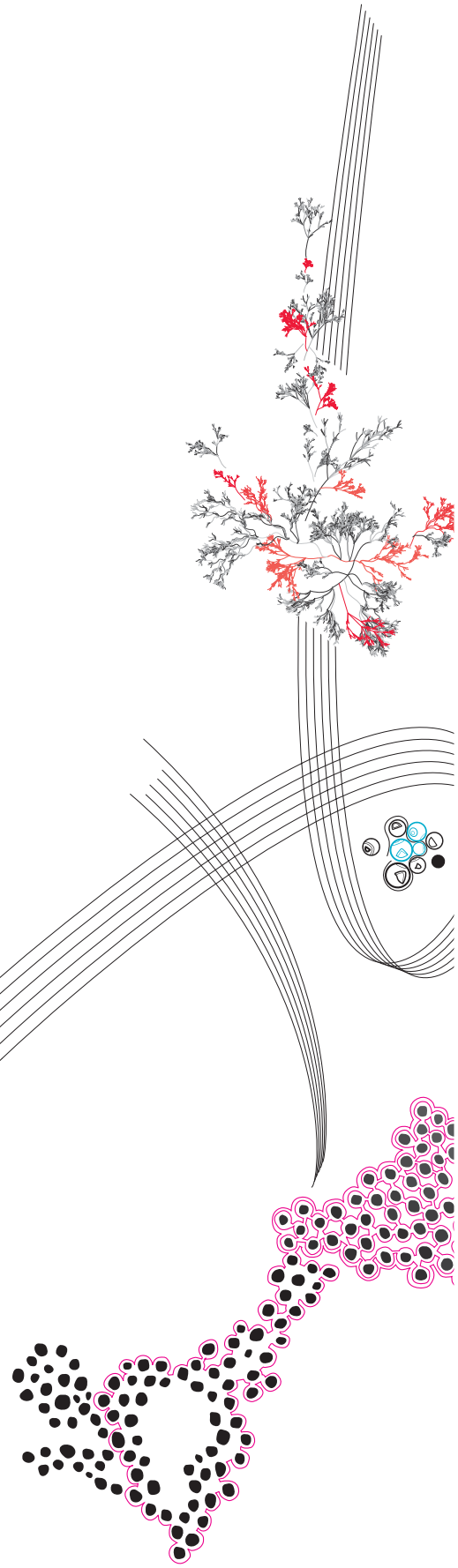


MSc Computer Science
Final Project

Comorbidity identification in clinical documents with medical terminology-based weak supervision.

Sylvain Brouwer

Supervisors:
prof. dr. ir. Maurice van Keulen
prof. dr. Johannes H. Hegeman

July, 2024

Department of Computer Science
Faculty of Electrical Engineering,
Mathematics and Computer Science,
University of Twente

UNIVERSITY OF TWENTE.

Abstract

Knowledge of patient comorbidities is crucial for effective healthcare decision-making and predictive modeling, particularly for vulnerable populations such as geriatric hip fracture patients. While electronic health records (EHRs) contain a wealth of data, data regarding comorbidities is often buried in unstructured text, posing a challenge for data extraction. The aim of this work was to evaluate the potential of machine learning (ML) and natural language processing (NLP) in extracting comorbidity data from EHRs, and thus work towards more complete representations of patient comorbidity in support of clinical care and research.

In this work we frame the task of identifying comorbidity in clinical documents as a multi-label classification problem. We aim to classify emergency department (ED) documents for elderly hip fracture patient into the categories of the Charlson Comorbidity Index (CCI), a well-established method of categorizing comorbidity. We first evaluated the performance of Naïve Bayes, Gradient Boosting, Random Forest, and a RoBERTa variant in a fully supervised setting based on ± 3200 documents. The performance of the fully supervised classifiers was hampered by the significant class imbalance for the CCI-categories. For all models except Naïve Bayes we observed f_1 scores above 0.8 for nearly all categories with an occurrence rate of 5% or higher, but performance dropped significantly with a decreasing occurrence rate under this 5% threshold. In the supervised learning scheme our best classification accuracy based on the full document label was 0.71, achieved with per-class binary Random Forests.

We attempted to mitigate the effects of the class imbalance by augmenting our training data with ± 20000 intake notes for patients outside the hip fracture cohort, using a weak supervision scheme. Weak labels were generated programmatically by checking for the presence of relevant terminology from SNOMED CT and the Unified Medical Language System (UMLS), two comprehensive medical terminology systems. The generated weak labels were supplemented with pseudo-labels generated by inference of the previously trained fully supervised Random Forest over the unlabeled documents. We find this approach to considerably improve classification performance for rare CCI categories, resulting in increases in the f_1 score of $0.05 - 0.35$ for categories with a prevalence under 5%. Random forest again was the most performant model, and document-level classification accuracy increased to 0.75 after inclusion of the weak- and pseudo-labeled documents.

Medical abstract

Objective: In this work, we apply machine learning (ML) to classify emergency department (ED) visit notes for elderly hip fracture patients into the groups of diagnoses of the Charlson Comorbidity Index (CCI). The CCI is a categorisation of groups of diagnoses, e.g. myocardial infarctions, cerebrovascular diseases, and renal diseases, and serves as a clinically validated predictor for 10-year patient mortality. The CCI has also found use as a powerful feature in other clinical predictive models, such as models that predict post-operative mortality. A machine learning model capable of identifying the CCI-categories from documents in the EHR would allow for a better overview of patient comorbidity in systems like EHRs, facilitating better clinical decision making. Furthermore, it would allow for easier inclusion of comorbidity as a factor in clinical research, resulting in more comprehensive analysis and improving the quality of the research.

Methods: We first evaluated the performance of four widely-used ML models (Naïve Bayes, Gradient Boosting, Random Forest and a transformer model), trained and evaluated based on ± 3200 manually labeled documents for hip fracture patients. We subsequently extended our dataset with ± 20000 ED documents for elderly patients with other types of fractures. The second set of documents was labeled by a computer program which checks for the presence of relevant terminology from SNOMED CT and the Unified Medical Language System (UMLS). These terminology-based labels were complemented by predictions of the Random Forest that was trained using the manually labeled data.

Results: When trained based on only the manually labeled documents, three models achieved f_1 performance scores above 0.8 for CCI-categories with a prevalence over 5% - however the limited amount of manually labeled data resulted in a gradual decrease towards 0 in the f_1 score for categories with a prevalence under 5%. The addition of the computer-labeled documents was effective in improving model performance for these rare CCI-categories. With the best-performing model, Random Forest, we observed increases in the f_1 score in the range $0.05 - 0.35$ for categories with a prevalence under 5%.

Conclusion: Of the tested models the Random Forest, after the introduction of the computer-labeled documents, performed best. In 75% of test cases, the Random Forest was able to predict all correct CCI-categories for a patient, and in 92% of test cases the predicted CCI-score was within 1 point of the correct score. While there is still room for improvement, particularly in classifying rare groups of diagnoses, our results offer a positive outlook for a more complete overview of comorbidity in EHRs, and the inclusion of comorbid condition as input for research and predictive models.

Keywords: weak supervision, machine learning, clinical NLP, medical terminologies, SNOMED CT, hip fractures, Charlson Comorbidity Index

Chapter 1

Introduction

Over the past two decades the use of electronic health records (EHRs) and health information systems (HIS) in healthcare has become widespread, to a point where the vast majority of hospitals and primary care physicians have now adopted an EHR of some form [47]. In the Netherlands all hospitals have chosen suppliers for a HIS/EHR and are currently using either an integrated EHR or a collection of subsystems [3].

Modern EHRs are a valuable tool in clinical research as they comprise a wealth of information [14], containing data such as test results, images, and patient vitals, as well as containing descriptions of the patient care process i.e., diagnoses, treatments and outcomes. However, most of this data is in unstructured form due to the nature of communication and reporting in medical practice [66]. About 80% of EHR data is unstructured [36]. Natural language processing (NLP) may offer solutions for processing this unstructured data for improving healthcare processes [66].

Ziekenhuisgroep Twente (ZGT) wishes to work towards models that can be used for improving and informing healthcare processes, patient comorbidities are important features for these models. Electronic health records contain a wealth of free-text clinical documents that contain information regarding comorbidities, however extracting this information manually is not reasonably feasible due to the significant time investment required. While automated labeling for individual studies based on ad-hoc rules is possible, it amounts to a significant amount of additional work for each study and results in inconsistent views on comorbidity across these studies. Furthermore, simple rule-based approaches are complicated by the intricacies of clinical reporting and text mining in general. For example, clinicians use a wide range of terms for the same concepts, and abbreviations are prevalent and may be ambiguous. These types of issues significantly limit the power of simple rule-based approaches. The goal of this work was to design a solution for automatically obtaining relevant comorbidities from clinical notes by leveraging natural language processing and machine learning methods, so that this solution may be used for efficient and consistent extraction of these conditions in further studies. More specifically we formulated the following research question:

RQ1 How can we design a machine learning approach or artifact for obtaining relevant comorbidities from clinical notes?

(a) Which comorbid conditions are relevant?

(b) How can the inherent structure of clinical notes be leveraged for improving model performance?

The prohibitive cost of manual annotation is an issue that also applies to this study. This study initially had just over 3200 manually annotated documents available for training and testing. This proved to be insufficient to create a model that was capable of extracting several groups of comorbid conditions in which we are interested at a sufficient level of accuracy. Taking into account the mentioned shortcomings for ad-hoc rule-based annotation approaches, and inspired by the availability of comprehensive medical terminology systems we pose the following questions:

RQ2 How can we leverage existing medical terminologies and ontologies in labeling sufficient training data?

(a) What are the shortcomings of training data labeled using medical terminologies compared to handlabeled data?

(b) How can we mitigate these shortcomings?

In addition to the main goal of working towards a ML-based solution, we have experimented with an approach to structuring research based on agile methodologies. While we will not fully cover this in this thesis, we do want to allude to it here as it impacted the research process. A full treatment of this topic can be found in appendix F. The main question we aimed to answer regarding the new approach is:

RQ3 How will adopting elements from Agile methodologies impact the research process in terms of effectiveness and efficiency?

The rest of this document is structured as follows:

- Chapter 2 provides an overview of the context of this research at ZGT.
- Chapter 3 provides background information on several topics and concepts relevant to this work.
- Chapter 4 discusses related work and literature.
- Chapter 5 covers the main aspects of our methodology regarding training ML models and discusses some challenges we identified for the problem at hand.
- Chapter 6 covers our first attempt at answering answering **RQ1**, based on the ±3200 manually annotated documents.
- Chapter 7 covers work in answering **RQ1** in conjunction with **RQ2**.
- Chapter 8 is a discussion and reflection on our obtained results.
- Finally, chapter 9 concludes this thesis.

Chapter 2

Context

2.1 ZGT

Care for older adults is a core expertise and spearhead for care, research and innovation at Ziekenhuisgroup Twente (ZGT). A specific area of research ZGT concerns itself with is that of the multidisciplinary care for elderly hip fracture patients. In order to improve quality of care for these patients, among others, the Centre for Geriatric Traumatology (CvGT) was established in 2008 at Ziekenhuisgroep Twente. At the CvGT patients aged 70 years and older are treated according to a multidisciplinary orthogeriatric care pathway. This pathway involves intensive co-management by a geriatrician, who is involved in performing a comprehensive geriatric assessment and evaluating patient treatment on a daily basis [79]. In support of the care delivered by the CvGT, ZGT puts effort into creating and evaluating metrics and models that may be used in improving prognosis and guiding the clinical pathway [46][78], determining risk of future fractures [71], as well as streamlining the care process, for example through automatic generation of radiology reports [48]. The data driving these models is often extracted from ZGT's EHR. In accordance with the number cited by Li et al. [36] about 80% of the EHR consists of unstructured data. This prevalence of unstructured data complicates the creation and training of models for prediction and classification.

2.2 Project Context

This specific research project originated from a desire at ZGT to build on the work of Yenodigan et al.[78] on predicting of post-operative mortality for elderly hip fracture patients. Comorbidities are significant contributing factors to mortality in both general and hip-fracture populations[41] and in the work of Yenidogan et al. they were therefore included as an input feature in their structured modality, however the completeness of the comorbidity feature was low. This low completeness is explained by the fact that comorbidities are not registered in the EHR in a structured manner unless the patient has previously received a diagnosis or treatment for the comorbid condition in a ZGT Hospital. Comorbid conditions can instead be found dispersed throughout communication between medical professionals and in clinical documentation, which are unstructured texts. Parallel to this desire to continue previous research, there is also the practical desire to complete the overview of patient in the EHR for clinical practice, either by completing the structured overview of conditions in the EHR or by providing some metric on patient comorbidity in a separate dashboard. Extracting the comorbidities from natural texts for use in a predictive model or augmentation of the EHR is a time consuming manual data extraction

task. This is not a desirable task in a healthcare setting, where healthcare professionals are already overloaded on reporting and paperwork, nor is it practical in a clinical research setting where datasets may be too large to manually process. This work therefore serves to facilitate a continuation of the work on predicting post-operative mortality as well as other future research projects by providing an efficient way of gathering a structured overview of comorbidities for the patient cohorts in those studies.

2.3 Research support and requirements

All experiments in this thesis were performed at the Information & Organization (I&O) department at ZGT. I&O provides access to internal computing resources and infrastructure dedicated to research purposes. An important requirement imposed on this project by ZGT and the I&O department is that it should be possible to host the resulting model on the internal infrastructure, ruling out the use of cloud-hosted solutions and very large language models like those in the GPT family. This requirement exists because moving data off-premise is undesirable as it raises significant privacy concerns. Medical expertise and domain knowledge was provided by prof. dr. J.H. Hegeman, a trauma surgeon at ZGT involved with the Centre for Geriatric Traumatology.

Chapter 3

Background

3.1 Comorbidity

There is discussion in the medical community regarding what constitutes comorbidity. In a broad sense there is some agreement that there is a conceptual split between the frameworks comorbidity and multi-morbidity, where comorbidity refers to a set of chronic conditions existing concurrently with a specific index condition and multi-morbidity to the existence of multiple chronic conditions in a general care setting. One can also take into account additional health-related and social factors to build the more general framework of patient complexity [65][44]. In both a care and research setting it matters which of the frameworks is chosen. Harrison et al. [26] note that while the comorbidity framework is useful in specialist care, adopting it a broader setting may lead to fragmented care where different parts of the healthcare system simply treat one index disease instead of the patient condition being treated in holistic manner. For research, they note that comorbidity and multi-morbidity frameworks require different approaches to sampling, as samples in comorbidity studies are skewed towards the index disease and conditions that cluster with the index disease and are thus not representative of patients with multi-morbidity in general.

As this work relates to a specific index condition — hip fractures — the comorbidity framework is most applicable in our case. However, even within that framework there may exist discussion as to what conditions should be considered a comorbidity for the given index condition. Research seems to mostly adhere to one of a number of previously defined measures for comorbidity, the most common of these being the Charlson Comorbidity Index (CCI) [10] followed by the Elixhauser Comorbidity Index [20]. In this work we will use a variant of the Charlson Comorbidity Index.

3.1.1 Charlson Comorbidity Index

The original CCI is a score sheet containing 19 conditions. Each condition is given a weight from 1 to 6 based on an estimate of 1-year mortality controlling for the contribution of coexistent conditions, these points are summed to obtain a score for a given patient. For each decade over 40 in the patients age an additional point is added to the score[10]. The resulting score is used to calculate a predicted 10-year survival rate using equation 3.1, the resulting values are plotted in figure 3.1.

$$survival = 0.987^{e^{0.9 \times CCI}} \tag{3.1}$$

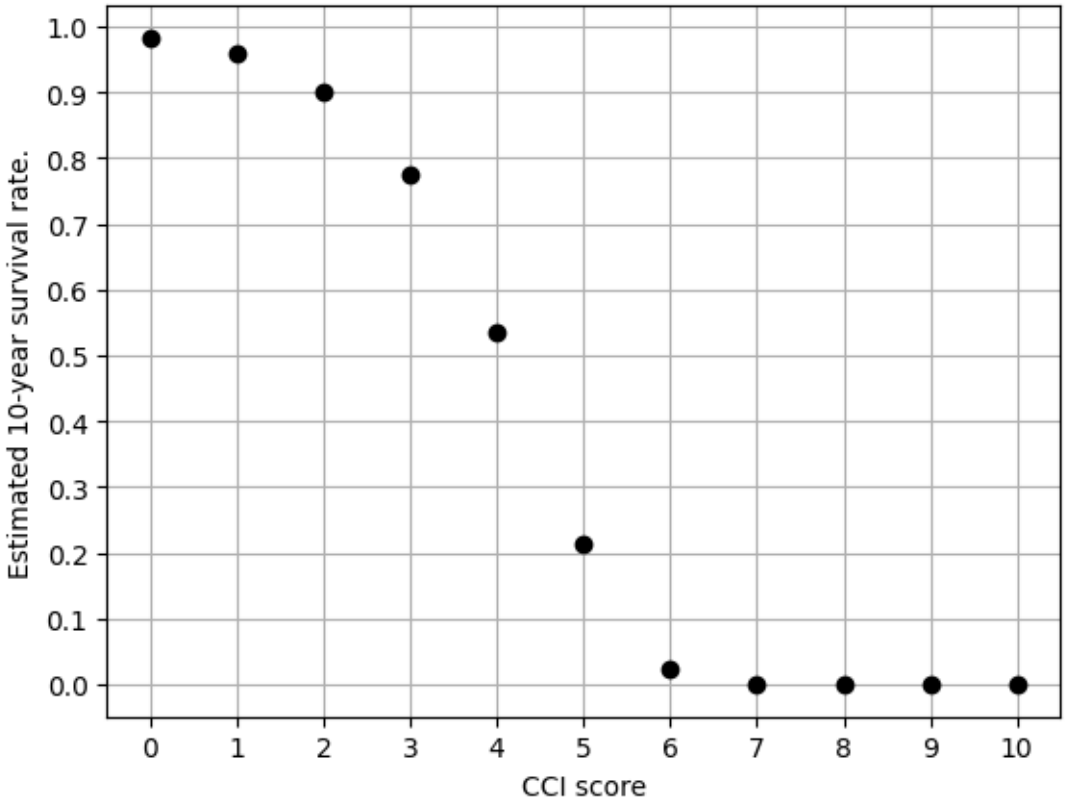


FIGURE 3.1: Estimated 10-year survival rate for CCI scores.

Deyo et al. [18] adapted the CCI to a set of 17 conditions which could be mapped to the clinical modification of the ninth revision of the International Classification of Diseases (ICD-9-CM). Quan et al. [53] introduced an adaptation for the tenth revision of the International Classification of Diseases (ICD-10) and enhanced the Deyo et al. ICD-9-CM adaptation through back-translation. Fortin et al. [22] adapted the Quan algorithm to use SNOMED-CT codes. These different coding algorithms may result in slightly different CCIs over the same dataset; Viernes et al. [70] reported a difference in mean CCI of 0.32 between the Quan et al. ICD-10 algorithm and a SNOMED-CT algorithm by Observational Health Data Sciences and Informatics (OHDSI) included in the R *FeatureExtraction* package.

Table 3.1 shows the categories in the Deyo adaptation of the CCI and the associated weights. It is important to note that there are three pairs of mutually exclusive categories in this index: *Diabetes, without chronic complications* and *Diabetes, with chronic complications*, *Mild liver disease* and *Moderate/severe liver disease*, and *Malignancy, except skin neoplasms* and *Metastatic solid tumor*. If the category with the higher weight applies to a patient, we no longer count the points from the less severe category.

3.2 Clinical Coding

Medical coding is the practice of assigning standardized codes to clinical concepts, these codes have found adoption in the healthcare systems for financial purposes such as billing, as well as as keywords for flagging and retrieving important diagnostic information in

TABLE 3.1: Conditions and associated scores in the Deyo-Charlson comorbidity index.

Weight	Condition
1	Peripheral vascular disease
	Dementia
	Myocardial infarction
	Chronic pulmonary disease
	Mild liver disease
	Congestive heart failure
	Peptic ulcer disease
	Cerebrovascular disease
	Diabetes, without chronic complications
	Rheumatic disease
2	Hemiplegia
	Renal disease
	Malignancy, except skin neoplasms
	Diabetes, with chronic complications
3	Moderate/severe liver disease
6	Metastatic solid tumor
	AIDS/HIV

EHRs. This section will now briefly cover three code systems that are commonly used in either financial reporting or clinical documentation and reporting.

3.2.1 SNOMED CT

SNOMED CT is a comprehensive clinical terminology system geared towards documentation and data analysis in support of clinical decision making and research. It models clinical concepts at multiple levels of granularity across 19 hierarchies, with each hierarchy covering a distinct category of clinical concepts such as clinical findings, body structures, procedures or events. The design of SNOMED CT was conceived with a set of desirable qualities for medical terminology systems, as laid out by James J. Simino[12], in mind. Two of these desiderata, a poly-hierarchical structure and the existence of multiple consistent views, are especially notable as they cause SNOMED CT to function significantly differently from other coding systems such as ICD.

Polyhierarchy
Within the 19 SNOMED CT hierarchies any concept may have more than one parent concept, parent child connections are modeled through hierarchical **IS A** relationships. When limited to concepts and hierarchical relationships the SNOMED CT data model forms a directed acyclic graph, figure 3.2 shows an example of this structure. A concrete example of polyhierarchy in SNOMED CT is the concept "Myocardial Infarction" (id: 22298006), which has two parent concepts: "Ischemic heart disease (414545008)" and "Myocardial necrosis" (id: 251061000).[1]

Multiple consistent views
Multiple consistent views refers to the requirement that concepts are accessible through

[1]For exploring these concepts and SNOMED CT in general see the official browsing tool.

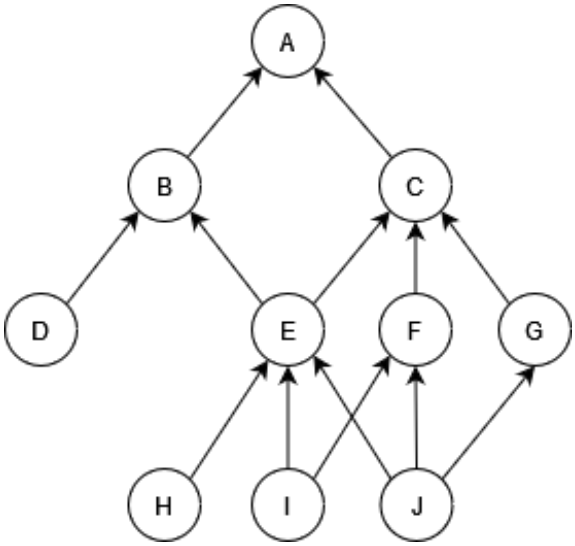


FIGURE 3.2: Example of a polyhierarchy

different paths in the data structure, but the choice of path should not influence the representation, meaning or interpretation of the concept. This applies to traversing the acyclic polyhierarchy but also in a broader sense as, in addition to **IS A** relationships, SNOMED CT contains **attribute** relationships which link nodes across hierarchies in a non-hierarchical manner and are used to represent certain aspects of the meaning of concepts. Examples of **attribute** relationships include the **finding site** relationship linking clinical findings and body structures, and the **procedure site direct** relationship linking procedures to body structures.

