing for a longer period of time. Diagnostics were used, which included a differential diagnosis with longer-standing fever and abdominal pain. To think: Infectious (mycobacterial, endocarditis lenta. Found HIV negative and no active EBV and CMV infection); Systemic disease (including GPA or other small vessel vasculitis in view of the respiratory and abdominal complaints); Metabolism: porphyria seen in abdominal pain attacks in combination with splenomegaly in triggering factors such as an infection (fever), stress; Malignant (lymphoma). The ENT doctor was also consulted to further analyze the vertigo, whereby a vestibular neuritis was excluded and no further clues were found. Given that the patient was not acutely ill, a PET-CT was requested at very short notice for further diagnosis, which will be followed up shortly afterwards. The patient was discharged from the AOA of the LUMC in good clinical condition.	Persistent fever, abdominal pain upper abdomen, cough with green sputum with sometimes some blood. Quickly dizzy when standing up. Eating very moderately drinking is still possible, no acid burns.

Table 2: Two columns with HPI data from the same patient and the same encounter, translated to English. The HPI on the left is taken from the GP letter, the HPI on the right is taken from the patient's EHR.

3.2 Data extraction

The data in the letters is written in the form of one long unstructured piece of text. Before the data can be used for the model, the HPI texts and the diagnosis need to be extracted from this letter first. All letters consist of the headings found in table 1, with the corresponding text underneath this heading in separate paragraphs. Different clinicians use different styles or orders, but all letters do contain these mandatory paragraphs. Therefore, rule-based functions were created to extract the HPI texts and diagnoses from the letters. The HPI texts will then be used as the input feature and the diagnoses as the labels.

3.3 Categorisation

The total number of GP letters available in the dataset is 72990. On average, every letter is linked to 2.8 diagnosis labels. The total number of unique diagnosis labels in the dataset is 1997. From these labels, 898 exist less than 10 times in the dataset. This is a problem with medical data for a deep learning model. There are relatively a lot of diagnoses that are very rare. In this dataset, 265 diagnosis labels only occurred one time in the entire dataset. Of the 1997 unique labels, only 55 percent occurred more than 50 times in the data. A deep learning model learns from examples, and diagnoses that have so few examples available, are never going to be learned by the model. There are multiple options to deal with this problem.

1. The first option is to put all the rare diagnoses together in one class, and call this the "rare diagnoses class" or simply "other". Then, the model will be able to predict either from all the residual diagnoses labels or predicts that the patient falls into this rare diagnoses group. Some information will be lost with this, but the model will most probably get higher performance and can still be of help for clinicians in the ER. However, since there exist so many rare diagnosis labels in the dataset, this group will become a huge group, with big differences between the input HPI texts, but with the same output label. This can confuse the model a lot since there is a big chance that a new HPI text will look like one of this group and then the model will just always predict this rare class.

2. Another option is to let clinicians manually put together clusters of diagnosis labels. Some diagnoses are close to each other, and will still be of value if the model can predict a group with closely-related diagnoses. However, only clinicians can know this and therefore have to go manually over all these 1854 diagnosis labels and see what categories can be made. This is very time-consuming and costly and therefore not affordable in this project.

3. The third option is to build a cluster embedding algorithm that can cluster the labels based on the text. This way, an algorithm is merging the labels, and clinicians do not need to do this work manually. However, it will take

a lot of time to build this algorithm, and before it can be used it still needs to be manually evaluated by clinicians, to see if it actually works well. This option was also too time-consuming and costly for this research.

4. Another option is to match diagnosis labels to ICD-10 codes. These codes are internationally known, and are organized in a hierarchy. In this way, once the ICD-10 codes of the diagnoses are available, certain codes can be put together in a category. The model will perform better since the amount of different labels is reduced and there exist more examples per category to learn from. Besides this, the model will still be of support for the clinicians in the ER since predicting a category is already of great value.

Considering time limits, for this project option 4 seemed the best option. When ICD-10 codes were available for the diagnoses in the data, a categorisation tool was used to put similar ICD-10 codes into the same category. The tool used for this is The Clinical Classifications Software Refined (CCSR) for ICD-10-CM Diagnoses [for Healthcare Research and Quality, 2022]. The CCSR tool classifies each ICD-10 code into a category. The size and distribution of the data after this categorisation can be found in the next subsection.

After extracting and categorizing the data, a clinician from the ER at the LUMC stated that not all categories from the CCSR tool were actual diagnoses because some of them were signs or symptoms. For this project, it is wanted that the model predicts diagnoses and not symptoms, and therefore these categories were left out. Furthermore, it was decided to leave out all diagnosis labels that still occurred less than 10 times in the data. As stated before, the model cannot learn from so few examples. It is realized that this is a form of bias. If the model is put into practice, patients might come in that do belong to one of these categories, but the model will not be able to predict this category since they were removed from the training dataset. However, this effect will be minimal, because the fraction of data points that were removed was less than 1 percent of the data.

3.4 Data size and distribution

After extracting the data, categorizing the labels, and leaving out considered labels, the amount of HPI texts left is 29871. The number of unique diagnosis labels (categories) now is 132. The full list of diagnosis labels can be found in Appendix A. The mean number of labels per HPI text is now 1.10. The distribution of the number of labels per HPI can be found in figure 1. Figure 2 shows in how many HPI texts the top 50 of diagnosis labels occur. In table 3 it can be seen that the labels are quite imbalanced since the number of documents per label differs quite a bit. However, since this dataset is multi-label, the sum of the frequencies of each label does not add up to the total number of HPI texts, since texts can contain more labels than one.

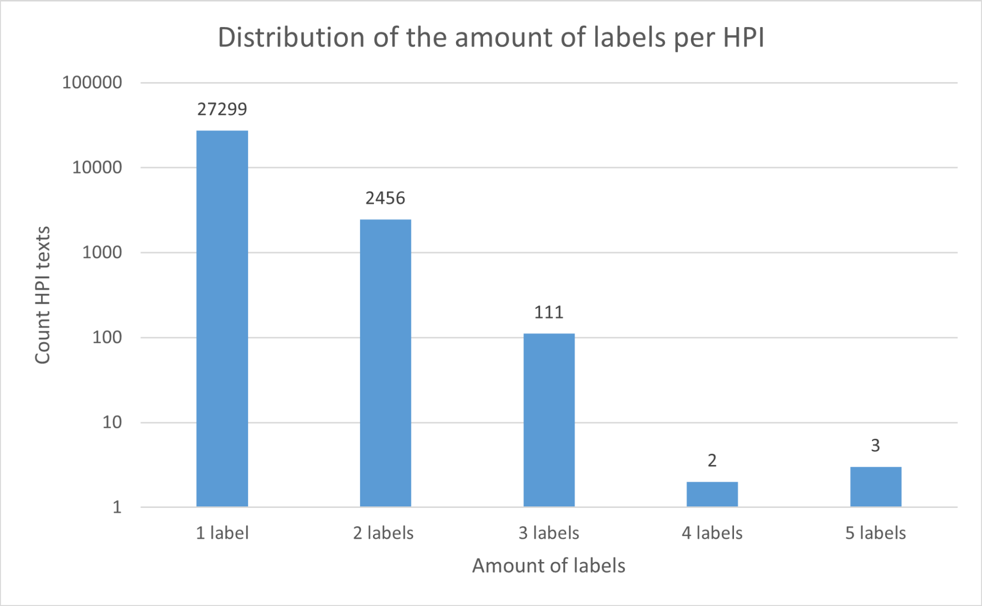


Figure 1: Distribution of the number of labels per HPI text, on a logarithmic y scale.

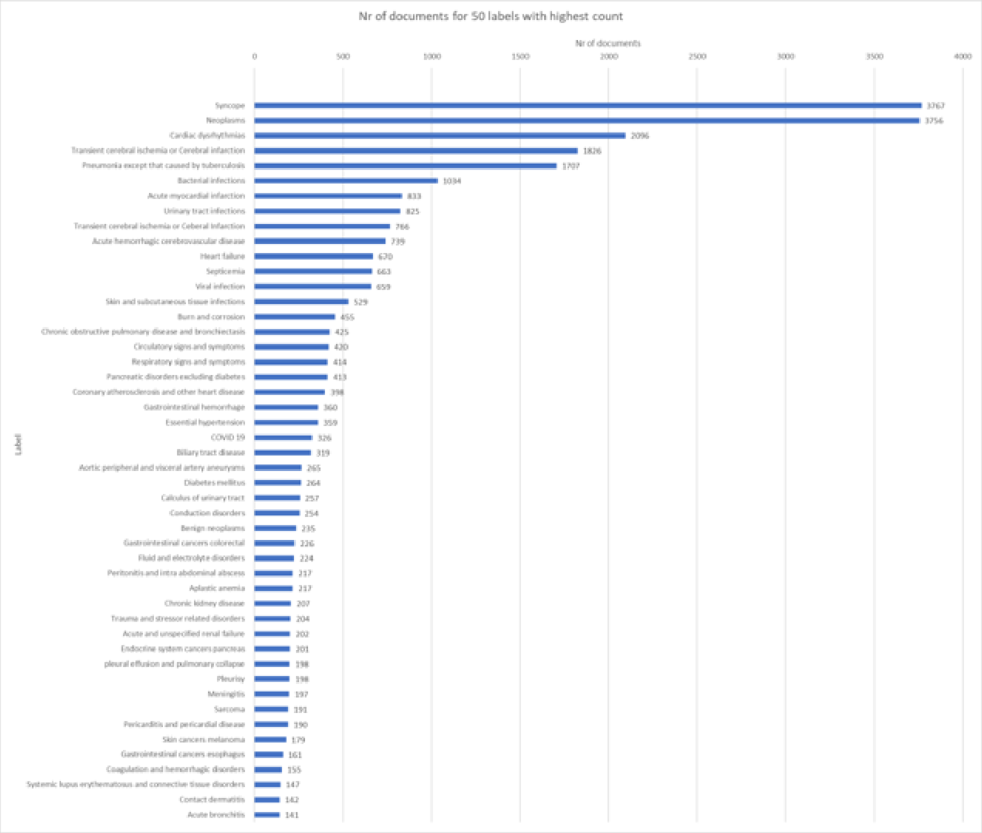


Figure 2: Count of the 50 most frequent labels.

Properties of labels	**Count**
Total count	132
Mean occurrence	246.72
Min occurrence	10
25 percentile	29.75
50 percentile	82.50
75 percentile	206.75
Max occurrence	3707

Table 3: Properties of the data labels

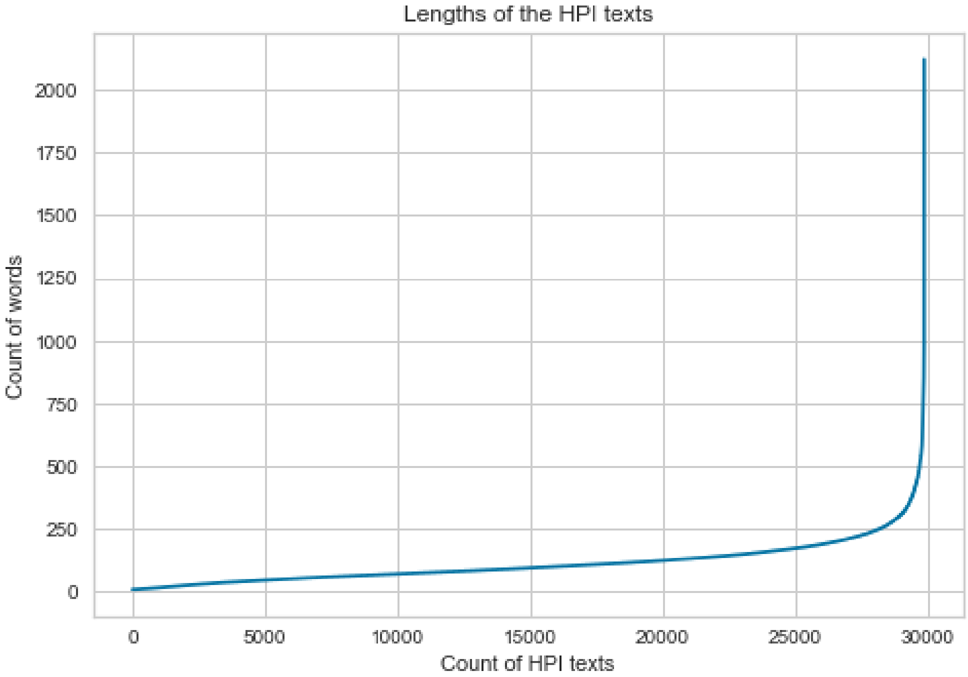


Figure 3: Lengths of the HPI texts.

Figure 3 shows the lengths of the HPI texts. Most of the texts are no longer than 250 words.

