A Comparative Study of Large Language Model Applications in Dutch Electronic Health Records for Symptom Identification

MSc Artificial Intelligence 06-2024 at Utrecht University | Written by *M.A. (Matthew) Scheeres* (4545966) | Supervised by *A.M. (Tuur) Leeuwenberg*, *I. (Isa) Spiero* and *M.P. (Marijn) Schraagen*

Abstract

Early identification of patients at risk of diseases like pneumonia is partly enabled through structured reporting of disease symptoms in Electronic Health Records (EHRs). However, this structured data is not always complete. Automated extraction of symptoms from unstructured text present in EHRs allows these records to be more exact and complete, resulting in more precise diagnoses. This report assesses the performance of Large Language Models (LLMs) in extracting lower respiratory tract infections (LRTI) from free-text sections of Dutch EHRs. The investigation involves the informed selection and comparison of promising LLMs, considering factors like local applicability, language compatibility, and model architecture. A search of relevant models is first performed, after which RobBERT and MedRoBERTa.nl are selected and evaluated across differing amounts of training samples. These models are both trained as direct classifiers and separately fine-tuned for few-shot prompt-based classification, with the goal of exploring the efficacy of the model types relating to the training (or multi-shot) samples provided. By employing a structured methodology and leveraging the capabilities of LLMs, the investigation seeks insights into the optimal utilisation of LLMs for effective symptom extraction in the context of Dutch EHR data. To increase generalisability, multiple target variables are selected to be extracted from the free-text samples (*fever*, *cough*, and *shortness of breath*). The classification performance is measured systematically by calculating metrics like precision, recall and F1-score. While the directly classifying MedRoBERTa.nl achieved F1-scores up to 0.88 with RobBERT closely following, the prompt-based models underperformed, suggesting limitations in their current design for this task.

Keywords

Large Language Models (LLMs) | Electronic Health Records (EHR) | NLP Applications | Multi-shot Classification | Symptom Extraction | Disease Prediction | Few-shot Learning | Clinical Text Analysis

Table of contents

1 Introduction

Over the last decade, Large Language Models (LLM) such as OpenAI's Generative Pretrained Transformer (GPT) and Google's Bard have shown promising potential in a plethora of language-oriented applications [1], [2], demonstrating performance close to task-specific systems in tasks ranging from prompt-based question answering to machine translation, without being specifically trained for that task [3].

One domain that could potentially benefit from the application of Large Language Models is the medical field, where massive amounts of relatively unstructured textual data are created daily in the form of digital clinical notes, or Electronic Health Records (EHRs).

An EHR contains medical data created and maintained by a medical professional like a General Practitioner or a surgeon. On a daily basis, large amounts of EHR entries are written and stored in secure databases. These records contain an abundance of information that could prove useful in research, but due to their unstructured format and strict privacy legislations, much of this information is not yet utilised to its fullest potential.

For this reason, over the past few decades, there has been research looking into the potential of Natural Language Processing (NLP) to extract structured information from these notes [4]. In the last few years, this research has largely focused on using Transformer-based Encoder models (e.g. Bidirectional Encoder Representations of Transformers, or BERT). Transformers differ from recurrent neural networks by, rather than using recurrence and convolutions to capture sequential information, solely relying on attention mechanisms to identify relational information between elements in a sequence [5].

While Transformer-based models hold great promise for NLP tasks, research specifically evaluating their effectiveness in extracting data from Dutch EHR data is currently relatively scarce. Many state of the art LLMs work via cloud services due to the large amounts of data and computational power needed to effectively run them. However, this reliance on cloud services raises privacy concerns especially since EHR data is privacy-sensitive by nature. Thus, locally applicable LLMs become more relevant in the context of this research. Besides this, as it is relatively expensive to create hand-labelled gold standard data, annotated EHR datasets to use for model training are rather scarce.

This report aims to address this by performing a domain search of locally applicable Large Language Models, after which these are applied to a LRTI symptom extraction task using labelled free-text notes from Dutch General Practitioner (GP) EHR data gathered through the Julius General Practitioners' Network in the region of Utrecht.

Following this, this report defines locally applicable models loosely as being a model that can reasonably be expected to successfully execute on consumer-available computers within an appropriate period of time. An important factor that this report aims to examine is the sample size needed for a local LLM to perform well, and how this number differs between LLMs that directly perform multiclass classification and LLMs that perform multi-shot classification through a prompt-based layer.

1.1 Research Goals

To summarise, the main research question this report attempts to answer is RQ: "*How do local LLMs perform when extracting LRTI-related symptoms from free-text Dutch GP clinical notes?*". RQ will be answered through the following subquestions:

- SQ1. *“What are promising small-scale LLMs that could be applied locally to extract symptoms from Dutch clinical notes data?”*
- SQ2. *"What is the classification performance of these models when extracting symptom presence from free-text clinical data, and how do their performances compare (precision, recall and F1-score)?"*
- SQ3, *“What is the impact of the sample size of available annotated data for model development or fine-tuning on the relative performance of the LMM approaches?”*

2. Background

This section provides a comprehensive overview of the current landscape of NLP with a specific focus on LLMs. It describes the current research into locally applicable LLMs in the healthcare sector, particularly in handling EHRs, and highlights the challenges and advancements associated with the application of LLMs both in general and in processing medical text data.

2.1 Natural language processing

Natural language processing is the subfield of Artificial Intelligence (AI) that focuses on enabling computers to process text from natural languages like English or Dutch - and interact with it in some analytically meaningful way. According to Liddy in 2001 [6], the goal of NLP is to create a full Natural Language Understanding (NLU) system that can 'accomplish human-like language processing'. Jurafsky and Martin define four base applications of NLP:

- Machine Translation: Using a computer to translate some text from one language to another;
- Question Answering and Information Retrieval: Answering questions by looking up information (*Information Retrieval*) or through knowledge and logical inference (*knowledge-based*);
- Chatbots and Dialogue Systems: Communicating with users in natural language, either to help them complete tasks or to mimic a human-like 'chat' for primarily entertainment-related purposes;
- Automatic Speech Recognition and Text-to-Speech: Recognising spoken language and being able to naturally reproduce it.

Nowadays, NLP techniques play a large role in many different industries, from speech-recognizing chatbots in consumer electronics - like Apple's Siri [7] - to automatic market forecasting through social media text mining [8].

2.2 Large Language Models

Large Language Models (LLMs) are a type of deep learning model, being a relatively new advancement in the field of NLP. These models are deemed 'large' due to their massive amounts of parameters, often ranging in the billions when looking at the current state of the art. In essence, a language model is a probability distribution, taking as input some sample of natural text, and assigning to it a probability $P(text|context)$. These models can be applied as text generators by determining the word with the highest probability (P_{max}), given some context in the form of, for example, an incomplete sentence [9].

Modern LLMs employ vast amounts of parameters and are trained on massive amounts of data, allowing for impressive performance in various tasks relating to natural language [3], [10]. However, due to their complexity, the models often prove to be very opaque [11]. Recently, innovations in computation have allowed models to be trained on extremely large datasets, sometimes containing trillions of words [12]. This has not only enabled LLMs to form more coherent responses, but recent models are also able to generate responses that are contextually relevant with regards to the input prompt. This enables these models to seemingly logically reason across a wide array of domains. Calling it *reasoning* is debatable, however, since the model does not directly apply domain knowledge or reasoning [13], [14], [15]. Rather, as mentioned, it simply functions as an extremely large probability model which, due to its extensive size, is able to respond in a manner that is generally deemed logical.

With the recent onset of large-scale LLM applications through (mainly) prompt-based front-end interfaces, general-purpose models that have an extremely high amount of parameters (often > 100B) show much promise for solving tasks such as information extraction [16], [17]. Concerning the medical domain, preliminary research into this topic has shown that GPT-4 is able to convert free-text clinical radiology notes into error-free structured JSON files containing the notes' key findings with relative satisfaction, outperforming state-of-the-art model medBERT.de in three out of four pathologic findings [18].

Early LLMs in 2019 and 2020 were mainly meant to be further developed using transfer learning for more specific tasks. With the introduction of GPT-3 by OpenAI, however, new methods have been explored that focus on general-purpose models, as its performance was already remarkable without fine-tuning [19].

Three general approaches of how LLMs could be applied are:

- *Training from scratch* involves completely designing the architecture of the LLM by oneself. Doing so allows for complete freedom in designing and altering the model, but comes at the cost of (generally) very high computational power needed and a need for an expansive dataset.

- *Using a pretrained model* is a simple approach when solving a problem using LLMs. As the name implies, this approach involves finding a model that has already been trained on either a general dataset or one related to the problem at hand, and directly applying it to said problem. This approach requires minimal effort and is the most straightforward, but the models used might consequently not perform optimally due to them not being optimised for the problem.

- *Building upon a model* involves using a pretrained model and fine-tuning it for the task at hand. This allows the model to adapt to the desired domain without the need for an extremely large dataset to train the model on. For these reasons, extending a pretrained model is often the first choice for LLM-related research in domain-specific appliances and has led to many models being built on top of already established language models, like BERT[20], [21].

To subsequently apply such a model to a specific task, like Information Extraction, the task is to be 'preloaded' in the input prompt, by writing e.g. "Using the input data, classify this sample to the most likely of the following classes: [cat, dog, hare]". A base LLM can also be trained to function directly as a classifier, without the need of a prompt-based 'layer' in its training architecture.

Besides the task description and the (in a classification problem) to-be-classified datapoint, it is also possible to include one or more examples, to give the model additional context to use when generating a response [19]. Doing so would turn the model from a zero-shot classifier into either a one- or multishot one. In Section 2.3, an explanation of these different 'shots' is provided.

One notable problem of LLMs is the occurrence of hallucinations in generated text [22]. Hallucinations occur when an AI system confidently responds with an answer that is not grounded in factual reality (or is not a product of its training data), which is, at the time of writing, still a major challenge in LLMs [23], [24]. While efforts to solve this issue have already been made [25], a perfect solution does not exist yet, nor one that does not add a need for significant additional computational power. The problem of hallucination is made even more complex due to the massive size of the training datasets for LLMs, which makes it even more difficult to find the cause of the hallucination. For many tasks involving question answering or conversational models, base LLMs are fine-tuned using a dataset consisting of pairs of input prompts and their corresponding responses, through, for example, reinforcement learning (the model gets a reward for generating a desired response) or supervised learning (the model is trained to predict the correct response for a given prompt).

However, as clinical notes generally contain privacy-sensitive, non-aggregated patient data, endeavouring to automatically extract data from them has proven rather difficult, since most large-scale models are closed-source and data used as input for the models is generally sent to an outside, nonlocal server. According to OpenAI, data such as chat history is saved by ChatGPT, for instance, for further training and improvement of its models. This data might even be subject to human review to improve OpenAI's systems [6].

Current State of the Art in LLMs outside of Medicine

Since the start of 2022, a type of machine learning model that has received widespread attention is what Wornow *et al.* call Foundational Models (FM) [26]; machine learning models that have been trained on large and diverse datasets to solve general-purpose tasks rather than being trained and evaluated for one specific task or use-case, which has often been the case before the first Foundational Models.

Perhaps the most widely known FMs in Natural Language Processing are OpenAI's GPT models, which have 176 billion and 1.7 trillion parameters respectively. This popularity is in large part due to ChatGPT: a general-purpose front-end application and API currently built on GPT3.5 and GPT4. In the first two months since its release, ChatGPT gained over 100 million users [27], with other interested parties quickly creating similar applications based on their own LLMs or GPT models (Google with Bard [28], Microsoft with Bing Chat [29] and Meta with LLaMa [30] and LLaMa 2 [31]).

Over the past year, development in the field of Foundational LLM models has been done at an extremely fast pace, with both datasets and parameter counts in models increasing exponentially between iterations that rapidly succeed each other. In this current field, there are three large identifiable challenges identified in a LLM overview paper by Kaddour *et al.* [11].

Firstly, datasets used for pretraining LLMs have become unfathomably large, often requiring millions to trillions of tokens. Needless to say, it is impossible for such a large amount of data to be thoroughly checked manually. Heuristics to overcome this challenge are being implemented, but e.g. checking for near-duplicates on such a large scale remains an extremely difficult process, while they are reported to degrade model performance significantly [32].

Secondly, pretraining such a large-scale LLM is computationally very taxing and thus requires extensive resources. The costs and time involved in pre-training one model can reach into the hundreds of thousands of compute hours, which can cost millions of dollars [33]. For both monetary and environmental reasons, the corresponding outcome of good results only being 'bought' by companies with the assets to do so is very unsustainable. This type of practice has been dubbed *Red AI* [34].

Thirdly, tokenizers used in training such LLMs introduce problems like computational overhead, language dependence and the handling of novel words. Efforts are being made towards reducing computational complexity, such as Byte-Pair Encoding[35], but the problem is far from resolved yet.

2.3 Information Extraction using Large Language Models

Large Language Models can be used to extract information from text in multiple manners. There are multiple forms of classification, described below.

In the conventional approach to model training in neural networks (via empirical risk minimization), a model is 'fit' to multiple training samples - i.e. through stepwise training iterations -, and the model's weights and biases are continuously updated, resulting in a trained model that uses its input data (here, a doctor's note in natural language) to make classifications based on the labels provided in the training set (here, data pertaining to the presence of a symptom in the text).

Another approach to this problem enabled by conversational LLMs is classification through pretrained prompt-based models. For these types of experiments, one of the following classification setup types can be applied [17]:

- Zero-shot classification: Having the model classify based solely on a task description (e.g. "Classify whether the patient in this text has a fever or not") and a prompt (e.g. "The individual shivers often and is feverish.");
- One-shot classification: The same as zero-shot, but here one sample is provided (e.g. "While the patient coughs constantly, there is no sign of elevated temperature" => "No fever");
- Few-shot or multishot classification: Similar to one-shot, but in this case more than one sample is provided to the model.

In none of these last classification types backpropagation is done, and thus no gradient updates are performed, leaving the trained model itself unchanged.

Using language models as x-shot classifiers has been proven to perform satisfactorily, especially when a model is pretrained or fine-tuned on relevant data, as exemplified by a study into suicide prevention by Varma *et al.*[36].

2.4 Electronic Health Records

Electronic Health Records have become a cornerstone of modern healthcare, revolutionising the way patient information is documented and managed. While parts of an EHR - like demographics and medications - are stored in a structured, indexable manner, others - such as progress notes or problem lists - are nearly always noted down by the medical professional in the form of free text. This unstructured data generally makes up about 80% of total EHR data, which by itself is difficult to process in a meaningful way outside of its original use [37].

As such, much information about a patient is saved in a way that is non-indexable and therefore cannot directly be used for e.g. aggregated patient research. The main reason for this is the documentation burden [38]. Medical professionals who need to make multiple entries a day into EHRs generally find it much easier and faster to enter a short piece of freeform textual information into a document than to file each piece of information into separated, structured fields multiple times over the course of a day.

However, as usage of EHRs has increased massively over the past few decades, EHRs can now be deemed ‘Big Data’. The large amounts of information contained in them are now practically impossible to be studied by individual humans, but do contain patterns that could be exposed and used computationally [39]. Because of this, the need for automated information extraction from these EHRs arises, which is a challenge that has been addressed by NLP in multiple ways.

Extracting Medical Information from EHRs

When it comes to extracting information from EHRs, there are multiple types of NLP tasks that can be performed, like the extraction of important study variables. Data extraction from free text sections in EHRs could lead to large amounts of useful structured data becoming available, which could be used for e.g. (partially) automated patient risk prediction for a given disease. This directly impacts the healthcare field in that it can strongly affect clinical decision making (as part of it can be automated by using algorithms that consider many more factors than humans realistically could) [39] and decrease the costs associated with it.

It goes without saying that EHRs contain very sensitive personal data. Research by Menger *et al.* shows how pattern matching can be applied to automatically anonymise or de-identify Dutch EHRs [40].

In as early as 2001, Aronsky *et al.* showed that combining probabilistic outcomes from a Bayesian Network with outcomes of a Natural Language Understanding System gathered using EHR data can significantly increase the performance of computerised decision support in predicting (or diagnosing) pneumonia [41]. The natural language understanding system in this research, created by Haug *et al.* [42], was made to work on radiological chest x-ray exams, and used a syntactical parser in combination with a rule-based approach to extract information from the texts.