We return to the example of the concept "Myocardial Infarction" (id: 22298006), and consider multiple queries that would return this concept. For example: a query for all children of one of the parents of "Myocardial Infarction", another query asking for all ancestors of "Acute Myocardial Infarction (id: 57054005)" which is a child of "Myocardial Infarction" and a query asking for all clinical findings with "Myocardium structure (id: 74281007)" as a finding site. The notion of multiple consistent views means that the representation and interpretation of the concept is stand-alone and does not depend of the chosen query.

Processing and practical use
SNOMED CT is designed to be computer processable and query-able, allowing it to be incorporated into information systems to support analytical tasks, support documentation and allow for querying data artifacts. Native tools support querying with the Expression Constraint Language (ECL), however the release format is compatible with other technologies such as relational or graph databases. Defining subsets from SNOMED CT through queries with tools such as the ECL is called *intensional* subset definition, while subsets presented as lists of individual SNOMED IDs are said to be *extensionally* defined.

3.2.2 ICD

ICD is a family of medical coding systems designed to be uses as a statistical classification managed by the United Nations System. The primary use case for ICD is in reporting and statistical analysis, but it has also found widespread use in clinical coding. ICD is designed to be mono-hierarchical, thus any concept has a single parent an can only be found at one

location in the coding system. This is in order to support statistical reporting as a concept being a member of multiple statistical categories makes reporting more complex and less transparent, however it requires strict design decisions to be made, such as whether to place "Tuberculosis of lung" under "Infectious diseases" or "Diseases of the respiratory system". This can make locating concepts that could fit in multiple categories difficult.

3.2.3 DBC

The standard for clinical coding for financial systems and reimbursement in the Netherlands is the diagnose-behandelcombinatie (DBC). DBC encodes combinations of diagnoses, treatments, medications and consults into 9-digit codes. DBC codes are assigned prices according to a country-wide standard, charged to patients that receive the associated set of treatments.

3.3 Weak Supervision

With the advent of large deep learning models and large language models the availability of labeled training data of sufficient quality has increasingly become a bottleneck in practical machine learning applications. Weak supervision is a machine learning paradigm that attempts to address this issue, it encompasses a range of methods that generate labels for unlabeled data in a cheap manner; at little time cost to individual experts and researchers. Labels generated by weak supervision methods are generally noisier and of lower quality than those created by experts, more specifically they may be incomplete, inexact or inaccurate [80].

Common methods that fall under the weak supervision paradigm include: *Active learning*, an approach in which a machine learning model can query a human user for input in labeling a set of samples. Typically the samples presented to the user are the set of most informative or difficult samples as identified by some heuristic. *Multiple instance learning*, a learning approach in which data samples are arranged in groups, called *bags*, and labels are assigned at a bag level. This allows for the creation of inexact labels at significantly reduced labeling cost. *Programmatic labeling* is an approach in which new samples are labeled based on some prior defined rule or heuristic, such as the presence of a certain word, or variables passing some combination of threshold values. Ratner et al.[56] introduce a programmatic approach called *Data Programming*, which allows an expert or user to define heuristics called *labeling functions*, and trains a generative model that accounts for interdependencies and conflicts among these labeling functions. The data programming paradigm can be applied using the Snorkel[57] tool.

Chapter 4

Related Work

4.1 Medical Coding

Numerous machine learning methods have been applied to the ICD coding problem, ranging from more traditional methods to state-of-the-art transformer-based architectures. Traditional methods have shown decent performance on shorter text descriptions, for example, Atutxa et al. [2] attempt to classify short diagnostic summaries across 1676 Spanish ICD codes. The diagnostic summaries were normalized in a number of different ways, based on SNOMED and web resources, and similarity features to a standard vocabulary are calculated for each normalization. Random forests (RFs) for each individual feature vector, RFs for concatenations of feature vectors, and weighted voting based on the individual RFs are then compared in their classification performance. Best performance was obtained by classifying based on the concatenation of all individual feature vectors, achieving a precision and recall of 0.92.

Xu et al. [77] take a multi-modal ensemble approach to multi-label classification for 32 common ICD codes, incorporating a structured modality, a short diagnostic text modality like those used by Atutxa et al. [2] and a unstructured text modality of full clinical notes. For their short diagnostic modality Xu et al. [77] use a character-level CNN for word representation combined with a bi-directional LSTM for encoding the sequence. The unstructured modality was processed using a CNN, of which the output features were subsequently enriched with TF-IFD features for relevant key words before being passed to a fully connected network. A decision tree was applied to the structured modality. Best performance was achieved by combining all three modalities with label smoothing regularization, which was introduced to adjust for the class imbalance in ICD codes and thus improve performance on rarer ICD codes.

Beyond regularization techniques like the one used by Xu et al. [77], code descriptions are often leveraged to improve performance for rare ICD codes. For example, Chapman et al. [9] introduce an approach for handling class imbalance that computes word-level attention between clinical documents and code descriptions using BERT-like encoders, resulting in an improvement in classifying uncommon ICD codes compared to a baseline of BERT and a fully connected classification layer.

Mullenbach et al. [42] perform multi-label classification on the ICD-9 codeset using a CNN combined with a per-class attention mechanism. They leverage code descriptions by adding a regularization term dependent on similarity of text embeddings to description

embeddings, the term scales with the frequency of true code occurrence such that similarity to descriptions is more important for rare codes.

Gao et al. [25] introduce KeyClass, a weakly-supervised approach for text classification based on solely on the class descriptions. It is trained on a number of common classification tasks, including assigning ICD-9 codes. Gao et al. [25] first create class vocabularies based on code descriptions, then key phrases in the corpus are identified and mapped to the most semantically similar class. Labeling functions are then created for each key phrase. These labeling functions vote for the associated class if the keyword is in the document, else they abstain. The vote distributions of these labeling functions are then used to probabilistically label documents. The downstream classifier is trained based on these probabilistic lables. KeyClass shows decent performance compared to other weakly-supervised and fully supervised models, however there is room for improvement with regards to precision and performance on low-frequency codes in the task of classifying clinical notes.

4.2 Classification tasks

Machine learning with EHRs offers opportunities for a wealth of classification tasks besides coding problems. A common example is predictive tasks, these are generally binary classification problems where the goal is to predict whether some medical outcome will occur. Target outcomes for prediction include mortality, hospital stay length and hospital readmission or specific medical problems like heart failure and cancer recurrence [36].

Poulain et al. [52] introduce CEHR-GAN-BERT, a model that leverages BERT for creating EHR representations, and train a downstream predictive model in a generative-adverserial setting in order to allow out-of cohort patients to be included when learning the representations. While they use BERT and Masked Language Modeling for extracting temporal relations in their data, a common approach in NLP, their work does not include a truly unstructured text modality, instead relying on sequences of tokens representing medical conditions and procedures for their input embedding, these have been obtained from pre-coded medical records [50]. CEHR-GAN-BERT seems to outperform other state-of-the-art models on a number of predictive tasks especially in problems with small patient cohorts.

Lovelace et al. [38] approach predictive problems through the avenue of medical coding, training a CNN augmented by an attention mechanism to extract intermediate "problem lists" from clinical notes. These problem lists consist of rolled up ICD-9 codes and phecodes[1] and serve a as human-understandable features for a downstream classifier.

An example of a non-predictive classification task on EHRs is the work by Si et al. [59], who introduce an adaptation of BioBERT that incorporates label embeddings for automatic triage using messages extracted from the EHR. Similarly to Chapman et al. [9], cross attention between class descriptions and input is considered as a way to give tokens relevant to a class a larger attention score.

[1] Phecodes: "Manually curated groups of ICD codes intended to capture clinically meaningful concepts for research." [4]

Chapter 5

Methodology

This chapter introduces the methodology applied during this research. First, section 5.1 touches upon the Agile approach we applied during this work, and how it affected the structure of this work. We then cover common methodological aspects: our framing of the task at hand (5.2), dataset and data annotation (5.3), and validation approach (5.4). Finally section 5.5 provides an overview of some problems we encountered during the course of this project, mostly related to working with real-world data.

5.1 Agile

The overarching goal of this project is design-focused: we wish to deliver a machine learning artifact or approach that allows for the extraction of comorbidities in clinical practice and clinical research projects. Given this goal of the delivery of an artifact, and the fixed time and resource constraints associated with this work, a traditional top-down research approach is likely not suitable for this project. Such a top-down approach would require the definition of rigid set of requirements and experiments and design of an "optimal solution" beforehand, something we do not believe to be possible. Rather, we believe that an iterative design approach which allows us to add complexity to our solution incrementally and steer the design process based on interim results is more suitable for this project. We therefore believe that Agile, a set of practices from software development is a good fit for this project.

Agile is a set of project management practices and principles created after the introduction of the Agile Manifesto[5] at the start of the century, largely as a reaction to the shortcomings of the dominant top-down Waterfall development approach of that time. The goal in Agile is to work in short development sequences in order to facilitate frequent delivery to customers, flexibility and management of changing and emerging requirements.

We have treated this research project as a case study on using Agile methodologies in research. We have re-framed elements from Agile and the related Scrum framework[62] from a software development context to a research context and analyzed their usefulness during this project. This thesis will not discuss this matter in detail, but it is important to note that we worked in sprints: time-boxed iterations of three weeks with a pre-defined set of tasks, each sprint ending with retrospective meeting, as depicted in figure 5.1. This approach resulted in this work breaking up into two clearly seperable phases, the first phase being a general exploration of the problem at hand, methods, and machine learning models and the second phase addressing a specific limitation of the machine learning

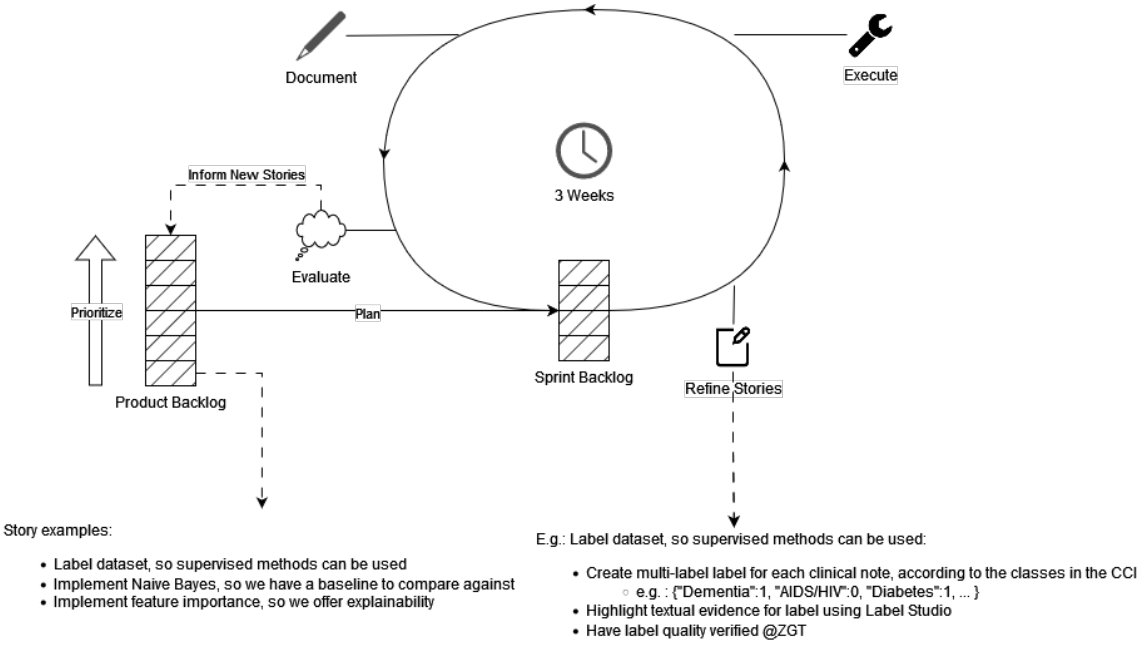


FIGURE 5.1: Depiction of the sprint process.

models observed during the fist phase. The rest of this document will present these phases seperately. For a full treatment of Agile and our experiences in using it during this project, we refer to appendix F.

5.2 Target Variable

The problem of obtaining comorbidities from clinical notes can be framed in two ways: as an entity extraction problem in which the goal is identify individual comorbidities in-text, or as a multi-label classification problem in which we pre-define classes of relevant comorbidities and assign document-level labels. We have chosen to adhere to the classification perspective on the problem as it is more compatible with accepted comorbidity measures like the CCI, firstly because multi-labels can map one-to-one to a score such as the CCI and secondly because several categories may be indicated by evidence spread throughout a note, rather than being stated as a single entity. Some concrete examples include:

- The CCI category "congestive heart failure" may be indicated by a combination of factors which, taken individually, are not problematic enough to be considered a comorbidity, like heart palpitations, an enlarged heart or valve insufficiency.

- Diabetes and relevant complications may be stated separately in a note. A classification between complicated and uncomplicated diabetes should thus be done at a document level.

- Certain medical procedures and medications may be strong indicators of comorbidity, but are not considered in a NER setting as they are not clinical findings. For example, stents and dotter procedures may indicate peripheral vascular disease, and insulin use may indicate diabetes.

We take the 17 categories of the Deyo adaptation[18] of the CCI as our target classes. Note that in the rest of this document the terms "category" and "class" will be used interchangeably when referring to these. The task at hand thus is a multi-label classification problem, with one caveat: the mutually exclusive classes as mentioned in section 3.1.1. We will handle these mutually exclusive classes by first approaching the task a pure multi-label problem and resolving conflicts post-prediction, i.e. if both classes have been predicted we will remove the predicted label for the less severe class.

hoofdklacht: pijnlijke r heup

anamnese:
patiënt is vanochtend in de trap gevallen op rechterzij, sindsdien niet meer op kunnen lopen. met hoofd tegen treden gekomen, geen bewustzijnsverlies. geen nekklachten, geen andere klachten.

medische geschiedenis:
dm type 2, hypertensie
[DATE] kniefractuur
[DATE] cataractoperatie extracapsulair phaco links
[DATE] cataractoperatie extracapsulair phaco rechts
[DATE] dementie

allergieën: geen

medicatie:
insuline
zolpidem [x]mg/dag
omeprazol [x]mg/dag
propanolol [x]mg/dag

lichamelijk onderzoek: r been verkort, in exorotatie.

lab:
hemoglobine [x]mmol/l; hematocriet [x]l/l; c-reactive protein [x]mg/l; leukocyten [x]/l; kreatinine [x]umol/l; bloedgroep -volgt-; antistof scr -volgt;
x-heup: collumfractuur rechts

diagnose: collumfractuur rechts

beleid:
-opname
-pijnstilling
-nuchter
-morgenochtend ok

FIGURE 5.2: Artificial example of a clinical note.
Note: for the sake of brevity this example is incomplete, it does not contain all elements of real ED notes.

5.3 Dataset

The training data used in this study consists of intake notes for patients with a fracture due to trauma from the emergency department (ED) at the ZGT locations in Hengelo and Almelo, collected from January of 2008 onward. Only documents concerning patients aged 70 and up and presenting with possible fractures have been included. Figure 5.2 shows an artificial example of what an ED intake note may look like. The documents are split into two datasets: one containing all notes for patients with hip fractures (**DATA-HIP**) and the other containing documents for all other fractures (**DATA-REST**). The documents in **DATA-HIP** hip have been annotated and serve as a golden standard to be used for learning and evaluation. **DATA-REST** has not been annotated, as these documents were included at a later stage in this study and are used for weak supervision. Weak supervision and the exact use of **DATA-REST** are discussed in section 7.2.1. We have limited our data to documents for fracture patients mainly because it fits within the project context as laid out in chapter 2, but it was also considered that fracture documents introduce little bias towards any individual CCI category, as opposed to the inclusion of e.g. cardiac patients.

5.3.1 Document pre-processing

A number of pre-processing steps were applied to all documents before they were used in further analysis and training. Most importantly an automatic anonymization process was applied to the documents in order to remove all mentions of personally identifiable information like names and addresses. Other pre-processing steps concerned maintaining consistent formatting across all documents, in order to accomplish this a number of special characters were removed or converted and document line termination was standardized to CRLF. Finally all texts were converted to lowercase.

5.3.2 Annotation Process

The task of hand-annotating the over 3000 documents in the **DATA-HIP** dataset with the 17 CCI categories requires a significant time investment. The complexity of the task, however, is rather limited given the short, structured and to the point nature of emergency department notes. Given these factors, it was decided to have the author label the documents based on a protocol agreed upon by a medical expert rather than burden clinical practitioners with a low complexity yet time consuming task.

The core of the labeling protocol is a terminology list containing conditions that should be included under each target class. In section 3.1 we have mentioned a number of works linking the CCI to medical ontologies, these may serve in constructing a terminology list. We have chosen to use the work of Fortin et al.[22] as a base reference during the annotation process, as it is based on the adaptation of the CCI used for the target variables, and provides a more fine grained list of conditions than the works of Deyo et al.[18] and Quan et al.[53].[1] The list of SNOMED CT codes provided by Fortin et al. was converted to a list of Dutch terms based on the SNOMED CT instance installed in ZGT systems and the Unified Medical Language System[64]. While the resulting list was quite comprehensive, it was decided to make a number of additional inclusions. These inclusions are conditions, encountered during the labeling process, that we consider to fall under or be

[1] Note that the original article included an incorrect code list, we used the list from the associated correction[23].

indicative of a CCI category, but were not included in the stricter definition of Fortin et al. Also included were certain medical procedures which, while not strictly comorbidities, are sufficiently indicative of a CCI class to warrant labeling the document with said class. The full overview of additional inclusions can be found in appendix B.

It stands to reason that negated terms, e.g. *Patient does not have X.*, should not be considered in labeling, as the condition X does not actually contribute to patient comorbidity in this case. Beyond negation, we take a rather strict position on when a condition should be considered, namely that documents should only be labeled if the mention of a relevant condition can conceivably be interpreted as a clinical diagnosis. This means that we not only exclude negations, but also conditions that are stated with some level of uncertainty, warnings of possible conditions and differential diagnoses. Table 5.1 provides an overview of words that would lead to a mentioned condition not being labeled.

TABLE 5.1: List of terms treated as negation.

Term	Interpretation
geen	no
niet	not
mogelijk	possible
cave	caveat (warning)
dd / dd. / d.d.	differential diagnosis
verdacht / verdenking	suspicious / suspicion
zonder	without
suspect	suspect

5.3.3 Comparing the Datasets

TABLE 5.2: Statistics on dataset size and note length.

Total number of notes (N)	24187
DATA-HIP	
Number of notes (N_h)	3290
Mean length (μ_h)	220
Standard deviation (σ_h)	112
DATA-REST	
Number of notes (N_r)	20897
Mean length (μ_r)	219
Standard deviation (σ_r)	144

Given that **DATA-HIP** and **DATA-REST** contain documents for different patient populations, it would be good to assert whether the data is similar across the two datasets. More specifically we would like to know whether medical histories as described in the clinical notes are similar between the two datasets, as this is the section likely to contain most information on comorbidities. We believe the rest of the document is more likely to vary between different types of fractures. If the contents of the medical history section differs greatly between **DATA-HIP** and **DATA-REST**, then the inclusion of **DATA-REST** in

training will mostly introduce noise. Table 5.2 shows basic statistics on note size and the number of notes in each set, revealing consistent note length across the datasets. Non hip-fracture documents show more variability in length, likely due to the remaining contents of the documents varying dependent on the fracture type, as we previously theorized.

Cosine similarity

A simple manner of evaluating the similarity between the two sets of documents is to isolate the medical histories form the documents, and encode the two datasets as vectors containing the word counts of all words in the isolated medical histories. We can then determine the cosine similarity between the resulting vectors. This results in a cosine similarity score of 0.98, indicating a high degree of similarity between the medical histories in **DATA-HIP** and **DATA-REST**. **DATA-REST** should thus be suitable for training a classifier.

5.4 Validation

Multiple models for classification may be developed during this project, in order to validate these models and compare their performances, a set of metrics is required. Once a comprehensive set of metrics has been defined we will evaluate model performance through a k-fold cross validations over the hand labeled data.

5.4.1 Metrics

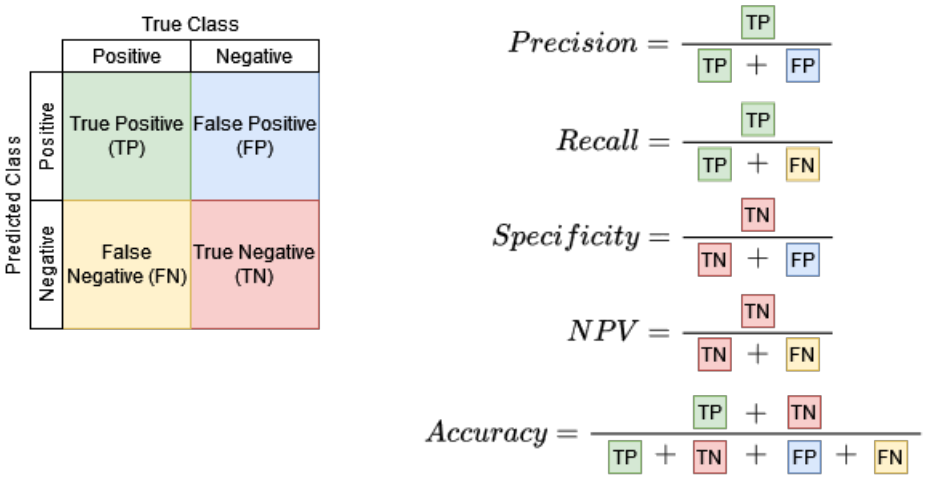


FIGURE 5.3: Confusion Matrix and associated metrics.