4 Methods

This thesis project focuses on making a prediction model to predict diagnoses of patients in the emergency room with the texts of History of Present Illness interviews, found in GP letters. Section 2 reviewed different classification techniques and deep learning models to use for this task. It was found that sequence-to-sequence models and BERT-based models might be appropriate models to use in this situation. The main questions now remain how these models perform on a real-life, imbalanced dataset and if they can be of support to doctors in the ER of a hospital. Therefore, it is necessary to find out if it is possible to predict diagnoses from HPI texts and if these two types of models will achieve high performance on this task together with a sort of explainability. For this, the models will be compared to a baseline model, a simple text classification model. Furthermore, the BERT-based model that has been pre-trained on Dutch medical notes (MedRoBERTa.nl) will be compared to a BERT model that has been pre-trained on Dutch general notes (BERTje), to see if the pre-training on medical notes has a higher impact than pre-training on general notes, and if the MedRoBERTa.nl model generalizes to other datasets, in this case, the dataset described in section 3. In this section, the methodology for answering these questions will be explained in more detail. Section 4.1 will explain the baseline model, section 4.2 the sequence-to-sequence model, and section 4.3 the BERT-based models. The explainability methods of the models will be explained in section 4.4, and the evaluation of the models in section 4.5.

4.1 Baseline model

The approach used as a baseline to compare the more complex models to is the Term Frequency - Inverse Document Frequency (TF-IDF) approach, together with a supervised machine learning classification model. TF-IDF is a good option because of its simplicity and the fact that it is often used for text classification [Artama et al., 2020]. The term frequency (TF) defines the frequency of a specific term in a document, the inverse document frequency (IDF) assigns a higher weight to words that only occur in a few documents. The TF-IDF approach measures therefore the importance of a term and can help identify if a specific term can be matched to a certain label. The TF-IDF creates feature vectors, which can be fed into a general machine learning classification model. The Linear-kernel Support Vector Machine (SVM) is chosen as the classification model. More options were considered, including a Decision Tree and a Logistic Regression classifier, but the SVM performed the best in a small experiment and therefore this model was chosen as the baseline classification model.

Before feature vectors were created with TF-IDF, the texts needed to be preprocessed. This cleaning of the text is explained in more detail in section 5.1. After cleaning up the data and creating the feature vectors, these vectorised documents could be fed into the SVM classifier. Because the dataset is multilabel, the one-vs-rest approach was chosen for the classifier. One-vs-rest means that instead of one classifier, there are as many classifiers as there are labels. For each classifier, the label is fitted against the other labels. Now, multiple independent classifiers are built and for each instance, every classifier predicts a probability score for that label in comparison to the rest of the labels. Another option is to use one-vs-one classifiers. With this approach, independent classifiers are built to compare every label with every other label in the dataset. Considering the fact that the dataset of this project consists of 132 different labels, this would result in 8646 classifiers and therefore the one-vs-rest approach was chosen. With multi-label classification, a threshold then needs to be set. If the probability of the one-vs-rest classifier then exceeds this threshold, this label is predicted by the model. If it does not exceed this threshold, the model will not predict this label.

4.2 Sequence-to-sequence model

A sequence-to-sequence model converts a sequence of text of one domain into a sequence of text of another domain. For this project, the input sequences consist of the HPI texts and the output sequences of the diagnosis labels. A seq2seq model consists of three parts: an encoder, a decoder and the class where these come together. The encoder turns the input into an internal hidden vector, which contains an input item plus its context. The decoder then takes this vector and turns it into the desired output, considering the context. Both the encoder and the decoder are separate Recurrent Neural Network (RNN) models, combined in the seq2seq class.

The seq2seq model was built using the PyTorch library. The encoder and the decoder were both made up of a bi-directional Gated Recurrent Unit (GRU) layer. The bi-directional GRU layer consists of two independent RNNs, where one takes as input the sequence in sequential order, and the other takes as input the sequence in reverse order. The outputs of these separate models are then concatenated.

One of the limitations of the seq2seq model is that it in general only works well and fast enough with texts of a maximum of 200 words. Longer texts can be used as input, but it is not recommended for the performance of the model. Most HPI texts in the dataset of this project are under 200 words, but some texts are over 200 words. This means that these data points either need to be deleted from the dataset, that these texts need to be cut off after word number 200, or that a summarization model is needed to first summarise these longer texts. The last option was not considered because of time constraints, but the first two options would mean that a lot of data would be lost. It was decided to deal with this differently. All HPI texts were split into different buckets (one bucket for texts with a length of fewer than 50 words, and one bucket for texts between 50 and 100 words, one for over 100 words and under 150 words, one bucket for texts between 150 words and 200 words, and one for the texts with a length over 200 words). These buckets were then fed into the model one by one, starting with the smaller texts. In this way, only with the last bucket it takes the model a long time to train.

Another limitation of the seq2seq model for this project is that, after categorizing the diagnoses, most labels are no longer a long sequence of text. It remains to be seen if the seq2seq model is the right choice in this situation, where the output sequence most of the time consists of only one or few words.

To pre-process the text before feeding it into the model, the same tokenizers were used for the BERTje model, which is explained in the next subsection. All hyperparameters of the model will be further explained in section 5.

4.3 BERT-based models

The baseline model of this project is very simple, simply counting occurrences of terms. As discussed in section 2, recently newer, more advanced language models have been created. One of the most promising and most recent is the Transformer. This is currently the best-performing type of model. For this project, the BERT model from Huggingface transformers will be used. A lot of different pre-trained BERT models exist, trained on different types of text in different languages. The models are openly available to download, and can therefore easily be used to finetune it on a new task on a new dataset.

As stated in section 2.3.2, one of the best BERT models for the Dutch language is BERTje. BERTje is trained on Dutch general texts, found in books, news articles, and Wikipedia pages. As stated in section 2.1, medical language can be quite different from general language, and therefore it will be interesting to see if the BERTje model will perform well on this task and this dataset.

Recently, a medical Dutch BERT model has been built, MedRoBERTa.nl.
MedRoBERTa.nl is trained on medical notes, including HPI notes, notes from doctors and nurses, letters from the hospital to the patient, and emergency room reports. Since this BERT model is pre-trained on similar texts as the dataset of this project, it will be interesting to see if the performance of MedRoBERTa.nl will be significantly greater than the performance of BERTje.

Transformer models have an encoder-decoder architecture, just like the sequence-to-sequence model. BERT is just an encoder and does not have a decoder. The pre-trained BERT model can read and process the text input but needs to be finetuned on a task to produce an actual prediction. To make the pre-trained BERT models ready to use for the task of this project, the output layer was changed to a classifier output layer which has an option for multi-label classification. The output layer used comes from the BertForSequenceClassification model, out of the Huggingface Library [HuggingFace, 2020].

A limitation of the BERT language model is that it can only process input texts no longer than 512 tokens. The dataset of this project does contain a few texts that are longer than 512 tokens, 146 texts to be precise, 0.49 percent of the texts. Compared to the seq2seq model, this is already a lot less documents that pass the maximum length. Since there are not so many texts exceeding this length, it was decided to truncate these texts to the length of 512 tokens. In this way, the data points are not lost, and these shorter texts will probably still contain enough information. Besides this truncation, it was considered to use a summarization model, to not lose any important information which is stated in the text after the 512 tokens. However, this would take some time to implement, and for the number of texts we have that exceed this 512 tokens limit, it was decided not worth implementing an extra model.

BERT models can become very large, with lots of parameters. This is another limitation of the language model because it can become very slow and requires a lot of computation because of its size. For this problem, a GPU is available to make the computations faster. However, to finetune and test the models on a dataset will still take a large amount of time, even when using a GPU. The models need to run for hours before they produce predictions. This is a limitation that the project just needed to work with, leaving the models running for multiple days.

To tokenize the data before feeding it into the models, the tokenizers from the models themselves were used. The Huggingface library provides both the BERTje and the MedRoBERTa.nl models with their tokenizers. The hyperparameters of the models are summed up in section 5.

4.4 Explainability

As stated before, besides a good performance, the models also need to be able to explain the black box: what is going on inside. If a model does not have a very high performance, clinicians in the ER need to understand what parts of the text the models focused on to come to a certain conclusion, to know whether the decision is reliable or not.

The BERT models have an intrinsic explainability, named attention. The model gives more attention to certain words, and after making a decision, these attention weights can be visualized with a special library called BERT Viz. The BERT Viz library visualizes attention in different ways: a head view, a model view, and a neuron view. It is quite hard to understand these visualizations, and not easy to quickly see what words the model paid more attention to. Therefore, it was decided for this project to not use intrinsic explainability but to use an extrinsic approach. The extrinsic explainability model used is LIME. As explained in section 2.4.2, LIME can be applied to any learning model and has a very easy and good way to immediately see what words the model paid most attention to, to come to a certain prediction. The LIME tool was therefore applied, to understand why the models came to a certain conclusion and to make the models a better support for the clinicians in the ER.

4.5 Model evaluation

To evaluate the model, a train-validation-test split will be implemented. All models are trained on 80 percent of the data, validated on 10 percent of the data, and in the end tested on 10 percent of the data. In this way, it is sure that the model is tested on new data that it has not seen before, and therefore the chance of getting a high performance because of overfitting will be smaller.

As stated in section 2.5, the micro and macro F1 scores are commonly used metrics for multi-label classification, so the evaluation of the models will be done by calculating these scores on the test set for each model. To better understand the performance of the models, the F1 score is also calculated per label. In this way, it can be seen if there are specific labels for the model that are more difficult to predict.

As will be explained more in the upcoming chapter, there will be made a few smaller samples of the data, to see if the models can predict specific diagnoses that are more difficult for doctors to distinguish from another. With these smaller samples, it is interesting to see the confusion matrix besides the F1 scores. In a confusion matrix, it is possible to see what labels were predicted as what label by the model. In this metric, it is easy to see what labels are predicted wrong, and which other labels are then predicted instead of the actual true label.

5 Experimental set-up

This section describes the setup of the experiments ran on the dataset described in section 3. All models described in section 4 were trained on the HPI texts to predict the corresponding diagnosis label(s). In section 5.1, the hyperparameter settings of the baseline model will be explained. Section 5.2 describes the settings used for the sequence-to-sequence and section 5.3 for the BERT models. Section 5.4 describes in more detail the experiments on the different samples of the dataset.

5.1 Baseline model

As was described in section 4.1, the baseline model consists of the TF-IDF vectorizer and the SVM classifier. The TF-IDF vectorizer is implemented with the scikit-learn library. Before building the feature vectors, the text was first preprocessed more. It was decided to remove special characters and stopwords since they are not containing any important information. For this, the NLTK library was used with their default Dutch stopword list. However, some words in this stopword list were considered important for this dataset (for example the word "not"). Those words were specifically removed from the stopword list first, so they would stay in the texts.

The TF-IDF score can be calculated for single words, or multiple words together. Even though single words are already very informative, it was decided to also implement bigrams and trigrams. This was decided because, for this dataset, bigrams can have very different meanings from unigrams. For example, using only unigrams can result in the word "sick", whereas using bigrams could result in "feeling sick" and trigrams in "not feeling sick". The difference of meaning here is huge between the trigram and the uni- or bigram, and because these types of bigrams and trigrams are very important in the dataset, it was decided to not only consider unigrams.

Furthermore, it was decided to leave out the TF-IDF scores for words that occur fewer than 3 times in the dataset. The reason for this is that sometimes the texts will contain spelling errors and in this way, this noise will be left out. Besides this, words that do not occur at least 3 times in the dataset might not be informative enough and will make it harder for the model to learn from. All other parameters were set to the scikit-learn's library default values.

For the SVM classifier, the following parameters were chosen. First, the choice was made to use a one-vs-rest classifier instead of a one-vs-one. This choice was explained in more detail in the previous section, and seemed to be the most logical option. As for the kernel of the classifier, multiple options are available, namely linear, polynomial, rbf and sigmoid. It was decided to use the linear kernel because this fitted best on this dataset. The multi-label classification threshold, as explained in the previous section, is set to 0.5. This means that if the probability of the label vs the rest of the labels is higher than the probability of the rest vs the one, this label is predicted by the model. In table 4, an overview of the parameters chosen for the baseline can be found.