Among more recent research, algorithms have been developed to extract data from EHRs to detect and identify different types of cancer and their symptoms [43], [44], [45], [46], derive lines of therapy for cancer treatment [47], identify suicidality in adolescents with autism [48], [49], [50], [51], and identify patient phenotypes [52].

Applying Large Language Models to Dutch EHR Data

When attempting to extract information from free-text sections of EHRs, some challenges arise. Firstly, EHRs tend to be filled with medical shorthand and terminology specific to the field. Not only does this influence the words used in the texts, but it also often leads to grammatical rules being ignored because of the manner in which medical terminology is used [4], [53]. This difficulty is made twofold because this jargon is also largely language-specific or even physician-specific, and the research in this report focuses on Dutch-language data.

Efforts have been made to apply Large Language Models to Dutch EHRs as zero- or few-shot information extractors. An example of this, relevant to this report's research, is MedRoBERTa.nl, created at Vrije Universiteit Amsterdam by Verkijk *et al.* [4]. This model was trained using the general Dutch RoBERTa model as a baseline, and training it using 13GB of Dutch EHR data. Compared to currently established models (like GPT-4 and LLaMa-2), this model is relatively small, only consisting of 117 million parameters, whereas company-hosted cloud-based LLMs nowadays tend to span in the hundred billions to trillions of parameters range. This difference in size, while possibly affecting its performance negatively, does however allow the pretrained model to be run locally with relative ease.

Verkijk *et al.* show that training their model from scratch on the data outperforms non-medical Dutch language models on an odd-one-out similarity task involving sentences from (a held-out part of) the EHR data.

3 Methodology

This section outlines the structure of the data, techniques and procedures used in the research of this report. This is done by first describing the aims of the experiments, followed by a description of the data and how it was used. Afterwards, a motivation of the target variables follows, together with the study design outlining the different experimental setups. Used performance measures are shortly described, and the computational setup of the experiments is named.

3.1 Aims

The experiments done in this paper aim to compare classification performance between direct classifiers and prompt-based models by looking at their behaviour when performing symptom extraction structured as multiclass classification across differing amounts of input training samples.

Furthermore, the goal of the rest of this report is to compare RobBERT and MedRoBERTa.nl (further explained in Section 3.4) and show how different training sample sizes result in different outcomes of direct classifiers and prompt-based classifiers respectively. This data allows for the second and third sub-questions (*SQ2* & *SQ3*) to be answered, whereas the first (*SQ1*) has been answered in Comparison of Models (Appendix).

Context of the research

In addition to being a standalone MSc Artificial Intelligence thesis report, the research done in this report is part of a larger prediction modelling project that aims to create a predictive model capable of reliably predicting hospital admission and mortality in patients diagnosed with lower respiratory tract infections (LRTI) [54].

3.2 Data

The dataset used in the experimentation of this report is a subset of the Julius General Practitioners' Network (JGPN) dataset [55]. The complete JGPN dataset is made up of data covering approximately 450,000 patients of general practitioners based in the region of Utrecht, the Netherlands.

As the symptom extraction task has been formulated as a supervised multiclass classification task, labelled data is needed to perform the experiments. This report will use the 1,000 labelled random samples created by Rijk *et al.* as a means of validating the symptoms extracted by the models [54]. The subset consists of EHR data of adult patients in primary care showing symptoms of LRTI. This dataset contains patients aged 40 years and older, who consulted their GP between 01/01/2016 and 31/12/2019 and for whom their GP recorded an International Classification of Primary Care (ICPC) code pertaining to subgroups of LRTI, such as acute bronchitis or pneumonia.

These clinical GP notes follow the SOEP structure (Subjective, Objective, Evaluation and Plan). In this research only subjective (S) and objective (O) records were used.

In the labelled data, two types of variables were hand-labelled, *Signs and symptoms* (ternary values that can be recorded as positive, recorded as negative or not reported in the text, e.g. shortness of breath) and *Measurements* (continuous values, e.g. heart rate). This report focuses solely on the extraction of data pertaining to fever, cough and shortness of breath, as mentioned in Section 3.3.

The distributions of all symptoms in the dataset are shown in Table 8 (Appendix). Figure 1 below shows how the length of the labelled EHR entries is distributed over the dataset, in terms of tokens.

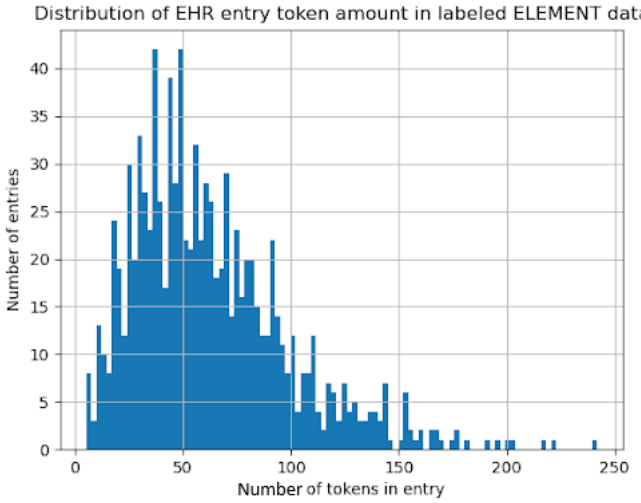

Figure 1. Distribution of tokens in labelled JGPN dataset entries

As can be seen, most entries have between 5 and 100 tokens, with a few outliers containing up to 240 tokens. The mean number of tokens present is 63.

Due to the RobBERT and MedRoBERTa.nl - chosen in A. Comparison of models (Appendix) - having a maximum input amount of 512 tokens, the maximum number of examples in evaluated prompts was chosen to be 3, in order to minimise the chance of overflow. This results in a mean prompt length of 310 tokens, leaving some room for longer samples. Should a prompt still overflow, then examples will be truncated evenly until the prompt fits within the acceptable input window.

Fine-tuning LLMs for Prompt-Based Tasks

To train RobBERT and MedRoBERTa.nl for prompt-based conversational abilities in Dutch, the HealthCareMagic-100k dataset is used. This dataset consists of a little over 100,000 patient-doctor conversations, and was used due to its success in training the original ChatDoctor model, for which it was created [56]. Due to the necessity of sequence-to-sequence capabilities specifically in Dutch, however, this dataset was machine-translated from English to Dutch using the Google Translate API. While these translations are not perfect, research suggests that Google Translate-translated texts show high correlation with human-translated texts on multiple fronts [57]. Moreover, as there is only a need for the models to classify - instead of giving full answers in correct Dutch - an automatic translation seems adequate for our research.

3.3 Target Variables

As mentioned before, Table 8 (Appendix) shows the full distribution of variables across the 1,000 labelled samples. In terms of classification, this data creates a multiclass classification task.

For some given input text *x*, symptom *y* can either be *recorded as negative (label 0), recorded as positive (label 1) or absent from the text (label 2)*. Generally, more balanced data is expected to yield better classifier performance than skewed data. Because of this, the ideal symptom has labels distributed equally, so each label is present in ± 33% of the samples.

The variable adhering to this the best is *Fever* ('Koorts'), which is recorded in 54.1% of samples, of which 57.7% are positive reports of the symptom (Table 1). Because of this, *Fever* is used as the target variable in this report's experiments. To ensure generalisability, *Cough* ('Hoesten') and *Shortness of breath* ('Kortademigheid') are also selected as target variables. These variables have been selected due to their high prevalence in the *positive if recorded* category (98.4% and 78.9% respectively) while also being comparatively prevalent in the *overall* category (76.6% and 52.2% respectively). Due to temporal and computational constraints, less prevalent symptoms were not used in the research.

Table 2 shows a fictional example record of the dataset.

	Recorded as positive	Recorded as negative	Not recorded
Cough	75.4	1.2	23.4
Fever	31.2	23.4	45.9
Shortness of Breath	37.7	15.5	46.8

Table 1. Distribution of target variables in dataset

Patnr	start_epi	start_icpc	SOEPcode	*Koorts*	*Hoesten*	*Kortademigheid*	DEDUCE_omschrijving
100020	2018-02-27	R90	SO	1	2	0	*"Pat heeft sterke verhoging, ademhaling goed"*

Table 2. Example record of labelled JGPN EHR entry

3.4 Study Design

The labelled data points used consist of individual Dutch free-text Electronic Health Record SOEP reports of patients' first LRTI-related GP consultation, and are labelled according to whether they a) contain a positive mention of the target variable (e.g. *coughing* is a confirmed symptom in the patient), b) contain a negative mention of the target variable (e.g. *coughing* is confirmed not to be a symptom of the patient), or c) contain no mention of the target variable (e.g. *coughing* is not mentioned in the input text). These three classes structure the classification as a multiclass classification task. The chosen target variables and the motivation for choosing them have been described in Section 3.3, and consist of the symptoms *fever*, *cough*, and *shortness of breath* (*Koorts*, *Hoesten*, and *Kortademigheid*).

To determine which models to evaluate on the data, a search of available models was performed, which is described in A. Comparison of Models (Appendix).

In total, four model setups will be evaluated on the three chosen symptoms, leading to twelve total setups as shown by Table 3. Two of each of the following types of models are used to perform classification.

Direct Classification

First, models that work as classifiers in the classic sense are applied: they are trained on a labelled training set using 5-fold cross-validation. As the BERT-based LLMs described in Final Selection (Appendix) are not prompt-based by themselves, they will be fine-tuned to function as direct classifiers. Fine-tuning involves retraining the models on the labelled dataset, on each examined symptom individually. Huggingface's Trainer object ensures the final layer has the appropriate amount of weights, based on the amount of different labels present in the dataset.

The models to be compared here are MedRoBERTa.nl and RobBERT. Both of these models will be applied to differing training set sizes using 5-fold cross-validation (using fixed folds across each experiment, with the validation set always containing 200 samples), with the following training or sample set sizes: 1, 3, 6, 12, 25, 50, 100, 200, 400 and 800.

Prompt-Based Classification

The second type of model used is prompt-based: these models will perform prompt-based multi-shot classification, i.e. forcing multiclass classification through prompt-based task preloading in a multi-shot sense. The models to be evaluated from this category are MedRoBERTa.nl and RobBERT, which are also evaluated using 5-fold cross-validation. These models will be fine-tuned for a 'conversational' sequence-to-sequence generation by using the HealthCareMagic-100k dataset as used by ChatDoctor [56]. This dataset comprises over 100,000 doctor-patient question-answer pairs, which were machine-translated using python-translate [58].

MedRoBERTa.nl and RobBERT only contain an encoder structure. However, to enable the models to generate text, a decoder is necessary as well. To allow for this, the models are merged with the base multilingual BERT decoder using a HuggingFace Transformers EncoderDecoderModel. A layer to concatenate these models is randomly initiated, and its weights and biases are also updated during fine-tuning.

A prompt is then used, describing the multiclass prediction task by instructing the model to classify the given sample based on some provided examples. The structure used for this prompt can be viewed under Example Input Prompt (Appendix).

As the maximum amount of input tokens for a prompt is limited to 512 for the BERT-based models, the maximum amount of input *samples* for these models is limited to 1, 2 and 3.

The used prompt-based models may not always generate one of the expected labels but instead answer the question in a completely different format, or not answer the question at all. To attempt to solve this issue, the predicted label is obtained by looking at the loss value gained by making a forward pass using the prompt as input for the model's encoder, and the different labels as inputs for its decoder. Then, the label corresponding to the lowest loss value is chosen. The loss values should be proportional to the transition scores for a full sequence (i.e. the input label), because of which they can be used to predict the chosen label as well (which had to be done, as HuggingFace Transformers is not able to calculate transition scores for an arbitrary given sequence when using an EncoderDecoderModel).

Table 9 (Appendix) shows the number of parameters corresponding to each model.

Setup name	***Model type***	***Model name***	***Target variable***
DC-MR-Koorts	Direct Classifiers	MedRoBERTa.nl	Koorts
DC-MR-Hoesten			Hoesten
DC-MR-Kortademigheid			Kortademigheid
DC-RB-Koorts		RobBERT	Koorts
DC-RB-Hoesten			Hoesten
DC-RB-Kortademigheid			Kortademigheid
PB-MR-Koorts	Prompt-Based Classifiers	MedRoBERTa.nl	Koorts
PB-MR-Hoesten			Hoesten
PB-MR-Kortademigheid			Kortademigheid
PB-RB-Koorts		RoBERT	Koorts
PB-RB-Hoesten			Hoesten
PB-RB-Kortademigheid			Kortademigheid

Table 3. Setup names for each experimental scenario

3.5 Performance Measures

This section briefly lists the performance measures that will be used to evaluate the models, as well as a short motivation for each one.

- Accuracy (A): While not very useful when dealing with imbalanced classes, accuracy is included in results for completeness. Is shows the proportion of correctly predicted classes, and is calculated by $A = \frac{TP+TN}{TP+TN+FP+FN}$;
- Precision (P): Precision measures the ratio of True Positive predictions to the total amount of positive predictions (a combination of True Positives (TP) and False Positives (FP)), i.e. the relative amount of correctly predicted positive samples, and is calculated by $P = \frac{TP}{TP+FP}$;
- Recall (R): Recall measures how well the model is able to identify positive cases, i.e. all cases where a symptom is present in the text. Recall is calculated using the formula $R = \frac{TP}{TP+FN}$;
- F1-score (F1): The F1-score provides a harmonic mean of precision and recall, leading to it being used as a balanced measure of the factors measured by the two. This is particularly useful when there is an inherent tradeoff present between precision and recall, since precision and recall measure 'false alarms' and 'missed symptoms' respectively. A high-precision model might predict only few positive cases but ensure few false positives are present, while a high recall might do the opposite by predicting many positive cases but including relatively many that are not actually positive. The F1-score is measured using the formula $F1 = 2\frac{P \times R}{P+R}$.

These metrics should, together, provide useful insight in the performance of each tested model across the different predicted variables outlined in Section 3.3.

Accuracy was originally also intended to be used as an evaluation metric, but was eventually omitted, since it does not show useful scores when dealing with an imbalanced dataset - which is the case with the used data, see Table 8 (Appendix).

3.6 Computational Setup

The experiments in this research will be run inside a virtual computer with 32 GB of RAM, an Intel(R) Xeon(R) Platinum 8272CL CPU without a GPU. The virtual computer uses Windows 10 as its Operating System, and pretrained models are evaluated and analysed within Python files, which make use of Python v3.11.5.

4 Results

This section contains an overview of the results gained from the experiments discussed in Section 3, along with visualisations showing the mentioned prediction metrics in the evaluated models and how they fluctuate with differing amounts of training samples in both the direct classifiers and prompt-based models. As the amount of experimental scenarios totals 120, an effort has been made to show the most relevant data, while an exhaustive overview of all results can be found in the Appendix. Since each experiment has been performed using five-fold cross validation, results in the graphs show the average of these five runs.

First, to enable a comparison to be made between the training processes, a selection of relevant loss plots for the directly classifying models are shown, depicting how the calculated loss of the models changes over the epochs. These values are then shown grouped by model and symptom.

Secondly, barplots are created for each combination of (prediction metric, symptom), showing on a per-model basis how these values differ when using different amounts of training samples. Besides this, the values of these metrics for individual scenarios are shown.

Each experiment is trained for five epochs on the full dataset of 1000 samples, as testing has shown that the models start overfitting shortly after this number - see Figure 6 (Appendix).

4.1 Direct Classifier Loss

Figure 2 below shows cross-entropy validation loss over the five epochs for each combination of model (RobBERT, MedRoBERTa.nl) and target variable (Koorts, Hoesten, Kortademigheid).