We can evaluate our model based on two views on model performance: evaluation at class level, and aggregated document-level evaluation. Traditionally, machine learning classification models are evaluated based on metrics derived from a confusion matrix, figure 5.3 depicts a binary confusion matrix along with several common derived metrics. We are dealing with a multi-label problem, which can be treated as one binary classification problem for each class. This view allows us to evaluate performance for each individual category in the CCI, which makes it possible to identify patterns across the categories and identify class-specific problems. We will use the f_1 score as our main metric in this

per-class view, the f_1 score is the harmonic mean of the precision and recall scores depicted in figure 5.3 and is calculated as given in equation 5.1.

$$f_1 = 2\frac{Precision * Recall}{Precision + Recall} \tag{5.1}$$

The second view on performance is to not consider individual classes but evaluate the models at a document level. The most important metric in this view is the classification accuracy, i.e. the fraction of documents for which all categories are predicted correctly. This approach is valuable as it provides a higher level overview of model performance that ties more closely to clinical practice compared to individual class performances. We can then also evaluate the ability to predict the correct CCI score for a given document, disregarding the exact labels. Comparing this precision in the CCI to the classification accuracy should give some insight into the likelihood of "flipped labels" resulting in a correct CCI score despite a wrong classification. For additional insight we will also determine the 1-off precision, that is the percentage of notes within 1 point of the correct CCI score, and provide a mean absolute error for the CCI, as calculated as in 5.2.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|CCI_i^{true} - CCI_i^{predicted}| \tag{5.2}$$

5.4.2 Cross validation

Model performance will be evaluated in a k-fold cross validation (CV), over **DATA-HIP**. K-fold CV randomly divides the dataset into k subsets and subsequently trains k models, leaving out one subset for evaluation each time. We thus obtain k sets of results based on the metrics introduced in section 5.4.1, allowing us to present either the whole spread of results, or aggregated results with a standard deviation. This approach will give a better insight into model variance than evaluation based on a single test-train split. Figure 5.4 shows a visualization of the process.

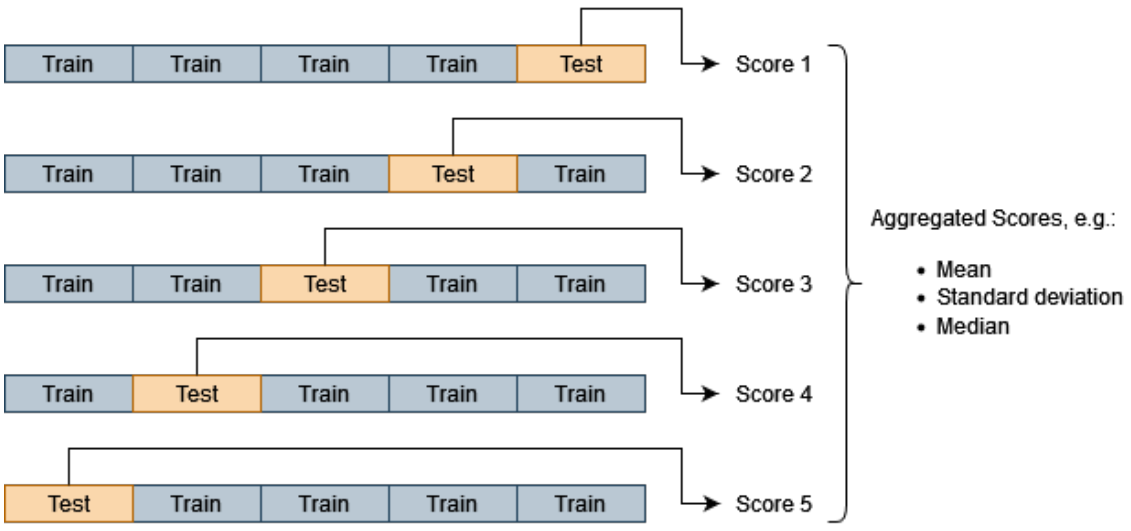


FIGURE 5.4: Example of a 5-fold cross validation.

In this work we have chosen to set $k = 10$ for model evaluation in all experiments, this value was chosen as it is large enough to get a decent interpretation of the spread of results and model variance but small enough for the test set to contain a decent amount of positive samples for most classes. Time was also a factor in our choice as each fold requires re-training, thus doubling the amount of folds approximately doubles the experiment time cost. In the experiments in chapter 7, where **DATA-REST** was included, **DATA-REST** was simply merged with the training folds in each iteration of the 10-fold validation over **DATA-HIP**.

5.5 Problem Identification

A persistent matter in the data science and artificial intelligence research community is the prevalence of work based on improving benchmarks, idealized problems and clean, accurate data. While this type of research is invaluable in driving innovation and prototyping models and approaches, it fails to take into account the imperfections of real world problems and the limitations of AI operating within the boundaries of complex human-centric domain. This often leads to AI-methods failing or performing sub-par when moved from the idealized situation to a real-world problem. Van der Schaar and Rashbass call these idealized methods petri-dish AI, as opposed to reality-centric AI[58]. Van der Schaar and Rashbass highlight five essential "pillars" for developing AI solutions that function in the real world, many of these concern model lifetime, deployment and operations, factors we are not concerned with at this stage. However they also highlight the necessity of model being able to operate on real-world data. On that topic, we would like to elaborate on several challenges relating to data quality in our dataset. Many of these challenges are common in applied clinical NLP and classification, and need to be considered during the rest of this work.

5.5.1 Class Imbalance

Imbalanced datasets are common in machine learning and NLP tasks, especially within the medical domain as most people are healthy and incidence rates can vary greatly between conditions. Imbalance can be problematic for machine learning tasks, as many machine learning methods will favour predicting majority classes due to being optimized for average performance metrics over a dataset.

The dataset in this study and the categories of the CCI exhibit such a class imbalance; table 5.3 shows the percentage of documents from **DATA-HIP** that have been labeled with each category from the CCI. It can be observed that the occurrence rates between the most common and rarest occurring classes differ by nearly two orders of magnitude, and the category *AIDS / HIV* is not present in the dataset at all. We will later see that the low occurrence rate for the rarer classes has a significant impact on classification performance.

5.5.2 Domain-specific Language

Domain-specific use of language is a common problem in NLP, different industries often have their own jargon, shorthand and distributions of words and language structures. Significant differences in language can affect the performance and generalizablity of NLP methods across application domains, thus different domains may require different processing and re-training of models. In recent years, the requirement for adaptation to a domain is clearly indicated by the rise of domain-specific language models, such as BioBERT[34] for biomedical text mining or BloombergGPT[75] for a variety of tasks on finance literature.

TABLE 5.3: Occurrence rates of CCI categories in **DATA-HIP**

Category	Occurrence rate
Cerebrovascular disease	0.188
Dementia	0.170
Congestive heart failure	0.153
Diabetes, without chronic complications	0.147
Malignancy, except skin neoplasms	0.146
Chronic pulmonary disease	0.136
Peripheral vascular disease	0.121
Renal disease	0.089
Rheumatic disease	0.086
Myocardial infarction	0.078
Diabetes, with chronic complications	0.047
Hemiplegia / paraplegia	0.024
Metastatic solid tumor	0.020
Peptic ulcer disease	0.020
Mild liver disease	0.009
Moderate / severe liver disease	0.003
AIDS / HIV	0.000

The healthcare industry is a prime example of an industry in which the use of jargon is particularly pervasive. The nature of the industry leads to the use of diagnostic and scientific terms which rarely occur in general language, each of which may have a number of associated abbreviations. Additionally, differences in language can be observed within the healthcare domain as reporting practices, certain terms and the interpretation abbreviation may differ between medical disciplines.

In this study the most problematic observed domain-specific language issues relate to the use of ambiguous abbreviations and limited context due to emergency department reporting practices.

Abbreviations and ambiguity

The use of abbreviations without expanded definition is common in clinical documentation[76]. This poses a problem for clinical NLP tasks as the interpretation of abbreviations can be context-dependent or ambiguous, and in some cases abbreviations can not be distinguished based on context at all[6]. Table 5.4 shows some examples of abbreviations with two interpretations, of which only one belongs to a CCI category. All of these examples were found in the **DATA-HIP** dataset.

TABLE 5.4: Abbreviations with CCI and non-CCI interpretations.

Abbreviation	CCI interpretation	Non-CCI interpretation
mi	myocardial infarction	mitral valve insufficiency
pta	dotterprocedure	staging for bladder cancer
hf	heart failure	heart frequency
all	acute lymphoblastic leukemia	allergies
ra	rheumatoid arthritis	right atrium

Limited context in ED notes

Documentation in the emergency department is generally short and to the point. As seen in table 5.2, the mean length of the used clinical notes was 220 words, which is only about half a page worth of text. This concise nature of the documentation is reflected in the way relevant medical conditions and procedures that could be considered as comorbidities are mentioned in the clinical notes. The majority of these features are captured under a medical history section of the note where they are presented in a list-wise or comma-separated manner. Complex conditions and more involved procedures may be given as a single sentence, but typically only a diagnostic term or procedure name is given. This means that most mentions of comorbid conditions are devoid of context, indicated by a single word or phrase. While this is not necessarily problematic in cases where clear diagnostic terms are used, it may complicate the disambiguation of ambiguous terms and abbreviations. It may also affect the performance of certain model types, as we may expect generative and context-sensitive models to offer little advantage in identifying context-free mentions of comorbidities.

Misspellings

Misspellings and typing errors are a pervasive problem in NLP tasks regardless of application domain, as an important feature being spelled incorrectly can lead to misclassification. NLP offers a number of approaches for dealing with the problem of misspelling, most based on fuzzy (inexact) matching to account for small differences in spelling. We will evaluate whether this is necessary in this study.

Chapter 6

Phase 1: Fully Supervised Learning

During the fist phase of this study we had two primary goals, both explorative in nature. Our first goal was to identify machine learning architectures that could serve as a platform for a final solution. Our second goal was to analyse the clinical documents with respect to their inherent structure in order to answer **RQ1 b)**.

6.1 Model Selection

With the goal of identifying a suitable base model in mind, we will compare the performances of four common machine learning models when applied to the problem of classifying clinical notes in a fully supervised learning scheme. The chosen models for this experiment are: Multinomial Naive Bayes, Random Forest, Gradient Boosted Trees and Transformer encoder architectures in the BERT family. All of these models are commonly applied to text classification tasks and are in theory suitable for the task at hand.

6.1.1 Tokenization and text representation

In natural language processing, tokenization is the process of breaking down text into smaller pieces. These pieces can be individual words, sequences of n words (*n-grams*), or even parts of words or syllables. The resulting pieces, referred to as tokens, subsequently serve in creating the input features for machine learning models.

Bag-of-Words vectorization

For all but one model architecture in this study, we have used a bag-of-words approach in creating input features. In the bag-of-words approach texts are tokenized into words or n-grams and then encoded as a vector of the occurrence counts of each vocabulary word in the text. We have also experimented with using TF-IDF as our vectorization approach, but found in preliminary experiments that this led to a significant bias towards not assigning any positive labels, therefore we chose to stick with bag-of-words early on during the research process. While this finding is informal on our part, it seems to be supported by some of the conclusions of Padurariu and Braeban[49] regarding the performance of various text representations with linear and decision tree models.

We have used the *CountVectorizer* implementation from *scikit-learn*[51]. We initially included a parameter search over the CountVectorizer hyperparameters in our further analysis, but it quickly became apparent that these converged to a single set of parameters regardless of the choice in model so we will report them here. Table 6.1 lists the found parameters for the vectorizer. In summary: our models mostly consider individual words,

TABLE 6.1: Found optimal parameters for the bag-of-words CountVectorizer

Parameter	Value
n-gram range	(1, 1)
binary	False
minimum document frequency	0.0
maximum document frequency	1.0

not larger n-grams and do not seem to discriminate based on word frequency across the documents.

Tokenization in transformers

Transformer models provide their own tokenization and embedding mechanisms, based on byte-pair encoding over the pre-training vocabulary of these models. This means that we do not need to tokenize our documents before passing them to a transformer model.

6.1.2 Models

As mentioned in section 5.2, the task at hand is a multi-label classification problem. From our four selected model architectures, Naive Bayes, Gradient Boosting and Random Forest do not inherently support multi-label tasks. We therefore need to use these architectures in a setup where we train a binary classifiers for each CCI category and then bundle the 17 classifiers for the full model. For these three models we will use implementations from *scikit-learn*[51].

Naive Bayes

Naive Bayes is a simple machine learning algorithm that models the i.i.d. probabilities of input features, in our case tokens, conditioned on each class and then selects the class that maximizes Bayes' theorem, which is given by equation 6.1. The prior class probability for a given class $c \in C$ is given by the fraction of documents in the training data that have the label for c assigned. The feature likelihood $P(D|C = c)$ can be calculated based on the occurrence rates of tokens in the subset of documents that belong to class c.

$$P(C|D) = \frac{P(D|C)P(C)}{P(D)} \propto P(D|C)P(C) \tag{6.1}$$

Naive Bayes does not have any parameters to search over, only the tokenizer parameter search we reported on in section 6.1.1 was performed for this model.

Gradient Boosting

Gradient Boosting is a ensemble machine learning method in which a sequence of "weak" classifiers are trained. Each successive classifier is trained to correct some of the residual errors of the previous classifier. A final prediction is then made by summing over the log-odds predictions of the weak classifiers, weighted by a learning rate. The most commonly used weak classifier is a shallow decision tree, this is what we will use.

Table 6.2 shows the parameter search space used for gradient boosting. We used the default setting of 100 weak learners and $lr = 0.1$ as a baseline and varied around these.

The range of number of estimators is capped at 400, as gradient boosting classifiers have a tendency to over-fit for large numbers of weak learners.

TABLE 6.2: Hyperparameter search space for Gradient Boosting

Parameter	Search Space
Number of weak learners	[50, 100, 200, 400]
Learning rate	[0.05, 0.1, 0.2, 0.4]

Random Forest

Random Forest is an ensemble method that trains multiple decision trees classifiers in parallel, with each tree using a random subset of features and data samples during training. Given our model input is a bag-of-words, this means each decision tree only considers a subset of the full vocabulary during training and trains on a random sample of documents rather than all training data. The predictions of the individual trees are then aggregated to come to a final prediction.

Table 6.3 shows the parameter search space for random forests. We put the range of number of individual estimators significantly higher compared to gradient boosting as this is less likely to result in over-fitting in random forests due to the independent, parallel construction of the estimators. The number of features selected for each decision tree was kept at the default setting, which is the square root of the vocabulary size.

TABLE 6.3: Hyperparameter search space for Random Forest

Parameter	Search Space
Number of decision trees	[100, 250, 500, 750]
Bootstrap sample size	[0.5, 0.75, 1]

Transformers

First introduced by Vaswani et al.[68], transformers are powerful deep learning architectures that are well known for their generative capabilities, but can also be applied to traditional NLP tasks like NER, sentiment analysis and text classification. A commonly-used branch of transformer models are those in the BERT[17]/RoBERTa[37] family, these models consist of only the encoder half of the full transformer architecture and have become very successful in non-generative NLP tasks.

A downside of transformer models is that they are "data-hungry", meaning that they require extensive pre-training to develop and internal model of language structures, even before being fine-tuned on specific tasks. It is generally beneficial if the documents used in pre-training are from the domain in which the ultimate NLP task takes place, this has led to the creation of a wide range of domain-specific language models. We have compared a number of pre-trained BERT-based models for the problem at hand, in the rest of this chapter we will only present the results for the best performing variant, which was MedRoBERTa.nl[69]. An overview of the tested variants can be found in appendix D.

Table 6.4 shows the parameter search-space for the used transformer model. Following the work of Devlin et al.[17] we defaulted to a learning rate of $2e^{-5}$ and batch size of 16, and decided to try a variation on the learning rate in both directions. The lower batch rate was tried as it may result in better accuracy and training stability, at the cost of a longer training time.

TABLE 6.4: Hyperparameter search space for Transformer models

Parameter	Search Space
Batch size	[8, 16]
Learning rate	[5e-5, 2e-5, 1e-5]

6.1.3 Parameter Selection

Hyperparameter tuning and model performance evaluation should be separated, as combining these tasks may lead to hyperparameter over-fitting which would lead to a poor generalization performance not reflected in the evaluation results[8]. We separate the two tasks by adapting the cross-validation approach described in section 5.4.2 into a nested cross-validation; for each iteration in the cross validation we perform a second cross validation over the training data. The inner validation is used to find hyperparameters, training and validation for the outer cross validation is then performed as normal using the found parameters. We have set the number of folds for the inner cross validation to $k_{inner} = 5$.

6.2 Analysis of Note Structures

During the annotation process of the **DATA-HIP** set, we observed that the clinical notes a sort of quasi-structure: medical professionals use headings to segment the clinical notes into sections. We have attempted to reflect this structure in the artificial example in figure 5.2. While types of sections observed are fairly consistent across the dataset, headings are not standardized and there is no strict order. The presence of this quasi-structure may offer possibilities for improving model performance and filtering, as the various sections may contribute differently to the set of comorbidities. We explore this inherent structure through the following process.

1. **Identify commonly used headings.**
   The general pattern observed for headings in the **DATA-HIP** set is as follows: $\backslash r\backslash n$*[heading_ text]*[*symbol*]$\backslash r\backslash n$, where *[heading_ text]* is the name of the section and *[symbol]* is a punctuation symbol, most commonly a semicolon or forward slash. Using a regular expression (regex) pattern, we can search the clinical documents for strings of text that match this pattern and aggregate all matches to find common heading text strings. While headings without punctuation were observed, we consider the terminating punctuation symbol non-optional during this step, as making it optional would result in commonly occurring list items, such as certain common medical conditions or medications, showing up in the results. It was also decided to adapt the regex pattern in order to account for two observed factors: firstly headings may consist of two words separated by a space or slash, secondly strings containing the terms *(consult)* or *(hoofdbehandelaar)* before the terminating symbol are matched, but these two terms are ignored in the match results. Accounting for these factors, the resulting regex pattern is as follows:

   \r\n(\w+[|/]*\w+?)\s?(?:\(consult\)|\(hoofdbehandelaar\))?[:|\\|/]\r\n

2. **Group headings indicating the same section type.**
   As the headings are not standardized, a variety of headings can be used to indicate the same section across different documents. For example, the heading for a medical

history section can be *medische geschiedenis:* or alternatively *voorgeschiedenis:*. We group these headings such that further analysis will only have to consider the type of section rather than individual headings. During this step we only consider headings from step 1 for which we found at least 30 matches, that is, all headings that have an occurrence rate of 1% or higher, under the assumption that headings occur only once in a given text.

3. **Segment clinical notes.**
   The sections identified in step 2 will be extracted from the documents in **DATA-HIP**, such that further analysis can be done on a per-section basis.

4. **Determine per-section feature importance.**
   We will determine the distribution of summed feature importance scores over the sections that were isolated in step 3. The tree-based classifiers are inherently interpretable, and feature importance scores can be derived directly from the decrease in purity for each feature during the training phase. For the transformer-based models feature importances may be derived based on the integrated gradients[61] approach. We use the implementation for the Gini importance for the Gradient Boosted model as provided by scikit-learn[1], and integrated gradients for MedRoBERTa.nl based on the transformers-interpret package[2].

6.3 Results

This section covers the results from the experiments described in sections 6.1 and 6.2. We will first discuss the obtained hyperparameters and model performance, and cover the note structure analysis second.

6.3.1 Model Selection Results

Found paramaters

The found optimal hyperparameters for the tested models are listed in table 6.5. The optimal number of estimators for the Random Forest model fluctuated across the validation folds, varying between the 250, 500 and 750 settings. As 500 was the most common value, and is a compromise value between the two other settings, we have chosen to use it in further experiments. All other found parameters were fully consistent across all cross validation folds.

TABLE 6.5: Found optimal parameters for each model type.

Model	Optimal Parameters
Gradient Boosting	{learning rate: 0.1, N learners: 400}
Random Forest	{max samples: 1.0, N estimators: 500}
Transformer	{batch size: 8, learning rate: $5e^{-5}$}

For the gradient boosting model it was found that the highest setting for number of learners performed best, this indicates that our search range for this parameter may have been somewhat conservative. With an even higher setting for the number of learners gradient boosting may see an increase in performance compared to our results, but we

[1] https://scikit-learn.org/
[2] https://pypi.org/project/transformers-interpret/0.3.0/

doubt that this would be very significant. Random forest notably performed best when all training samples were included for every estimator, meaning that no bootstrapping was used. For the transformer model we see a preference for a small batch size and a larger leaning rate, these findings are expected and go hand-in-hand, as an increase in training stability due to using smaller batches allows for the larger learning rate.

Model performance

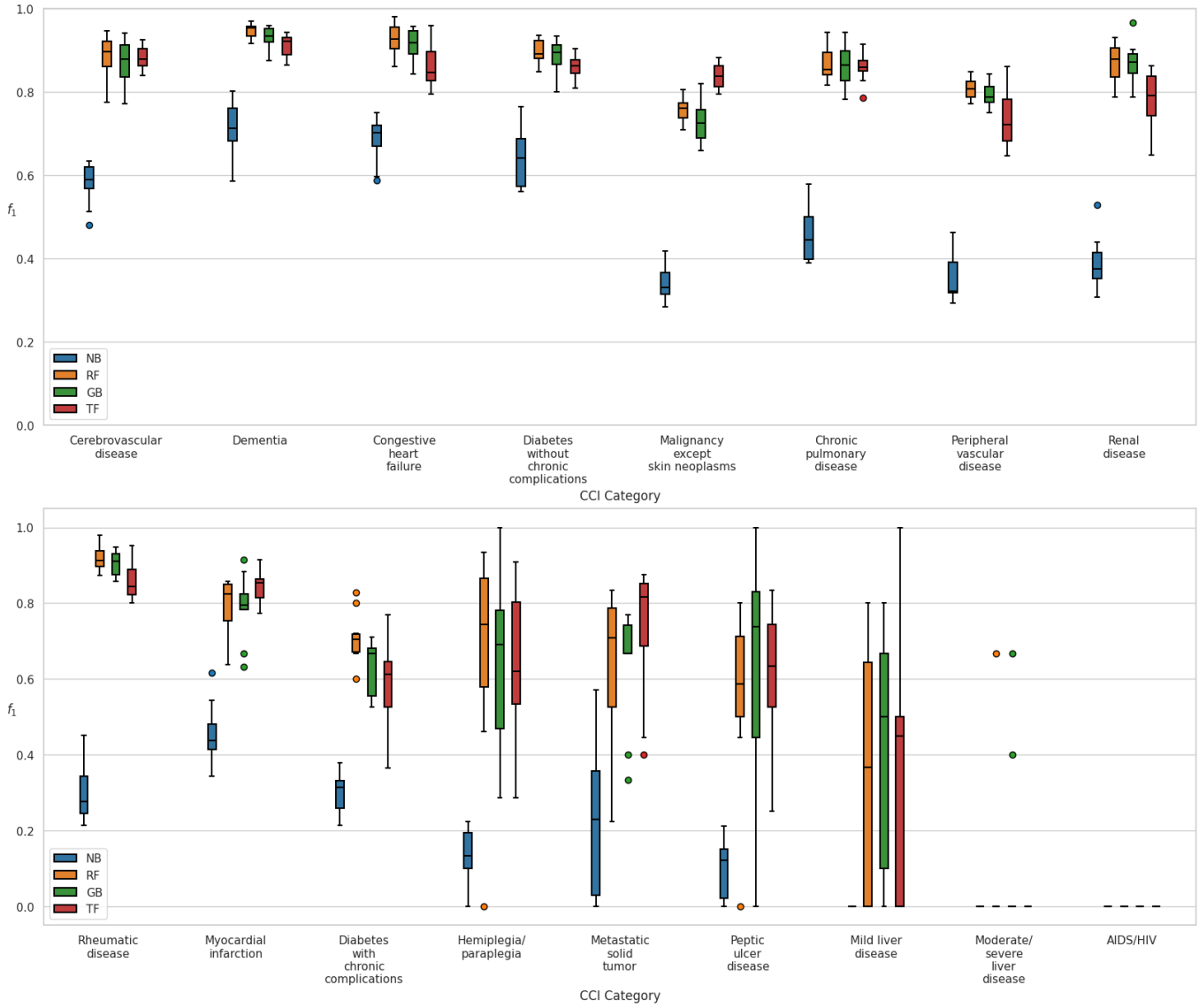


FIGURE 6.1: f_1 score distributions for fully supervised models. Categories have been ordered by prevalence from the top left to the bottom right.