Parameter	**Baseline**
Vectorizer	TF-IDF
ngrams	Unigrams, bigrams, and trigrams
Minimum frequency	3
Maximum frequency	95%
Classifier	SVM
Multi-label classifier	One-vs-rest
Kernel	Linear
Threshold	0.5

Table 4: Overview parameter choices baseline model

5.2 Sequence-to-sequence model

The sequence-to-sequence model was coded with the PyTorch library. Before feeding the text into the model, special character removal and stopword removal were done. After this, the tokenizer used for the BERTje model was also used for the seq2seq, since this tokenizer tokenizes the text into subwords instead of only into words. Tokenization into subwords is important because for example the word "running" will be tokenized into "run" and "#ing". In this way, the word run will be counted, instead of only the word running, so the model can understand this is the same meaning as when a document contains the word run. Furthermore, it was decided to set the minimum frequency of words here to 10. This is to let the model have enough information about all the words that it sees in the training data. The infrequent words and words that appear for the first time in the test set were converted to "UNK", a token for unknown words. For the training of the model, a batch size of 4 was used, due to memory issues with bigger batch sizes. The model was finetuned using the AdamW optimizer, with an initial learning rate of 0.001. The loss of the model was calculated with a cross-entropy loss function and the number of epochs was 10. Validation showed that after 10 epochs, the model performance did not increase anymore on the validation set and the loss did not decrease any more.

Both the encoder and the decoder of the model consist of a bi-directional GRU layer and a linear layer, with dropout set to 0.5. Dropout is used to reduce overfitting, randomly dropping out certain layers of the model, so it will need to find other ways to reduce the loss and cannot simply rely always on the same path. This way the training process becomes noisy. A dropout of 0.5 means that a layer has a 0.5 probability of being dropped and is mostly used as a dropout rate [Baldi and Sadowski, 2013].

5.3 BERT models

The BERT models were also coded with the PyTorch library. Both the BERTje and the MedRoBERTa.nl models were available for download via the Huggingface library, including their tokenizers. For all parameters, it was decided to use the same for BERTje as for MedRoBERTa.nl. In this way, the results of the models can be compared, and the only difference will be the texts which the models were pre-trained on. This means that any difference in results will be because of this pre-training, and cannot be attributed to any other difference between the two models.

For training, a batch size of 4 was used, again because of memory constraints. The max length of the text was set to 512, and all texts longer than this were truncated. The finetuning was done with the AdamW optimizer and a learning rate of 0.00001. The losses are calculated with the Binary cross-entropy (BCE) with logits loss because this generates the probabilities per label. The default dropout of the BERT models is 0.1. For the BERT models, the multi-label threshold was also set to 0.5, for the same reason as why this threshold was chosen for the baseline model. Furthermore, the number of epochs was 4, after this number the performances did not increase anymore. In table 5, an overview is shown of the parameter choices for the seq2seq model and the BERT models.

Parameter	**Seq2Seq**	**BERTje**	**MedRoBERTa.nl**
Tokenizer	BERTje tokenizer	BERTje tokenizer	MedRoBERTa.nl tokenizer
Max length	N/A	512	512
Optimizer	AdamW	AdamW	AdamW
Initial learning rate	0.001	0.00001	0.00001
Batch size	4	4	4
Loss function	Cross entropy loss	BCE with logits loss	BCE with logits loss
Epochs	10	4	4
Dropout	0.5	0.1	0.1

Table 5: Overview parameter choices seq2seq model and BERT models

5.4 Samples

Since the dataset contains a lot of different diagnosis labels, and some labels do not occur very frequently in the dataset, it can be difficult for a language model to have a high performance for all labels. A clinician of the ER of the LUMC hospital in The Netherlands described that a model that can choose from a small number of specific diagnoses from only the HPI text will already be helpful in the ER. In this situation, once a patient is admitted to the ER, the doctors could do a quick assessment to determine which diagnoses are most likely, after which the trained model that discriminates between those diagnoses can then run on the text of the HPI interview. As will be shown in the results section, some specific labels were hard for the models to distinguish from one another. However, when trained on only those labels, the distinction became easier. Therefore, it was decided to set up smaller samples of the data, consisting of small groups of diagnoses that are difficult for clinicians in the ER to distinguish from one another. It will be interesting to see if the language models have fewer difficulties with distinguishing these specific diagnoses for patients. The generated samples with their distribution can be found in table 6.

Sample nr	**Diagnosis labels**	**Nr of documents**	**Total**
1	Pneumonia except that caused by tubercolosis	1707	2708
	Acute phlebitis thrombophlebitis and thromboembolism	13	
	Heart failure	670	
	Covid-19	326	
1A	Heart failure	669	995
	Covid-19	326	
2	Gastrointestinal and bilinary perforation	31	2094
	Pancreatic disorders excluding diabetes	413	
	Biliary tract disease	321	
	Acute myocardial infarction	833	
	Arterial dissections	13	
	Peritonis and intra abdominal abscess	217	
	Aortic peripheral and cisceral artery aneurysms	266	
2A	Biliary tract disease	321	587
	Aortic peripheral and cisceral artery aneurysms	266	
3	Acute hemorrhagic cerebrovascular disease	739	935
	Meningitis	196	

Table 6: Generated samples in cooperation with clinicians from the ER including the number of documents.

6 Results

In this section, the results of the experiments discussed in the previous sections are reported. The experiments consist of training a baseline model and three deep learning models on the dataset explained in section 3, and on samples of this data that were shown in section 5.4. Section 6.1 shows the results of the models on the full dataset, section 6.2 shows the results of the models on the samples, and section 6.3 shows the findings of the explainability model.

6.1 All data

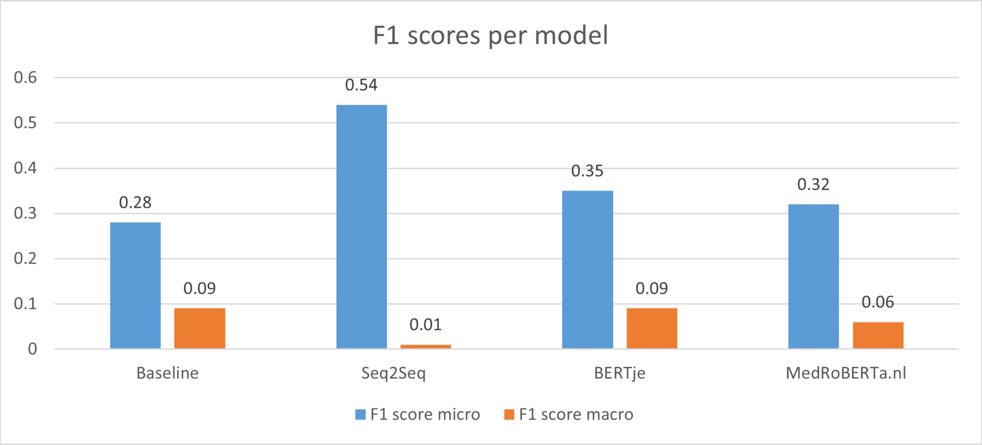


Figure 4: The mean micro and macro F1 score calculated for each model on the whole dataset.

Because the amount of labels is 132, the full table of all labels and their F1 scores per model can be found in appendix A. The ten labels with the highest F1 scores are shown in table 7.

Label	**Baseline**	**Seq2Seq**	**BERTje**	**MedRoBERTa.nl**
Burn and corrosion	0.62	0	**0.76**	0.54
Syncope	0.53	0.18	**0.56**	0.50
Cardiac dysrhythmias	0.43	0.1	**0.51**	0.49
Mental disorders	**0.56**	0.14	0.35	0.39
Neoplasms	0.19	0.14	**0.49**	0.46
Skin and subcutaneous tissue infections	0.33	0	0.41	**0.45**
Nerve and nerve root disorders	0.38	0	**0.44**	0.24
Cornea and external disease	0.20	0	**0.52**	0.33
Calculus of urinary tract	0.26	0	**0.39**	0.34
Essential hypertension	0.30	0	**0.36**	0.29

Table 7: F1 score per label per model for the 10 labels with the highest scores.

The F1 scores reported previously are calculated assuming that the model outputs only the labels where the probability exceeds the threshold. Alternatively, we can take the K most likely diagnoses according to the model, for each HPI text. A true positive is then recorded if the true diagnosis is within this top K of diagnoses, output by the model. Figure 5 shows the mean micro and macro F1 score per model, calculated for different K's. K is here the number of labels the model gives as output. The expectation is when a model outputs for example 10 labels (K = 10), there is a higher chance that the right label is within these 10 labels and therefore will the performance of the model be higher.

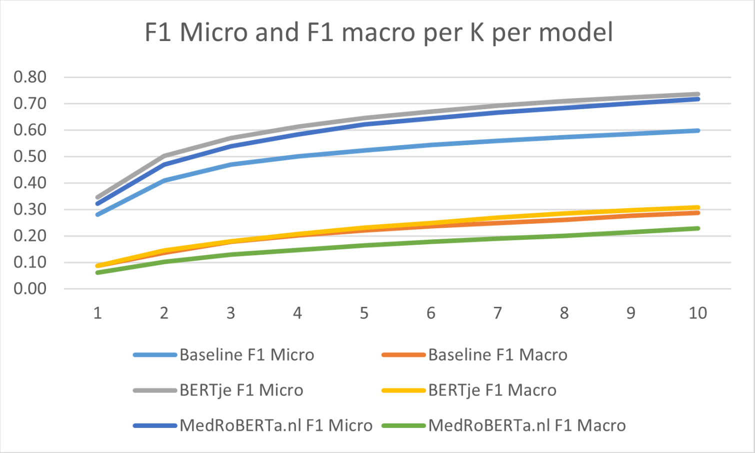


Figure 5: The mean micro and macro F1 score calculated for each model per K for a range of 1-10.

6.2 Data samples

The confusion matrices of all models of the samples can be found in Appendix B.

6.2.1 Sample 1

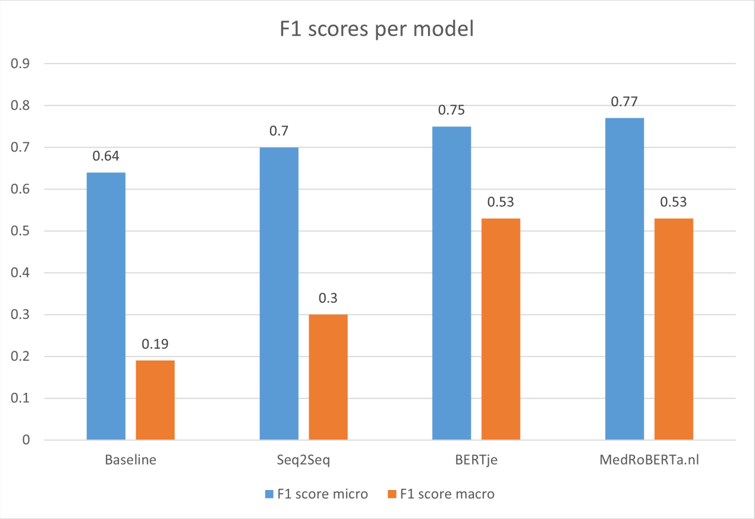


Figure 6: The mean micro and macro F1 score calculated for each model on sample 1.

Label	**Baseline**	**Seq2Seq**	**BERTje**	**MedRoBERTa.nl**	**Nr of documents**
Pneumonia	0.78	0.47	0.82	**0.83**	1707
Acute phlebitis thrombophlebitis and thromboembolism	0	0	0	0	13
Heart failure	0	0.34	0.64	**0.71**	670
Covid-19	0	0	**0.65**	**0.57**	326

Table 8: F1 score per label per model of sample 1

6.2.2 Sample 1A

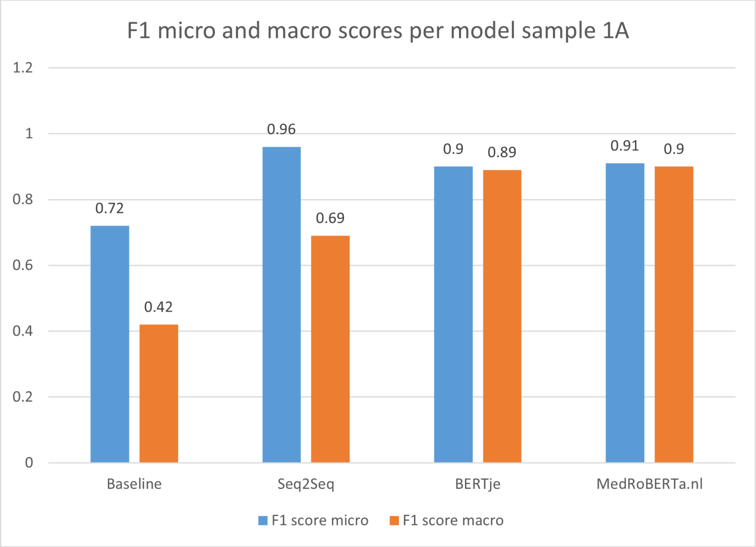


Figure 7: The mean micro and macro F1 score calculated for each model on sample 1A.