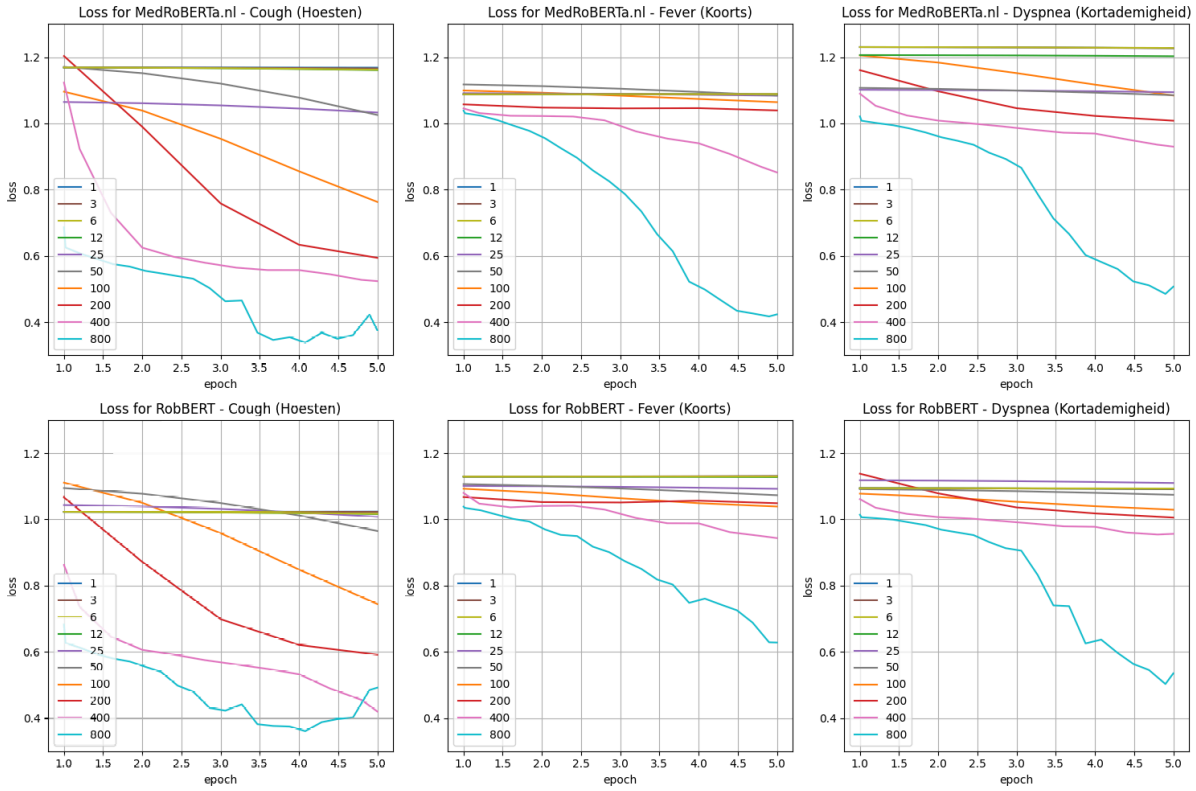

Figure 2. Validation loss of direct classifier experiments, using 5-fold cross-validation

Based on the graphs in this figure, the use of 800 samples seems to reliably lead to the lowest loss value in every model, for each target variable. Models classifying the symptom ‘Hoesten’ generally have the best performance in terms of loss, which is likely due to this symptom being the most evenly distributed in the dataset.

Below in Figure 3, the cross-entropy loss value is visualised per epoch for each combination of model and symptom. These values represent the models when trained on the full dataset of 800 samples (excluding a validation set of 200).

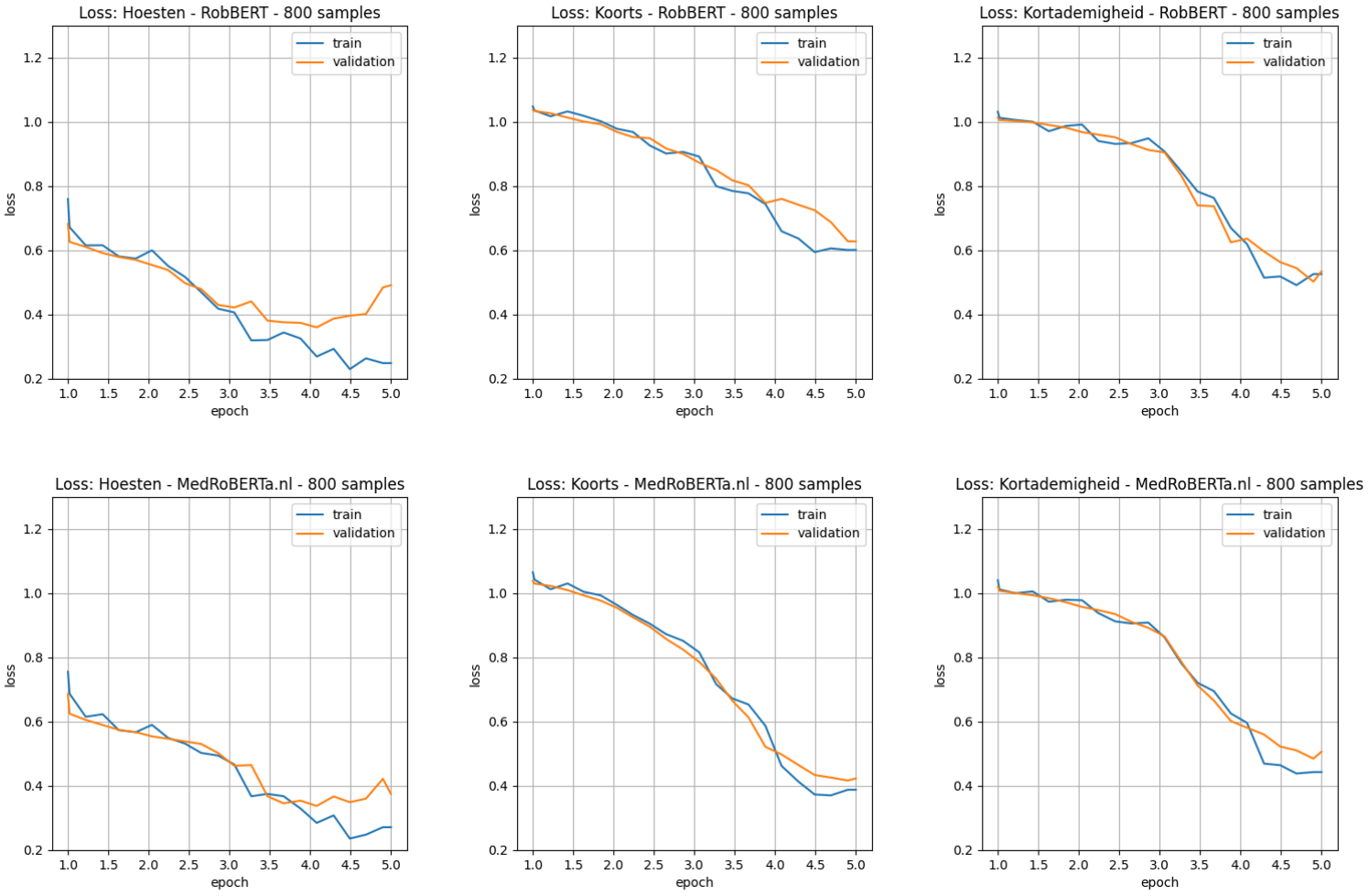

Figure 3. Training and validation loss of direct classifiers trained on 800 samples, using 5-fold cross-validation

As can be seen in the loss plots, the training and validation loss always follow each other relatively closely, indicating that the models are not strongly overfitting. Both models seem to converge relatively reliably around five epochs of training. Table 4 displays the lowest and final loss values for each of these setups.

	Hoesten	**Koorts**	**Kortademigheid**
Lowest validation loss			
RobBERT	0.360	0.628	0.503
MedRoBERTa.nl	0.338	0.417	0.485
Final validation loss			
RobBERT	0.492	0.628	0.535
MedRoBERTa.nl	0.375	0.423	0.507

Table 4. Lowest and Final validation loss of direct classifiers trained on 800 samples, using 5-fold cross-validation

Table 4 shows that MedRoBERTa.nl reaches lower validation loss for each examined symptom than RobBERT in the direct classification symptom extraction task. The lowest loss value is reached for '*Hoesten*' in both models, with MedRoBERTa.nl achieving the lowest overall value with the '*Hoesten*' symptom.

4.2 Classification Performance

This subsection shows accuracy, precision, recall and F1-score for each run of both the prompt-based and direct classifier models. Below, Figure 4 visualises the results of experiments corresponding to the former.

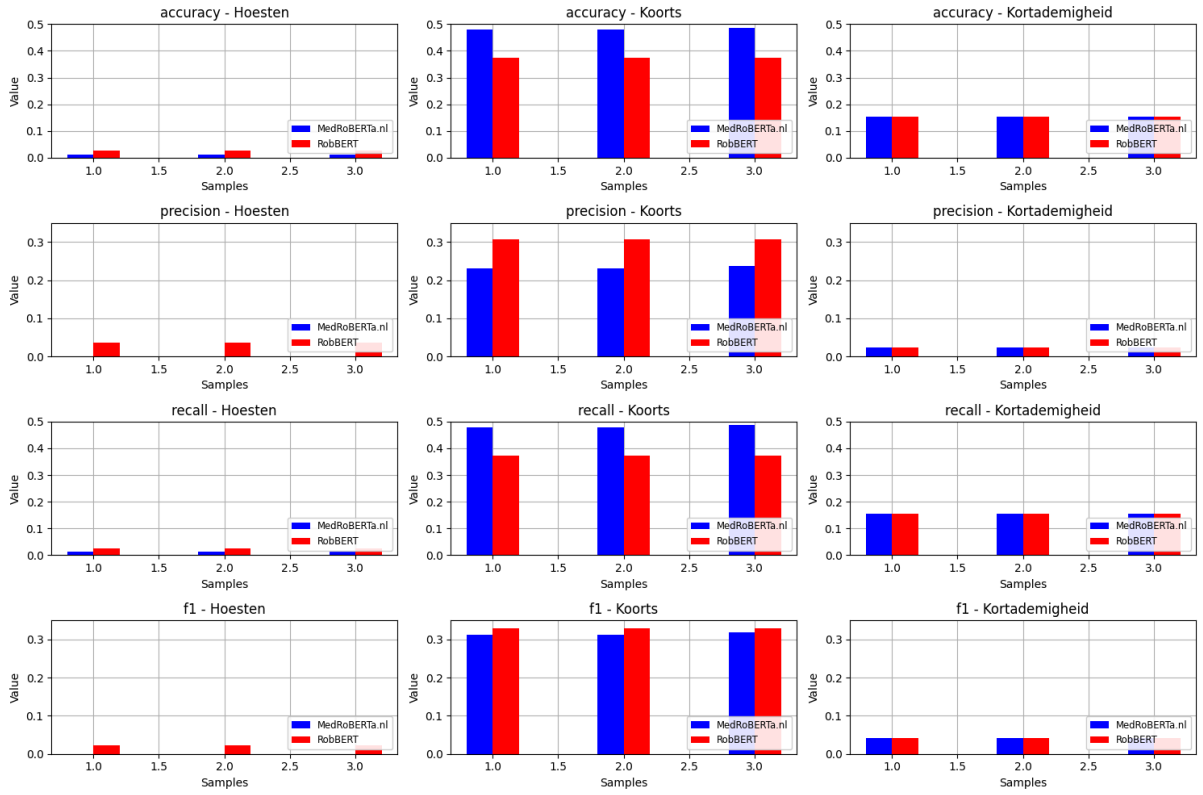

Figure 4. Prediction metrics of prompt-based model experiments, using 5-fold cross-validation

Figure 4 shows metrics varying by symptom, with *Fever* (*'Koorts'*) showing the highest performance with F1-scores of 0.329 for RobBERT and 0.312 for MedRoBERTa.nl. While precision for this symptom is higher for RobBERT as well, MedRoBERTa.nl outperforms RobBERT in accuracy and recall by 0.108 in both metrics. Besides this, adding more samples does not seem to lead to a visible change in any of the tested samples.

Figure 5 shows the same classification metrics, but corresponding to the experiments performed on the direct classifier models versions of MedRoBERTa.nl and RobBERT. To make the image more clear, the x-axes of subplots in this figure have been scaled logarithmically for equidistant distribution of bars.

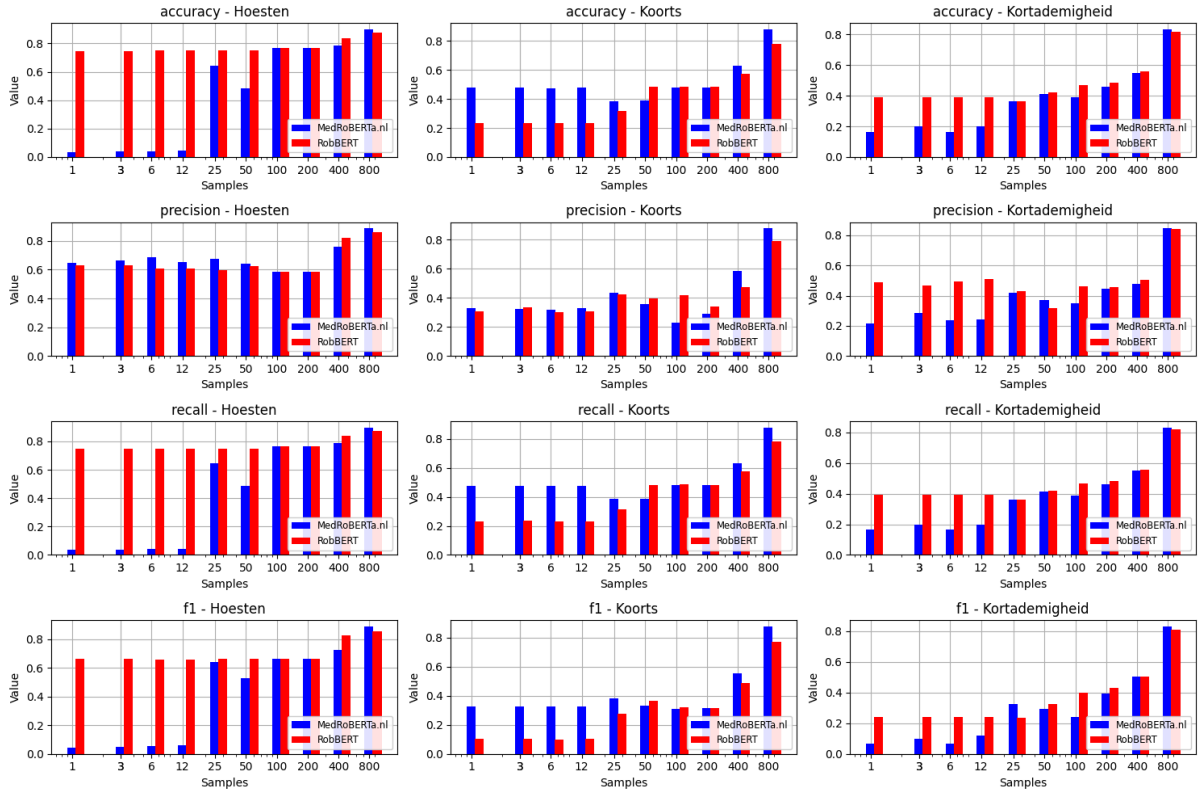

Figure 5. Prediction metrics of direct classifier experiments, using 5-fold cross-validation

Looking at the largest dataset size consisting of 800 samples, MedRoBERTa.nl outperforms RobBERT on all measured prediction metrics independent of the symptom. For F1-score, MedRoBERTa.nl scores 4.2%, 14.0% and 2.9% better than RobBERT in '*Hoesten*', '*Koorts*' and '*Kortademigheid*' respectively. Concerning the other dataset sizes, the results vary more per symptom. In general, RobBERT outperforms MedRoBERTa.nl in '*Hoesten*' and '*Kortademigheid*'. The most equally distributed variable, however, '*Koorts*' is still predicted most accurately by MedRoBERTa.nl.

Table 5 below shows the different prediction metrics for each setup type, using the optimal amount of samples found by the loss plots. Setup names correspond to Table 3, located in Study Design.