Figure 6.1 graphs the f_1 scores resulting from the 10-fold outer validation for the four chosen models. Naive Bayes performs poorly across all classes. The other three models all show similar performance patterns, random forests has a slight edge in the majority of classes but is outperformed by a decent margin by transformer models for a select few categories. Overall it can be observed that the performance for RF, GB and TF is promising for all categories with a prevalence of over 5%, that is all categories up to *Myocardial in-*

farction. After this threshold performance drops off significantly to near zero for the rarest present category, *Moderate/severe liver disease*. It can also be observed that performance variance is significantly larger for the rarer categories, that is, the model is inconsistent across training folds. Note that the last listed category, *AIDS/HIV*, does not occur in our dataset at all, we chose to represent this as a 0 performance, this will remain the same in the other experiments in this thesis.

Table 6.6 shows the model performances at a document level, confirming the results we observer in the per-class analysis: Random Forest, Gradient Boosting and the transformer perform very similarly, all having an overall classification accuracy around 70%, with a slight edge for the random forest, as reflected in the mean absolute error. It is notable that there is only a 0.01 difference between the percentage of correctly classified notes, and the percentage of correct CCI scores, this indicates that there are few cases of "flipped" labels, that is, few documents for which a correct label was missed and replaced by another label.

TABLE 6.6: Document-level metrics (mean±std over 10 folds)

model	CCI MAE	Classification accuracy	% CCI correct	% CCI within 1
NB	2.19 ± 0.13	0.28 ± 0.02	0.32 ± 0.02	0.54 ± 0.02
RF	0.44 ± 0.04	0.71 ± 0.03	0.72 ± 0.03	0.89 ± 0.01
GB	0.47 ± 0.05	0.69 ± 0.04	0.70 ± 0.04	0.88 ± 0.01
TF	0.46 ± 0.05	0.71 ± 0.02	0.72 ± 0.03	0.89 ± 0.01

6.3.2 Note structure analysis

TABLE 6.7: Identified section coverage in clinical notes.

Heading (English)	Coverage
medical history	85%
additional examination	99%
physical examination	97%
anamnesis	99%
chief complaint	69%
diagnosis	57%
laboratory results	77%
medication	81%
policy/therapy	96%
vitals	22%
allergies	36%
ECG	58%
radiology	82%

Based on steps 1 and 2 from section 6.2, we have identified 13 commonly occurring sections. The identified sections are listed in table 6.7, along with the percentage of documents in which they could be found. An overview of the headings included for each section and the match counts for these headings can be found in appendix A. It can be observed that most sections are present in a significant number of notes. With sections which one would expect to contain important information regarding comorbidities, such as *medical history* and *physical examination* being present in more or less all notes. Notable outliers

with relatively low coverage are *diagnosis*, *vitals*, *allergies* and *ECG*.

Figure 6.2 shows the distribution of feature importance scores derived from the Gradient Boosted model in step 4 of 6.2. The medical history section contributes the most feature weight, about 45% of the summed feature importance over the **DATA-HIP** set. The remaining feature weight is spread out more evenly, with the second most important section, anamnesis contributing only about 10%. It can also be observed that the distribution in the number of informative features is completely different, with the medical history contributing under 10%. This indicates that the medical history section is made up of a relatively small amount of highly informative features, and the rest of the document contains mostly features of a lower importance.

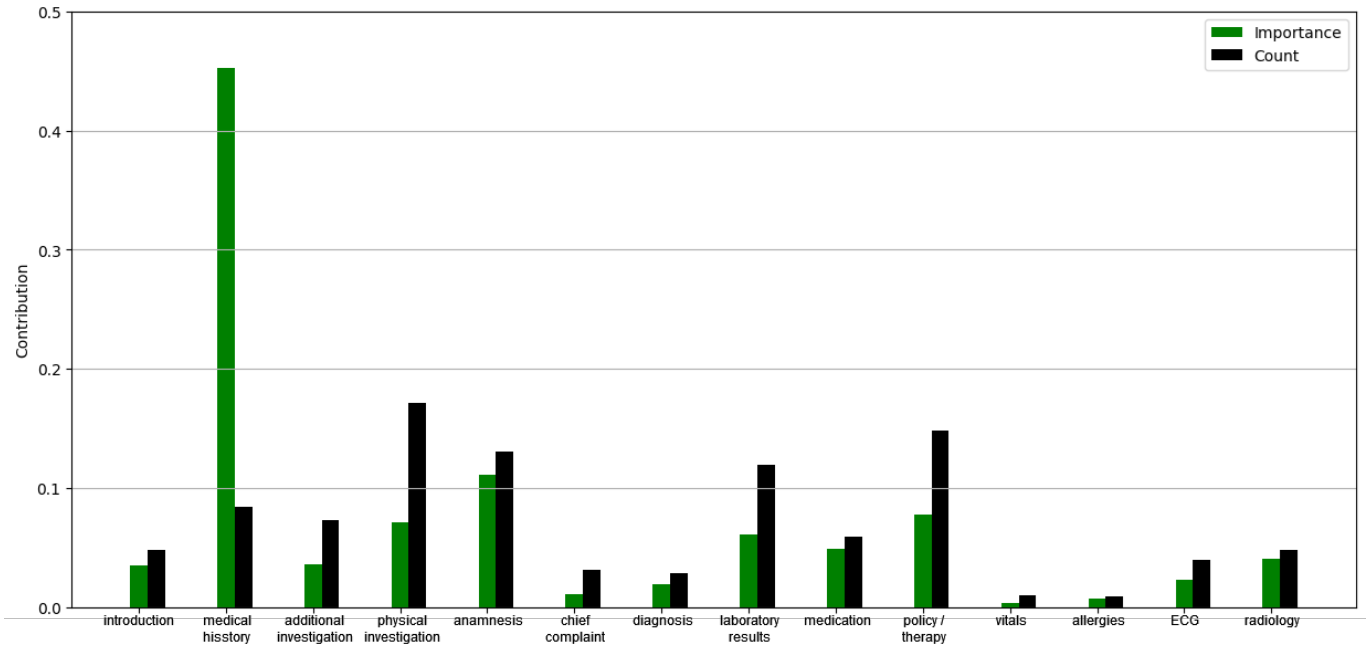


FIGURE 6.2: Summed feature importance and feature counts per section

While the integrated gradients method we attempted for the transformer model returned relevant results for individual sections, correctly identifying important features when they were present, it did not allow for interpretable aggregation of results. The scaling of the integrated gradient scores was not consistent when comparing two isolated sections; an isolated section containing words that indicate a comorbidity can be assigned the same score as an isolated section that contain no relevant text at all.

6.4 Discussion

6.4.1 Model performance

With the exception of Naive Bayes, all tested classifiers showed good performance on the more common classes in the Charlson comorbidity index, with the median f_1 score being over 0.8 in nearly all cases for categories with a prevalence of over 5%. After this 5% threshold median performance dropped rapidly and variability increased significantly. While the rarity of these poor performing classes meant that mean performance in terms of the CCI score was decent, it should be noted that these rare classes are mostly categories

with higher (> 1) weights in the CCI, that is, the set of most severe comorbidities. Therefore our performance is lowest on documents for the most fragile set of patients. This is problematic in a research setting, as high comorbidity patients are a subset which is some cases may be studied separately or be considered the most important sub-population, and even more so in a care setting, where we do not want to structurally under-report the comorbidity load for fragile patients. We therefore believe that the low performance for rare categories should be addressed.

We believe that the poor overall performance for Naive Bayes is due to the large amount of feature noise present in the documents; comorbidities are indicated by a handful of highly indicative words in each document, with the great majority of text being noise, Naive Bayes seems to deal with this especially poorly. Regarding the comparative performance of the other three classifiers: a slight edge in performance in favour of the tree-based model over the transformer architecture can be observed for most categories. This may in part be due to most comorbidities being stated as stand alone entities within the documents. The power of transformer based models lie in their ability to consider entire sentences and textual context, however in the absence of such context this may result in spurious correlations. This seems to be confirmed by the fact that the category for which the transformer does outperform the tree based models significantly: *Malignancy, except skin neoplasms*, is somewhat contextual. To illustrate this point, consider a document that contains the sentence *"carcinoom van de huid" (carcinoma of the skin)*, but no mentions of other cancers. The transformer model is able to process this phrase in it's entirety, taking into account the fact that *"carcinoom"* and *"huid"* are part of the same phrase. The tree-based classifiers however, use 1-gram BoW vectorization, and consider the words in the phrase as completely independent input features. It is therefore plausible that tree-based models are more likely to erroneously label the document under *Malignancy, except skin neoplasms*.

6.4.2 Note structure analysis

Based on conversations with medical experts at ZGT, we can anecdotally confirm that the results in table 6.7 is as expected based on the reporting practice for ED intake notes at ZGT. The five most common headers — *medical history*, *anamnesis*, *physical examination*, *additional examination*, and *policy/therapy* — are elements that should always be reported on in ED intake notes. Other sections may be missing from any given report, either because the relevant information is not yet known or because the information is reported elsewhere. Examples of information that may not yet be known are *medication*, which may have to be obtained from a general practitioner, or an *ECG* which may not have been taken yet. *Laboratory results* and *vitals* are examples of data recorded elsewhere, as these are registered in structured form in the EHR thus stating these result in the clinical note is double reporting. After visual inspection, we believe that the observed lower occurrence rate of the *diagnosis* section is due to the diagnosis frequently being merged into *policy/therapy*, rather than being given separately.

The observed pattern for feature importance and informative feature counts lines up with our expectations based on manual inspection of the documents and patterns observed during the labeling process. The medical history sections in the clinical notes contained the most relevant terms during the labeling process by a wide margin, and this section was often stated in the form of a list of stand-alone diagnostic terms or short sentences. This observation for the medical history agrees with the observed result of a high feature

importance based on relatively few features in figure 6.2.

The observed distribution for feature importance scores makes it difficult to make any recommendations as to improving model performance or efficiency based on the document structure. The medical history is generally structured as a list, a possible approach would be to evaluate whether the items in this list are relevant comorbidities on a item-by-item basis. However, we doubt that this would lead to significant performance gains over classifying the document as a whole especially in the case of the context insensitive, classifiers such as Random Forests and Gradient Boosting, given that the list items are largely context free. Also, while the medical history is the most important section by a margin, it contains under 50% of the overall feature weight, and the remaining feature weight is spread out, making it difficult to trim documents significantly without compromising classifier performance.

Chapter 7

Phase 2: Weak Supervision

The second phase of this study is concerned with addressing the issue of low performance for rare classes. Following the agile methodology this phase consisted of five sprints covering an exploration of options for addressing the issue, gathering of resources and set-up, refinement of the chosen approach, and execution. Figure 7.1 shows an overview of the process, the various design decision made and gives the main rationalization for each design decision. Section 7.1 briefly covers three approaches that were considered. Section 7.2 covers the general principle and methods of the chosen approach. 7.3 discusses a number of refinements of the chosen approach, addressing some of the issues mentioned in section 5.5 and resulting in a number of the design decisions in figure 7.1. Finally 7.4 provides an overview of the fully composed approach.

7.1 Exploration options

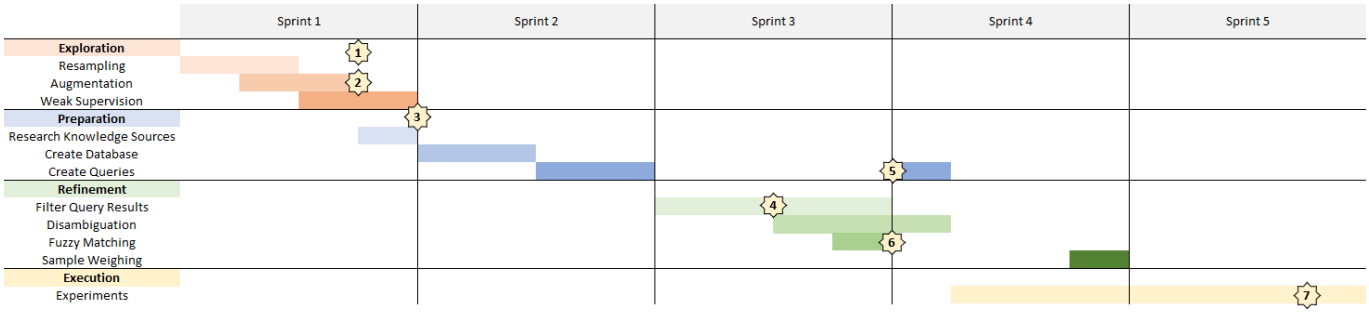
Three options for improving performance for rare classes were explored: resampling, data augmentation using random insertion and weak supervision. We performed performed some informal, exploratory experiments to assess the viability of these approaches. For the sake of brevity we will only discuss the conclusions and our ultimate choice of method rather than giving these experiments a full treatment.

7.1.1 Resampling

Resampling is the practice of changing the number of samples available per class in order to achieve a better class balance in a training dataset. Common schemes for resampling include: random under-sampling which excludes a random subset of a majority class from training, random over-sampling which duplicates random samples from a minority class, and SMOTE[11], a technique that creates synthetic minority samples.

We investigated the use of random over- and under-sampling to improve classification performance for our use case by retraining classifiers after resampling and comparing performance to model performance in phase 1. It should be noted that the transformer architecture was not included in this exploration as resampling for a true multi-label classifier is significantly more complex compared to resampling for the 17 binary architectures used for Naïve Bayes, Gradient Boosting and Random Forest.

We found that while resampling affected performance, it did not strictly improve it. Under-sampling resulted in increased recall for several categories at the cost of of a drop in precision, conversely over-sampling resulted in increased precision for rare categories at a significant penalty in recall. Given that we err on the side of precision over recall



1. **Use weak supervision as main approach for improving performance for rare classes.**
   - The primary limiting factor for performance is data availability.
   - The current number of positive samples for rare classes is too small for effective re-balancing and augmentation approaches.
2. **Extend dataset with non-hip fracture documents for patients in the same age cohort.**
   - No more documents for hip fracture patients were available at ZGT.
   - Medical histories for patients in the same cohort with different fractures are sufficiently similar to provide useful information in training.
3. **Programmatic weak supervision using aggregated terminologies.**
   - Most comorbidities are indicated in-text a limited set of key terms, which can likely be found in ontologies / terminologies.
   - The Fortin et al.[22] code list insufficiently matches language used in practice.
4. **Disambiguate abbreviations using active labeling and random forests.**
   - Abbreviations are prone to mislabeling, due to having multiple context-dependant interpretations.
5. **Explicitly exclude skin cancers.**
   - *Malignancy, except skin neoplasms* has a subset excluded from its definition.
   - Terms for concepts higher in the SNOMED hierarchy may label more fine-grained concepts, thus we require an extra check for the fine grained concepts.
6. **Do not perform fuzzy matching or misspelling correction.**
   - The occurrence rate of misspellings is fairly low.
   - Naive fuzzy matching or misspelling correction may introduce new errors.
7. **Augment weak labels with pseudo labels generated by random forest.**
   - The amount of weakly-labeled data is significantly larger than the amount of hand-labeled data.
   - Poor recall of the label model for a given category will drown out the hand-labeled positive samples for that class.

FIGURE 7.1: Design decisions and rationalization.

in this study we ruled out the use of under-sampling based on these results. We are also reluctant to rely on over-sampling as our main avenue for addressing class imbalance, while it increased precision for rare classes the penalty in recall was significant. We also have significant doubts regarding the generalization of model performance when trained on over-sampled data based on a very limited amount of positive samples.

7.1.2 Augmentation

Augmentation is the process of increasing the amount and diversity of available training data by modifying existing samples. We explored a simple augmentation technique based on random insertion of class-indicative terms: we created new positive samples by randomly injecting terms indicative of rare classes from the Fortin et al.[22] code list into copies of existing samples. The classifiers were retrained after extending the training dataset with the artificial examples and subsequent performance was compared to performance in phase 1. This process was performed for various amounts of artificial examples and imputation into random samples and exclusively existing positive samples.

While we noted increased recall for vary rare classes, i.e. those with near zero performance in phase 1, performance for other classes was negatively impacted, results were also inconsistent and very sensitive to the random choices of injected terms. We again have doubts regarding model generalization when using this technique, as we are implicitly performing over-sampling by creating copies of existing samples and as it would require generating very large amounts of artificial examples to cover the full range of terms that can be injected.

7.1.3 Weak supervision

Unlike resampling and augmentation, weak supervision could not be evaluated using an ad-hoc experiment that fit within a sprint. Nonetheless, considering the exploratory results for resampling and augmentation and taking into account that the primary limiting factor in phase 1 was availability of samples for rare classes we believe weak supervision to be the most promising option. As mentioned in section 3.3 a wide range of weak supervision methods exist, given the wealth of existing medical terminologies and ontologies and inspired by Snorkel[57] and KeyClass[25] we see potential in programmatic labeling approach that leverages those terminologies.

7.2 General approach and methods

We take a programmatic approach to assigning weak labels, this normally requires subject matter experts to design labeling heuristics. We attempt to bypass this requirement by leveraging medical coding- and terminology systems. A natural labeling heuristic would be to check for the occurrence of terms that correspond to a target class within each document. For each relevant term $t \in L_C$ where L_C is the constructed lexicon for CCI category C, a function of the form in listing 7.1 is created, where $VOTE$ indicates a vote for class C, such a function is called a labeling function. All labeling functions for each class are then applied to the documents and we assign a document the class label C if a sufficient number of labeling functions vote for said class. The check as to whether a t occurs in the document is done by performing a regular expression match.

Snorkel provides a generative label model over the labeling function votes, considering the true label as a latent variable, this should allow the generation of probabilistic labels based on the entire set of labeling functions[57][60]. However the open-source version of

Snorkel does not support multi-label problem out of the box. We experimented with binary label models for each class, but found that in almost all cases this results in a document being labeled if any labeling function for a given class votes.

```
function labeling_function_t(document){
    if (t in document) {
        return VOTE;
    } else {
        return ABSTAIN;
    }
}
```

LISTING 7.1: Labeling function prototype.

7.2.1 Dataset

A weakly supervised approach required collection of additional unlabeled samples, this posed a minor issue as the **DATA-HIP** set contained all ED intake documents for geriatric hip fracture patients that were available at ZGT at the time of this research. As an alternative we have chosen to include documents for patients with any other type of fracture, subject to the same constraint of patients being aged 70 years and up. The resulting documents form the **DATA-REST** dataset. We have limited the additional documents to traumatic fracture patients under the assumption that doing so should not introduce significant bias towards any of the comorbidities in the CCI, this is an assumption we could not make if we had included patients listed under other disciplines such as cardiology or nephrology. We also assume that given the age constraint, medical histories for all fracture patients should be sufficiently similar for the additional documents to provide useful information in training, this assumption is supported by the high cosine similarity found in section 5.3.3. We do realize that hip fracture patients are typically more fragile than the average patient, therefore we expect somewhat lower incidence of comorbidities in **DATA-REST** compared to **DATA-HIP**.

7.2.2 External Knowledge sources

One may note that the weak labeling approach described at the start of this section is analogous to the manual labeling process performed for **DATA-HIP**, as during manual labeling the documents were matched against the code list created by Fortin et al.[22] which is ultimately derived from SNOMED-CT. However the Fortin et al. code list is not exhaustive in terms of synonyms, does not include all descendant elements for every higher-level SNOMED-CT concept included, and the language in the chosen SNOMED-CT definitions does not map perfectly onto the language used in clinical practice. During manual labeling these issues can be easily resolved through human judgement, but this is not the case in programmatic labeling. In order to alleviate these issues as best possible it was decided to aggregate multiple external knowledge sources, and extract the relevant terms from the aggregation, rather than to rely on solely the Fortin et al. code list.

We have chosen to use the December of 2023 International Edition of SNOMED-CT as a skeleton system onto which other knowledge sources are mapped. SNOMED-CT is the most comprehensive terminology system available and its polyhierarchical structure and attribute model offer excellent support for querying and subset definition, making

it the best choice for a core terminology system. Dutch language terms from the Dutch SNOMED-CT reference set was available via ZGT systems.

The US National Library of Medicine offer access to a number of medical vocabularies through the UMLS[64], and where available it also provides mappings onto SNOMED-CT concepts. From the UMLS a number of Dutch vocabularies were obtained and mapped onto the SNOMED-CT. Table 7.1 lists the vocabularies that were included.

TABLE 7.1: List of vocabularies obtained from the UMLS.

UMLS code	Vocabulary name
ICD10DUT	ICD, 10th revision, Dutch Translation
ICPCDUT	International Classification of Primary Care, Dutch Translation
ICPC2EDUT	International Classification of Primary Care, 2nd Edition, Dutch Translation
LNC-LN-NL	Logical Observation, Identifiers, Names and Codes (LOINC), Dutch
MDRDUT	Medical Dictionary for Regulatory Activities (MedDRA), Dutch
MSHDUT	Medical Subject Headings (MeSH), Dutch

The mentioned knowledge sources are all freely available either on a UMLS licence[1] or a licence with official distributors in SNOMED member countries[2] In addition to these resources a proprietary list of synonyms for common medical terms was supplied by ZGT. Model performance with the inclusion of the proprietary list in the weak labeling pipeline is relevant to ZGT as a stakeholder, we realise that this would make our experiments non-reproducible therefore we will also report results without inclusion of the list[3].

7.2.3 Storage and querying

The SNOMED-CT International Edition was loaded into a Neo4j[43] graph database instance. The SNOMED data model is inherently a graph thus this is a natural fit; every SNOMED concept is represented as a node, labeled according to its SNOMED hierarchy, and hierarchical (is a) and attribute relationships are the edges between these nodes. Each node contains an unique SNOMED id and descriptions of the concept as properties. This makes the process of adding terms from other vocabularies to the graph simple: for each term for which UMLS provides a mapping to SNOMED, the correct node is identified by its id and the term is added to the properties of the node.

The aggregated graph model can be queried using the CYPHER query language for Neo4j. This is equivalent to intensional subset definition using the SNOMED-CT Expression Constraint Language (ECL), thus for every ECL query, a equivalent Cypher query can be constructed. As an example, figure 7.2 shows the CYPHER and ECL queries this study uses for the myocardial infarction CCI category. The list of ECL queries for all 17 categories and a reference to the CYPHER queries can be found in Appendix C, equivalency of the CYPHER and ECL queries was asserted by comparing the number of nodes returned in Neo4j to the number of concepts returned by the ECL query in the IHTSDO SNOMED Browser[32].