Label	**Baseline**	**Seq2Seq**	**BERTje**	**MedRoBERTa.nl**	**Nr of documents**
Heart failure	0.84	**0.96**	0.92	0.94	669
Covid-19	0	0.11	0.85	**0.87**	324

Table 9: F1 score per label per model of sample 1A

6.2.3 Sample 2

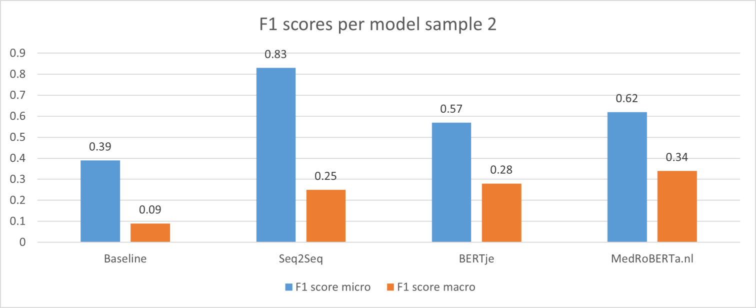


Figure 8: The mean micro and macro F1 score calculated for each model on sample 2.

Label	**Baseline**	**Seq2Seq**	**BERTje**	**MedRoBERTa**.nl	**Nr of documents**
Gastrointestinal and biliary perforation	0	0	0	0	31
Pancreatic disorders excluding diabetes	0	0.08	0.51	**0.55**	413
Biliary tract disease	0.03	0.02	0.22	**0.31**	319
Acute myocardial infarction	0.56	0.8	0.86	**0.87**	833
Arterial dissections	0	0	0	0	13
Peritonitis and intra abdominal abscess	0	0.03	0	**0.23**	217
Aortic peripheral and visceral artery aneurysms	0.04	0.08	0.38	**0.44**	265

Table 10: F1 score per label per model on sample 2.

The confusion matrices of the models can be found in Appendix D.

6.2.4 Sample 2A

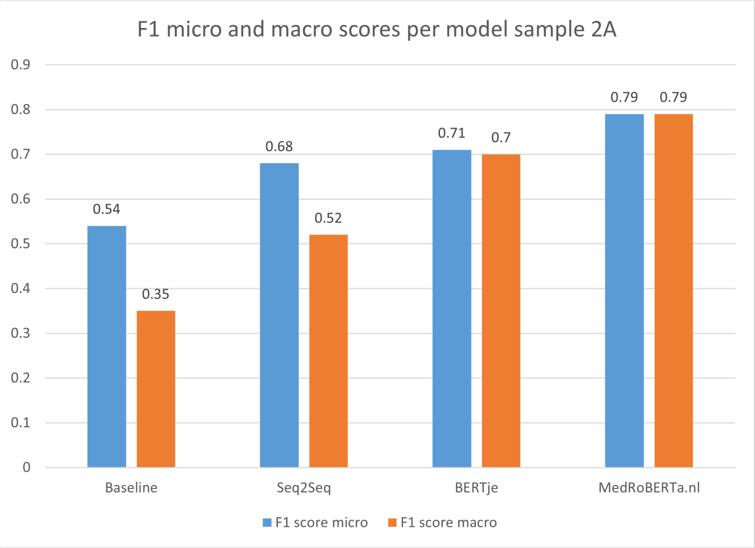


Figure 9: The mean micro and macro F1 score calculated for each model on sample 2.

Label	**Baseline**	**Seq2Seq**	**BERTje**	**MedRoBERTa.nl**	**Nr of documents**
Biliary tract disease	0.7	0.51	0.74	**0.83**	321
Aortic peripheral and visceral artery aneurysms	0	0.5	0.67	**0.75**	266

Table 11: F1 score per label per model on sample 2A.

6.2.5 Sample 3

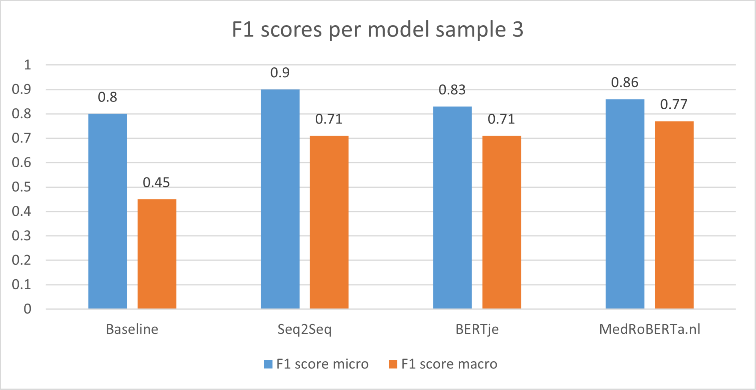


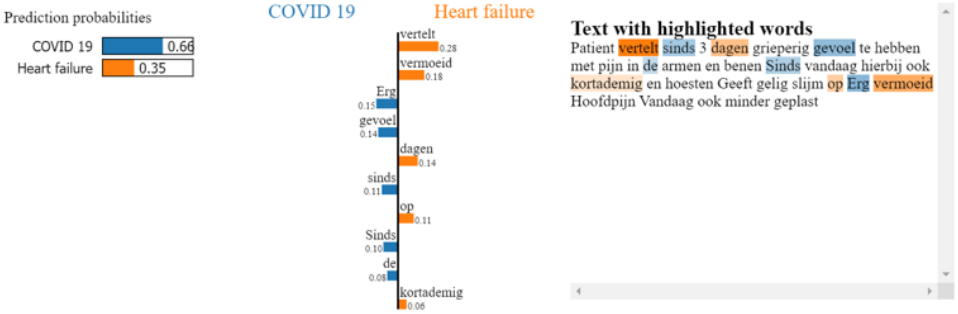
Figure 10: The mean micro and macro F1 score calculated for each model on sample 3.

Label	Baseline	Seq2Seq	BERTje	MedRoBERTa.nl	Nr of documents
Acute hemorrhagic cerebrovascular disease	0.89	0.98	0.9	0.91	739
Meningitis	0	0.28	0.51	**0.63**	196

Table 12: F1 score per label per model on sample 3.

6.3 Explainability

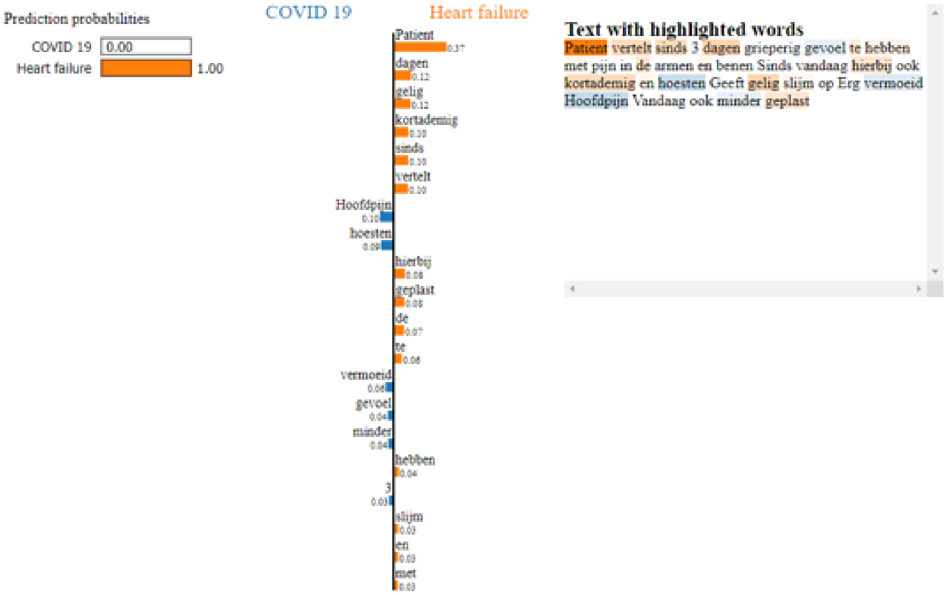
To show how the explainability of the models with LIME looks, two examples are shown here. The explainability of the same HPI text, with the actual label “Heart failure”. In figure 11 the explainability of the baseline is shown and in figure 12 the explainability of MedRoBERTa.nl.



Text with highlighted words

Patient vertelt sinds 3 dagen grieperig gevoel te hebben met pijn in de armen en benen Sinds vandaag hierbij ook kortademig en hoesten Geeft gelig slijm op Erg vermoeid Hoofdpijn Vandaag ook minder geplast

Figure 11: Visualisation of the explainability of the baseline model using LIME.



Text with highlighted words

Patient vertelt sinds 3 dagen grieperig gevoel te hebben met pijn in de armen en benen Sinds vandaag hierbij ook kortademig en hoesten Geeft gelig slijm op Erg vermoeid Hoofdpijn Vandaag ook minder geplast

Figure 12: Visualisation of the explainability of the MedRoBERTa.nl model using LIME.

7 Discussion

This section will discuss the results presented in section 6. The experiments included training a baseline model and three deep learning models on a multi-labelled dataset. Section 7.1 will discuss the analysis of the results in further detail. Section 7.2 will answer the research questions based on this analysis. Section 7.3 will discuss the limitation of the research, accompanied by options for further research on the topic.

7.1 Results analysis

This section provides an analysis and discussion of the results shown in section 6. Figure 4 shows the macro and micro F1 scores per model. As discussed in section 2.5, the macro F1 score is computed on the individual labels first and then averaged. The micro F1 score is computed on the individual data points first. This means that the micro F1 score is influenced by labels that have more instances than other labels, and the macro F1 score is not.

The overall micro F1 and macro F1 scores are low, which means that none of the models is capable of correctly predicting the labels in the whole dataset. Against expectations, the macro F1 scores of the deep learning models are not higher than the macro F1 score of the baseline model, which means that the deep learning models do not a well performance on a bigger amount of labels than the baseline model does. The micro F1 score is higher for the deep learning models compared to the baseline model. This means that the deep learning models do predict a bigger number of data points correctly. As can be seen in figure 4, for all models the micro F1 score is significantly higher than the macro F1 score. This could mean that the models do well on predicting the bigger classes, but not well on predicting the classes with fewer data points. When looking at appendix A, it can be seen that this is indeed the case. Most models perform better on for example the labels Syncope and Neoplasms, but for smaller labels, the F1 score of all models is simply 0. When further inspecting the F1 score per label, it can be seen that both the Baseline model and the BERTje model perform the best on most classes. MedRoBERTa.nl and seq2seq perform fewer times the best for a class, whereas the seq2seq model seldom performs the best for a class.

The dataset is imbalanced, and this could be the reason for the aforementioned problem of the lower macro F1 scores. This would mean that on the samples explained in section 5.4, the models should have better performance since those datasets are less imbalanced than the whole dataset. To understand why these F1 scores are so low, the F1 scores of the samples can be inspected. When inspecting the F1 scores of sample 2A (table 11), where the class is least imbalanced with a distribution of 321 data points for class A and 266 data points for class B, it can indeed be concluded that the macro F1 scores are higher for every model, and also come closer to the micro F1 scores. This means that when the class is less imbalanced, the models do perform better. However, when looking at the confusion matrices of the samples in Appendix B, it can be concluded that the baseline model still only predicts the biggest class. The same goes for the sequence-to-sequence model. It does not solely predict the biggest class, but it does most of the time. However, when looking at table 11, it can be seen that both BERT models do predict both classes and even wrongly predict both classes almost as much. This cannot be only concluded for sample 2A, but all samples in general. The baseline model and the sequence-to-sequence model only perform well on a specific class (for example, the class Acute myocardial infarction in sample 2, seen in table 10), whereas the BERT models perform better on almost all classes. This means that, even though all models do not perform so well on the whole dataset, the BERT-based models do learn something from the data when the classes are less imbalanced, whereas the baseline and the seq2seq model do not seem to learn anything. An example of this can be found in figure 11 and figure 12. This figure shows an HPI text from sample 1A with the label Heart failure. Figure 11 shows that the prediction of the baseline was Covid-19 and that the baseline model based the prediction of the label Covid-19 on the words “erg”, “gevoel”, “sinds” and “de”, which mean “very”, “feeling”, “since” and “the”. A doctor would never base their prediction on these words. However, as can be seen in figure 12, the MedRoBERTa.nl model based the prediction of Heart failure on for example the words “kortademig”, “gelig slijm” and “geplast”, which mean “shortness of breath”, “yellow phlegm”, and “peed”, which is backed up by a clinician from the ER that he would base the prediction of this text also mostly on these specific symptoms. Furthermore, you can see that the MedRoBERTa.nl model also recognizes the words “headache”, “coughing”, and “tired” as features for the label Covid-19.

When comparing the MedRoBERTa.nl model with the BERTje model, the following can be observed. When trained on the full dataset, even though the differences are not very big, BERTje outperforms MedRoBERTa.nl on most labels. This also explains the slightly better F1 micro and F1 macro score of BERTje compared to MedRoBERTa.nl. But when trained on the smaller samples, it can be observed that MedRoBERTa.nl outperforms BERTje in all samples on all labels. When trained on a big, imbalanced dataset, the fact that MedRoBERTa.nl was pre-trained on medical texts compared to the general texts that BERTje was pre-trained on does not seem to matter. However, on the task of predicting a label from a smaller sample, it does seem that MedRoBERTa.nl learns more than BERTje does. This can also be seen in the differences between the confusion matrices in Appendix B.