Setup name	***# of samples***	**Accuracy**	**Precision**	**Recall**	**F1-score** ***(macro)***
DC-MR-Koorts	*800*	0.877	0.878	0.877	0.877
DC-MR-Hoesten	*800*	0.898	0.887	0.898	0.890
DC-MR-Kortademigheid	*800*	0.832	0.845	0.832	0.830
DC-RB-Koorts	*800*	0.780	0.792	0.780	0.770
DC-RB-Hoesten	*800*	0.874	0.864	0.874	0.854
DC-RB-Kortademigheid	*800*	0.817	0.840	0.817	0.807
PB-MR-Koorts	*1*	0.486	0.237	0.486	0.318
PB-MR-Hoesten	*1*	0.012	0.000	0.012	0.000
PB-MR-Kortademigheid	*1*	0.155	0.246	0.155	0.042
PB-RB-Koorts	*1*	0.374	0.307	0.374	0.329
PB-RB-Hoesten	*1*	0.027	0.038	0.027	0.024
PB-RB-Kortademigheid	*1*	0.155	0.025	0.155	0.042

Table 5. Prediction metrics for experiments using optimal training set size, using 5-fold cross-validation

Looking at this table, it is again observed that utilising the full dataset reliably leads to optimal classification performance in both RobBERT and MedRoBERTa.nl when examining direct classification..

MedRoBERTa.nl performs the best overall on each symptom, when fine-tuned as a direct classifier. A difference in F1-score averaging ~*7%* is observed when comparing this model to RobBERT applied as a direct classifier.

Results for *Hoesten* and *Kortademigheid* are shown to be much lower than *Koorts* for the prompt-based models. This might, again, be due to large parts of the prompt remaining unchanged independently of given samples, while still influencing the outcome, leading to one class being predicted disproportionately often.

5 Conclusions

This study has investigated the performance of locally applicable direct classifiers and prompt-based LLMs, for the task of multiclass symptom extraction from Dutch Electronic Health Records. The findings show that small-scale fine-tuned LLMs, such as MedRoBERTa.nl, are able to achieve good performance on this task, with F1-scores up to 0.89, provided enough data is used to fine-tune the model.

The fact that MedRoBERTa.nl reliably outperforms RobBERT when trained as a direct classifier shows that fine-tuning a model on Dutch EHR data has a positive effect on its predictive capabilities in symptom extraction.

From the data shown in Figure 4, it can be gathered that adding more samples to the prompt-based setups does not reliably lead to better results. Prediction metrics do not seem to change with added samples. A likely explanation for this is that about two-thirds of the input prompt does not change. As these local models are relatively limited in their capabilities, due to comparatively small vocabulary sizes, their inherent language reasoning capabilities are equally restricted.

The used prompt-based MedRoBERTa.nl and RobBERT models, though interesting conceptually, have shown limited effectiveness in their current state. Compared to the same models fine-tuned to classify directly, the prompt-based models performed markedly worse, which might be a consequence of their limited size compared to current state of the art prompt-based LLMs, the manner in which they were constructed or fine-tuned, or the fact that these models were not trained as an encoder-decoder model from the ground up.

Concerning the subquestions of the main research question *RQ*, the first, *“What are promising small-scale LLMs that could be applied locally to extract symptoms from Dutch clinical notes data?*” has been answered through the model search covered in Section A of the Appendix. The models used in this research are MedRoBERTa.nl and RobBERT, and the results show that MedRoBERTa.nl shows optimal performance when applied as a direct classifier to our subset of the JGPN dataset.

Subquestion 2, "*What is the current classification performance of these models when extracting symptom presence from free-text clinical data, and how do their performances compare (Precision, Recall and F1-Score)?*" is answered in the same manner: while prompt-based methods applied in our experiments do not show results that could be deemed adequate, both MedRoBERTa.nl and RobBERT achieve desirable results when fine-tuned on (part of) the dataset using five-fold cross-validation.

Next, subquestion 3, “*What is the impact of the sample size of available annotated data for model development or fine-tuning on the relative performance of the LMM approaches?*” is answered. In our approach of prompt-based few-shot classification, changing the amount of samples provided in the input prompt did not seem to influence the classification performance of the models. As mentioned previously, this might be due to the models’ limited size and tendency to assign labels to one class due to the prompt’s structure always being the same. This would render the models as they are applied in this thesis incapable of interpreting the prompt effectively.

Concluding, RQ "*How do local LLMs perform when extracting LRTI-related symptoms from free-text Dutch GP clinical notes?*" is answered as follows. The experiments performed in this research have pointed out the following: A larger dataset size leads to better predictive classification performance when the tested local LLMs are used as direct classifiers in the task of symptom extraction from Dutch EHR data, as taken from the labelled subset of the JGPN dataset. This increase might halt at some larger amount of used training data, but due to the limited amount of labelled data, confirming this remains out of scope for this research.

When the models are made to classify in a prompt-based few-shot setting where the training set is expressed as samples in the prompt, this increase in performance is not directly visible. However, the limited size and fine-tuning of the prompt-based models might be creating an upper ceiling for the classification performance of these models.

Lastly, fine-tuning LLMs on medical data before they are trained as direct classifiers might lead to higher classification performance, as suggested by the increase of ~7% on average when comparing MedRoBERTa.nl to RobBERT.

5.1 Discussion

While other research into symptom extraction through either direct classification or prompt-based methods using MedRoBERTa.nl were not found, the original MedRoBERTa.nl paper by Verkijk *et al.* does fine-tune the model to perform direct classification of EHR data on ICF categories [4]. Micro-level F1-scores in this task ranged from 0.40 to 0.69, compared to the (macro-level) range of 0.77 to 0.89 found in direct classifier experiments pertaining to this report.

A direct comparison between these experiments cannot be made however, as both datasets and target variables differ. No published works on clinical variable extraction or classification have been found using RobBERT as a base model.

Compared to the wealth of research done in English clinical NLP, the field of Dutch clinical NLP is decidedly smaller, especially when narrowing it further down to research focusing on LLMs. This disparity might partly be attributed to most state-of-the-art LLMs being closed-source, cloud-based and thus not privacy-friendly [59], [60]. Additionally, issues associated with the lack of readily available and high-quality EHR datasets, especially in Dutch, might further complicate this issue [54], [61], [62].

(Ethical) Implications and Considerations

This subsection contains a summation of implications and considerations of this report's research and outcomes, both concerning ethical standards and other relevant ones.

The first and main concern of the research done in this paper is the used EHR data being highly privacy-sensitive. While data is in large part de-identified [54], the risk of re-identification attacks (i.e. recombining different data to re-identify an individual based on a dataset) should always be considered when using privacy-sensitive data.

Moreover, a factor that should be considered when applying models trained on this dataset is the risk of bias. For instance, the geographic locations of general practitioners recording the data might influence the variety of patient demographics and socioeconomic groups present in the dataset, potentially reducing generalisability.

Thirdly, due to the complexity arising from the used models' architectures, these models are opaque, providing a largely black-box solution to the problem. This makes the reasoning behind the models' results much more difficult to explain, even when resorting to algorithms developed for this reason.

One final consideration in terms of replicability is the fact that the used training data is not publicly available. This makes replicating the experiments of this thesis near impossible by any group or individual who is not affiliated with the research institution this thesis was completed at (University Medical Center Utrecht).

Limitations and Future Iterations

This subsection will concern itself with limitations of the research carried out, as well as what could have been added or done differently.

As has already been mentioned, the main limiting factors of this research are the availability of only a small dataset of 1,000 labelled instances, bundled with the limited computational resources available in the virtual environment the experiments were executed in. Even though the data is anonymised [54], it was still not allowed to leave the secure digital environment, severely limiting the relevant and applicable models to a small subset of smaller, older and thus not optimally performing models.

Future iterations should address this issue. Additionally, larger and newer models should be applied, as they are expected to provide much more adequate results, given the recent success of LLMs in both online and offline settings. A higher classification performance on prompt-based few-shot symptom extraction can likely be reached using larger, state-of-the-art models, some of which could also be applied locally, given more computational resources outside of a virtual environment. This would have the additional benefit of allowing for larger sample sizes in prompt-based model testing, as the maximum prompt length generally increases with the amount of parameters and/or vocabulary a model has.

Besides this, a more thorough data preparation stage could also increase results. For instance, creating a mapping of domain-specific terminology and abbreviations to more widely known terms might allow for a larger proportion of the text being present in the models' vocabularies.

In a future iteration of this research, less prevalent symptoms could be addressed as target variables as well. Due to both the duration and computational limitations, the selection of target variables was kept to three symptoms in this research.

Finally, different types of prompts could be tested through prompt-tuning in order to examine what constitutes an 'optimal' prompt. As prompt-based models are not fine-tuned for the classification task themselves but instead perform classification based on their inherent linguistic capabilities, the main influence that is easy to manipulate and iterate on without large amounts of data and computational power needed is the prompt that is used, and its structure. Optimising the prompt might lead to better results.

6 Acknowledgements

I would like to express my sincere gratitude to the following people, who have been instrumental in the completion of this thesis.

First and foremost, Tuur Leeuwenberg, my primary supervisor. There were some moments of self-doubt during my thesis, but your insight and enthusiasm for the subject allowed me to get through these, and complete my thesis. I am also grateful to Isa Spiero for her additional support and supervision during the research process.

I would also like to extend my gratitude to Dr. Marijn Schraagen, my secondary examiner, for providing comprehensive feedback on my research proposal. Furthermore, I am thankful to Merijn Rijk and the entire JGPN team for providing the dataset used in this report's experiments: Without them, this thesis would not have been possible.

Finally, I would like to express my appreciation to my family and friends in and around Utrecht. Their support throughout this process has been vital. I could always count on them to lend a listening ear, whether I was discussing why I find my research so interesting, or feeling a bit overwhelmed.

7 References

[1] OpenAI, ‘GPT-4 Technical Report’. arXiv, Mar. 27, 2023. doi: 10.48550/arXiv.2303.08774.

[2] A. E. Nicholson *et al.*, ‘BARD: A structured technique for group elicitation of Bayesian networks to support analytic reasoning’, *Risk Anal.*, vol. 42, no. 6, pp. 1155–1178, Jun. 2022, doi: 10.1111/risa.13759.

[3] Y. Moslem, R. Haque, J. D. Kelleher, and A. Way, ‘Adaptive Machine Translation with Large Language Models’. arXiv, May 09, 2023. doi: 10.48550/arXiv.2301.13294.

[4] S. Verkijk and P. Vossen, ‘MedRoBERTa.nl: A Language Model for Dutch Electronic Health Records’, *Comput. Linguist. Neth. J.*, vol. 11, pp. 141–159, Dec. 2021.

[5] A. Vaswani *et al.*, ‘Attention Is All You Need’. arXiv, Aug. 01, 2023. Accessed: Jun. 18, 2024. [Online]. Available: http://arxiv.org/abs/1706.03762

[6] E. D. Liddy, ‘Natural Language Processing’.

[7] ‘Hey Siri: An On-device DNN-powered Voice Trigger for Apple’s Personal Assistant’, Apple Machine Learning Research. Accessed: Nov. 20, 2023. [Online]. Available: https://machinelearning.apple.com/research/hey-siri

[8] A. Khadjeh Nassirtoussi, S. Aghabozorgi, T. Ying Wah, and D. C. L. Ngo, ‘Text mining for market prediction: A systematic review’, *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7653–7670, Nov. 2014, doi: 10.1016/j.eswa.2014.06.009.

[9] ‘What are Large Language Models? - LLM AI Explained - AWS’, Amazon Web Services, Inc. Accessed: Dec. 13, 2023. [Online]. Available: https://aws.amazon.com/what-is/large-language-model/

[10] D. V. Veen *et al.*, ‘Clinical Text Summarization: Adapting Large Language Models Can Outperform Human Experts’, *Res. Sq.*, doi: 10.21203/rs.3.rs-3483777/v1.

[11] J. Kaddour, J. Harris, M. Mozes, H. Bradley, R. Raileanu, and R. McHardy, ‘Challenges and Applications of Large Language Models’. arXiv, Jul. 19, 2023. doi: 10.48550/arXiv.2307.10169.

[12] P. Villalobos, A. Ho, J. Sevilla, T. Besiroglu, L. Heim, and M. Hobbhahn, ‘Will we run out of data? Limits of LLM scaling based on human-generated data’. arXiv, Jun. 04, 2024. Accessed: Jun. 19, 2024. [Online]. Available: http://arxiv.org/abs/2211.04325

[13] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, ‘On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜’, in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, in FAccT ’21. New York, NY, USA: Association for Computing Machinery, Mar. 2021, pp. 610–623. doi: 10.1145/3442188.3445922.

[14] M. Nezhurina, L. Cipolina-Kun, M. Cherti, and J. Jitsev, ‘Alice in Wonderland: Simple Tasks Showing Complete Reasoning Breakdown in State-Of-the-Art Large Language Models’. arXiv, Jun. 05, 2024. doi: 10.48550/arXiv.2406.02061.

[15] L. Chen, M. Zaharia, and J. Zou, ‘How is ChatGPT’s behavior changing over time?’ arXiv, Jul. 18, 2023. doi: 10.48550/arXiv.2307.09009.

[16] Y. Zhu *et al.*, ‘Large Language Models for Information Retrieval: A Survey’. arXiv, Aug. 15, 2023. doi: 10.48550/arXiv.2308.07107.

[17] M. Agrawal, S. Hegselmann, H. Lang, Y. Kim, and D. Sontag, ‘Large Language Models are Few-Shot Clinical Information Extractors’. arXiv, Nov. 30, 2022. doi: 10.48550/arXiv.2205.12689.

[18] L. C. Adams *et al.*, ‘Leveraging GPT-4 for Post Hoc Transformation of Free-text Radiology Reports into Structured Reporting: A Multilingual Feasibility Study’, *Radiology*, vol. 307, no. 4, p. e230725, May 2023, doi: 10.1148/radiol.230725.

[19] T. B. Brown *et al.*, ‘Language Models are Few-Shot Learners’. arXiv, Jul. 22, 2020. doi: 10.48550/arXiv.2005.14165.