[1] Can be obtained from the US National Library of Congress[64].
[2] In the the Netherlands SNOMED CT is distributed by Nictiz[45].
[3] See the result for the **NO-PROP** run in section 7.5.2

```
//Find descendants of Myocardial infarctions
MATCH (mi:ClinicalFinding)-[:IS_A*0..]->(:ClinicalFinding {id:22298006})
//Find descendants of Acute ischemic heart disease
MATCH (aihd:ClinicalFinding)-[:IS_A*0..]->(:ClinicalFinding {id:413439005})
WITH collect(mi) + collect(aihd) AS results_list
UNWIND results_list as results
RETURN DISTINCT results;
```

```
<<22298006 |Myocardial infarction (disorder)|
OR
<<413439005 |Acute ischemic heart disease (disorder)|
```

FIGURE 7.2: CYPHER and equivalent ECL queries for the Myocardial Infarction class.

7.3 Refinement

As described at the start of this chapter, we iterated on our solution for weak labeling evaluating and adding a number of refinements. This section covers the elements we evaluated.

7.3.1 Misspelling correction

The first element we want to mention concerns the problem of misspellings. We experimented with fuzzy matching methods in order to account for potential misspellings in our dataset, but ultimately decided not to include it in our final approach. This is because we found that inexact matching introduced more erroneous labels than it fixed. A notable example of an introduced error concerns the terms **neuropathie** and *nefropathie*, the first is not necessarily relevant to any of our CCI-categories, while the second is part of *Renal disease*. Because there is only one letter difference between the two terms, fuzzy matching at the strictest possible setting .allowing for an edit distance of 1. would mislabel all occurrences of **neuropathie** as *Renal disease*.

7.3.2 Filtering Terminology

The class lexicons resulting from our queries on the graph database were rather extensive, containing several thousands of concept descriptions for some classes. Cross referencing these thousands of terms with the over 20000 documents in **DATA-REST** is a computationally intensive task, which we had to execute multiple times while iterating on this work. In order to streamline this process we applied a number of filters to the lexicons.

First of all we chose to remove concept description that consisted of more than five tokens after undergoing the tokenization and stop word removal process that was also applied to the documents. We believe that we can apply this filter without significant loss of power for our approach, as the longer descriptions removed by this filter are unlikely to occur in the concise ED notes, clinicians will likely use a shorted description for the same concept instead. Secondly we apply a process in which we "fold up" the class lexicons, that is we remove descriptions which are a superset of another description also present in the lexicon. For example, the lexicon for the class *Myocardial infarction* contained the terms "hartinfarct" and "acuut hartinfarct", we remove the latter because any occurrence of the

latter term in a document would also be labeled by the former. Finally we observed that the lexicons of some mutually exclusive classes contained a limited number of identical descriptions, we resolved this by only keeping these descriptions for the less severe of the two CCI categories.

7.3.3 Exclusions

Two mechanisms for excluding a term from being labeled under a class lexicon L_C can be identified. The first and rather obvious way is to not include the term in L_C to begin with, we will call this an *implicit* exclusion. While implicit exclusion is sufficient in most cases, another mechanism is required when a target class is defined as being some set of medical concepts minus a narrowly defined subset. In order to clarify this point consider again the poly-hierarchical structure of SNOMED CT depicted in figure 7.3 and say we have defined some class to consist of concept C and all its descendant except for concept H. If we choose to rely only on implicit exclusion, then it has to be taken into account that terms associated with higher-level in the hierarchy are more general and may also label concepts lower in the hierarchy, thus we may have to exclude terms associated with concepts C and E from our lexicon to avoid labeling concept H.

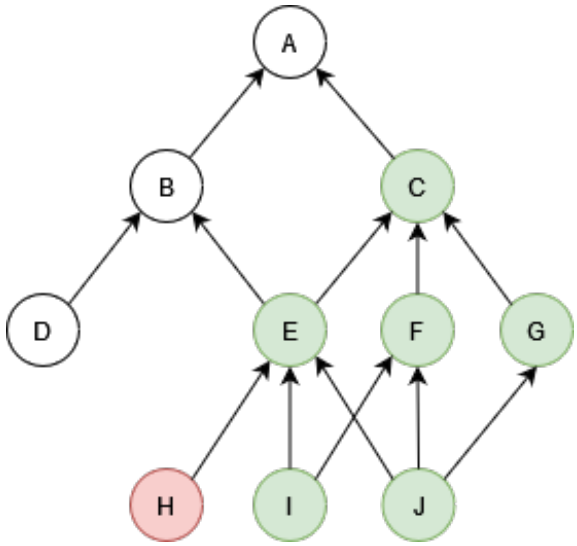


FIGURE 7.3: Polyhierarchy with defined class (C minus H).

A solution to this issue is to *explicitly* exclude concept H, meaning that whenever a document is identified as containing a term belonging to C or its descendants a check is performed to assert that the term is not a term for H. It should be noted that explicit exclusion should be used sparely as it slows down labeling by significantly increasing the number of requited matching operations.

In this study we use explicit exclusion to remove skin cancers form the class *malignancy, except skin neoplasms* as generic terms have significant labeling power for this class but would erroneously label skin cancers, for example: "carcinoma" would label "basal cell carcinoma".

7.3.4 Negation handling

As was the case during the manual annotation process for **DATA-HIP**, negated terms should be ignored by the weak labeling approach. Programmatic labeling, being a rule-

based approach is insensitive to context cues like negation, therefore we need to manually correct for negation. We do so by prepending tokens between a negation and the next punctuation mark with "NOT_" and capitalizing these token, this excludes these tokens from being labeled by any labeling function. We extend this behaviour to cover the terms from table 5.1 that indicate a differential diagnosis, an unsure diagnosis or a suspicion, as was the case for the manual annotation process.

7.3.5 Disambiguation of abbreviations

In table 5.4 we provided a number of examples of abbreviations with multiple interpretations, identified in the **DATA-HIP** set. These types of abbreviations are also present in the final lexicons for several CCI categories. This is problematic for the chosen weak labeling approach as the associated labeling functions do not account for context and thus will be prone to mislabeling. In order to disambiguate labeling functions for abbreviations, we add a second stage to the labeling process in which potentially problematic labeling function are reevaluated based on active input from a user.

We have limited the labeling functions that should be reevaluated to those with a keyword consisting of four or fewer character that vote for the associated class on more than 50 documents. These thresholds were chosen such that the most problematic cases are caught, while the number of labeling functions that need reevaluation is kept limited. Each labeling function that fits these criteria is replaced through the following process:

Given a labeling function λ_t for term t belonging to class C, where $\#votes_{\lambda_t} > 50$ and $|t| < 5$:

1. The set of contexts for t is identified by extracting the sentences containing t from the documents for which λ_t votes.
2. $N = 20$ contexts are sampled randomly.
3. A user is prompted to provide active input as to whether t indicates C in each of the 20 contexts.
4. A small instance of a random forest classifier is fit to the hand labeled examples.
5. λ_t is replaced with a function that identifies sentences containing t and then performs inference over the sentence using the random forest classifier.

After all problematic labeling functions have been replaced, the new labeling functions are applied to all documents. While we realise that a sample size of 20 is very small we are not too concerned with the classifier over-fitting for the purposes of this study. In most cases only one interpretation of an abbreviation was present in the dataset, in these cases the random forest will always provide the correct label. When multiple interpretations were present in significant proportions, the context were generally distinct, so the classifier should be suitable. Furthermore we have a preference for precision over recall in the weak labeling process, in that regard the classifier is very unlikely to perform worse compared to the stage 1 labeling function.

7.3.6 Pseudo-labeling

While we have tried to improve the recall of the label of the weak labeling pipeline by including multiple available terminologies, a mismatch between the language used in practice and the terms obtained from the medical terminologies may still exist. This can be

due to a variety of factors, such as the use of more informal terms for certain conditions or because more complex description are broken up or written in some other word order than the associated term in medical terminologies. We observe the first factor for the *Hemiplegia / paraplegia* category, as common indicative terms for this category include *"hemibeeld l/r" "zwakte l/r"* and *"hemi l/r"*, but these terms are absent from all used terminologies. We observe the second factor very strongly for *Diabetes, with chronic complications*, as diabetes and the associated complication are regularly mentioned in different lines in the clinical documents, and to a lesser degree for *Peripheral, vascular disease*, *Malignancy, except skin neoplasms* and *Renal disease*.

As our training data consists of significantly more weakly-labeled than hand-labeled documents, low recall for the weak labels will likely have an impact on model performance. We address this issue by augmenting the weak labels with pseudo-labels. Pseudo-labels are generated by inferencing a fully supervised classifier over **DATA-REST**, and using the resulting predictions as labels in training. We use the Random Forest classifier trained in phase 1 for generating pseudo-labels, while performance for this classifier was not perfect by any stretch, it encodes a substantial amount of information regarding informal en non-standard terms used for comorbidities in the more common categories, and should be suitable for augmenting the weak labels. The weak- and pseudo-labels for each document are merged by taking the logical *OR* of the two labels.

7.4 Overview of full approach

Figure 7.4 **A** shows an overview of the full training pipeline including weak supervision for **DATA-REST**. As illustrated **DATA-HIP**, and it's associated manual labels are used twice: they are included in training the final classifier, and used to train a supervised classifier for pseudo-labeling. Labels for **DATE-REST** are created in two ways: using the supervised classifier, and based on the class lexicons from SNOMED CT and the UMLS. These labels are merged and subsequently included in training the final classifier.

Figure 7.4 **B** zooms in on the weak labeling element of the pipeline, illustrating it for a single CCI-category, this process is applied independently for all 17 categories. The labeling functions that simply check for the presence of terminology from class lexicons are first applied to all unlabeled documents, resulting in a matrix of labeling function votes. Based on this vote matrix we identify the labeling functions for short labels that label often. The labeling functions that are flagged based on our chosen thresholds of a term length under 5 character, and 50 minimum votes are then replaced by a Random Forest classifier as specified in section 7.3.5. We then determine the new vote matrix which can be converted to document labels for the given CCI-category. With our chosen thresholds and requirement for labeling 20 samples, the overall time required for the active input step when executing this process for all 17 categories is 10-15 minutes.

7.5 Experiments & Results

To validate our proposed weak labeling approach we performed two experiments, first we compare the performance of weakly supervised classifiers to the performance of the fully supervised classifiers from chapter 6 and second we assess the impact of design elements in our weak labeling pipeline on the classifier performance.

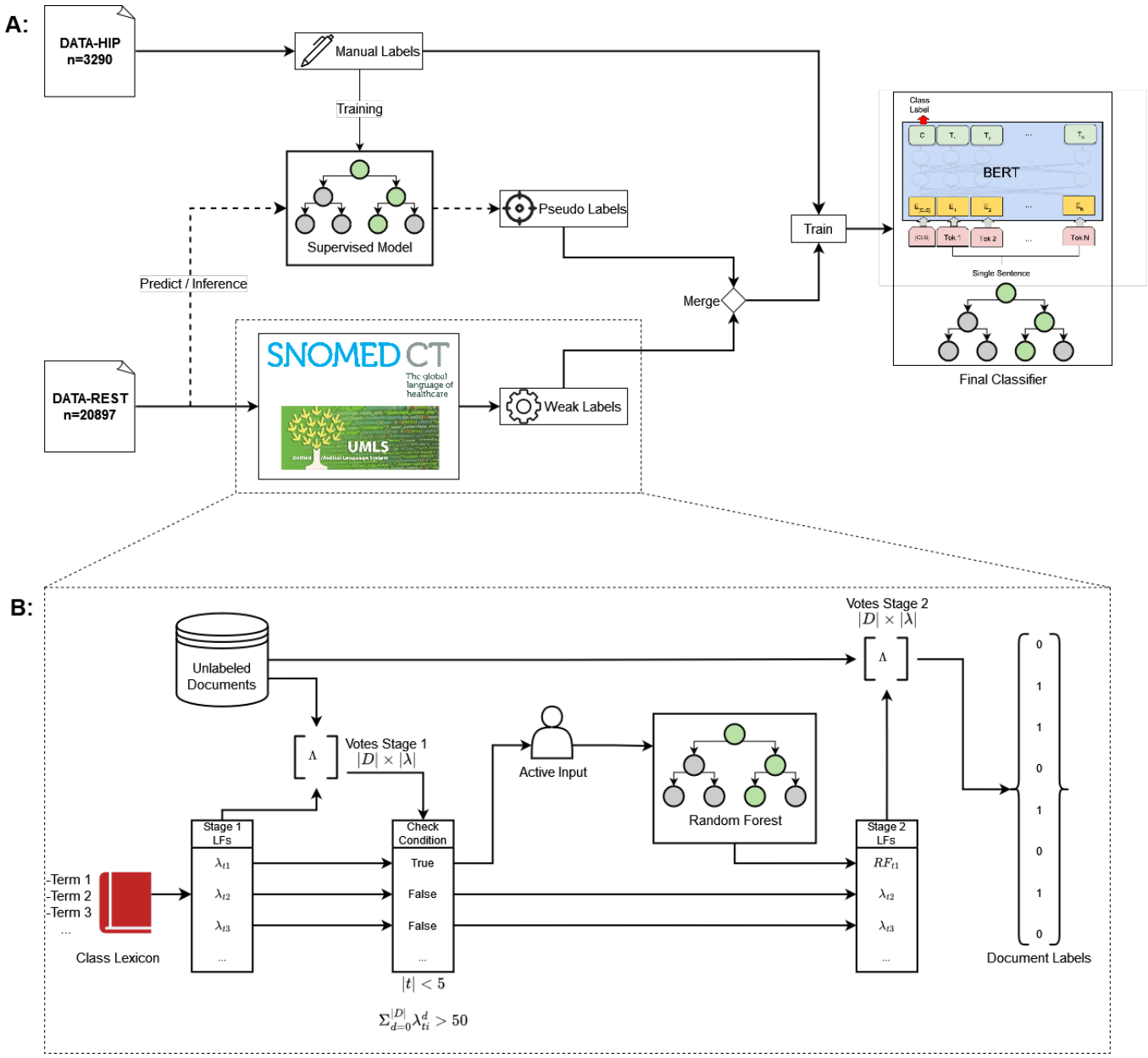


FIGURE 7.4:
A: Training pipeline for final classifier.
B: Weak labeling pipeline.

7.5.1 Experiment 1: Comparison with full supervision

We re-trained the two best performing classifiers from chapter 6, Random Forests and MedroBERTa.nl, on the combination of **DATA-HIP** with manually created labels and **DATA-REST** with weak labels generated by the full weak labeling pipeline as described by section 7.4. As before, the models are validated using a 10-fold validation over **DATA-HIP**, thus for each of the ten folds the training data consists of 90% of the documents in **DATA-HIP** and all of **DATA-REST**. We do not perform a nested cross-validation in this case, as we copy over the found optimal hyperparameters from the experiments in chapter 6. We believe that it is acceptable to copy the hyperparameters as the problem

domain, types of documents and model architectures do not change. To allow for the best possible comparison, the 10-fold split applied to **DATA-HIP** is identical to the split used in chapter 6.

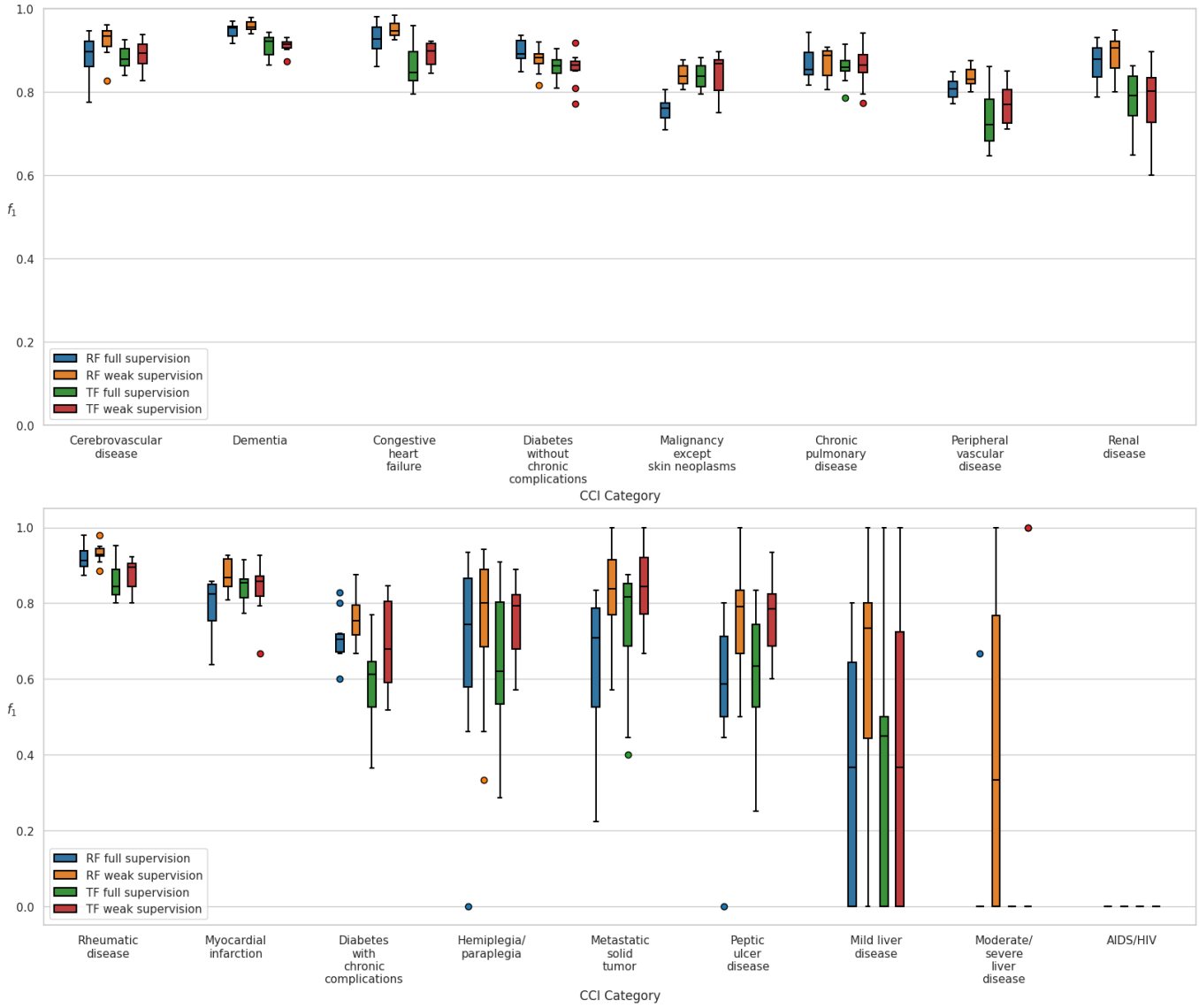


FIGURE 7.5: Performance of fully and weakly supervised random forest and MedRoBERTa.nl.

Figure 7.5 graphs the per-class f_1 score of the weakly- and fully-supervised Random Forest and MedRoBERTa.nl models. For both architectures, the weakly supervised scheme outperforms full supervision for CCI categories with occurrence rates below 5%, and minor increases in average performance can be observed for the more common categories. The comparative performance of Random Forests and MedRoBERTa.nl under the weakly supervised scheme is similar to what was observed under the supervised scheme: the random forests have a slight edge in most categories and perform significantly better in the two rarest categories present.

Table 7.2 shows the document-level results for the weakly- and fully-supervised Random Forest and MedRoBERTa.nl models. A notable result here is that while the classification accuracy rate for Random Forest improved by 4%, we see no significant improvement for MedRoBERTa.nl, despite the noted increases in f_1 score for several categories for MedRoBERTa.nl, this is also reflected in the mean absolute errors. This may be due to two factors: first of all improvements in classifying rarer categories are not well reflected in the overall accuracy rate, precisely because these conditions are rare, and secondly in figure 7.5 we see larger performance gains on common categories for Random Forest compared to MedRoBERTa.nl. Again we note little difference between the classification accuracy and the CCI precision, indicating that most correctly predicted CCI scores correspond to correct predictions for all labels. It can also be observed that for the Random Forest, the fraction of predicted CCI scores within 1 point of the correct CCI score increased from 0.89 to 0.92, this mirrors the increases observed for categories with a prevalence under 5%, 4 out of 6 of these categories have CCI weights larger than 1 assigned to them, thus wrongly classifying these categories would affect this metric.

TABLE 7.2: Full vs. weak supervision: document-level metrics (mean±std over 10 folds)

model	CCI MAE	Classification accuracy	% CCI correct	% CCI within 1
RF full supervision	0.44 ± 0.04	0.71 ± 0.03	0.72 ± 0.03	0.89 ± 0.01
RF weak supervision	0.35 ± 0.03	0.75 ± 0.02	0.76 ± 0.02	0.92 ± 0.01
TF full supervision	0.46 ± 0.05	0.71 ± 0.02	0.72 ± 0.03	0.89 ± 0.01
TF weak supervision	0.46 ± 0.06	0.72 ± 0.02	0.73 ± 0.02	0.89 ± 0.02

7.5.2 Experiment 2: Labeling pipeline ablation testing

TABLE 7.3: Overview of runs in ablation testing

Run	Description
FULL	Full pipeline as described by section 7.4.
NO-PROP	Proprietary list of synonyms has been excluded; weak labeling only based on SNOMED-CT and UMLS.
NO-SUP	No supervised labels are used in training; weak labeling pipeline has been applied to **DATA-HIP**. *N.B.*: pseudo-labeling was not applied to **DATA-HIP** as this would amount to data leakage given that the pseudo-labeling model was trained on **DATA-HIP**
NO-ACTIVE	Disambiguation of abbreviations / active input is excluded.
NO-PSEUDO	No augmentation with pseudo-labels; only weak labels are used for **DATA-REST**.

We further evaluate our approach through "leave-one-out" ablation testing, that is, we train a classifier multiple times, excluding one component from the labeling and training approach each time. This process should help to understand the importance and contribution of each tested component to the overall model performance. Table 7.3 lists an

overview of all training runs performed in this experiment. The ablation tests have been performed using random forests as a base model, as this was the best performing model in all previous experiments. As before, a 10-fold validation over **DATA-HIP** is used for experimental validation.

FIGURE 7.6: Ablation testing performance for random forest.