Looking at the sequence-to-sequence model’s performance, it can be concluded that this model is by far the worst performing model of all four models. A possibility of a reason for this bad performance is that after processing the diagnosis labels, the deleting and the categorization, there was not much of a sequence of text left for the model to predict. Before this processing, the output of the model was a long text, like a sequence. Now it is more like a specific class, consisting of only a few words. For this research, it can be concluded that the model does not learn much. For the whole dataset, as well as the samples, the model only performs above zero for the biggest label(s), and even on those labels, the model does not perform well. In table 9 it can be seen that the seq2seq model does outperform the other models on the label heart failure, in sample 1A. However, it performs very bad on the residual label, Covid-19, where the BERT-based models perform slightly worse on heart failure but way better on Covid-19. When inspecting the confusion matrices of this sample (Appendix B), it can be seen that the test data of the seq2seq model consisted of mostly the heart failure label, and therefore almost only predicting heart failure gave the seq2seq model this high performance on this specific label, but the low performance on the Covid-19 label.

Something that is also interesting to see is how much it affects the performance of the models when the models do not predict only a few labels with a high probability, but when it predicts a top k of labels with the highest probability. According to ER clinicians, a model is still of support when it predicts a list of 3, 5, or even 10 diagnosis labels, as long as the performance is then very good (a very high chance that the right diagnosis is within this list). The seq2seq model is not in this graph, since this model predicts a sequence and not probabilities per label. Looking at the micro F1 score, taking a k of 10 does affect the performance of the models significantly. It can also be seen that both BERT models increase more than the Baseline model, meaning that the BERT models predict the right diagnosis for a datapoint more often with a k of 10 than the baseline does. However, the micro F1 score is not much affected by taking a k of 10. This means that the smaller classes are still not predicted right mostly. The performance of the bigger classes is therefore affected by taking a bigger k, but the performance of the smaller classes is not. For the micro F1 scores, the BERT-based models do not outperform the baseline model anymore when taking a bigger k.

It is interesting to note that, during validation, some specific labels were confused for all models, when trained on the full dataset. An example of this was the confusion between the labels “Covid-19” and “Pneumonia except that caused by tuberculosis”. When trained on the full dataset, all models always predicted the label Pneumonia except that caused by tuberculosis when the actual label was Covid-19. However, when the models were trained on only the data of these two labels, both BERT-based models could distinguish these labels way better. Similar to sample 1A of the results. Therefore, it was decided to make these samples, to see how much the performances of the models would improve. Apparently, in a small dataset, the models are capable of achieving higher performances, and the models can better distinguish labels from each other than when more labels are involved.

7.2 Conclusion

The main question to answer with this thesis was how to make a model to predict diagnoses of patients to support clinicians in the emergency room. Several of sub-questions were suggested to answer this research question. These sub-questions will be answered in this section, after which the answer to the main research question is discussed.

Firstly the question was raised about what is required of a diagnosis prediction model from a clinical perspective to be helpful and of support to clinicians in the ER. As found in previous literature on predictive models in the medical domain, as well as in questionnaires answered by clinicians, it was found that firstly the performance of the model needs to be high enough. Diagnostic errors can result in serious harm, and therefore the model needs to be able to accurately predict the right diagnosis most of the time. However, since the model is never meant to replace a clinician, but only to support them, the performance does not need to be 100 percent, since the clinician is also still there to make the decision. To be of support, the model needs to be able to make specific decisions that are difficult for clinicians. For example, the decision between two specific diagnoses where the clinician might doubt between. So the models' performance needs to be high enough on the whole dataset, or on specific samples of diagnosis labels that are harder to distinguish from the first interview for clinicians. However, only the performance of the model is not enough. Clinicians clearly stated that the model also needs to have some sort of explainability, for the clinicians to trust the model. To accept the models' decision, the clinician needs to understand where the certain conclusion is coming from. If the model cannot explain itself, it might draw a conclusion from the wrong reasons, and therefore the decision cannot be trusted.

The second sub-question was what kind of metrics are best to use for the evaluation of the model. As stated in previous literature, the metrics that are best to use for a multi-label classification problem are the micro F1 score, the macro F1 score, and the F1 score per label. This is because the data is very imbalanced, and the F1 score can capture this. Besides this, the micro and macro scores are averaged (over either data points or labels), and can therefore say something about the performance of multi-label classification. The F1 scores per label show the performance of a model per specific label. This is interesting for the analysis of the model, whether the model performs well on all labels, or only on specific labels. Besides the F1 scores, the confusion matrices are interesting for further analysis of the results. However, with a dataset as big as the one of this research, this matrix would become too big and is therefore only interesting when analyzing the results of smaller samples of the data.

The third sub-question raised was about explainability. Whether an intrinsical model could be used or if it is necessary to add an extrinsic explainability model to the prediction model. Literature showed great potential in the attention mechanism, an intrinsic part of the architecture of both the BERT-based models and the sequence-to-sequence model. This mechanism gives weight to specific relationships among concepts, which can then be interpreted as the features that the model paid the most attention to. However, visualizing this attention results in an environment that can be very difficult to understand. The models need to be of support to clinicians, and therefore the explainability needs to be easily interpreted. Therefore, the intrinsic attention mechanism might not be the best fit for this research. The external model that turned out to be best visualizing the explainability of the models is LIME. As shown in figures 11 and 12, LIME visualisations are very easy to interpret the explainability. In one glance, the clinician can see which words in the HPI text the model paid the most attention to to come to a certain conclusion, and the clinician can then see for themselves whether this is a logical reason for the prediction.

Fourth, the sub-question was raised about whether a sequence-to-sequence model can be used for this task. This means, under the criteria of sub-question 1, that the performance of the whole dataset needs to be high or the performance on smaller interesting samples needs to be high, and the model needs to be explainable. In the first set of data that was used, the model needed to predict a lot of labels, formed in a sequence. In this way, the model performed quite okay, and with the use of an extrinsic explainability model, this model could be used for the task of this research. However, the data that was eventually used for this research, did not contain many diagnosis labels per text, and therefore not much of a sequence needed to be predicted. In this case, the performance of the model was very bad on the whole dataset as well as on all created samples. Therefore the seq2seq model is not the best choice for this task.

The last sub-question was whether a BERT-based model can be used for the task and if the pre-trained clinical Dutch language model MedRoBERTa.nl is generalizable to other datasets, and in particular to the dataset of this study. When training the model on the whole dataset, the MedRoBERTa.nl model does not achieve very high results. Against expectations, it performs slightly worse than the BERTje model, which is a similar model but pre-trained on general texts instead of medical texts. This would mean that pre-training the model on domain-specific texts would not necessarily result in a better performance on a task of that domain. However, the MedroBERTa.nl model does perform better than all other models when trained on smaller samples of the data. The model seems to be the best choice for the task of predicting a diagnosis when there are a few diagnosis labels to choose from. This then means that when the task consists of distinguishing diagnoses that are very similar where clinicians might have doubts, it is better to use a model that is pre-trained on domain-specific texts. However, since the performance only is slightly better than the BERTje model, it can also be concluded that pre-training a BERT-based model endlessly does not necessarily always end in much higher performance and is therefore not always the best option.

Coming back to the main research question of how to make a model to predict diagnoses of patients to support clinicians in the emergency room, it can be concluded that the MedRoBERTa.nl model is the best model to use in this situation. However, it is only of support for clinicians when used in combination with an extrinsic explainability model like LIME, and in the specific situation of when a clinician is in doubt between a small number of diagnoses. No model of this thesis is capable of predicting diagnoses when choosing from all available diagnosis labels. The MedRoBERTa.nl model had an average micro F1 score of 0.79 and an average macro F1 score of 0.66 (averaged over the samples), but a micro F1 score of 0.32 and a macro F1 score of 0.06 when trained on the full dataset.

7.3 Limitations and further research

This project was the first to research different prediction models to predict diagnoses of patients presenting in the emergency room from Dutch HPI texts. This study compared four different models for this task and evaluated these models on the F1 scores. The results found in this research are subject to several limitations. These limitations are acknowledged in this section, accompanied by further research that can be conducted to overcome these limitations.

7.3.1 Imbalanced dataset

A limitation of this study is the number of different diagnoses present in the dataset. Working with a real-life dataset like the one in this project means that the data might not look ideal. In this case, there are many different diagnoses a patient can have, and therefore the number of unique labels was very high. Also, a big part of the diagnoses appeared less than 10 times in the dataset, and a lot of these even appeared only one time. Real-life datasets results a lot of times in a highly imbalanced dataset, and no model can learn from examples that only appear one time in a dataset.

In section 3.3, multiple solutions were suggested to overcome this problem. However, they all include categorizing the diagnoses in smaller categories. A limitation of the categorization of the diagnoses is that it loses specificity. Predicting a category a patient's diagnosis falls into is of less support to a clinician than predicting a very specific diagnosis. Instead of categorizing the data, further research could inspect if there are specific models that might work well on imbalanced datasets. Or a different solution could be expanding the dataset. In further research, the same research could be done with the data of not only one hospital, but the data of multiple hospitals. In this way, the dataset might still be imbalanced, but there is a higher chance that all diagnosis labels will occur enough times in the dataset for models to learn from.

To overcome the problem of the imbalanced dataset, this study also created smaller, less imbalanced samples, to see if the models would perform better.

However, these samples are not generalizable to the real-life situation. A patient that comes into the ER can have any possible diagnosis, not just one out of a few. What could be interesting to find out is whether a two-step model would work in this situation. As shown in figure 5, the performance of the models did increase when predicting a top-k amount of labels. If you have a model where it is 100 percent sure that the right label is within a specific top-k of labels, the model could run another time, on only these top-k labels instead of all labels, to predict the right label from. Further research could be conducted to see if a two-step model like this would increase performances. Something else that could be done to overcome an imbalanced dataset is downsampling. With downsampling, you simply make the biggest labels less big, in order for the dataset to become more balanced. This was not done in this study because you then throw out potentially useful data, but in further research, it could be interesting to see whether this would be of advantage or not.

7.3.2 Input features

In this study, purposely only the HPI interviews were used to predict the diagnoses of patients in the ER. This means that the results of this study can say something about how predictive this first HPI interview is. But, clinicians take more into consideration when diagnosing a patient than only this first HPI interview. Examples of other matters that clinicians take into account in this decision are a physical examination, blood tests, medical history and additional examinations. Besides these, the clinician sees, hears, smells and feels the patient, on which they (unconsciously) might base conclusions. Further research could include these extra features that clinicians use, to maximize the performances of the models. This might result in models with higher performance in predicting the diagnosis of the patient but cannot say anything about the predictiveness of the HPI interview only.

7.3.3 Threshold

As explained in section 4.1, the threshold chosen for whether the model predicts a label or not is in this project set to 0.5. This is a default threshold since it is logical to say that if the probability of the label being right is higher than the probability of the label being wrong, the label is predicted. However, there are certain methods to find out the perfect threshold during validation. One method is using the Receiver Operating Characteristic Area Under the Curve (ROC AUC). This is a measure of how well the model can distinguish between the labels, based on the true positive rate and false positive rate. The Sklearn library has a function to compute the best threshold for the highest ROC AUC. Further research could find this optimal threshold and see if the performance of the models will significantly chance with this threshold.

7.3.4 Ethics

Last but not least, this study did not dive deeper into ethics. However, this is very important to consider when an artificial intelligence model is wanted to implement in a hospital because of two reasons. First, hospitals work with real patients and as discussed in section 1.1, diagnostic error can be of enormous effect on a patient. Second, the model will work with real patient data, which is strictly private data. Before a model can be implemented in practice, multiple ethical questions need to be considered. For example, who or what is responsible for the diagnostic error of the model? Does there need to be informed consent to use all patients' data? Another ethical subject to consider is bias. The model is trained on real-life data, and decisions made by humans. This bears the risk of biases. Biases can for example occur regarding age, sex, or race. Further research should consider all safety, legal, privacy and bias challenges.

References

[Artama et al., 2020] Artama, M., Sukajaya, I. N., and Indrawan, G. (2020). Classification of official letters using tf-idf method. *Journal of Physics: Conference Series*, 1516(1):012001.

[Baldi and Sadowski, 2013] Baldi, P. and Sadowski, P. (2013). Understanding dropout.

[Balogh EP, 2015] Balogh EP, Miller BT, B. J. (2015). *Committee on Diagnostic Error in Health Care; Board on Health Care Services; Institute of Medicine; The National Academies of Sciences, Engineering, and Medicine*. National Academies Press (US).