[20] F. Li, Y. Jin, W. Liu, B. P. S. Rawat, P. Cai, and H. Yu, ‘Fine-Tuning Bidirectional Encoder Representations From Transformers (BERT)–Based Models on Large-Scale Electronic

Health Record Notes: An Empirical Study', *JMIR Med. Inform.*, vol. 7, no. 3, p. e14830, Sep. 2019, doi: 10.2196/14830.
[21] L. Rasmy, Y. Xiang, Z. Xie, C. Tao, and D. Zhi, 'Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction', *Npj Digit. Med.*, vol. 4, no. 1, Art. no. 1, May 2021, doi: 10.1038/s41746-021-00455-y.
[22] A. T. Kalai and S. S. Vempala, 'Calibrated Language Models Must Hallucinate'. arXiv, Mar. 19, 2024. doi: 10.48550/arXiv.2311.14648.
[23] H. Alkaissi and S. I. McFarlane, 'Artificial Hallucinations in ChatGPT: Implications in Scientific Writing', 2023.
[24] R. Azamfirei, S. R. Kudchadkar, and J. Fackler, 'Large language models and the perils of their hallucinations', *Crit. Care*, vol. 27, no. 1, p. 120, Mar. 2023, doi: 10.1186/s13054-023-04393-x.
[25] P. Manakul, A. Liusie, and M. J. F. Gales, 'SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models'. arXiv, May 07, 2023. Accessed: Aug. 31, 2023. [Online]. Available: http://arxiv.org/abs/2303.08896
[26] M. Wornow *et al.*, 'The shaky foundations of large language models and foundation models for electronic health records', *Npj Digit. Med.*, vol. 6, no. 1, pp. 1–10, Jul. 2023, doi: 10.1038/s41746-023-00879-8.
[27] G. Eysenbach, 'The Role of ChatGPT, Generative Language Models, and Artificial Intelligence in Medical Education: A Conversation With ChatGPT and a Call for Papers', *JMIR Med. Educ.*, vol. 9, no. 1, p. e46885, Mar. 2023, doi: 10.2196/46885.
[28] 'An important next step on our AI journey', Google. Accessed: Aug. 15, 2023. [Online]. Available: https://blog.google/technology/ai/bard-google-ai-search-updates/
[29] Y. Mehdi, 'Announcing the next wave of AI innovation with Microsoft Bing and Edge', The Official Microsoft Blog. Accessed: Aug. 15, 2023. [Online]. Available: https://blogs.microsoft.com/blog/2023/05/04/announcing-the-next-wave-of-ai-innovation-with-microsoft-bing-and-edge/
[30] H. Touvron *et al.*, 'LLaMA: Open and Efficient Foundation Language Models'. arXiv, Feb. 27, 2023. doi: 10.48550/arXiv.2302.13971.
[31] H. Touvron *et al.*, 'Llama 2: Open Foundation and Fine-Tuned Chat Models'. arXiv, Jul. 19, 2023. doi: 10.48550/arXiv.2307.09288.
[32] K. Lee *et al.*, 'Deduplicating Training Data Makes Language Models Better'. arXiv, Mar. 24, 2022. doi: 10.48550/arXiv.2107.06499.
[33] D. Patterson *et al.*, 'The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink', *Computer*, vol. 55, no. 7, pp. 18–28, Jul. 2022, doi: 10.1109/MC.2022.3148714.
[34] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, 'Green AI'. arXiv, Aug. 13, 2019. doi: 10.48550/arXiv.1907.10597.
[35] C. Wang, K. Cho, and J. Gu, 'Neural Machine Translation with Byte-Level Subwords', *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 05, Art. no. 05, Apr. 2020, doi: 10.1609/aaai.v34i05.6451.
[36] 'Few shot learning with fine-tuned language model for suicidal text detection'. Accessed: Oct. 24, 2023. [Online]. Available: https://www.researchsquare.com
[37] E. Hossain *et al.*, 'Natural Language Processing in Electronic Health Records in Relation to Healthcare Decision-making: A Systematic Review'. arXiv, Jun. 22, 2023. Accessed: Aug. 30, 2023. [Online]. Available: http://arxiv.org/abs/2306.12834
[38] R. A. Bush, C. Kuelbs, J. Ryu, W. Jiang, and G. Chiang, 'Structured Data Entry in the Electronic Medical Record: Perspectives of Pediatric Specialty Physicians and Surgeons', *J. Med. Syst.*, vol. 41, no. 5, p. 75, May 2017, doi: 10.1007/s10916-017-0716-5.
[39] H. Naveed *et al.*, 'A Comprehensive Overview of Large Language Models'. arXiv, Aug. 18, 2023. doi: 10.48550/arXiv.2307.06435.
[40] V. Menger, F. Scheepers, L. M. van Wijk, and M. Spruit, 'DEDUCE: A pattern matching method for automatic de-identification of Dutch medical text', *Telemat. Inform.*, vol. 35, no. 4, pp. 727–736, Jul. 2018, doi: 10.1016/j.tele.2017.08.002.

[41] D. Aronsky, M. Fiszman, W. W. Chapman, and P. J. Haug, ‘Combining decision support methodologies to diagnose pneumonia’, *Proc. AMIA Symp.*, pp. 12–16, 2001.
[42] P. J. Haug, S. Koehler, L. M. Lau, P. Wang, R. Rocha, and S. M. Huff, ‘Experience with a mixed semantic/syntactic parser’, *Proc. Symp. Comput. Appl. Med. Care*, pp. 284–288, 1995.
[43] H. Xu *et al.*, ‘Extracting and integrating data from entire electronic health records for detecting colorectal cancer cases’, *AMIA Annu. Symp. Proc. AMIA Symp.*, vol. 2011, pp. 1564–1572, 2011.
[44] A. W. Forsyth *et al.*, ‘Machine Learning Methods to Extract Documentation of Breast Cancer Symptoms From Electronic Health Records’, *J. Pain Symptom Manage.*, vol. 55, no. 6, pp. 1492–1499, Jun. 2018, doi: 10.1016/j.jpainsymman.2018.02.016.
[45] B. J. Kenner *et al.*, ‘Early Detection of Pancreatic Cancer: Applying Artificial Intelligence to Electronic Health Records’, *Pancreas*, vol. 50, no. 7, pp. 916–922, Aug. 2021, doi: 10.1097/MPA.0000000000001882.
[46] E. Menasalvas Ruiz *et al.*, ‘Profiling Lung Cancer Patients Using Electronic Health Records’, *J. Med. Syst.*, vol. 42, no. 7, p. 126, May 2018, doi: 10.1007/s10916-018-0975-9.
[47] A. Swaminathan *et al.*, ‘Extraction of Unstructured Electronic Health Records to Evaluate Glioblastoma Treatment Patterns’, *JCO Clin. Cancer Inform.*, vol. 8, p. e2300091, Jun. 2024, doi: 10.1200/CCI.23.00091.
[48] J. Downs *et al.*, ‘Detection of Suicidality in Adolescents with Autism Spectrum Disorders: Developing a Natural Language Processing Approach for Use in Electronic Health Records’, *AMIA. Annu. Symp. Proc.*, vol. 2017, pp. 641–649, Apr. 2018.
[49] M. Raja, ‘Suicide risk in adults with Asperger’s syndrome’, *Lancet Psychiatry*, vol. 1, no. 2, pp. 99–101, Jul. 2014, doi: 10.1016/S2215-0366(14)70257-3.
[50] E. Simonoff, A. Pickles, T. Charman, S. Chandler, T. Loucas, and G. Baird, ‘Psychiatric disorders in children with autism spectrum disorders: prevalence, comorbidity, and associated factors in a population-derived sample’, *J. Am. Acad. Child Adolesc. Psychiatry*, vol. 47, no. 8, pp. 921–929, Aug. 2008, doi: 10.1097/CHI.0b013e318179964f.
[51] Y. Barak-Corren *et al.*, ‘Predicting Suicidal Behavior From Longitudinal Electronic Health Records’, *Am. J. Psychiatry*, vol. 174, no. 2, pp. 154–162, Feb. 2017, doi: 10.1176/appi.ajp.2016.16010077.
[52] H. Wu *et al.*, ‘Knowledge Driven Phenotyping’, *Stud. Health Technol. Inform.*, vol. 270, pp. 1327–1328, Jun. 2020, doi: 10.3233/SHTI200425.
[53] R. Leaman, R. Khare, and Z. Lu, ‘Challenges in clinical natural language processing for automated disorder normalization’, *J. Biomed. Inform.*, vol. 57, pp. 28–37, Oct. 2015, doi: 10.1016/j.jbi.2015.07.010.
[54] M. H. Rijk *et al.*, ‘Incomplete and possibly selective recording of signs, symptoms, and measurements in free text fields of primary care electronic health records of adults with lower respiratory tract infections’, *J. Clin. Epidemiol.*, vol. 166, Feb. 2024, doi: 10.1016/j.jclinepi.2023.111240.
[55] H. M. Smeets *et al.*, ‘Routine primary care data for scientific research, quality of care programs and educational purposes: The Julius General Practitioners’ Network (JGPN)’, *BMC Health Serv. Res.*, vol. 18, no. 1, Sep. 2018, doi: 10.1186/s12913-018-3528-5.
[56] Y. Li, Z. Li, K. Zhang, R. Dan, S. Jiang, and Y. Zhang, ‘ChatDoctor: A Medical Chat Model Fine-Tuned on a Large Language Model Meta-AI (LLaMA) Using Medical Domain Knowledge’. arXiv, Jun. 24, 2023. Accessed: Oct. 14, 2023. [Online]. Available: http://arxiv.org/abs/2303.14070
[57] H. Li, A. C. Graesser, and Z. Cai, ‘Comparison of Google Translation with Human Translation’.
[58] T. Yin, ‘translate: This is a simple, yet powerful command line translator with google translate behind it. You can also use it as a Python module in your code.’ Accessed: Jun. 12, 2024. [OS Independent]. Available: https://github.com/terryyin/google-translate-python

[59] A. Névéol, H. Dalianis, S. Velupillai, G. Savova, and P. Zweigenbaum, ‘Clinical Natural Language Processing in languages other than English: opportunities and challenges’, *J. Biomed. Semant.*, vol. 9, no. 1, p. 12, Mar. 2018, doi: 10.1186/s13326-018-0179-8.
[60] L. Li *et al.*, ‘A scoping review of using Large Language Models (LLMs) to investigate Electronic Health Records (EHRs)’. arXiv, May 22, 2024. doi: 10.48550/arXiv.2405.03066.
[61] M. K. Kim, C. Rouphael, J. McMichael, N. Welch, and S. Dasarathy, ‘Challenges in and Opportunities for Electronic Health Record-Based Data Analysis and Interpretation’, *Gut Liver*, vol. 18, no. 2, pp. 201–208, Mar. 2024, doi: 10.5009/gnl230272.
[62] C. Xiao, E. Choi, and J. Sun, ‘Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review’, *J. Am. Med. Inform. Assoc.*, vol. 25, no. 10, pp. 1419–1428, Oct. 2018, doi: 10.1093/jamia/ocy068.
[63] L. Martin *et al.*, ‘CamemBERT: a Tasty French Language Model’, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7203–7219. doi: 10.18653/v1/2020.acl-main.645.
[64] W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim, ‘BERTje: A Dutch BERT Model’. arXiv, Dec. 19, 2019. doi: 10.48550/arXiv.1912.09582.
[65] T. Pires, E. Schlinger, and D. Garrette, ‘How multilingual is Multilingual BERT?’ arXiv, Jun. 04, 2019. doi: 10.48550/arXiv.1906.01502.
[66] P. Delobelle, T. Winters, and B. Berendt, ‘RobBERT: a Dutch RoBERTa-based Language Model’. arXiv, Sep. 16, 2020. doi: 10.48550/arXiv.2001.06286.

8 Planned Schedule

The schedule in Table 6 provides an overview of the tasks and objectives on a per-week basis. It mainly serves an illustrative purpose, as the actual progress of individual tasks and objectives happened less linearly than the table would make it seem, and smaller tasks were excluded from it.

Week	***Task / Objective***
50 (11/12/2023 - 17/12/2023)	Process comments on proposal; Create example visualisations
51 (18/12/2023 - 24/12/2023)	Step-by-step / roadmap of experiments; Presentation of step-by-step
52 (25/12/2023 - 31/12/2023)	*Christmas Holiday*
1 (01/01/2024 - 07/01/2024)	*Christmas Holiday*
2 (08/01/2024 - 14/01/2024)	Determining what LLMs to use
3 (15/01/2024 - 21/01/2024)	Determining what LLMs to use
4 (22/01/2024 - 28/01/2024)	Setting up scripts for usage of LLMs
5 (29/01/2024 - 04/02/2024)	Finding HealthCareMagic dataset
6 (05/02/2024 - 11/02/2024)	Preparing machine-translation script for HealthCareMagic dataset
7 (12/02/2024 - 18/02/2024)	fine-tuning prompt-based models using translated HealthCareMagic dataset
8 (19/02/2024 - 25/02/2024)	fine-tuning prompt-based models using translated HealthCareMagic dataset
9 (26/02/2024 - 03/03/2024)	Evaluating models on fake data
10 (04/03/2024 - 10/03/2024)	Accessing data via anDREa
11 (11/03/2024 - 17/03/2024)	Data exploration and preparation
12 (18/03/2024 - 24/03/2024)	Data exploration and preparation
13 (25/03/2024 - 31/03/2024)	Setup direct classifier experiments
14 (01/04/2024 - 07/04/2024)	Running direct classifier experiments
15 (08/04/2024 - 14/04/2024)	Running direct classifier experiments
16 (15/04/2024 - 21/04/2024)	Setup prompt-based experiments
17 (22/04/2024 - 28/04/2024)	Running prompt-based experiments
18 (29/04/2024 - 05/05/2024)	Running prompt-based experiments
19 (06/05/2024 - 12/05/2024)	Update report based on experiment changes
20 (13/05/2024 - 19/05/2024)	Rerunning direct classifier experiments
21 (20/05/2024 - 26/05/2024)	Results generation
22 (27/05/2024 - 02/06/2024)	Visualisation of results
23 (03/06/2024 - 09/06/2024)	Visualisation of results
24 (10/06/2024 - 16/06/2024)	Write Results and Conclusion; Hand in draft for proofreading
25 (17/06/2024 - 23/06/2024)	Create final version of report
26 (24/06/2024 - 30/06/2024)	Final Thesis Report

27 (01/07/2024 - 07/07/2024)	Thesis Defence (Preliminary)

Table 6. Schedule of most important tasks per week

Appendix

A. Comparison of Models

To address the first research question (*RQ1*), a selection of viable Large Language Models (LLMs) had to be made. These selections are made based on what local models are available at the time of writing.

Model Search

To find a list of relevant models, a search of relevant literature was combined with a direct search for models on the HuggingFace website. These websites provide comprehensive filtering options, enabling us to find models that can be run offline rather easily.

Relevant models were located using both relevant literature and search engines present on the HuggingFace website. This website provides comprehensive filtering options, enabling the user to find models that can be run offline with relative ease. One difficult measure is determining when a model will be truly available to be run locally, as exact hardware requirements are generally not specified on HuggingFace. In this search, lighter (i.e. less computationally intense) models with good performance are preferred over heavy models with extremely good performance.

In the context of this research, the search of models has been kept limited to LLMs that show promise in classification tasks. Classification can be orchestrated in multiple ways. For example, a conversational LLM is also able to classify when correct instructions are provided, which is what experiments in this paper also focus on.

Selection Criteria

Before commencing the search for these models, however, some important aspects considering what would make a model viable for this research had to be formulated. This resulted in the following set of factors (must-haves are underlined, other factors are nice-to-haves):

- *Local Execution Capability*: As mentioned in Background, clinical notes contain a multitude of sensitive data points that can be used to identify individuals. Therefore, it is mandatory that the model is able to completely execute on local hardware, as no EHR data is allowed to be transmitted from the machine it is stored on. For this reason, in this research the decision was made to only utilise open-source, locally applicable LLMs in an effort to minimise the probability of a data leak.

- *Availability of a pre-trained model*: The model must be pretrained, as this report's research mainly looks into the efficacy of LLMs as zero-shot, one-shot and few-shot classifiers;

- *Parameter count of a model*: The amount of parameters in a model represents the size and complexity of a model and thus correlates with the computational power required to run it. As computational power is a limiting factor due to the dataset's privacy regulations, it is of utmost importance to strike a balance between having enough parameters to get suitable results, while not having so many that the model cannot run on the virtual desktop;

- *Dutch language competency*: It is vital for the model to be able to handle Dutch language data, whether that be through optimization for the Dutch language or it being completely trained on a Dutch dataset;

- *Medical text familiarity*: Models trained or fine-tuned on medical texts are more likely to be able to correctly process domain-specific knowledge and jargon, and are thus preferred.

As models that adhere to all of these factors to a good standard are very specific to the problem at hand, they are expected to perform rather well, but are likely few and far between for the same reason.

To find the models via HuggingFace, the following filters were applied during a search performed on 18/10/2023:

- Tasks: Text Classification, Zero-Shot Classification, Conversational, Text Generation, Fill-Mask
- Libraries: PyTorch, TensorFlow, Transformers, Sentence Transformers, Adapter Transformers,
- Datasets: *Not specified*
- Languages: Dutch
- Licences: *Not specified*
- Other: *Not specified*

In the context of this research, the search of models has been kept limited to LLMs that show promise in classification tasks. Classification can be orchestrated in multiple ways, e.g. by directly training the model to classify data using a labelled dataset, or by providing instructions to a prompt-based conversational LLM. The experiments performed in this paper focus on both of these options, and compare them across different sample sizes.

Final Selection

Through a search of available models, the following models are selected as the most relevant. These models have been found by using both relevant literature and searching on GitHub and Huggingface, basing the searches on the factors mentioned at the start of A: Comparison of Models (Appendix). As mentioned there, it is not a must-have for a model to be (partly) trained on medical data, although there is a preference for models that are; they are ranked higher in the chosen list.

Table 7 shows all considered models and a comparison of their relevant features, resulting in the following selection. Below is a list of the four models chosen for experimentation, together with a short description and motivation on why they are expected to provide interesting results.

1. MedRoBERTa.nl (117M)
    - Description: A Dutch medical model created by researchers at Vrije Universiteit Amsterdam, who used 13GB of text data from Dutch hospital notes to train an altered version of the RoBERTa model. This model has shown to perform better on this data in odd-one-out tasks when compared to general Dutch LLMs, and is thus likely to perform well in our research.