Figure 7.6 shows the per-class f_1 score results in the ablation test. It can be observed that both disambiguation through active input and pseudo-labeling have an impact on the performance of specific categories. We find that the significant drops in performance observed in the **NO-ACTIVE** run correlate with erroneous labeling functions observed in the active labeling stage, i.e. those labeling functions for which almost none of the 20 prompts constituted real examples of the associated class, therefore this result is as expected. The results of the **NO-PSEUDO** run are also as expected, as the most significant drops in performance line up with the categories for which we observed a mismatch in language between terminologies and practice, as described in section 7.3.6. From the results of the **NO-PROP** run it can be seen that exclusion of the proprietary list of synonyms

has little impact on model performance, this is likely because samples the weak labeling pipeline would miss after exclusion of this list are accounted for by pseudo-labeling. Replacement of the supervised labels with weak labels in **NO-SUP** resulted in minor decreases in performance for several classes, mostly lining up with the observed decreases for **NO-PSEUDO**. We believe this is a strong indicator that pseudo-labeling based on a reasonably accurate supervised classifier, and adding supervised data has largely the same effect. In this instance the label quality of the classes for which we see a decrease in performance has been diluted as **DATA-HIP**, which accounts for about 13% of data in any training fold has been neither pseudo-labeled nor hand labeled.

Table 7.4 displays the results at a document level. It can be observed that **NO-ACTIVE** and **NO-PSEUDO** have a significantly lowered classification accuracy rate compared to the full training pipeline. In fact the achieved accuracy in these experiments is lower than the accuracy we observed in chapter 6 for all fully supervised models except Naive Bayes. This result highlights the necessity of the active disambiguation and pseudo-labeling steps in bridging the language gap between medical terminologies and clinical practice. Again we note little difference between the classification accuracy and the CCI precision, though we see a larger discrepancy for **NO-ACTIVE**.

TABLE 7.4: Ablation test: document-level metrics (mean±std over 10 folds)

model	CCI MAE	Classification accuracy	% CCI correct	% CCI within 1
FULL	0.35 ± 0.03	0.75 ± 0.02	0.76 ± 0.02	0.92 ± 0.01
NO-PROP	0.35 ± 0.02	0.75 ± 0.02	0.76 ± 0.02	0.91 ± 0.01
NO-SUP	0.38 ± 0.03	0.72 ± 0.02	0.74 ± 0.02	0.91 ± 0.01
NO-ACTIVE	0.59 ± 0.04	0.61 ± 0.02	0.64 ± 0.02	0.85 ± 0.02
NO-PSEUDO	0.46 ± 0.04	0.69 ± 0.03	0.70 ± 0.02	0.89 ± 0.01

Chapter 8

Discussion and Reflections

8.1 Model performance

We found that regardless of training approach, the approach using a binary random forest per class was the most performant, outperforming even modern transformer-based language models like BERT and RoBERTa. We theorise that the success of the random forest model is in part due to nature of reporting in emergency department and the structure of the resulting documents. The section of these documents describing patient history contains the majority of information regarding comorbidities, and is generally structured in the form of a list in which the individual entries are often context-free. The majority of comorbid condition are therefore indicated by stand-alone keywords or n-grams in each documents, which means that there is not much performance to be gained by using a context-sensitive model such as a transformer. For that reason, we do not believe that more modern, larger, transformer architectures would lead to significant performance gains either.

The introduced combined weak- and pseudo-labeling approach was effective at generating additional informative training samples, allowing us to improve classification performance for rarer CCI categories considerably while achieving on-par or slightly improved performance on the categories that were already performant under the fully supervised scheme. As demonstrated in the ablation testing, the active input component of our weak labeling approach is effective at mitigating the effects of problematic abbreviation in the class lexicon, at the cost of a very small time investment to a user. Optionally, this approach can be made more powerful by broadening the thresholds and criteria for assessment of a labeling function or increasing the number of labeled samples as desired. The augmentation of our weak labels with pseudo-labels, generated by a classifier trained using a set of supervised data, allowed us to "fill in" gaps between the language used in medical terminologies and practice.

8.2 Clinical relevance and applications of our model

Keeping in mind that our purpose in creating a classifier was to allow for the identification of comorbidities as features for clinical research, as well as to facilitate a more complete view on patient comorbidity in ZGT information systems such as the EHR, we will now discuss whether the achieved results are sufficient and relevant for these stated goals.

We believe that our classifier is suitable for the task of feature extraction for clinical research and predictive modeling. While there is still significant room for improvement over our best achieved classification rate of 75%, the error in predicted CCI score is typically small with on average 92% of test cases being within 1 point of the correct CCI score. As was shown in figure 3.1, 1 point difference in the CCI score can translate to a significant difference is estimated survival rate, but we have to consider that without a system to identify comorbidity features, the CCI score for many patients will be missing or set to 0. In older patient populations with, on average, a larger number of comorbidities, such as the patient populations treated by the CvGT, setting many such 0 values will very likely be further from the truth than the CCI score predicted by our system.

We are significantly more cautious regarding the use of our classifier in clinical practice, and integration into information systems or electronic health records. The achieved 75% classification rate is likely insufficient for completing structured modalities or generating problem lists of individual conditions in the EHR, given that the chance of missing conditions, or more problematically assigning incorrect conditions is significant. Using the classifier in the EHR to provide some overall measure of comorbidity such as the aggregated CCI score or a color-coded warning system for high comorbidity is a more acceptable implementation.

8.3 Limitations

The presented work and solutions for comorbidity identification are subject to a number of limitations, arising from the assumptions and design decisions made during the course of this work. We would like to now discuss the most important of these limitations.

Generalizability of our solution

The main limitation of this work concerns the generalizability of the methods and solutions in this work. While our solution is theoretically applicable to a broader set of medical documents, we have restricted the application domain in two ways: firstly by constraining the patients in our dataset to elderly traumatic fracture patients, and secondly by constraining the used document types to emergency department intake notes.

The first constraints impacts the distribution of comorbid conditions on our dataset, which is evident from the complete absence of the *HIV/AIDS* category in our datasets, indicating a prevalence much lower than the ±1% prevalence in the general Dutch population[33]. It also impacts the choice of which conditions are considered relevant, the Charlson index is typically used for general elderly patient populations, other condition may be relevant in younger population or populations with certain specific index conditions. Note that while this first constraint affects the generalizability of specifically our final model, we believe that the overall approach does generalize to different patient populations, conditions and comorbdidity indices.

The second mentioned constraint is more impactful as to the overall approach. Emergency department notes are short and, as previously mentioned, have a somewhat consistent structure, style and use of language. Document from different disciplines and departments are likely structured differently, and may give rise to language-related issues that are not present or not evident in our dataset. Possible issues include: more widespread

use of informal descriptions, mixed interpretations for an abbreviation in a single document, or more complex and contextual sentence structures complicating keyword-based programmatic labeling.

Limited power of validation for rare classes

The prevalence of the rarest CCI categories in our dataset is such that there are only a handful of samples available for these classes. This limits the value of our validation for these categories for a number of reasons. Firstly, our dataset may not cover the full range of conditions in these categories. Secondly, a very low number of samples means that validation performance may be impacted significantly by the choice of cross-validation folds.

Post-hoc creation of queries

Another matter that should be noted is that the queries against the terminologies in the weak labeling approach were created after the authors had seen all documents in the dataset during manual annotation. We have attempted to keep the queries as general as possible by constructing the queries based on the annotation protocol as described in section 5.3.2, rather than on the observed samples and refraining from in-depth optimization of the weak labels against the hand created labels. However, we cannot fully guarantee that the queries have not been influenced by the fact that the authors observed the test samples. Ideally the proposed weak labeling approach should be evaluated by isolating the two matters, for example by having one person label test samples while a second person creates the queries based on a given protocol or a separate set of data.

8.4 Future work

Investigating the generalizability of our approach is in our opinion the main avenue for future research. Our work was limited to emergency department documents for fracture patients, we strongly suggest evaluating the approach based on a larger dataset of documents from different medical disciplines, as the use of a more broader set of documents may bring to light shortcomings of our approach and favour the use of different parameters or base models. We also believe that training and evaluating based on a larger dataset would significantly reduce the observed variability in results for rarer categories.

Application of our approach for a different target variable, for example the Elixhauser Comorbidity Index, or identification of entirely different clinical concepts, such as medications or procedures, may also be explored.

Another possibility for future work would be to investigate whether a self-training approach, in which a classifier is iteratively trained and used for pseudo-labeling, rather than pseudo-labeling occurring once based on a supervised classifier, could lead to performance gains.

From an applied point of view, we see opportunities for integrating a model trained according to our approach into the EHR. We noted in section 8.2 that given our obtained document-level accuracy it would be prudent to represent comorbidity as an aggregated score. Given that our choice of model, the random forest, is inherently interpretable, a score that can be linked to individual conditions mentioned in specific documents may also be possible. We also believe that an automated version of the introduced weak labeling pipeline could be useful, either within the context of another research project or as a tool

in its own right. We envision a tool with which people unfamiliar with SNOMED CT and/or query languages can label sets of documents by simply selecting the desired sets of conditions from a comprehensive menu.

Chapter 9

Conclusions

The overarching goal of this work was to design a machine learning treatment that could aid in the extraction of information regarding patient comorbidity from unstructured clinical text in support of clinical research and clinical decision making. Literature shows that the definition of the term "comorbidity" is complex and highly dependent on medical perspective and professional interpretation, thus the problem of extracting comorbidities could be as complex as identifying individual medical conditions in-text. We believe that shifting the problem from identifying single conditions to identifying broader categories from established comorbidity indices such as the Charlson Comorbidity Index is a good compromise, as this significantly reduces the problem dimensionality while maintaining a strong link with medical practice and research.

In the first part of this work we compared a number of machine learning methods for classifying clinical notes of geriatric hip fracture patients according to the Charslon Comorbidity index in a fully supervised scheme. We found that given sufficient labeled training data, both tree-based ensemble methods and transformer-based models show promising performance. We observed f_1 scores above 0.8 for our target classes with an occurrence rate over 5%, but for classes under this threshold performance decreased significantly. Overall classification accuracy is hampered by the fact that chance for prediction errors cascades over the 17 categories; all categories need to be prediction correctly for a correct classification. We found a best classification rate of 71% with the Random Forest model in this supervised learning scheme. We also explored the inherent structure of the clinical notes and the distribution of features over this structure. While we found that the documents exhibit a clear structure, and that much of the feature mass was concentrated in one section of the document, we were not able to incorporate this knowledge into our classification approach.

In the second part of this work we presented a weak-labeling approach that leverages existing medical terminologies. We used this approach to generate additional training data from clinical documents of elderly patients with traumatic fractures other than hip fracture. Our goal was to attempt to increase classification performance for classes under the previously mentioned 5% occurrence rate threshold. We found that augmenting our training data with these weakly supervised samples considerably increases performance for these rare CCI categories, however care should be taken to bridge the gap in language between medical terminologies and practice, we achieved this with a pseudo-labeling approach. Furthermore, issues arising due to the rule-based nature of the labeling approach, such as mislabeling ambiguous abbreviations, should be addressed. We observed increases

in the f_1 score of 0.05-0.35 for categories under the 5% threshold, as well as minor improvements for more common classes. Overall we found a best classification rate of 75% using the Random Forest model in the weakly supervised learning scheme.

9.1 Research Questions

RQ1 How can we design a machine learning solution for obtaining relevant comorbidities from clinical notes?
A: When framing the problem as a multi-label classification problem, both traditional machine learning approaches such as Random Forests and Gradient Boosting and more modern transformer-based models are very decent solutions, given sufficient labeled training data. However, due to the inherent imbalance in the prevalence of medical conditions, hand-labeling documents is prohibitively expensive if one wishes to gather sufficient positive samples for rarer comorbidities. By pairing these base classifiers with a weak-supervision scheme based on well-established medical terminologies and ontologies, it is possible to create classifiers that are suitable for and perform sufficiently well for research and, to a degree, for information management tasks.

The best performing approach in our study consisted of binary Random Forest classifiers for each category in the CCI, paired with the introduced weak labeling approach including all introduced refinements: active disambiguation of abbreviations, pseudo-labels based on a supervised classifier, and explicit exclusion of skin cancers.

(a) Which comorbid conditions are relevant?
A: In general the choice of conditions should be limited to chronic conditions that significantly impact patient risk or complexity of clinical management. The exact choice of conditions may vary dependent on the clinical context in which the solution or model is deployed, as different index conditions will consider different comorbid conditions relevant. In a general setting or setting with an index condition with few index-condition specific comorbidities, for example hip fractures, well established and scientifically validated comorbidity indices such as the Charlson Comorbidity index or the Elixhouser comorbidity index are excellent choices.

(b) How can the quasi-structure inherent to the clinical notes be leveraged for improving model performance?
A: While the clinical notes exhibit a clear structure, and much of the important information is concentrated in one section within that structure, we do not see clear avenues for using this structure in order to improve performance. One could imagine a solution in which the various sections are processed differently according to the patterns observed for these sections, such as the tendency to a list for the medical history section or the more narrative nature of the anamnesis. However, these patterns are not fully consistent, therefore it is dubious whether such a tailored approach would lead to a significant performance gain, especially given that application of the existing models to the full documents without pre-processing involving the note structure was shown to already have decent performance. Furthermore, the applicability of tailor-made solutions is limited to the domain or specific document type for which they were designed, in our case emergency department notes. A solution based on a classifier for

the document as a whole is more likely to generalize to, for example, notes from internal medicine. Also, we found that outside the medical history section, features are spread throughout the document fairly evenly, thus there is no way of trimming the documents without information loss.

RQ2 How can we leverage existing medical terminologies and ontologies in labeling sufficient training data?

(a) What are the limitations of using training data labeled using medical terminologies compared to handlabeled data?
A: The main limitations of training data labeled based on terminology systems arise from the language gap between those systems and the language used in practice. In practice, clinicians regularly use diagnostic terms and abbreviations that can not be found in medical terminology systems. This results in labeling mechanisms based on terminology systems missing occurrences of comorbid condition where such "non-standard" language is used by clinicians. Furthermore, labeling systems based on terminology systems can be prone to mislabeling if terminology is ambiguous, as was the case for abbreviation in this study, or when the terms extracted from the terminology system do not match the intended target diagnosis group well.

(b) How can we mitigate these shortcomings?
A: The language gap between practice and terminology systems can be overcome by including information derived from some amount of handlabeled data in the labeling and training process. We achieved this by augmenting the terminology-based weak labels with pseudo-labels generated by a supervised classifier. Other option may include the direct inclusion of handlabeled data in training, with oversampling applied for problematic classes, or maintaining a list of common "non-standard" terms.

RQ3 How will adopting elements from Agile methodologies impact the research process in terms of efficiency and effectiveness?
A: See appendix F.

Bibliography

[1] Emily Alsentzer et al. *Publicly Available Clinical BERT Embeddings*. 2019. arXiv: 1904.03323 [cs.CL]. URL: https://arxiv.org/abs/1904.03323.

[2] Aitziber Atutxa, Alicia Pérez, and Arantza Casillas. "Machine Learning Approaches on Diagnostic Term Encoding With the ICD for Clinical Documentation". In: *IEEE Journal of Biomedical and Health Informatics* PP (Aug. 2017), pp. 1–1. DOI: 10.1109/JBHI.2017.2743824.

[3] Autoriteit Consument & Markt. *ZIS/EPD-systemen: marktproblemen en oplossingsrichtingen*. Introductory research to Case nr: ACM/21/052741 Doc nr.: ACM/UIT/567852. The Netherlands, Dec. 20, 2021.

[4] Lisa Bastarache. "Using Phecodes for Research with the Electronic Health Record: From PheWAS to PheRS". In: *Annual Review of Biomedical Data Science* 4.1 (2021). PMID: 34465180, pp. 1–19. DOI: 10.1146/annurev-biodatasci-122320-112352. eprint: https://doi.org/10.1146/annurev-biodatasci-122320-112352. URL: https://doi.org/10.1146/annurev-biodatasci-122320-112352.

[5] Kent Beck et al. *Manifesto for Agile Software Development*. 2001. URL: http://www.agilemanifesto.org/.

[6] Jules J Berman. "Pathology abbreviated: a long review of short terms". In: *Archives of pathology & laboratory medicine* 128.3 (2004), pp. 347–352.

[7] G. Maarten Bonnema, Karel T. Veenvliet, and Jan F. Broenink. *Systems design and engineering: facilitating multidisciplinary development projects*. English. CRC Press (Taylor & Francis), 2016. ISBN: 978-1-4987-5126-1. DOI: 10.1201/b19135.

[8] Gavin C. Cawley and Nicola L. C. Talbot. "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation". In: *J. Mach. Learn. Res.* 11 (2010), pp. 2079–2107. URL: https://api.semanticscholar.org/CorpusID:1858029.

[9] Kathryn Annette Chapman and Günter Neumann. "Automatic ICD Code Classification with Label Description Attention Mechanism". In: *IberLEF@SEPLN*. 2020.

[10] Mary Charlson et al. "A new method of classifying prognostic comorbidity in longitudinal studies: development and validation." In: *Journal of chronic diseases* 40 5 (1987), pp. 373–83.

[11] N. Chawla et al. "SMOTE: Synthetic Minority Over-sampling Technique". In: *ArXiv* abs/1106.1813 (2002). URL: https://api.semanticscholar.org/CorpusID:1554582.

[12] James J. Cimino. "Desiderata for Controlled Medical Vocabularies in the Twenty-First Century". In: *Methods of Information in Medicine* 37 (1998), pp. 394–403. URL: https://api.semanticscholar.org/CorpusID:39094303.

[13] Danijela Ciric et al. "Agile Project Management beyond Software Development: Challenges and Enablers". In: *9th International Scientific and Expert Conference under the auspices of the International TEAM Society, 10–12th October 2018 University of Novi Sad Faculty of Technical Sciences Department of Industrial Engineering and Management*. May 2021.

[14] Martin Cowie et al. "Electronic health records to facilitate clinical research". In: *Clinical research in cardiology : official journal of the German Cardiac Society* 106 (Jan. 2017). DOI: 10.1007/s00392-016-1025-6.

[15] Pieter Delobelle, Thomas Winters, and Bettina Berendt. "RobBERT: a Dutch RoBERTa-based Language Model". In: (Nov. 2020), pp. 3255–3265. DOI: 10.18653/v1/2020.findings-emnlp.292. URL: https://www.aclweb.org/anthology/2020.findings-emnlp.292.

[16] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[17] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *North American Chapter of the Association for Computational Linguistics*. 2019. URL: https://api.semanticscholar.org/CorpusID:52967399.

[18] Richard A. Deyo, Daniel C. Cherkin, and Marcia Aparecida Ciol. "Adapting a clinical comorbidity index for use with ICD-9-CM administrative databases." In: *Journal of clinical epidemiology* 45 6 (1992), pp. 613–9.

[19] W. Alex Edmonds and Thomas D. Kennedy. "1: A Primer of the Scientific Method and Relevant Components". In: *An Applied Guide to Research Designs: Quantitative, Qualitative, and Mixed Methods*. Second. Thousand Oaks, California, June 2017. DOI: 10.4135/9781071802779. URL: https://doi.org/10.4135/9781071802779.

[20] Anne Elixhauser et al. "Comorbidity measures for use with administrative data." In: *Medical care* 36 1 (1998), pp. 8–27. URL: https://api.semanticscholar.org/CorpusID:29229635.

[21] Patricia Farrugia et al. "Practical tips for surgical research: Research questions, hypotheses and objectives". In: *Canadian journal of surgery. Journal canadien de chirurgie* 53 (Aug. 2010), pp. 278–81.

[22] Stephen Fortin, Jenna Reps, and Patrick Ryan. "Adaptation and validation of a coding algorithm for the Charlson Comorbidity Index in administrative claims data using the SNOMED CT standardized vocabulary". In: *BMC Medical Informatics and Decision Making* 22 (Oct. 2022). DOI: 10.1186/s12911-022-02006-1.

[23] Stephen Fortin, Jenna Reps, and Patrick Ryan. "Correction to: Adaptation and validation of a coding algorithm for the Charlson Comorbidity Index in administrative claims data using the SNOMED CT standardized vocabulary". In: *BMC Medical Informatics and Decision Making* 23 (June 2023). DOI: 10.1186/s12911-023-02205-4.

[24] Clara Franco et al. "Introducing ScrumAdemia: An Agile Guide for Doctoral Research". In: *PS: Political Science & Politics* 56.2 (2023), pp. 251–258. DOI: 10.1017/S1049096522001408.

[25] Chufan Gao et al. "Classifying Unstructured Clinical Notes via Automatic Weak Supervision". In: *Proceedings of the 7th Machine Learning for Healthcare Conference*. Ed. by Zachary Lipton et al. Vol. 182. Proceedings of Machine Learning Research. PMLR, May 2022, pp. 673–690. URL: https://proceedings.mlr.press/v182/gao22a.html.

[26] Christopher Harrison et al. "Comorbidity versus multimorbidity: Why it matters". In: *Journal of Multimorbidity and Comorbidity* 11 (Mar. 2021), p. 263355652199399. DOI: 10.1177/2633556521993993.

[27] O. Hazan and S. Toziq. *Agile Research*. URL: https://www.infoq.com/articles/agile-academic-research/. (accessed: 18.05.2023).

[28] Michael Hicks and Jeffrey Foster. "Adapting Scrum to Managing a Research Group". In: (Sept. 2010).

[29] Enric Senabre Hidalgo. "Adapting the scrum framework for agile project management in science: case study of a distributed research initiative". In: *Heliyon* 5 (2019).

[30] Enric Senabre Hidalgo. "Management of a Multidisciplinary Research Project: A Case Study on Adopting Agile Methods". In: *Journal of Research Practice* 14 (2018), p. 1.

[31] Enikő Ilyés. "Create your own agile methodology for your research and development team". In: *2019 Federated Conference on Computer Science and Information Systems*. Sept. 2019, pp. 823–829. DOI: 10.15439/2019F209.

[32] SNOMED International. *IHTSDO SNOMED CT browser*. URL: https://browser.ihtsdotools.org/?.

[33] L Kayaert et al. *Sexually transmitted infections in the Netherlands in 2022*. 2023. DOI: 10.21945/RIVM-2023-0161. URL: http://hdl.handle.net/10029/626792.

[34] Jinhyuk Lee et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4 (Sept. 2019). Ed. by Jonathan Wren, pp. 1234–1240. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btz682. URL: http://dx.doi.org/10.1093/bioinformatics/btz682.

[35] Dean Leffingwell. *Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise*. Addison-Wesley Professional, 2010. ISBN: 9780321635846.

[36] Irene Li et al. "Neural Natural Language Processing for Unstructured Data in Electronic Health Records: a Review". In: *Comput. Sci. Rev.* 46 (2021), p. 100511.

[37] Yinhan Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *ArXiv* abs/1907.11692 (2019). URL: https://api.semanticscholar.org/CorpusID:198953378.

[38] Justin Lovelace et al. "Dynamically Extracting Outcome-Specific Problem Lists from Clinical Notes with Guided Multi-Headed Attention". In: *Proceedings of the 5th Machine Learning for Healthcare Conference*. Ed. by Finale Doshi-Velez et al. Vol. 126. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 245–270. URL: https://proceedings.mlr.press/v126/lovelace20a.html.