[Blinov et al., 2020] Blinov, P., Avetisian, M., Kokh, V., Umerenkov, D., and Tuzhilin, A. (2020). Predicting clinical diagnosis from patients electronic health records using bert-based neural networks.

[Bologna and Hayashi, 2017] Bologna, G. and Hayashi, Y. (2017). Characterization of symbolic rules embedded in deep dimlp networks: A challenge to transparency of deep learning. *Journal of Artificial Intelligence and Soft Computing Research*, 7:265–286.

[Bresnick, 2018] Bresnick, J. (2018). What is deep learning and how will it change healthcare? *Health IT Analytics*.

[Chapman et al., 2011] Chapman, W. W., Nadkarni, P. M., Hirschman, L., D'Avolio, L. W., Savova, G. K., and Uzuner, O. (2011). Overcoming barriers to nlp for clinical text: the role of shared tasks and the need for additional creative solutions. *Journal of the American Medical Informatics Association : JAMIA*, 18:540–543.

[Chen, 2020] Chen, L. C. J. Y.-W. (2020). *Deep Learning in Healthcare: Paradigms and Applications*, volume 171. Springer, 1 edition.

[Chen et al., 2019] Chen, M., Zhang, M., Li, R., Zhang, X., Zhao, H., and Zhang, S. (2019). A novel deep neural network model for multi-label chronic disease prediction. *Frontiers in Genetics — www.frontiersin.org*, 1:351.

[Chen et al., 2018] Chen, M. C., Ball, R. L., Yang, L., Moradzadeh, N., Chapman, B. E., Larson, D. B., Langlotz, C. P., Amrhein, T. J., and Lungren, M. P. (2018). Conclusion: A deep learning cnn model can classify radiology free-text reports with accuracy equivalent to or beyond that of an existing traditional nlp model. *Original research n Computer AppliCAtions Radiology*, 286.

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

[Dernoncourt et al., 2017] Dernoncourt, F., Lee, J. Y., Uzuner, O., and Szolovits, P. (2017). De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association : JAMIA*, 24:596–606.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., Google, K. T., and Language, A. I. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

[Education, 2020] Education, I. C. (2020). Deep learning.

[Ekbal et al., 2016] Ekbal, A., Saha, S., and Bhattacharyya, P. (2016). Deep learning architecture for patient data de-identification in clinical records.

[Esteva et al., 2017] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature Publishing Group*.

[for Healthcare Research and Quality, 2022] for Healthcare Research, A. and Quality, Rockville, M. (2022). Healthcare cost and utilization project (hcup).

[Goh et al., 2021] Goh, K. H., Wang, L., Yeow, A. Y. K., Poh, H., Li, K., Yeow, J. J. L., and Tan, G. Y. H. (2021). Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare.

[Hahn and Oleynik, 2020] Hahn, U. and Oleynik, M. (2020). Medical information extraction in the age of deep learning. *Yearb Med Inform*, 2020:208–228.

[Holzinger et al., 2019] Holzinger, A., Langs, G., Denk, H., Zatloukal, K., and Müller, H. (2019). Causability and explainability of artificial intelligence in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9.

[Huang et al., 2020] Huang, K., Altosaar, J., and Ranganath, R. (2020). Clinicalbert: Modeling clinical notes and predicting hospital readmission.

[HuggingFace, 2020] HuggingFace (2020). Bert.

[Hussain et al., 2019] Hussain, F., Cooper, A., Carson-Stevens, A., Donaldson, L., Hibbert, P., Hughes, T., and Edwards, A. (2019). Diagnostic error in the emergency department: Learning from national patient safety incident report analysis. *BMC Emergency Medicine*, 19:1–9.

[Johnson et al., 2016] Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L. W. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Scientific Data 2016 3:1*, 3:1–9.

[Kim, 2021] Kim, J. (2021). Automated assignment of icf functioning levels to clinical notes in dutch.

[Li et al., 2020] Li, Y., Rao, S., Solares, R. A., Hassaine, A., Ramakrishnan, R., Canoy, D., Zhu, Y., Rahimi, K., Salimi-Khorshidi, G., and Rao, S. 2020). Behrt: transformer for electronic health records.

[Lin et al., 2019] Lin, K., Hu, Y., and Kong, G. (2019). Predicting in-hospital mortality of patients with acute kidney injury in the icu using random forest model. *International Journal of Medical Informatics*, 125:55–61. Mortality prediction ML models (random forest), no nlp¡br/¿.

[Liu et al., 2014] Liu, S., Liu, S., Cai, W., Pujol, S., Kikinis, R., and Feng, D. (2014). Early diagnosis of alzheimer's disease with deep learning. *2014 IEEE 11th International Symposium on Biomedical Imaging, ISBI 2014*, pages 1015–1018.

[Liu et al., 2017] Liu, Z., Tang, B., Wang, X., and Chen, Q. (2017). De-identification of clinical notes via recurrent neural network and conditional random field. *Journal of biomedical informatics*, 75S:S34–S42.

[Locke et al., 2021] Locke, S., Bashall, A., Al-Adely, S., Moore, J., Wilson, A., and Kitchen, G. B. (2021). Natural language processing in medicine: A review. *Trends in Anaesthesia and Critical Care*, 38:4–9.

[Lundervold and Lundervold, 2019] Lundervold, A. S. and Lundervold, A. (2019). An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127. Special Issue: Deep Learning in Medical Physics.

[Maier et al., 2019] Maier, A., Syben, C., Lasser, T., and Riess, C. (2019). A gentle introduction to deep learning in medical image processing. *Zeitschrift für Medizinische Physik*, 29(2):86–101. Special Issue: Deep Learning in Medical Physics.

[Miller, 1994] Miller, R. A. (1994). Medical diagnostic decision support systems–past, present, and future: a threaded bibliography and brief commentary. *Journal of the American Medical Informatics Association : JAMIA*, 1:8–27.

[Novakovi et al., 2017] Novakovi, J. D., Veljovi, A., Ili, S. S., Papi, Z., and Tomovi, M. (2017). Evaluation of classification models in machine learning.

[Pauker et al., 1976] Pauker, S. G., Gorry, G., Kassirer, J. P., and Schwartz, W. B. (1976). Towards the simulation of clinical cognition: Taking a present illness by computer. *The American Journal of Medicine*, 60(7):981–996.

[Putelli et al., 2020] Putelli, L., Gerevini, A. E., Lavelli, A., Olivato, M., and Serina, I. (2020). Deep learning for classification of radiology reports with a hierarchical schema. *Procedia Computer Science*, 176:349–359. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 24th International Conference KES2020.

[Rasmy et al., 2021] Rasmy, L., Xiang, Y., Xie, Z., Tao, C., and Zhi, D. (2021). Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *npj Digital Medicine 2021 4:1*, 4:1–13.

[Smiti, 2020] Smiti, A. (2020). When machine learning meets medical world: Current status and future challenges. *Computer Science Review*, 37:100280.

[Sorower, 2010] Sorower, M. S. (2010). A literature survey on algorithms for multi-label learning.

[Stiglic et al., 2020] Stiglic, G., Kocbek, P., Fijacko, N., Zitnik, M., Verbert, K., and Cilar, L. (2020). Interpretability of machine learning-based prediction models in healthcare. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10:e1379.

[Vaswani et al., 2017] Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Łukasz Kaiser, and Polosukhin, I. (2017). Attention is all you need.

[Verkijk et al., 2021] Verkijk, S., Vossen, P., and Nl, P. T. J. M. V. (2021). Medroberta.nl: A language model for dutch electronic health records.

[Vries et al., 2019] Vries, W. D., Cranenburgh, A. V., Bisazza, A., Caselli, T., Noord, G. V., and Nissim, M. (2019). Bertje: A dutch bert model.

[Xiao et al., 2018] Xiao, C., Choi, E., and Sun, J. (2018). Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review.

[Zhou et al., 2021] Zhou, L., Zheng, X., Yang, D., Wang, Y., Bai, X., and Ye, X. (2021). Application of multi-label classification models for the diagnosis of diabetic complications. *BMC Medical Informatics and Decision Making*, 21:1–10.

Appendix A: F1 score per label per model

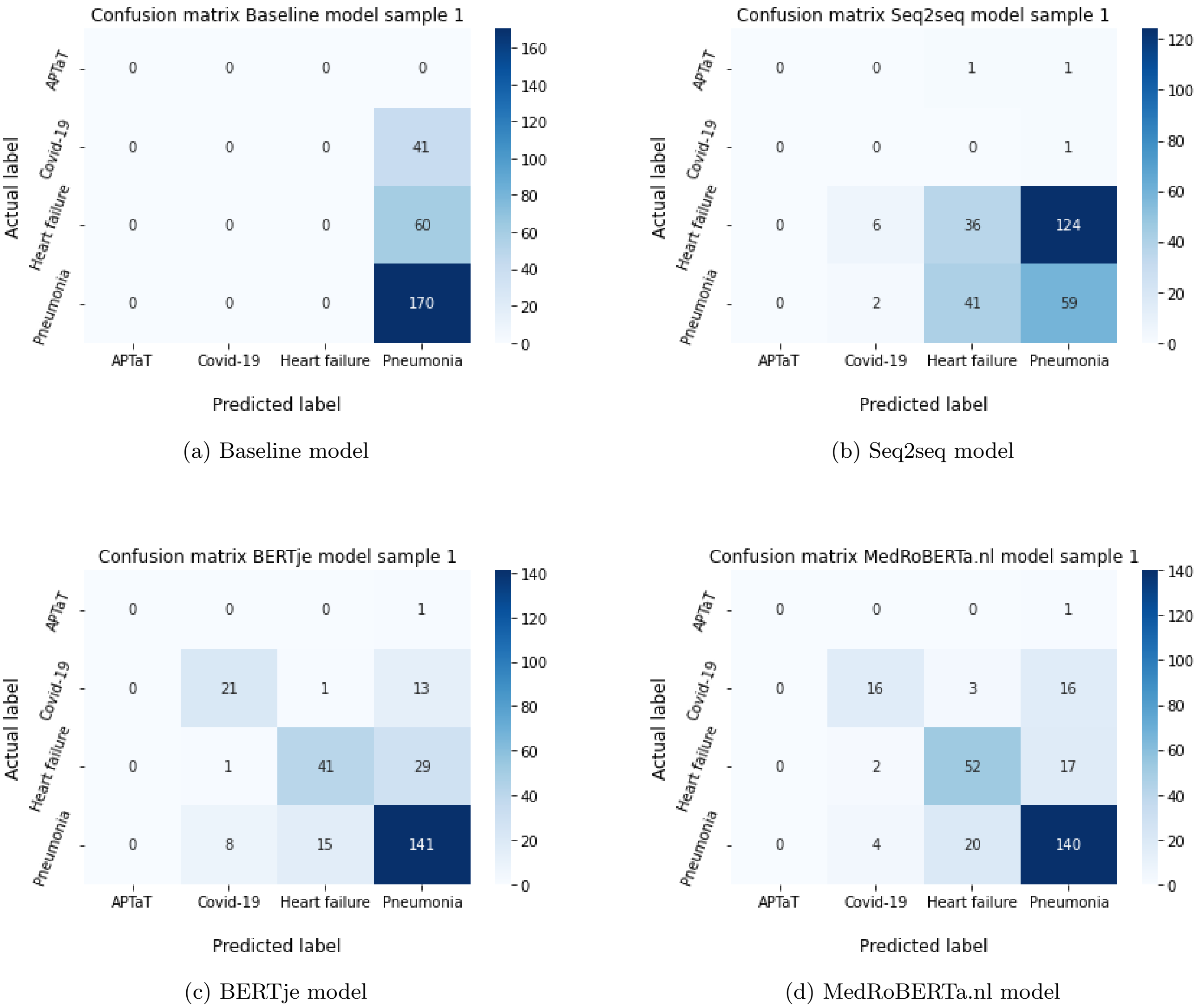
Label	**Baseline**	**Seq2Seq**	**BERTje**	**MedRoBERTa**	**Distribution**
Syncope	0.53	0.18	**0.56**	0.50	3767
Neoplasms	0.19	0.14	**0.49**	0.46	3754
Skin and subcutaneous tissue infections	0.33	0	0.41	**0.45**	529
Nerve and nerve root disorders	0.38	0	**0.44**	0.24	139
Pneumonia except that caused by tuberculosis	0.24	0.07	**0.34**	0.29	1707
Acute myocardial infarction	0.23	0	0.33	**0.36**	833
Respiratory signs and symptoms	0.15	0.02	**0.35**	0.32	414
Covid-19	**0.47**	0.01	0.35	0	326
Pancreatic disorders excluding diabetes	0.17	0	**0.35**	0.29	413
Urinary tract infections	0.19	0.01	0.30	**0.31**	825
Transient cerebral ischemia or Ceberal Infarction	0.19	0.03	**0.37**	0.15	2592
Acute and chronic tonsillitis	**0.38**	0	0.26	0	60
Acute hemorrhagic cerebrovascular disease	0.11	0.03	**0.27**	0.18	739
Viral infection	0.10	0	**0.25**	0.19	659
Acute bronchitis	**0.38**	0	0.05	0	141
Chronic obstructive pulmonary disease and bronchiectasis	**0.18**	0	0.13	0.04	425
Bacterial infections	0.09	0.04	**0.11**	0.09	1034
Pericarditis and pericardial disease	**0.18**	0	0.12	0.04	190
Personality disorders	**0.17**	0	0.14	0	133
Respiratory cancers	**0.29**	0	0	0	15
Paralysis other than cerebral palsy	**0.25**	0	0	0	27
Schizophrenia spectrum and other psychotic disorders	0.10	0	**0.11**	0	122
Biliary tract disease	**0.12**	0	0.08	0.01	319
Sickle cell trait anemia	**0.18**	0	0	0	22
Uveitis and ocular inflammation	**0.13**	0	0.02	0	54
Aortic peripheral and visceral artery aneurysms	**0.10**	0	0.04	0.01	265
Peritonitis and intra abdominal abscess	0.05	0	**0.07**	0	217
Trauma and stressor related disorders	0.04	0.02	**0.06**	0	204
Skin cancers melanoma	**0.06**	0	0.04	0.01	179
Allergic reactions	**0.12**	0	0	0	48
Aplastic anemia	**0.07**	0.01	0.02	0	217
Bone cancer	**0.07**	0	0	0	109
Thyroid disorders	**0.07**	0	0	0	118
Blindness and vision defects	0	0	**0.07**	0	93
Sarcoma	0.03	0	**0.04**	0	191
Spondylopathies spondyloarthropathy including infective	**0.05**	0	0.02	0	75
Fluid and electrolyte disorders	0.03	0.01	**0.03**	0	224
Endocarditis and endocardial disease	**0.07**	0	0	0	63
Septicemia	**0.06**	0	0.01	0	663
Acute and unspecified renal failure	**0.04**	0	0.02	0	202
Gastrointestinal cancers esophagus	**0.03**	0	0.02	0	161
Coagulation and hemorrhagic disorders	**0.04**	0	0.01	0	155
Systemic lupus erythematosus and connective tissue disorders	**0.05**	0	0	0	147
Vasculitis	**0.04**	0	0.01	0	115
Benign neoplasms	**0.04**	0	0.01	0	235