- Motivation: Of all chosen models, MedRoBERTa.nl is the only one directly trained on Dutch medical data. Consequently, this model is expected to perform very well on the dataset. One limitation is that the parameter count of the model is rather low by modern standards, which, while being a positive factor in terms of computational efficiency, might reduce the quality of its responses.

2. RobBERT (117M)

    - Description: RobBERT is a model built upon the foundation of BERT, serving as the core architecture. BERT was released as a multilingual model, and RobBERT was fine-tuned as a language-specific model as research points out that doing so generally results in higher performance.[63], [64], [65]

    - Motivation: RobBERT's strong performance on various NLP tasks and proven ability to handle Dutch text make it a suitable candidate for our investigation.[66]

These models will be used for the experiments described in Methodology. While other viable models were also experimented with, most had to be kept out of consideration due to computational and temporal limitations. Consequently, using this list of chosen models, SQ1 *"What are promising small-scale Large Language Models that can be evaluated locally to medical data?"* is answered

B. Table of Considered Models

Below, Table 7 is shown, depicting all models that were taken into consideration for the comparison made in this report's research.

Model	***Base Model***	***#Parameters***	***Prompt-based model available***	***Local?***	***Pretrained?***	***Dutch?***	***Medical?***	***Open-source?***	***Relevant paper***	***Model link***	***Suitable?***
BERTje	BERT	109M	No	Yes	Yes	Yes	No	Yes	Link	GH | HF	Yes
GPT-2 (recycled for Dutch)	GPT-2	129M or 369M	No	Yes	Yes	Yes	No	Yes	Link	HF small, medium	Yes
GPT-2 XL	GPT-2	1.5B	No	Yes	Yes	Multilingual	No	Yes	Link	HF	Yes
DistilBERT-nl	BERT	69M	No	Yes	Yes	Yes	No	Yes	Link	HF	Yes
GPT-3.5/4	GPT-3.5/4	154B / 1.76T	Yes	No	Yes	Multilingual	No	No	Link3.5 / Link4	NA	No
Legal BERT	BERT	295M	No	Yes	Yes	Yes	No	Yes	Link	HF	No
medroBERTa.nl	BERT	117M	Yes	Yes	No (possibly via contact)	Yes	Yes	Yes	Link	GH	Yes
Google Bard	Lambda	137B	Yes	No	Yes	Multilingual	No	No	NA	NA	No

Model	***Base Model***	***#Parameters***	***Prompt-based model available***	***Local?***	***Pretrained?***	***Dutch?***	***Medical?***	***Open-source?***	***Relevant paper***	***Model link***	***Suitable?***
LLaMa	LLaMa	65B	Yes	Yes	Yes	Multilingual	No	No	Link	HF	Yes
LLaMa 2	LLaMa2	7B to 70B	Yes	Yes	Yes	Multilingual	No	No	Link	HF	Yes
ChatDoctor	LLaMa-7B	7B	Yes	Yes	No	Multilingual	Yes	No	Link	GH	Yes
RoBERTa	BERT	355M	Yes	Yes	Yes	Multilingual	No	Yes	Link	HF	Yes
RobBERT-2023	BERT	117M	Yes	Yes	Yes	Yes	No	Yes	Link	HF	Yes
DialoGPT	None	117M, 345M or 762M	Yes	Yes	Yes	Multilingual	No	Yes	Link	GH	Yes, but not used

Table 7. Overview of all considered LLMs

C. Table of Dataset Distribution

Below, Table 8 shows the complete distribution of different ‘Signs and symptoms’ variables present in our labelled EHR dataset. Information present in this table was taken from Rijk *et al.* [54].

	Pneumonia	Acute Bronchitis	Overall	Recorded as pos. (and [*overall*])	Recorded as neg. (and [*overall*])	Not recorded
Patient reported						
Cough	71.9	82.6	76.6	98.4 [*75.4*]	1.6 [*1.2*]	23.4
Fever	57.0	50.3	54.1	57.7 [*31.2*]	43.3 [*23.4*]	45.9
Shortness of Breath	54.4	51.7	53.2	70.9 [*37.7*]	29.1 [*15.5*]	46.8
Sputum	26.1	30.7	28.1	91.1 [*25.6*]	8.9 [*2.5*]	79.9
Chest pain	22.9	13.5	18.8	78.7 [*14.8*]	21.3 [*4.0*]	81.2
Chills	5.7	2.7	4.4	95.5 [*4.2*]	4.5 [*0.2*]	95.6
Patient/GP reported						
Confusion	5.3	1.1	3.5	17.1 [*0.6*]	82.9 [*2.9*]	96.5
GP reported						
Crackles (auscultation)	83.7	89.2	86.1	26.6 [*22.9*]	75.4 [*64.9*]	13.9
Ill appearance	39.1	24.7	32.8	39.3 [*12.9*]	60.7 [*19.9*]	67.2

Table 8. Distribution of signs and symptoms in labelled JGPN dataset

D. Example input prompt

“
Classificeer de volgende teksten op basis van de aanwezigheid van het symptoom 'hoesten' als volgt:

Als 'hoesten' wordt vermeld als voorkomend bij de patiënt, label het als 'Aanwezig.'
Als 'hoesten' wordt vermeld als niet voorkomend bij de patiënt, label het als 'Niet aanwezig.'
Als 'hoesten' helemaal niet wordt vermeld in de tekst, label het als 'Niet vermeld.'

Print exclusief het label bijbehorende aan de volgende tekst:
<insert to-be-classified text here>
“

“
Classificeer de volgende teksten op basis van de aanwezigheid van het symptoom 'hoesten' als volgt:

Als 'hoesten' wordt vermeld als voorkomend bij de patiënt, label het als 'Aanwezig.'
Als 'hoesten' wordt vermeld als niet voorkomend bij de patiënt, label het als 'Niet aanwezig.'
Als 'hoesten' helemaal niet wordt vermeld in de tekst, label het als 'Niet vermeld.'

Voorbeeld:
<insert samples here>

Print exclusief het label bijbehorende aan de volgende tekst:
<insert to-be-classified text here>
”

E. Optimal Amount of Training Epochs in Direct Classifiers

To maximise the amount of experiments to be run within the timeframe of the thesis, MedRoBERTa.nl was trained for 10 epochs. As can be seen by its loss values shown in Figure 6, the model starts overfitting - validation loss rising while training loss continues to decrease - after around epoch five. For this reason, five epochs were chosen as the amount of training epochs for the direct classifier models.

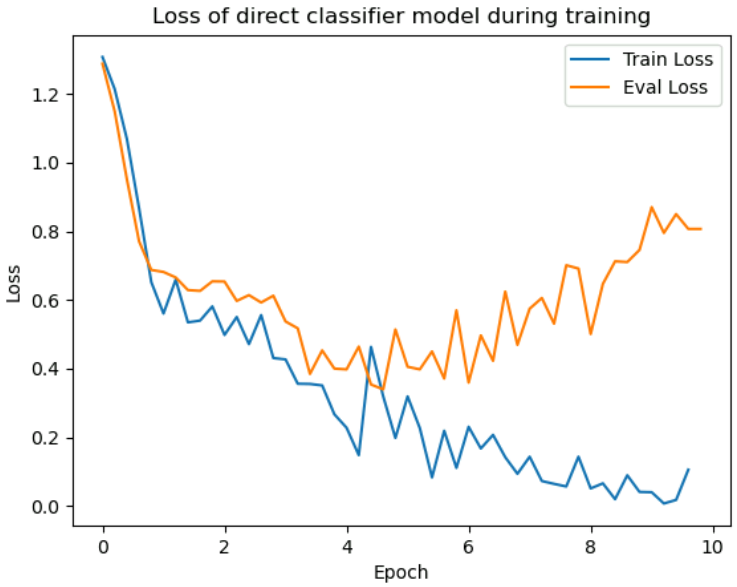

Figure 6. Loss of MedRoBERTa.nl when training for 10 epochs on the full dataset

F. Amount of Parameters in Each Model

Table 9 shows the number of parameters for each model used in this report's research. Prompt-based models show an increase in parameters due to extra layers being added by the EncoderDecoder model (with the base multilingual BERT model as decoder).

Model name	**Model type**	**Number of parameters**
RobBERT	Direct classifier	116,764,419
MedRoBERTa.nl		125,980,419
RobBERT	Prompt-based	254,643,258
MedRoBERTa.nl		263,859,258

Table 9. Parameter counts for each model

G. Individual Loss Plots

Below, a graph is shown for every individual experimental scenario, ordered by increasing training set size, then symptom. Due to the cross-entropy loss not being calculated by HuggingFace Transformers Trainer objects at epoch 0, it might look like some lines represent the model not learning over the epochs (especially the more flat lines, shown in the first few graphs). However, these values likely start higher as the model is initialised, and do improve between epoch zero and one.

One Sample

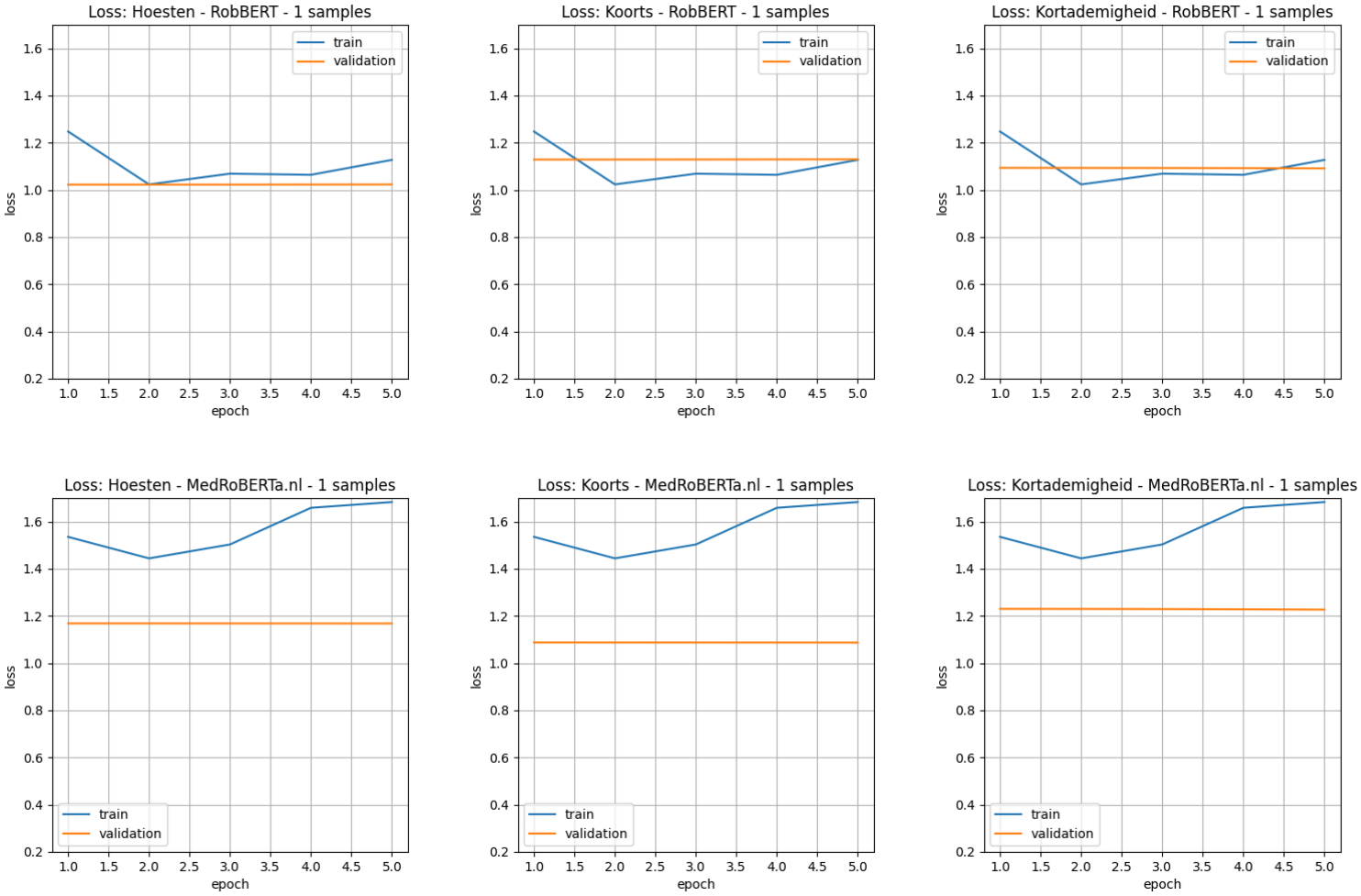

Figure 7. Training and validation loss of direct classifiers trained on one sample, using 5-fold cross-validation

Three Samples

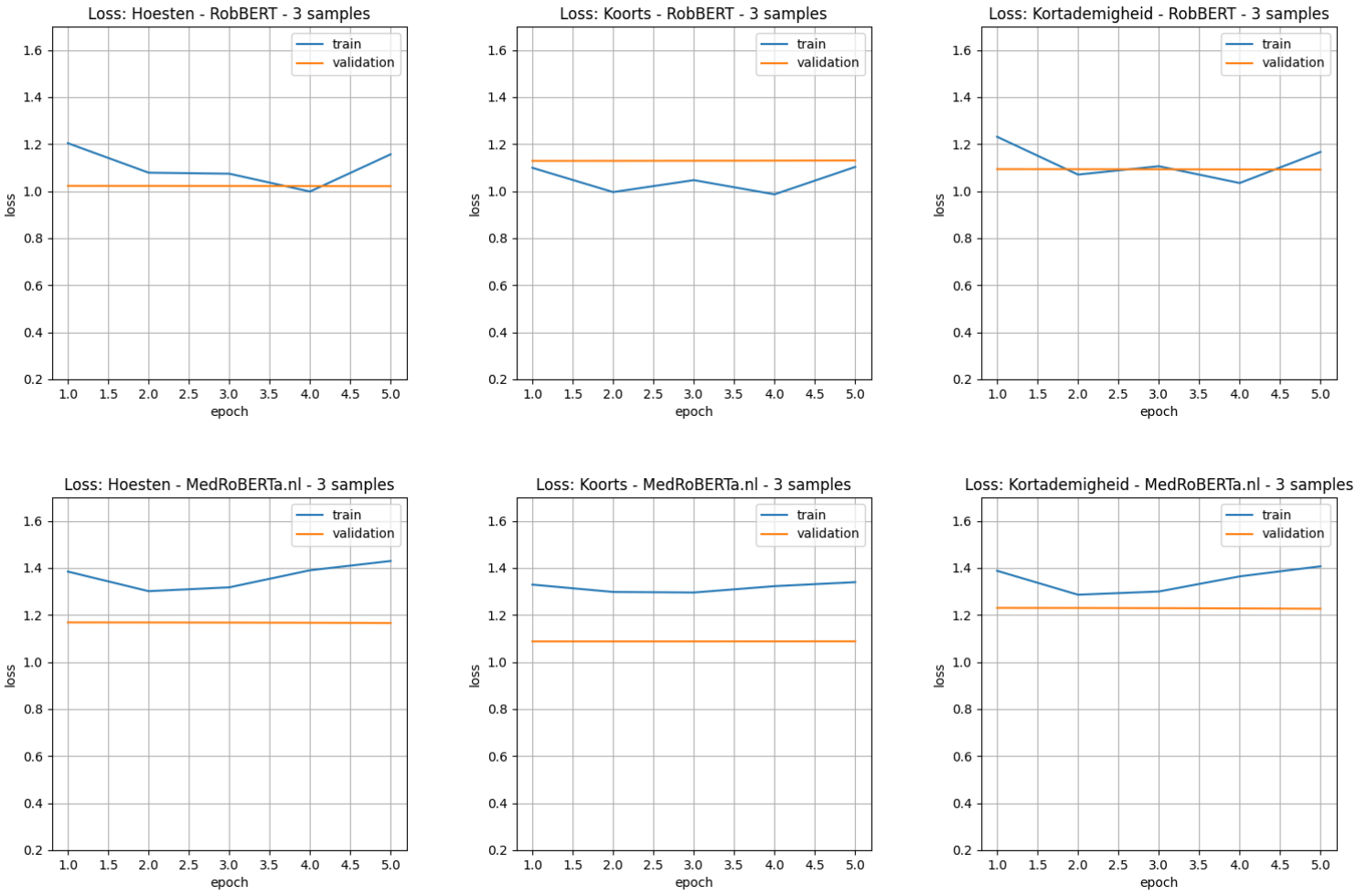

Figure 8. Training and validation loss of direct classifiers trained on three samples, using 5-fold cross-validation