[39] Chathuragnga Manamendra et al. "Improvements for agile manifesto and make agile applicable for undergraduate research projects". In: *Computer Science & Education (ICCSE), 2013 8th International Conference on*. Apr. 2013, pp. 539–544. ISBN: 978-1-4673-4464-7. DOI: 10.1109/ICCSE.2013.6553969.

[40] Michele Marchesi et al. “Distributed Scrum in Research Project Management”. In: *Agile Processes in Software Engineering and Extreme Programming, 8th International Conference, XP 2007*. June 2007, pp. 240–244. ISBN: 978-3-540-73100-9. DOI: 10.1007/978-3-540-73101-6_45.

[41] L. Melton et al. “Predictors of Excess Mortality After Fracture: A Population-Based Cohort Study”. In: *Journal of Bone and Mineral Research* 29 (July 2014). DOI: 10.1002/jbmr.2193.

[42] James Mullenbach et al. “Explainable Prediction of Medical Codes from Clinical Text”. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 1101–1111. DOI: 10.18653/v1/N18-1100. URL: https://aclanthology.org/N18-1100.

[43] *Neo4j*. URL: https://neo4j.com/.

[44] Kathryn Nicholson et al. “Multimorbidity and comorbidity revisited: Refining the concepts for international health research”. In: *Journal of Clinical Epidemiology* 105 (Sept. 2018). DOI: 10.1016/j.jclinepi.2018.09.008.

[45] *nictiz SNOMED-CT Browser*. URL: https://terminologie.nictiz.nl/art-decor/snomed-ct.

[46] W.S. Nijmeijer et al. “Prediction of early mortality following hip fracture surgery in frail elderly: The Almelo Hip Fracture Score (AHFS)”. In: *Injury* 47.10 (2016), pp. 2138–2143. ISSN: 0020-1383. DOI: https://doi.org/10.1016/j.injury.2016.07.022. URL: https://www.sciencedirect.com/science/article/pii/S002013831630300X.

[47] Jillian Oderkirk. “Readiness of electronic health record systems to contribute to national health information and research”. In: *OECD Health Working Papers* 99 (2017). DOI: https://doi.org/https://doi.org/10.1787/9e296bf3-en. URL: https://www.oecd-ilibrary.org/content/paper/9e296bf3-en.

[48] Olivier Paalvast et al. “Radiology report generation for proximal femur fractures using deep classification and language generation models”. English. In: *Artificial intelligence in medicine* 128 (June 2022). Publisher Copyright: © 2022 The Authors. ISSN: 0933-3657. DOI: 10.1016/j.artmed.2022.102281.

[49] Cristian Padurariu and Mihaela Elena Breaban. “Dealing with Data Imbalance in Text Classification”. In: *Procedia Computer Science* 159 (2019). Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 23rd International Conference KES2019, pp. 736–745. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2019.09.229. URL: https://www.sciencedirect.com/science/article/pii/S1877050919314152.

[50] Chao Pang et al. “CEHR-BERT: Incorporating temporal information from structured EHR data to improve prediction tasks”. In: *CoRR* abs/2111.08585 (2021). arXiv: 2111.08585. URL: https://arxiv.org/abs/2111.08585.

[51] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[52] Raphael Poulain, Mehak Gupta, and Rahmatollah Beheshti. "Few-Shot Learning with Semi-Supervised Transformers for Electronic Health Records". In: *Proceedings of the 7th Machine Learning for Healthcare Conference*. Ed. by Zachary Lipton et al. Vol. 182. Proceedings of Machine Learning Research. PMLR, May 2022, pp. 853–873. URL: https://proceedings.mlr.press/v182/poulain22a.html.

[53] Hude Quan et al. "Coding Algorithms for Defining Comorbidities in ICD-9-CM and ICD-10 Administrative Data". In: *Medical care* 43 (Dec. 2005), pp. 1130–9. DOI: 10.1097/01.mlr.0000182534.19832.83..

[54] David Brito Ramos et al. "On the use of Scrum for the management of research-oriented projects". In: *Nuevas Ideas en Informática Educativa, Volumen 12*. 2017.

[55] Simmi K. Ratan, Tanu Anand, and John Ratan. "Formulation of Research Question – Stepwise Approach". In: *Journal of Indian Association of Pediatric Surgeons* 24 (Jan. 2019), p. 15. DOI: 10.4103/jiaps.JIAPS_76_18.

[56] Alexander Ratner et al. *Data Programming: Creating Large Training Sets, Quickly*. 2017. arXiv: 1605.07723 [stat.ML]. URL: https://arxiv.org/abs/1605.07723.

[57] Alexander Ratner et al. "Snorkel: rapid training data creation with weak supervision". In: *Proceedings of the VLDB Endowment* 11.3 (Nov. 2017), pp. 269–282. ISSN: 2150-8097. DOI: 10.14778/3157794.3157797. URL: http://dx.doi.org/10.14778/3157794.3157797.

[58] Mihaela van der Schaar and Andrew Rashbass. *The case for reality-centric Ai // Van der Schaar lab*. Dec. 2023. URL: https://www.vanderschaar-lab.com/the-case-for-reality-centric-ai.

[59] Shijing Si et al. "Students Need More Attention: BERT-based Attention Model for Small Data with Application to Automatic Patient Message Triage". In: *Proceedings of the 5th Machine Learning for Healthcare Conference*. Ed. by Finale Doshi-Velez et al. Vol. 126. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 436–456. URL: https://proceedings.mlr.press/v126/si20a.html.

[60] *Snorkel*. URL: https://www.snorkel.org/.

[61] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 3319–3328. URL: https://proceedings.mlr.press/v70/sundararajan17a.html.

[62] *The Scrum Guide*. URL: https://www.scrum.org/resources/scrum-guide.

[63] C.G. Thomas. "1.8: The Scientific Method". In: *Research Methodology and Scientific Writing*. Springer International Publishing, 2021. ISBN: 9783030648640. URL: https://books.google.nl/books?id=XVX6zQEACAAJ.

[64] *UMLS Metathesaurus Browser*. URL: https://uts.nlm.nih.gov/uts/umls/home.

[65] JM Valderas et al. "Defining Comorbidity: Implications for Understanding Health and Health Services". English. In: *Annals of Family Medicine* 7 (July 2009), pp. 357–363. ISSN: 1544-1709. DOI: 10.1370/afm.983.

[66] Maurice Van Keulen et al. "Exploiting Natural Language Processing for Improving Health Processes". English. In: *Proceedings of the 7th International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA 2017)*. Ed. by Paolo Ceravolo, Maurice van Keulen, and Kilian Stoffel. CEUR Workshop Proceedings. 7th IFIP WG 2.6 International Symposium on Data-Driven Process Discovery and Analysis, SIMPDA 2017, SIMPDA ; Conference date: 06-12-2017 Through 08-12-2017. CEUR, Dec. 2017, pp. 145–146. URL: http://simpda2017.di.unimi.it/.

[67] C. Van Wyngaard, Jan-Harm Pretorius, and Leon Pretorius. "Theory of the triple constraint — A conceptual review". In: *2012 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. Dec. 2012, pp. 1991–1997. ISBN: 978-1-4673-2945-3. DOI: 10.1109/IEEM.2012.6838095.

[68] Ashish Vaswani et al. "Attention is All you Need". In: *Neural Information Processing Systems*. 2017. URL: https://api.semanticscholar.org/CorpusID:13756489.

[69] Stella Verkijk and Piek Vossen. "MedRoBERTa.nl: A Language Model for Dutch Electronic Health Records". In: *Computational Linguistics in the Netherlands Journal* 11 (Dec. 2021), pp. 141–159. URL: https://www.clinjournal.org/clinj/article/view/132.

[70] Benjamin Viernes et al. *SNOMED CT Disease Hierarchies and the Charlson Comorbidity Index (CCI): An analysis of OHDSI methods for determining CCI.*

[71] B. C. S. de Vries et al. "Comparing three machine learning approaches to design a risk assessment tool for future fractures: predicting a subsequent major osteoporotic fracture in fracture patients with osteopenia and osteoporosis". English. In: *Osteoporosis international* 32.3 (Mar. 2021). Publisher Copyright: © 2021, International Osteoporosis Foundation and National Osteoporosis Foundation., pp. 437–449. ISSN: 0937-941X. DOI: 10.1007/s00198-020-05735-z.

[72] Wietse de Vries et al. *BERTje: A Dutch BERT Model*. arXiv:1912.09582. Dec. 2019. URL: http://arxiv.org/abs/1912.09582.

[73] Bill Wake. *XP123*. URL: https://xp123.com/. (accessed: 10.07.2023).

[74] Roel Wieringa. *Design Science Methodology for Information Systems and Software Engineering*. Jan. 2014, pp. 1–332. ISBN: 978-3-662-43838-1. DOI: 10.1007/978-3-662-43839-8.

[75] Shijie Wu et al. *BloombergGPT: A Large Language Model for Finance*. 2023. arXiv: 2303.17564 [cs.LG].

[76] Huan Xu, Peter D. Stetson, and Carol Friedman. "A Study of Abbreviations in Clinical Notes". In: *AMIA ... Annual Symposium proceedings. AMIA Symposium* (2007), pp. 821–5. URL: https://api.semanticscholar.org/CorpusID:4010681.

[77] Keyang Xu et al. "Multimodal Machine Learning for Automated ICD Coding". In: *Proceedings of the 4th Machine Learning for Healthcare Conference*. Ed. by Finale Doshi-Velez et al. Vol. 106. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 197–215. URL: https://proceedings.mlr.press/v106/xu19a.html.

[78] Berk Yenidogan et al. "Multimodal Machine Learning for 30-Days Post-Operative Mortality Prediction of Elderly Hip Fracture Patients". In: *IEEE International Conference on Data Mining Workshops, ICDMW*. Dec. 2021, pp. 508–516. DOI: 10.1109/ICDMW53433.2021.00068.

[79] ZGT. *ZGT-Traumatology*. Web. URL: https://www.zgt.nl/aandoening-en-behandeling/onze-specialismen/wetenschap/wetenschappelijk-onderzoek/medische-disciplines/traumatologie/.

[80] Zhi-Hua Zhou. "A brief introduction to weakly supervised learning". In: *National Science Review* 5 (2018), pp. 44–53. URL: https://api.semanticscholar.org/CorpusID:44192968.

Appendix A

Identified Headings

Header	# occurrences
anamnese	1990
aanvullend onderzoek	1896
lichamelijk onderzoek	1827
diagnose	1827
laboratorium onderzoek	1309
medicatie	1184
beleid/therapie	1070
medische geschiedenis	941
hoofdklacht	932
beleid / therapie	798
vitale functies	653
allergie	432
vervolgbeleid	374
thuismedicatie	255
voorgeschiedenis	211
vg	100
beleid	65
lab	64
reden consult	54
naam behandelaar	51
ecg	42
allergieën	40
radiologie	37

Table A.1: 20 most common headers

Header	English translation
voorgeschiedenis|vg|medische geschiedenis	medical history
aanvullend onderzoek	additional examination
lichamelijk onderzoek	physical examination
anamnese	anamnesis
hoofdklacht|reden consult	chief complaint
diagnose	diagnosis
laboratoriumonderzoek|lab	laboratory results
medicatie|thuismedicatie	medication
beleid/therapie|beleid / therapie|vervolgbeleid|beleid	policy/therapy
vitale functies	vitals
allergie|allergieën	allergies
ecg	ECG (electrocardiogram)
radiologie	radiology

TABLE A.2: Identified headings and translations

Appendix B

Additional CCI inclusions

B.1 Hemiplegia / Paraplegia

Conditions

- hemi-/paraparese

B.2 Peripheral Vascular Disease

Conditions

- chronische veneuze insufficiëntie
- spataderen / varices
- tromboflebitis
- trombose (peripheral), including:
  - dvt
- ulcus criris

Procedures

- fem-pop
- broekprothese
- thoracic endovascular aortic repair (TEVAR)

B.3 Metastatic Solid Tumour

n/a

B.4 Dementia

n/a

B.5 Renal Disease

Conditions

- nefrotisch syndroom
- schrompelnier
- cystenieren

Procedures

- dialyse
- nefrectomie

B.6 Myocardial Infarction

Conditions

- acuut coronair syndroom (acs)

B.7 Malignancy, Except Skin Neoplasms

Conditions

- myelodysplastisch sydroom, including
  - RARS

Procedures

- prostatectomie

B.8 Chronic Pulmonary Disease

Conditions

- longembolieën
- small airway disease
- atelectase
- pleurale afwijkingen door asbestcontact
- respiratoire insufficiëntie
- restrictieve longfunctiestoornissen, including
  - longfibroze

B.9 Mild Liver Disease

Conditions

- hepatomegalie

B.10 AIDS / HIV

n/a

B.11 Congestive Heart Failure

Procedures

- PTCA/PCI
- CABG

B.12 Peptic Ulcer Disease

Conditions

- ulceratieve gastritis
- ulceratieve bulbitis
- cascademaag
- zollinger-ellison syndroom

B.13 Cerebrovascular Disease

Procedures

- carotis- endarteriëctomie (CEA)

B.14 Moderate / Severe Liver Disease

n/a

B.15 Diabetes, With Chronic Complications

n/a

B.16 Diabetes, Without Chronic Complications

n/a

B.17 Rheumatic Disease

Conditions

- jicht
- pseudojicht
- jeugdreuma
- oligoarthritis
- ziekte van Bechterew

Appendix C

Terminology Queries

This appendix provides the full SNOMED CT ECL queries for the definitions of the Charlson Comorbidity Index categories used in this work. The equivalent Neo4j Cypher queries, as well as instruction for loading SNOMED CT into Neo4j can be found on the authors Github page.[1]

C.1 Hemiplegia / Paraplegia

```
<<372310001 |Paralysis due to lesion of spinal cord (disorder)|
OR
<<192970008 |Cauda equina syndrome (disorder)|
OR
(
<<29426003 |Paralytic syndrome (disorder)|:
    ((
    [0..0] 363698007 |Finding site (attribute)| =
        (
                <<49549006 |Structure of visual system (body
                    structure)|
                OR <<89837001 |Urinary bladder structure (body
                    structure)|
                OR <<25238003 |Cranial nerve structure (body
                    structure)|
        )
    )
        AND [0..0] 371881003 |During (attribute)| = 236973005 |
            Delivery procedure (procedure)|
        AND [0..0] 246454002 |Occurrence (attribute)| = 255407002
            |Neonatal (qualifier value)|
    )
)
```

C.2 Peripheral Vascular Disease

[1] https://github.com/SylvainBrouwer/neo4j-snomed-cci

```
(
<<27550009 |Disorder of blood vessel (disorder)|
MINUS
(
        (
        <<404684003 |Clinical finding (finding)|:
                116676008 |Associated morphology (attribute)| =
                        (
                        <<12856003 |Uneven venous ectasia (
                           morphologic abnormality)|
                        OR
                        50960005 |Hemorrhage (morphologic
                           abnormality)|
                        )
        )
        OR
        (
        <<404684003 |Clinical finding (finding)|:
                363698007 |Finding site (attribute)| =
                        (
                        <<299717005 |Structure of carotid and/or
                           cerebral and/or subclavian artery (
                           body structure)|
                        OR
                        <<281232002 |Vascular structure of head
                           and/or neck (body structure)|
                        OR
                        <<846601002 |Structure of blood vessel of
                            thoracic cross-sectional segment of
                           trunk (body structure)|
                        OR
                        15825003 |Aortic structure (body
                           structure)|
                        )
        )
)
)
OR
<<63491006 |Intermittent claudication (finding)|
OR
(
(
<<71388002 |Procedure (procedure)|:
        {
        405813007 |Procedure site - Direct (attribute)| =
                (
                (
                <<306954006 |Regional blood vessel structure (
                   body structure)|
```

```
                OR
                <<51833009 |Peripheral vascular system structure
                    (body structure)|
                )
                MINUS
                (
                <<299717005 |Structure of carotid and/or cerebral
                     and/or subclavian artery (body structure)|
                OR
                <<281232002 |Vascular structure of head and/or
                    neck (body structure)|
                OR
                <<846601002 |Structure of blood vessel of
                    thoracic cross-sectional segment of trunk (
                    body structure)|
                OR
                15825003 |Aortic structure (body structure)|
                )
                ),
        260686004 |Method (attribute)| =
                <<257903006 |Repair - action (qualifier value)|
        }
)
MINUS
(
<<71388002 |Procedure (procedure)|:
        405813007 |Procedure site - Direct (attribute)| =
                (
                <<299717005 |Structure of carotid and/or cerebral
                     and/or subclavian artery (body structure)|
                OR
                <<281232002 |Vascular structure of head and/or
                    neck (body structure)|
                OR
                <<846601002 |Structure of blood vessel of
                    thoracic cross-sectional segment of trunk (
                    body structure)|
                OR
                15825003 |Aortic structure (body structure)|
                )
)
)
OR
5431005 |Percutaneous transluminal angioplasty (procedure)|
```

C.3 Metastatic Solid Tumour

```
<<14799000 |Neoplasm, metastatic (morphologic abnormality)|
```

```
OR
(
<<404684003 |Clinical finding (finding)|:
    116676008 |Associated morphology (attribute)| =
        <<14799000 |Neoplasm, metastatic (morphologic abnormality
            )|
)
```

C.4 Dementia

```
<< 52448006 |Dementia (disorder)|
```

C.5 Renal Disease

```
(
<<90708001 |Kidney disease (disorder)|
MINUS
<<79131000119100 |Kidney lesion (disorder)|
)
OR
(
<<71388002 |Procedure (procedure)|:
    363702006 |Has focus (attribute)| =
        <<90708001 |Kidney disease (disorder)|
)
OR
<<175905003 |Total nephrectomy (procedure)|
```

C.6 Myocardial Infarction

```
<<22298006 |Myocardial infarction (disorder)|
OR
<<413439005 |Acute ischemic heart disease (disorder)|
```

C.7 Malignancy, Except Skin Neoplasms

```
<<363346000 |Malignant neoplastic disease (disorder)|:
    (
    [0..0] 363698007 |Finding site (attribute)| =
        <<39937001 |Skin structure (body structure)|
    AND
    [0..0] 116676008 |Associated morphology (attribute)| =
        <<14799000 |Neoplasm, metastatic (morphologic abnormality
            )|
    )
```

C.8 Chronic Pulmonary Disease

```
<<17097001 |Chronic disease of respiratory system|
OR
<<24417004 |Environmental lung disease (disorder)|
OR
<<59282003 |Pulmonary embolism (disorder)|
OR
<<195967001 |Asthma (disorder)|
```

C.9 Mild Liver Disease

```
(
<<235856003 |Disorder of liver (disorder)|
OR
<<13920009 |Hepatic encephalopathy (disorder)|
OR
<<75183008 |Abnormal liver function (finding)|
OR
82403002 |Cholangitis (disorder)|
)
MINUS
(
<<59927004 |Hepatic failure (disorder)|
OR
<<93870000 |Malignant neoplasm of liver (disorder)|
)
```

C.10 AIDS / HIV

```
<<19030005 |Human immunodeficiency virus (organism)|
OR
(
<<404684003 |Clinical finding (finding)|:
    246075003 |Causative agent (attribute)| =
        <<19030005 |Human immunodeficiency virus (organism)|
)
```

C.11 Congestive Heart failure

```
<<84114007 |Heart failure (disorder)|
OR
(
<<71388002 |Procedure (procedure)|:
        405813007 |Procedure site - Direct (attribute)| =
        <<41801008 |Coronary artery structure (body structure)|,
```

```
        260686004 |Method (attribute)| =
        <<257903006 |Repair - action (qualifier value)|
)
```

C.12 Peptic Ulcer Disease

```
<<13200003 |Peptic ulcer (disorder)|
OR
54051005 |Cascade stomach (disorder)|
```

C.13 Cerebrovascular Disease

```
<<62914000 |Cerebrovascular disease (disorder)|
OR
(
<<404684003 |Clinical finding (finding)|:
    255234002 |After (attribute)| =
        <<62914000 |Cerebrovascular disease (disorder)|
)
OR
(
<<404684003 |Clinical finding (finding)|:
    42752001 |Due to (attribute)| =
        <<62914000 |Cerebrovascular disease (disorder)|
)
OR
<<1386000 |Intracranial hemorrhage (disorder)|
OR
<<66951008 |Carotid endarterectomy (procedure)|
```

C.14 Moderate / Severe Liver Disease

```
<<59927004 |Hepatic failure (disorder)|
OR
<<34742003 |Portal hypertension (disorder)|
OR
<<91109007 |Gastric varices (disorder)|
OR
<<28670008 |Esophageal varices (disorder)|
```

C.15 Diabetes, With Chronic Complications

```
<<404684003 |Clinical finding (finding)|:
        42752001 |Due to (attribute)| =
        <<73211009 |Diabetes mellitus (disorder)|
```

C.16 Diabetes, Without Chronic Complications

```
<<73211009 |Diabetes mellitus (disorder)|
```

C.17 Rheumatic Disease

```
(
<<85828009 |Autoimmune disease (disorder)|:
        363698007 |Finding site (attribute)| =
                21793004 |Connective tissue structure (body
                    structure)|
)
OR
(
<<404684003 |Clinical finding (finding)|:
        42752001 |Due to (attribute)| =
                55464009 |Systemic lupus erythematosus (disorder)
                    |
)
OR
<<3723001 |Arthritis (disorder)|
OR
<<400130008 |Temporal arteritis (disorder)|
OR
<<52661003 |Extra-articular rheumatoid process (disorder)|
OR
<<396230008 |Dermatomyositis (disorder)|
OR
<<31384009 |Polymyositis (disorder)|
OR
<<65323003 |Polymyalgia rheumatica (disorder)|
OR
<<276657008 |Overlap syndrome (disorder)|
```

Appendix D

Comparison of Transformer Variants

We compared the performance of five BERT and RoBERTa variants in a fully supervised setting. The chosen variants are:

- MedRoBERa.nl[69]
- BERTje[72]
- RobBERT[15]
- ClinicalBERT[1]
- Multilingual BERT[16]

Figure D.1 shows the average per-class f_1 scores for the tested models.