Label	**Baseline**	**Seq2Seq**	**BERTje**	**MedRoBERTa**	**Distribution**
Esophageal disorders	**0.04**	0	0	0	112
Conduction disorders	0	0	**0.01**	0	254
Male reproductive system cancers prostate	0	0	**0.01**	0	126
Abdominal hernia	**0**	**0**	**0**	**0**	78
Acute phlebitis thrombophlebitis and thromboembolism	**0**	**0**	**0**	**0**	13
Anxiety and fear related disorders	**0**	**0**	**0**	**0**	34
Aortic and peripheral arterial embolism or thrombosis	**0**	**0**	**0**	**0**	28
Arterial dissections	**0**	**0**	**0**	**0**	13
Aspiration pneumonitis	**0**	**0**	**0**	**0**	30
Bipolar and related disorders	**0**	**0**	**0**	**0**	14
Cancer of other sites	**0**	**0**	**0**	**0**	24
Cardiac and circulatory congenital anomalies	**0**	**0**	**0**	**0**	82
Cataract and other lens disorders	**0**	**0**	**0**	**0**	19
Chronic kidney disease	**0**	**0**	**0**	**0**	207
CNS abscess	**0**	**0**	**0**	**0**	36
Diabetes mellitus with complication	**0**	**0**	**0**	**0**	16
Digestive congenital anomalies	**0**	**0**	**0**	**0**	11
Diseases of mouth excluding dental	**0**	**0**	**0**	**0**	16
Disorders of lipid metabolism	**0**	**0**	**0**	**0**	18
Drug induced or toxic related condition	**0**	**0**	**0**	**0**	38
Endocrine system cancers thyroid	**0**	**0**	**0**	**0**	68
Feeding and eating disorders	**0**	**0**	**0**	**0**	14
Female reproductive system cancers cervix	**0**	**0**	**0**	**0**	46
Female reproductive system cancers uterus	**0**	**0**	**0**	**0**	23
Gastroduodenal ulcer	**0**	**0**	**0**	**0**	78
Gastrointestinal and biliary perforation	**0**	**0**	**0**	**0**	31
Gastrointestinal cancers liver	**0**	**0**	**0**	**0**	49
Gastrointestinal cancers stomach	**0**	**0**	**0**	**0**	72
Genitourinary congenital anomalies	**0**	**0**	**0**	**0**	15
Head and neck cancers hypopharyngeal	**0**	**0**	**0**	**0**	10
Head and neck cancers laryngeal	**0**	**0**	**0**	**0**	11
Head and neck cancers lip and oral cavity	**0**	**0**	**0**	**0**	11
Head and neck cancers throat	**0**	**0**	**0**	**0**	11
Hemolytic anemia	**0**	**0**	**0**	**0**	34
Hepatitis	**0**	**0**	**0**	**0**	35
Hypertension with complications and secondary hypertension	**0**	**0**	**0**	**0**	62
Immunity disorders	**0**	**0**	**0**	**0**	79
Inflammatory diseases of female pelvic organs	**0**	**0**	**0**	**0**	15
Menstrual disorders	**0**	**0**	**0**	**0**	17
Myocarditis and cardiomyopathy	**0**	**0**	**0**	**0**	34
Myopathies	**0**	**0**	**0**	**0**	24
Nephritis nephrosis renal sclerosis	**0**	**0**	**0**	**0**	67
Nervous system congenital anomalies	**0**	**0**	**0**	**0**	12
Neurodevelopmental disorders	**0**	**0**	**0**	**0**	52
Non Hodgkin lymphoma	**0**	**0**	**0**	**0**	13

Label	**Baseline**	**Seq2Seq**	**BERTje**	**MedRoBERTa**	**Distribution**
Nonmalignant breast conditions	**0**	**0**	**0**	**0**	51
Nonrheumatic and unspecified valve disorders	**0**	**0**	**0**	**0**	49
Nutritional deficiencies	**0**	**0**	**0**	**0**	18
Obesity	**0**	**0**	**0**	**0**	10
Other specified and unspecified hematologic conditions	**0**	**0**	**0**	**0**	75
Other specified and unspecified liver disease	**0**	**0**	**0**	**0**	65
Parasitic	**0**	**0**	**0**	**0**	43
Parkinson s disease	**0**	**0**	**0**	**0**	55
Peripheral and visceral vascular disease	**0**	**0**	**0**	**0**	100
Polyneuropathies	**0**	**0**	**0**	**0**	44
Pulmonary heart disease	**0**	**0**	**0**	**0**	20
Refractive error	**0**	**0**	**0**	**0**	16
Sinusitis	**0**	**0**	**0**	**0**	14
Transient Ischemic Attack	**0**	**0**	**0**	**0**	22
Urinary system cancers all other types	**0**	**0**	**0**	**0**	29
Urinary system cancers bladder	**0**	**0**	**0**	**0**	28
Mental disorders	**0.56**	0.14	0.35	0.39	3754
Cardiac dysrhythmias	0.43	0.1	**0.51**	0.49	2096
Heart failure	0.16	0.02	**0.33**	0.28	670
Burn and corrosion	0.62	0	**0.76**	0.54	455
Circulatory signs and symptoms	0.06	0.01	0.18	**0.22**	420
Coronary atherosclerosis and other heart disease	**0.25**	0	0.21	0.05	398
Gastrointestinal hemorrhage	0.21	0.03	**0.30**	0.23	360
Essential hypertension	0.30	0	**0.36**	0.29	359
Diabetes mellitus	0.14	0	**0.17**	0.14	264
Calculus of urinary tract	0.26	0	**0.39**	0.34	257
Diabetes mellitus without complication	0.10	0	**0.21**	0.21	248
Gastrointestinal cancers colorectal	0	0	0.06	**0.06**	226
Endocrine system cancers pancreas	**0.14**	0	0.04	0.05	201
Pleurisy	0	0	**0.04**	0	198
pleural effusion and pulmonary collapse	0	0	**0.02**	0	198
Meningitis	0.06	0	**0.07**	0	197
Contact dermatitis	0.24	0	**0.3**	0.13	142
Depressive disorders	0.14	0.03	**0.21**	0.20	139
Nutritional anemia	0	0	**0.01**	0	133
Pituitary disorders	0	0	**0.02**	0	125
Intestinal obstruction and ileus	**0.09**	0	0	0	110
Cornea and external disease	0.20	0	**0.52**	0.33	106
Noninfectious gastroenteritis	**0.04**	0	0.01	0	104
Nervous system cancers brain	0	0	**0.03**	0	94
Pneumothorax	**0.06**	0	0	0	83
Head and neck cancers all other types	**0.06**	0	0	0	81
Nervous system cancers all other types	0	0	**0.02**	0	67
Gastrointestinal cancers all other types	**0.08**	0	0	0	49
Encephalitis	**0.18**	0	0	0	33
Intestinal infection	**0.18**	0	0	0	20

Appendix B: Confusion matrices for all models on all samples

APTaT stands for Acute Phlebitis Thrombophlebitis and Thromboembolism. Pneumonia stands for Pneumonia except that caused by tubercolosis.