Six Samples

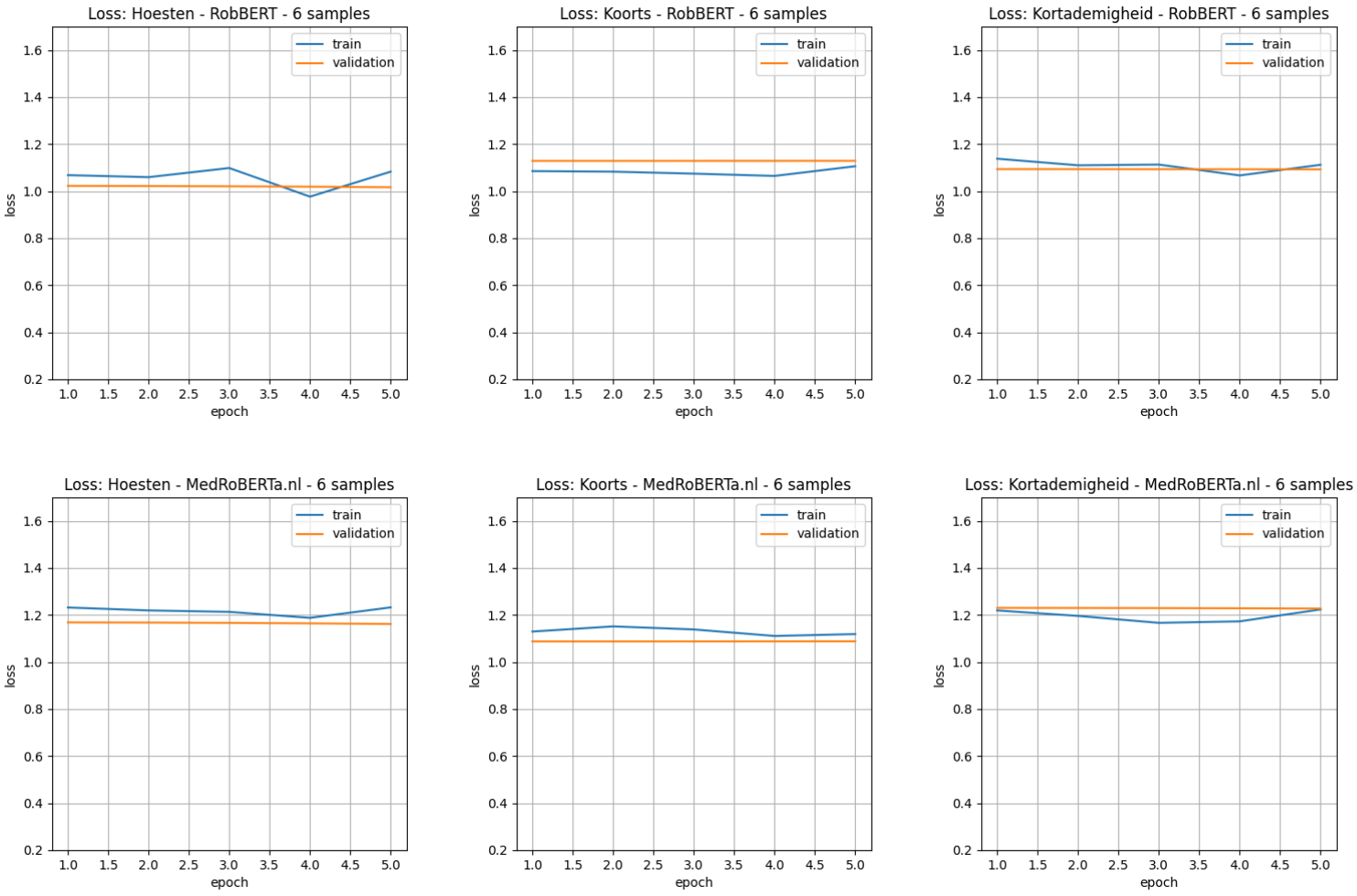

Figure 9. Training and validation loss of direct classifiers trained on six samples, using 5-fold cross-validation

Twelve Samples

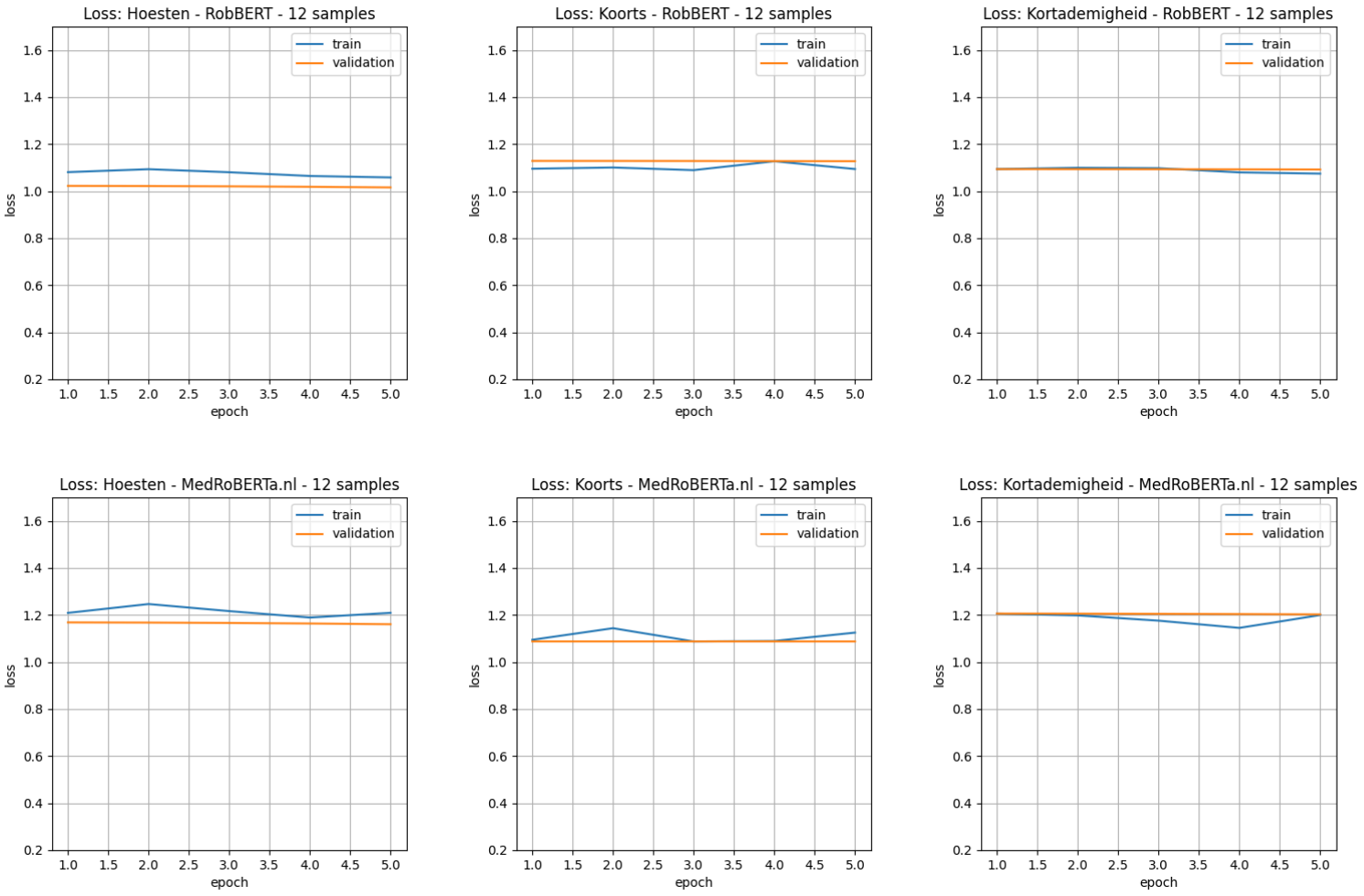

Figure 10. Training and validation loss of direct classifiers trained on twelve samples, using 5-fold cross-validation

25 Samples

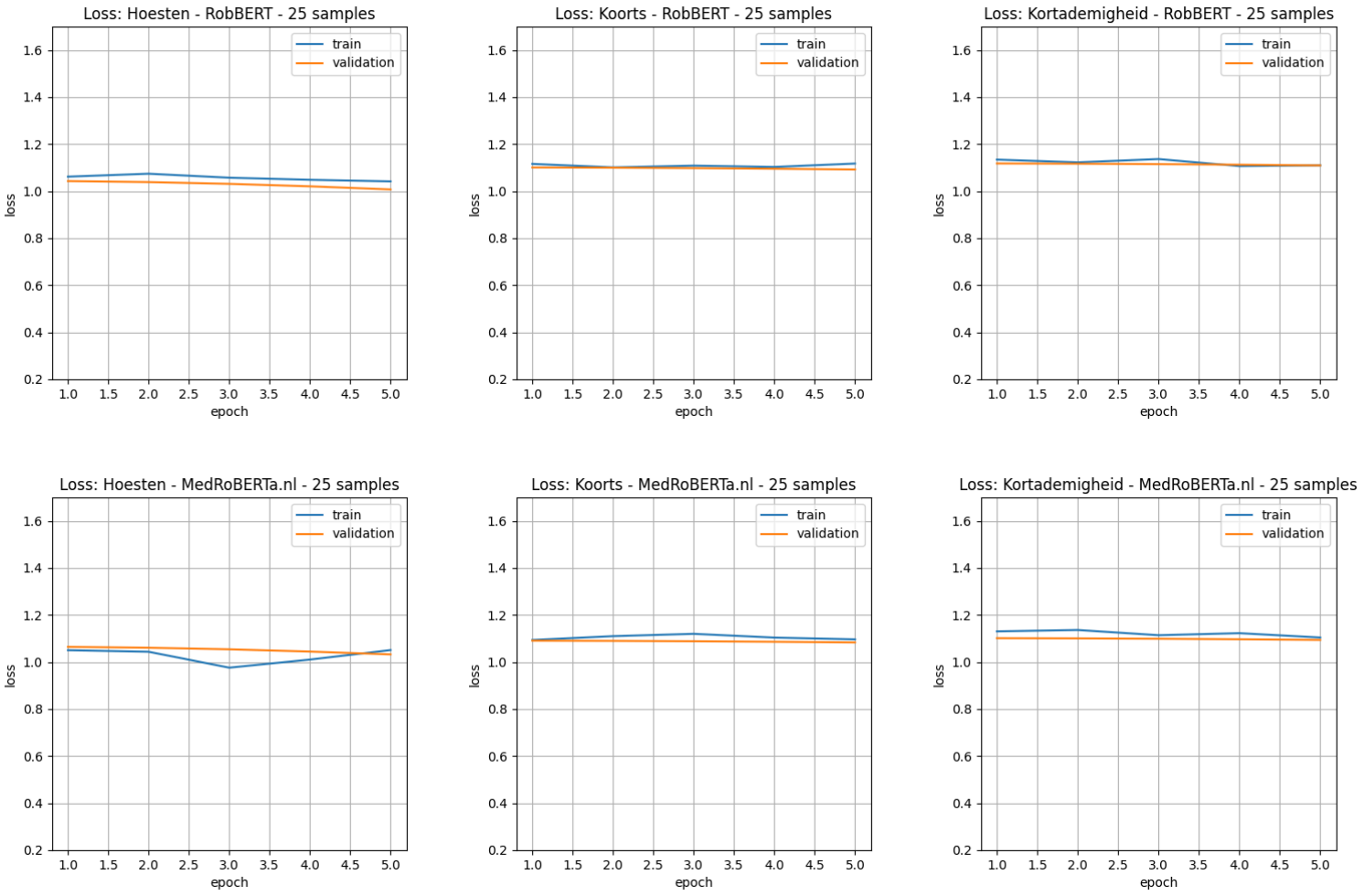

Figure 11. Training and validation loss of direct classifiers trained on 25 samples, using 5-fold cross-validation

50 samples

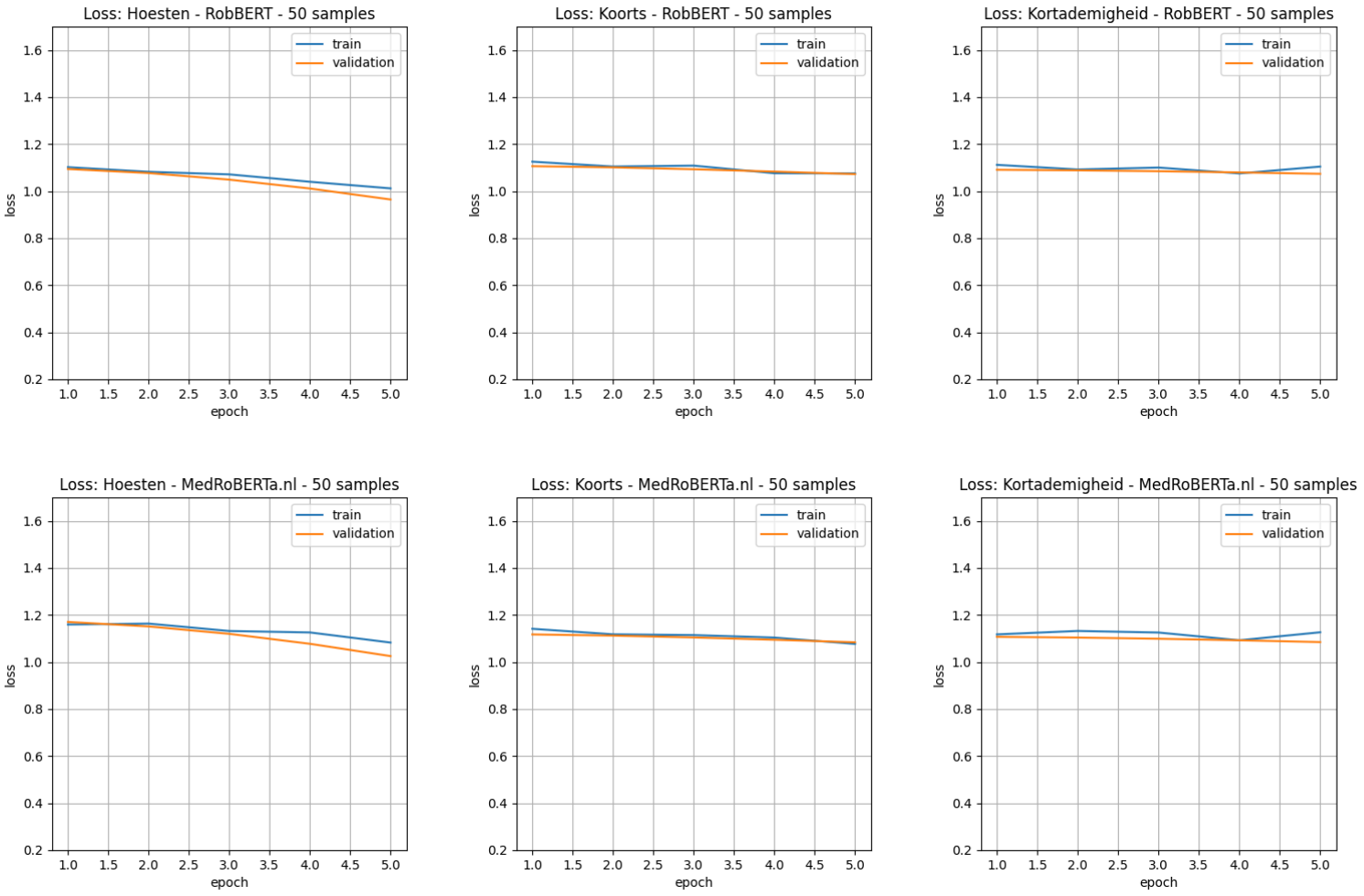

Figure 12. Training and validation loss of direct classifiers trained on 50 samples, using 5-fold cross-validation

100 samples

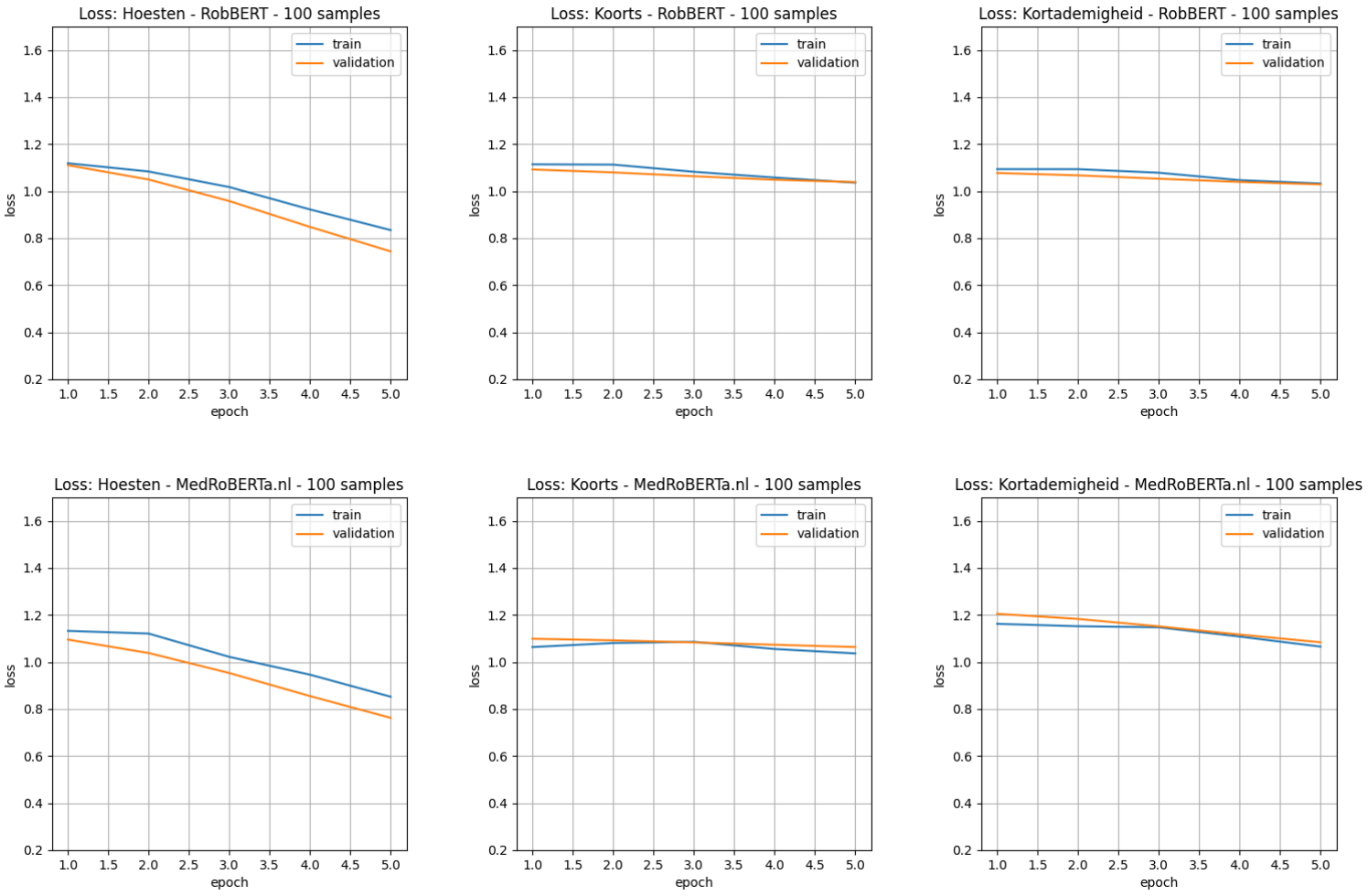

Figure 13. Training and validation loss of direct classifiers trained on 100 samples, using 5-fold cross-validation

200 samples

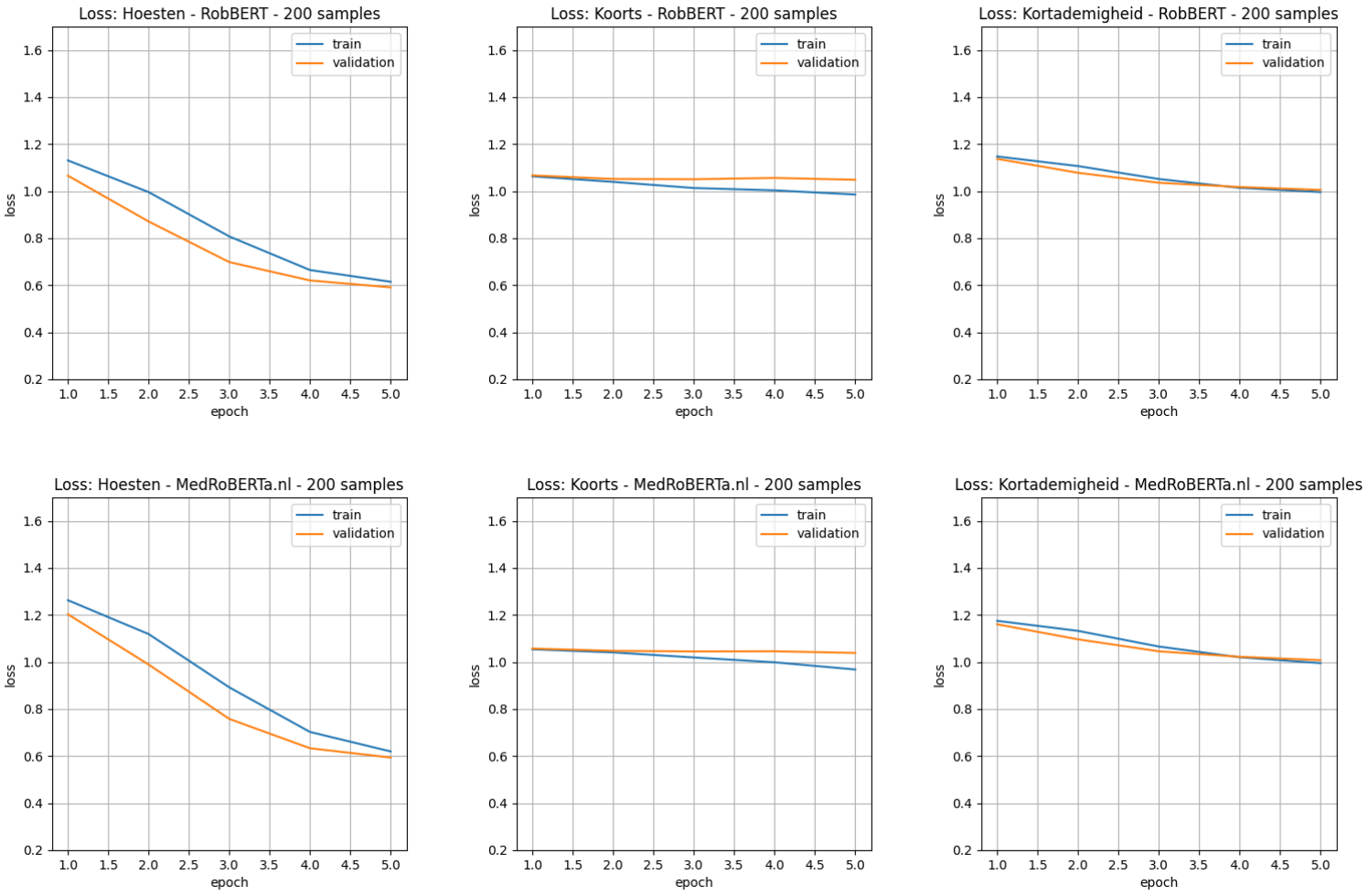

Figure 14. Training and validation loss of direct classifiers trained on 200 samples, using 5-fold cross-validation

400 samples

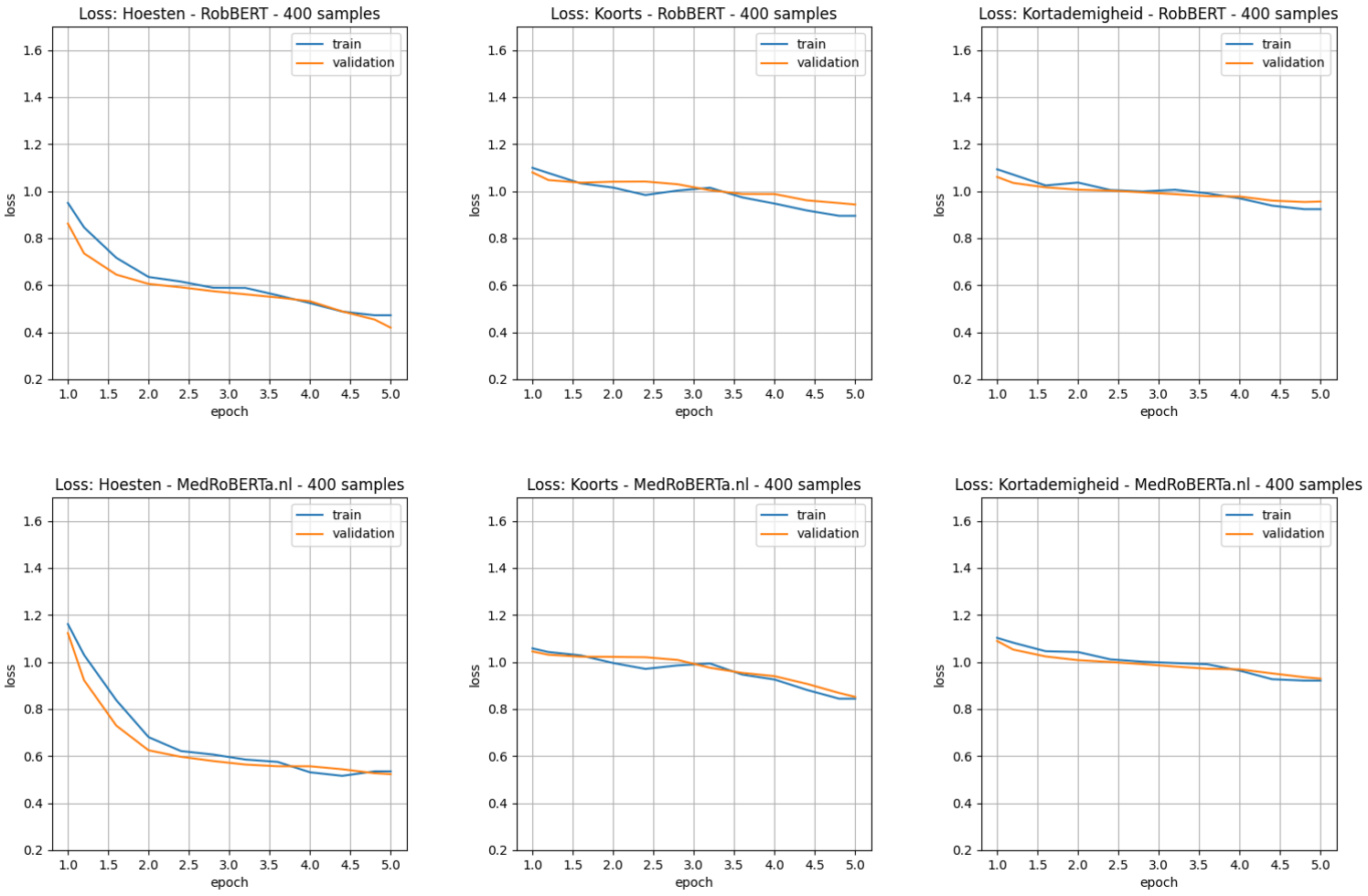

Figure 15. Training and validation loss of direct classifiers trained on 400 samples, using 5-fold cross-validation

800 samples

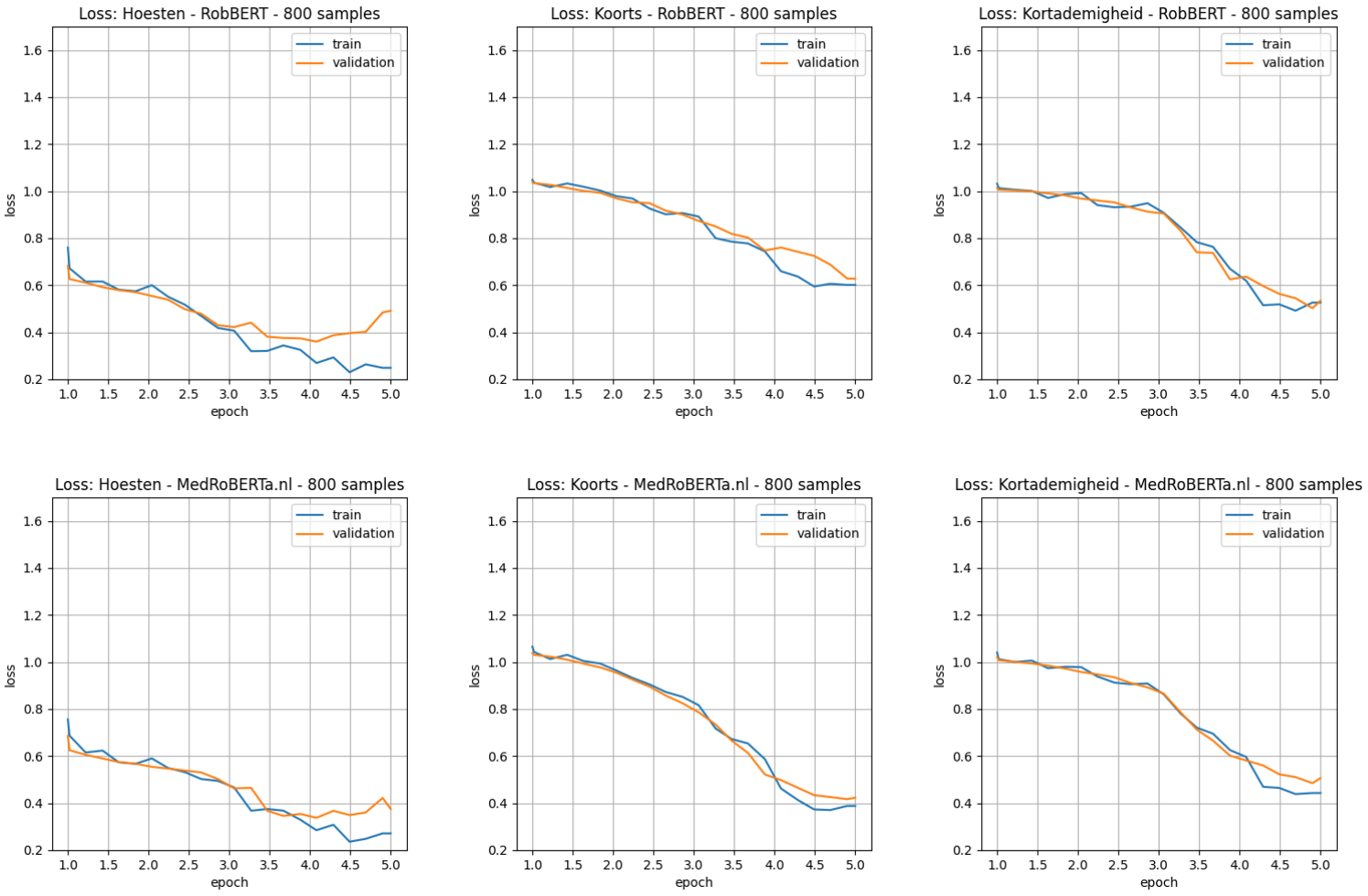

Figure 16. Training and validation loss of direct classifiers trained on 800 samples, using 5-fold cross-validation