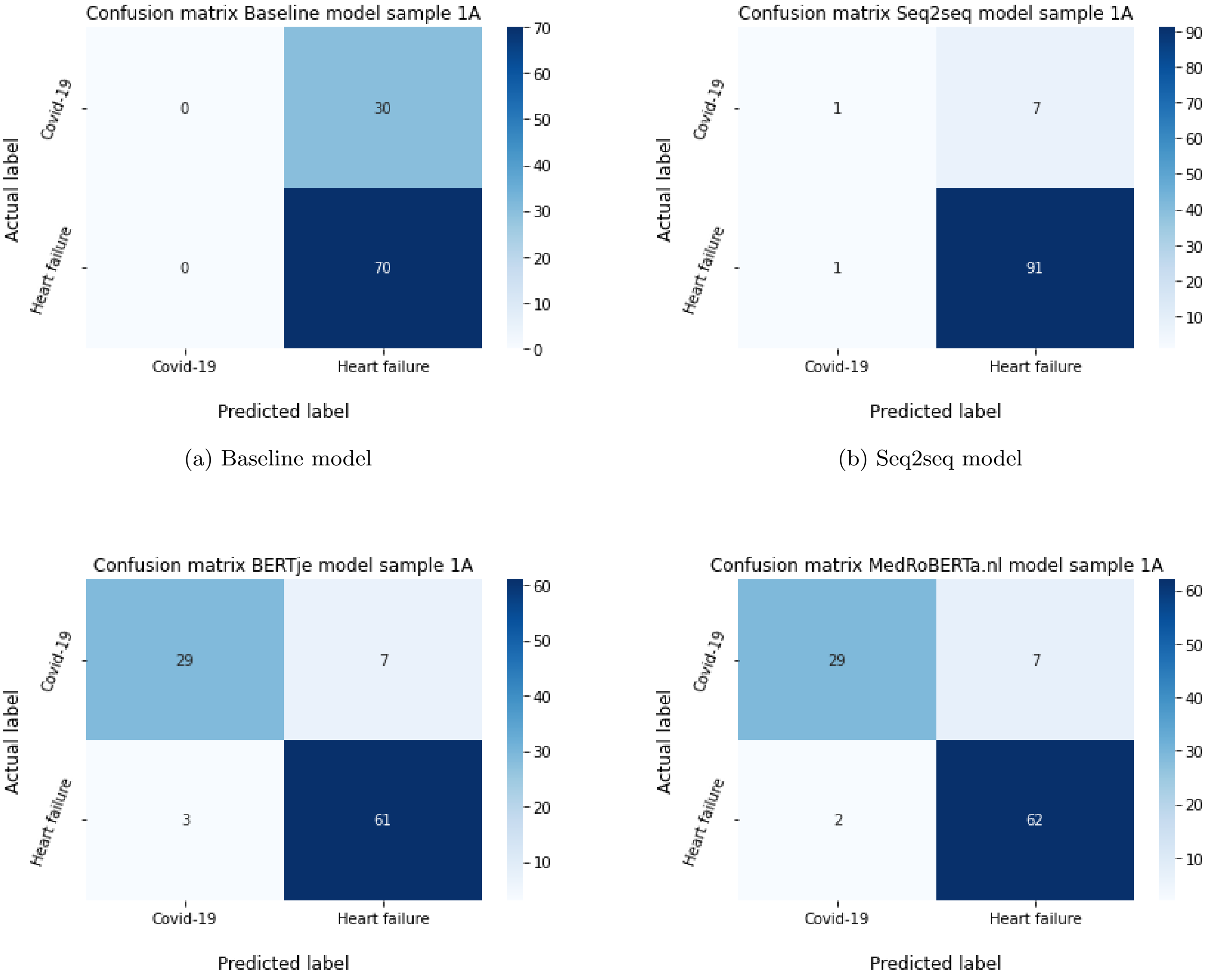
(a) Baseline model

(b) Seq2seq model

(c) BERTje model

(d) MedRoBERTa.nl model

Figure 13: Confusion matrices for all models on sample 1



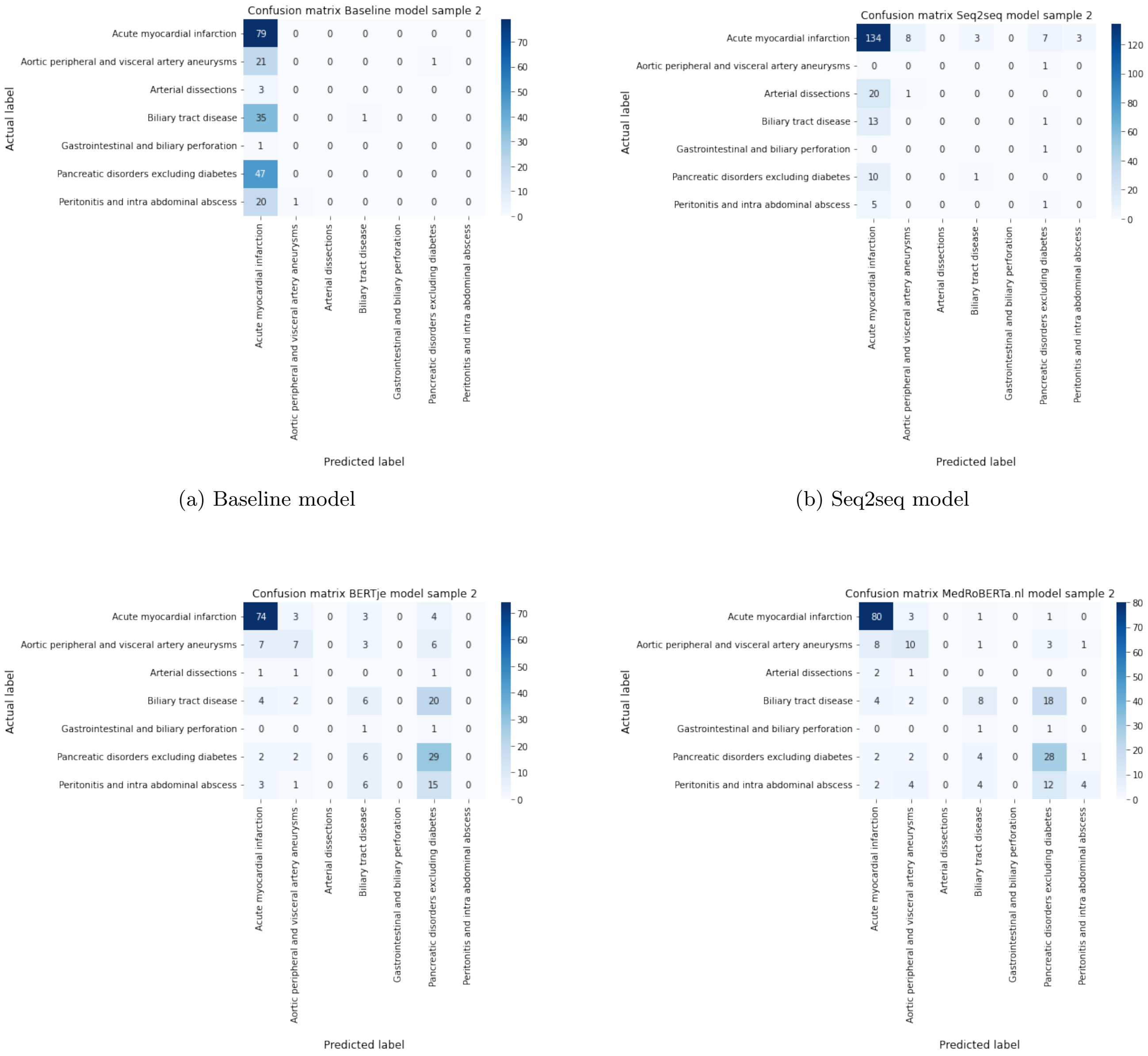
(a) Baseline model

(b) Seq2seq model

(c) BERTje model

(d) MedRoBERTa.nl model

Figure 14: Confusion matrices for all models on sample 1A



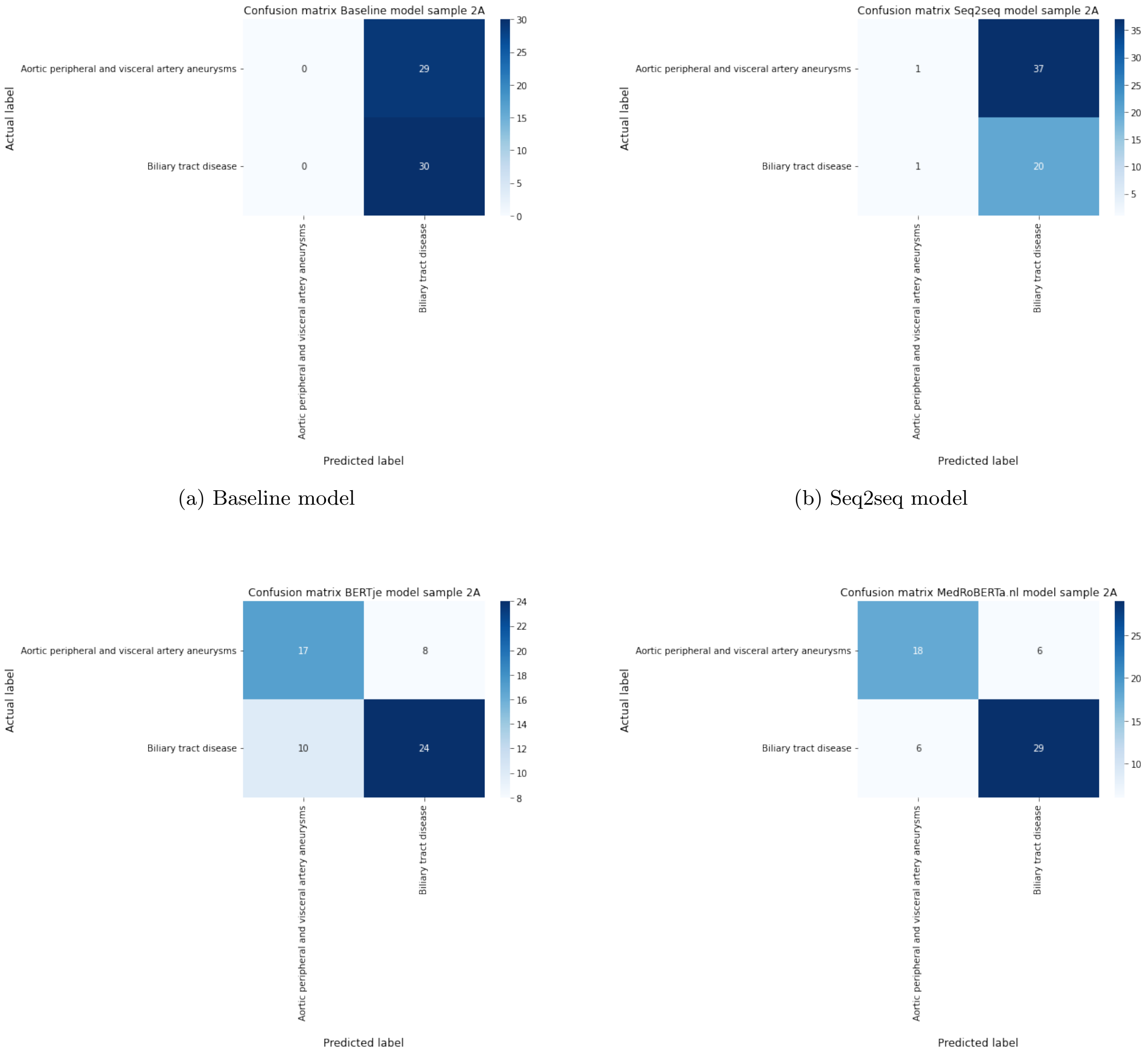
(a) Baseline model

(b) Seq2seq model

(c) BERTje model

(d) MedRoBERTa.nl model

Figure 15: Confusion matrices for all models on sample 2



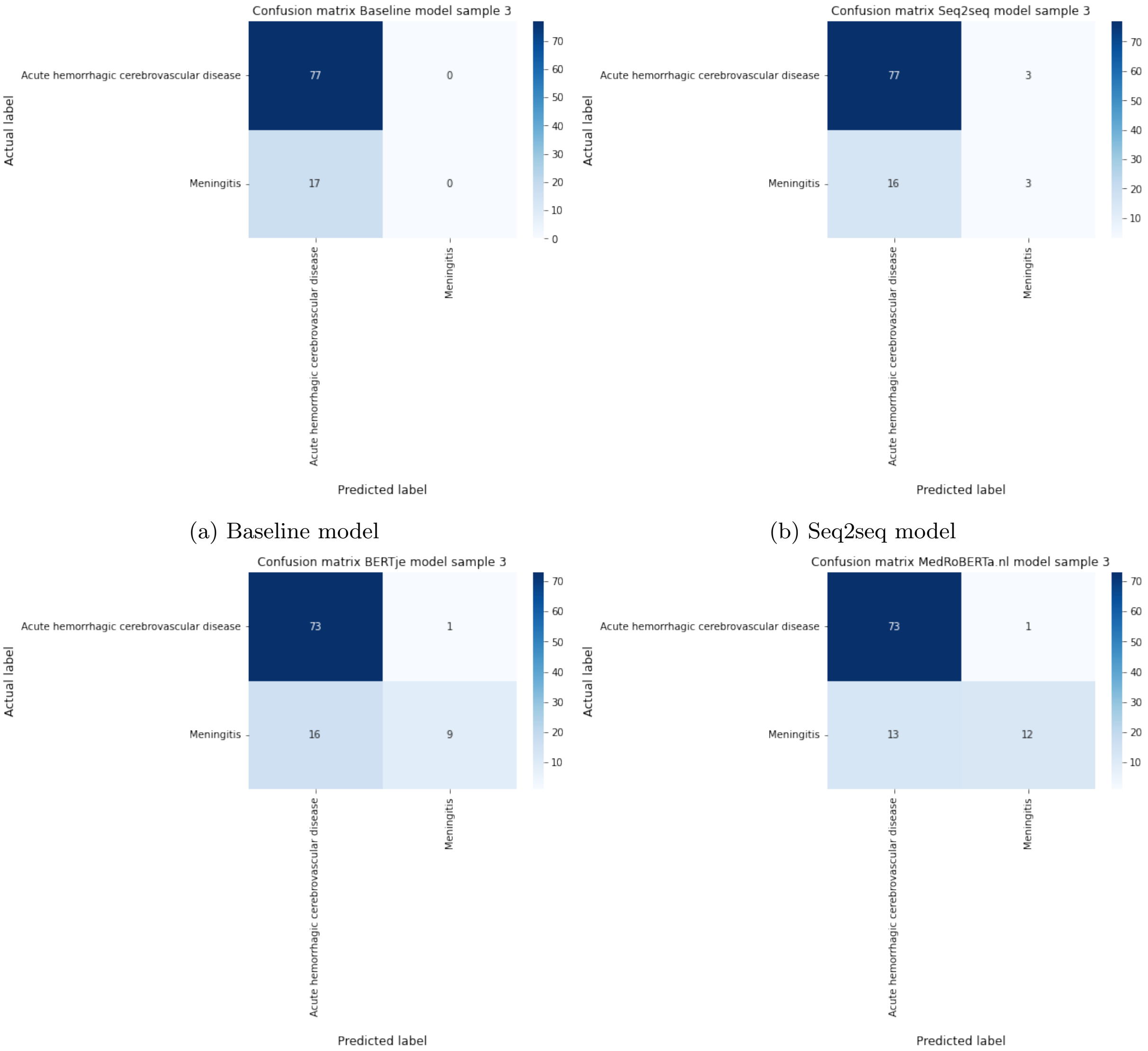
(a) Baseline model

(b) Seq2seq model

(c) BERTje model

(d) MedRoBERTa.nl model

Figure 16: Confusion matrices for all models on sample 2A



(a) Baseline model

(b) Seq2seq model

(c) BERTje model

(d) MedRoBERTa.nl model

Figure 17: Confusion matrices for all models on sample 3