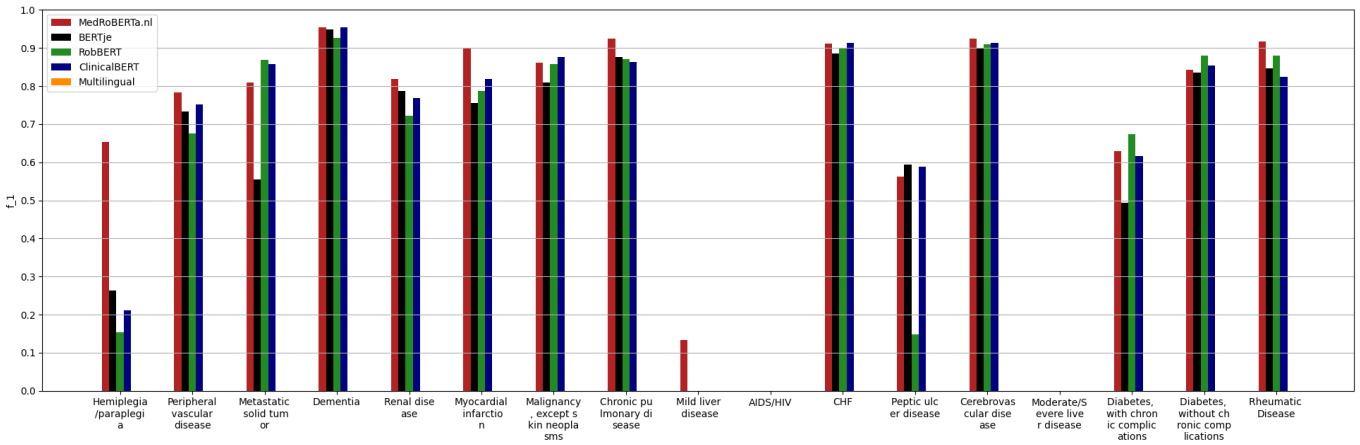


FIGURE D.1: BERT/RoBERTa variants

Appendix E

Label model performance

Table E.1: Performance metrics for weak labeling stage as a rule-based model.

Category	Occurrence rate	Precision	Recall
Cerebrovascular disease	0.188	0.92	0.85
Dementia	0.170	0.97	0.82
Congestive heart failure	0.153	0.94	0.88
Malignancy, except skin neoplasms	0.146	0.92	0.75
Diabetes, without chronic complications	0.147	0.79	0.93
Chronic pulmonary disease	0.136	0.84	0.88
Peripheral vascular disease	0.121	0.83	0.64
Renal disease	0.089	0.84	0.61
Rheumatic disease	0.086	0.82	0.79
Myocardial infarction	0.078	0.93	0.78
Diabetes, with chronic complications	0.047	0.88	0.29
Hemiplegia / paraplegia	0.024	0.81	0.31
Metastatic solid tumor	0.020	0.92	0.89
Peptic ulcer disease	0.020	1.0	0.64
Mild liver disease	0.009	0.69	0.67
Moderate / severe liver disease	0.003	1.0	0.70
AIDS / HIV	0.000	-	-

Appendix F

Agile: In depth treatment and conclusions

F.1 Introduction

At the start of this thesis project, Ziekenhuisgroep Twente (ZGT) outlined a singular goal, namely to work towards a solution for extracting comorbidities from clinical documentations. The desire for such a solution was driven by practical needs: the need for comorbidities as inputs for further clinical research, and the augmentation of structured information in ZGTs electronic health record. While some details, such as the dataset to be used, were implicit from the project context, i.e. the ongoing work regarding post-operative mortality prediction for elderly hip fracture patients, the start of this project was mostly a blank sheet; very few requirements for the solution were laid out, ZGT did not pose any prior research questions on the topic, and there was no inkling as to what a suitable solution could look like. As a result this project was mostly design-focused, contained a significant exploratory aspect. This is not necessarily problematic for scientific research, but we found that it complicates organizational aspects and management of research. We initially found it difficult to define focused research questions, we had no idea of what problems we would encounter, and thus could not clearly outline any future steps, experiments, or necessary elements for a solution. This made following a traditional approach traditional approach to structuring a research project rather difficult.

In an attempt to overcome the mentioned difficulties and to facilitate the exploratory nature thesis project, we experimented with an iterative research approach inspired by Agile practices from the software development industry. We used the this project on comorbidity identification as a case study for trying out the approach. This appendix lays out our motivation for borrowing from agile practices, our approach and our reflections. In this case study we ask the following research question:

RQ1 How will adopting elements from Agile methodologies impact the research process in terms of effectiveness and efficiency?

F.2 Background

F.2.1 The Iron Triangle and Scope

A core concept in project management is that of the *Iron Triangle*, or *Triple Constraint*, as shown in Figure F.1. The central idea behind this concept is that project delivery

and quality is mainly constrained by three factors: scope, resources and time; these three constraints are intertwined - changing one of the constraints affects the others [67]. For example: bringing a project deadline forward and thus reducing the time constraint requires that the scope of the project is reduced, that more resources are allocated (human, financial, equipment), or both in order to maintain project quality.

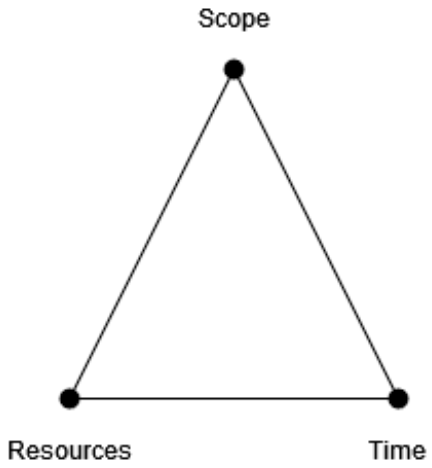


FIGURE F.1: The Iron Triangle

In a given project, these constraints can be either fixed or have some degree of flexibility. It is common for one of the three constraints to be fixed, in that case this constraint is the driver constraint of the project. A driver time constraint manifests itself as a short inflexible deadline, a driver resource constraint as a tight budget. Good examples of projects with scope as a primary driver are construction projects like building a bridge or a road, as these have rigid pre-defined requirements. In some cases two constraints may be fixed, a small-scale example of this is student projects within the context of a course, these often have a fixed deadline and limited human, financial and equipment resources, thus care needs to be taken in managing the only flexible variable — scope — such that the project can be completed. Attempting to fix all three requirements of a project up-front is generally ill-advised, unless the set scope is very limited or the fixed resources and time substantial.

Traditionally, research projects have a large focus on pre defining the scope of the research, this is intimately tied to the process of defining research questions, which is vital step in the research process [21][55]. Well-defined research questions frame the project and inform the methodology and research design. Note that even when research scope is predefined in this manner it is not necessarily fully fixed, in more complex real-world projects a set of research questions may be altered during the process based on initial experimental results and insights. Naturally real-world research project may also deal with fixed time and resource constraints, in these cases one can define a desired scope and simply "see what gets done". Alternatively one can choose an approach where a minimal scope is defined and expanded continually.

The proposed project on extracting comorbidities is in fact fixed in time and resources. As the desired output of the project is not just the answer to a research question, but

a artifact or model that can extract comorbidities from clinical notes, framing this as a project with a pre-defined desired scope requires us to design an "ultimate" solution beforehand. This is not possible as unforeseen issues related to the problem at hand may arise during the research process, and new insights into designing a better solution may be gained. We thus feel the best way of approaching this project is with a continually expanding scope. In short, one can say that this project is variable in scope, and fixed in resources and time.

F.2.2 Research Design

Traditional research design aligns closely with the practice of pre-defining scope through research questions that was mentioned in section F.2.1; research is often designed in a top-down, waterfall like manner: based on a problem or knowledge gap, research questions are defined, these then inform a methodology which is implemented by a (set of) experiments. The entire research life-cycle can be imagined as a V-model, as shown in Figure F.2: experimental results are discussed and linked back to the methodology to discuss potential limitations, conclusions relating to the research question are then drawn, and these conclusion together form a contributions to the problem domain [19][63]. Note that the V-model constitutes a single pass through the scientific method for a set of clearly defined research questions. As mentioned in section F.2.1, in real-world research projects the set of research questions may evolve during the execution of the project, however for each individual research question the V-model still applies, we can thus imaging such a project consisting of multiple sequential or staggered V-models.

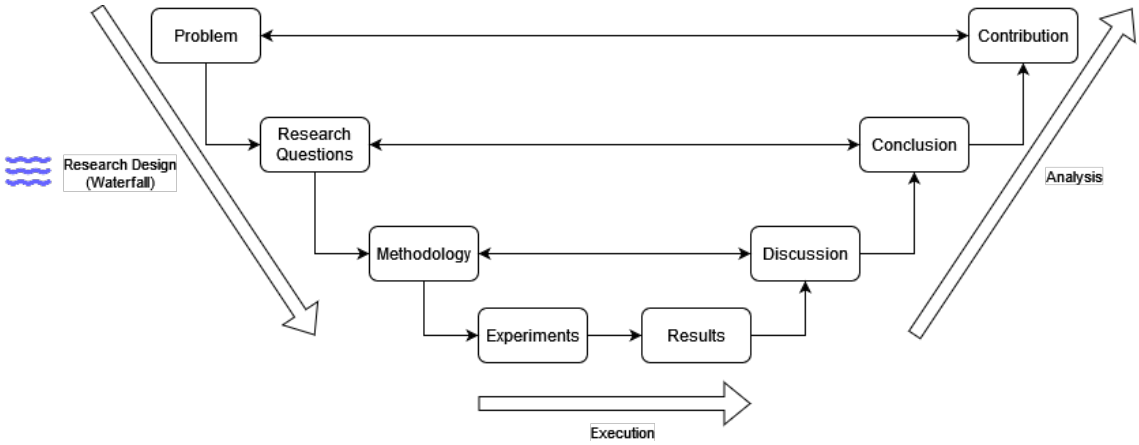


FIGURE F.2: V-model for question-based research, adapted from [7].

F.2.3 Why Choose an Agile Methodology?

A methodology based on Agile principles was chosen because we see parallels between the dichotomy of top-down and iterative research design, as mentioned in section F.2.2, and the evolution of software design methodologies. The early days of the software industry was dominated by a waterfall model, the assumption was that strict software requirements could be scoped during contract negotiation with clients, and based on these requirements project cost and schedule could be estimated. This assumption was flawed when it came to software development, as requirements could be misunderstood or change with time,

thus leading to significant percentages of projects failing [35]. Agile is a set of methodologies that were created as reaction to the failings of the waterfall model, the overarching beliefs of these methodologies were defined in the Agile Manifesto in 2001 [5]. Agile shifts the focus from plan- and process-driven development to responsive, collaborative and teamwork-driven development. By working in short time-boxed iterations Agile flips the triple constraint compared to the waterfall model: as illustrated in Figure F.3 scope is now the estimated variable within a fixed-time and fixed-resource iteration, one is able to estimate score for individual iterations, and adapt the scope of the entire project based on the results achieved within an iteration[35].

We propose that given the variable-scope nature of the thesis project, Agile practices will offer a better approach to executing this project as compared to a traditional waterfall interpretation of the scientific method.

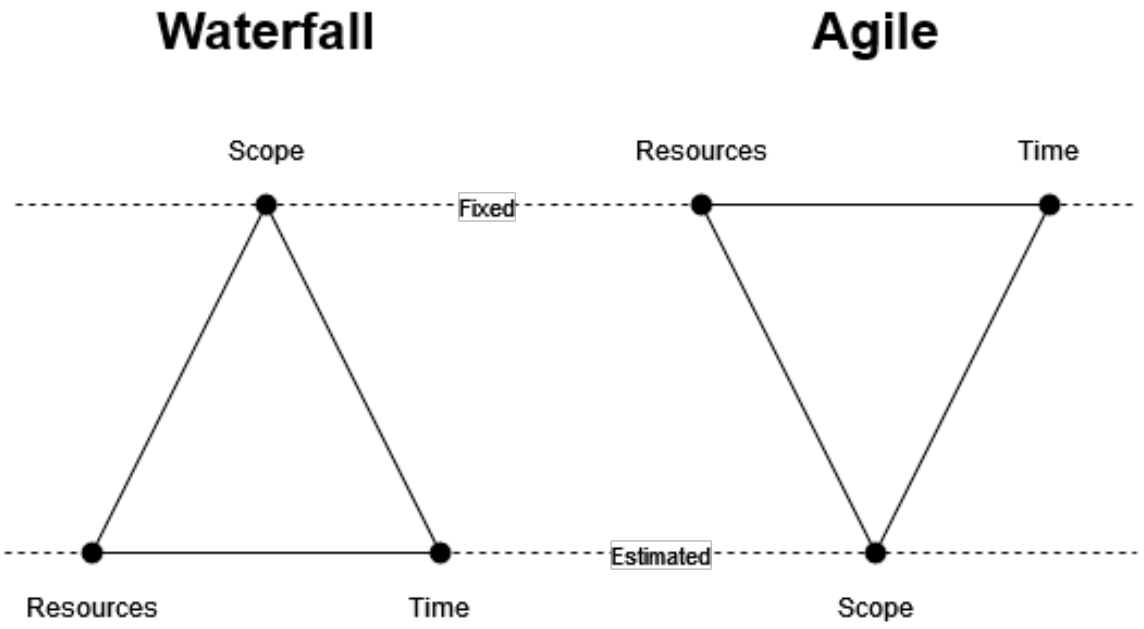


FIGURE F.3: Triple constraint: Waterfall vs. Agile [35]

F.3 Literature

F.3.1 Design science research

In his work *Design Science Methodology*[74], prof. R.J, Wieringa extensively discusses the notion of design-focused or design science research, describing it as projects in where "the object of study is an artifact in context, and its two major activities are designing and investigating this artifact in context". Wieringa also notes that in design science research, there are two types of research problems: *design problems*, the requirement for some solution in the real world, and *knowledge questions*, which aim to bridge some knowledge gap. While these concepts are closely knit, as attempting to solve a design problem may create new knowledge questions and vice versa, Wieringa notes that they require fundamentally different approaches. Design problems are solved through an "engineering cycle" of investigation, design, validation and implementation steps, while knowledge questions are

approached analytically or based on experimental validation. The overall process for design science research is therefore an iteration over design problems and knowledge questions, where the various iterations of the design problem and answers to knowledge questions are stepping stones towards a ultimate "treatment" for the problem at hand.

When we evaluate the project in this case study through the lens of design science research, we can see that we were initially presented with a design problem - design a system for extracting comorbidities from clinical documents - but no knowledge questions. This may explain why we initially found it difficult to define research questions and experiments, as these are scientific tools more suited to answering empirical knowledge questions. While Wieringa does not prescribe a specific execution process for the engineering cycle, he does mention that the process is inherently iterative, and that a waterfall-approach is only suitable if it is known in advance that only a single iteration will be performed. We believe this supports our choice for an alternative, iterative approach to the project in this case study.

F.3.2 Application of Agile in research

In the two decades since the formulation of the Agile Manifesto, Agile practices have found adoption in projects outside software development [13]. However literature on the adoption of Agile practices in a research context is rather sparse. We do find some adoption in research which has a core software development aspect or as a part of an informatics program [31][54][39], though we also see some adoption in a multidisciplinary and broader research context[40][30][29]. Another identified use-case is student management and supervision [28][24]. A consistent thread throughout adaptation of Agile in a research context is a strong focus on the social and teamwork aspect of Agile rather than on incremental delivery and responsiveness. Improvements in communication and interpersonal dynamics is often reported, as well as improvements in workflow and productivity. The lack of focus on incremental methodology often comes with a loose adherence to some agile practices, ceremonies like regular Scrum meetings, retrospectives and reviews are common however aspects like backlog management and careful iteration planning are not as common.

F.4 Methodology: Research Process for this Project

As we noted at the end of section F.2.2, we believe this project would benefit from an incremental approach built upon an expanding scope and frequent feedback. We believe that Agile, a set of practices form software development is a good fit for this project. This section discusses which Agile concepts and practices we plan on using, as well as how we conceptualize these in the context of this project, at the end of this section a short overview of the intended research process is given.

F.4.1 Sprint

The sprint is the core component of Agile, it is a time-boxed, self-contained iteration in which a development teams aims to achieve a specific, limited set of goals. The overarching idea is that at the end of each iteration a self contained increment of the project should be completed. At the end of a sprint the output product is reviewed (Sprint Review), subsequently the sprint process is reviewed as well (Sprint Retrospective). We aim to adhere to this definition of a sprint fairly strictly, and propose structuring sprints in the following manner:

- Sprints consist of a are a three week time-box.
- Once a week an in-person status-meeting with a supervisor will take place. If deemed necessary intermediate status meetings can be performed digitally (e-mail).
- Given the 1-person size of the "development team", as well as limited number of stakeholders we propose merging sprint review, retrospective and planning into a single meeting.

F.4.2 Research Questions as Features

Software *features* are services or behaviour provided by a system to meet a stakeholder need, typically this is higher-level or more abstract system behaviour. Any software system can be described in terms of its set of features, therefore they serve as excellent milestones for Agile development teams to work towards and are the main deliverables in Agile release planning. Agile teams are often feature-oriented, as features represent a vertical slice of a full software solution this means that teams are responsible for the entire technology stack in delivering their feature.[35]

We think that in a research setting, research questions are the correct analogue for features under development. As with describing software by its features we can describe any piece of completed research by a set of research questions with associated answers and evidence. Furthermore, a research question with answer and evidence is a vertical slice of a full research product as while there may be some overlapping steps in answering multiple research questions, a single research question can stand on its own.

F.4.3 Stories

In Agile development system features are subdivided into smaller elements of functionality, these elements are called *user stories*. Small features may consist of a single user story, but they are often composed of multiple. These user stories are the main artifact used for identifying system behaviour and value for stakeholders; the finer-grained nature of user stories compared to features allows for discussion about requirements between developers and customers. Still, a user story should not prescribe requirements for system behaviour, it serves as a placeholder for these requirements and finer details to be discussed and developed. Usually a user story is stated in the form:

As a [role], I can [activity], so that [value].[35]

Unlike software development research is not generally done with a "user" perspective in mind, therefore we will simply refer to the user story analogue in research as a *story*. For the same reason we find the *[role]* aspect of the user story statement not to be very important in research, however we think that the statement of intent in the sentence is important, as it requires the researcher to assign a purpose to any action they perform. We thus propose that any activities that are performed are first formulated in the following story form:

[research activity], so that [value].

Here the *[research activity]* can be any research task: literature review of a particular topic, implementing an additional method or experiment, labeling a dataset etc.. The

[value] would be any contribution towards answering a research question or delivering a complete research product, for example: theoretically supporting a method used, or obtaining results relevant to answering (part of) a research question.

The *INVEST* acronym is often used to describe the citeria for a good user story, which are as follows: [73]

- **I**ndependent - The story should be able to be completed on its own and deliver value independent from other stories. It should be kept in mind that the independent user story is an ideal and that it is nearly impossible to remove all dependencies; in software development some functionality may be built on top of other functionality and in research a next step may be conditional on the outcome of an experiment. Still, dependencies should be kept to a minimum, and if any do exist they should be sequential and the order op execution should be obvious.
- **N**egotiable - As previously mentioned a stories should not be prescriptive; details regarding methods, tools and implementations should be negotiated with supervisors and any other members of the research team. This criterion is natural to a collaborative research setting, but in order to maintain flexibility in regard to the research direction care should still be taken not to detail too much in advance. Ideally story detail should be worked out in a just-in-time manner.
- **V**aluable - In software development this criterion is tied to the user story statement, it reinforces the fact that a story should deliver some concrete business value to a project stakeholder, the *[role]* in the user story statement, by delivering (part of) a feature. It may seem that this criterion needs to be redefined for research as the *[role]* aspect of user stories has been eliminated, however given that we've framed research questions as features the original definition still holds. We consider a story to be valuable if it contributes towards answering a research question, either by having an output that offers insight or by facilitating the research process. The argument for allowing facilitating stories is that Agile in software development allows for analogous "Technical User Stories" which do not contribute towards functionality, but improve the development process or some nonfunctional requirement, examples of this are major code refactors and component upgrades.
- **E**stimable - A research team should be able to estimate the complexity and work required for completion of a story.
- **S**mall - The story should be able to be completed within an iteration.
- **T**estable - In software development this criterion refers to the fact that all created software artifacts should be tested by comprehensive unit and feature tests. Code is not considered completed if it hasn't been tested, and stories that cannot be tested are consider ill-defined. In research one may deal with stories that don't describe functionality, for example stories relating to literature research. In order to accomodate these types of stories we suggest broadening this criterion to *Evaluable*. All stories need a concrete output that can be evaluated in some fashion: for example, a literature research story has a paragraph or chapter of writing as an output which can be evaluated with regard to its quality and relevance, and an experiment has results and an associated discussion as an output which can again be evaluated.

Examples of user stories:

- Implement classifier X, **so that** comparative performance with our baseline can be obtained.

- Research label embedding techniques, **so that** we know whether it can be incorporated in classifier X.
- Re-evaluate performance of classifier X with section type Y removed from dataset, **so that** influence of section type Y on performance can be evaluated.

F.4.4 Backlog

We propose using a backlog in a standard Agile manner, this means that two separate backlogs are maintained: a *product backlog* containing all defined stories in a prioritized manner and *sprint backlog* serving as a "to-do" list for the current sprint. Sprint backlog stories are pulled from the product backlog and further elaborated during sprint planning. An elaborated story includes descriptions of tasks that need to be completed

F.4.5 Definition of done

The definition of done is a list of criteria a story must adhere to in order to be considered "done". Considering that most research tasks are already fairly atomic, e.g. reporting on an research experiment would never be considered "done" without a discussion we propose defining done as follows:

Definition of Done:
- Defined story tasks completed.
- Story acceptance criteria met.
- Story output documented.
- Relevant writing integrated in an evolving draft.
- Documentation properly referenced.
- Documentation quality reviewed.

F.4.6 Acceptance criteria

In addition to the definition of done per-story acceptance criteria may be defined. Where the definition of done relates mostly to steps in the research process, the acceptance criteria define implementation-specific criteria stories need to adhere to in order to be considered done.

Example Acceptance Criteria:
For Story: Re-evaluate performance of classifier X with section type Y removed from dataset, **so that** influence of section type Y on performance can be evaluated.
- From the dataset D, the subset $D_{del} \in D$ of notes which contain a section Y has been identified.
- The performance of classifier X over D and D_{del}, before and after deletion of Y, has been determined.
- Performance change over each comorbidity class, after deletion of Y of had been computed.

Kanban and Limiting WIP

We will adopt the use of a Kanban board and limit concurrent WIP in order to maintain focus and workflow during a sprint.

F.4.7 Overview of Sprint Process

Figure F.4 shows an overview of the entire sprint process as performed in this project. An overall product backlog with relevant research stories is kept. Grooming the backlog, meaning prioritizing stories and deleting stories that are no longer relevant, is often done at a distinct event, however this is not strictly Agile practice. In this project it is expected that the product backlog will stay modest in size, thus it is more appropriate to maintain it in an ad-hoc manner, rather than scheduling a meeting. The only constraint to backlog grooming in this project is that it has to be done before sprint planning.

Sprint planning then takes place, during which stories relevant to the sprint goal are pulled from the product backlog. Some refinement of these stories may already take place during this stage, where stories are broken down into tasks and acceptance criteria are defined. However the more intricate details of story implementation are refined in a just-in-time manner during the sprint.

Once the stories have been executed they are documented in accordance with the definition of done. At the end of a sprint, stories from the sprint backlog are evaluated based on the definition of done and specified acceptance criteria. The current research increment is evaluated with regard to quality and performance on the classification task. Results of these evaluations are used to generate ideas for improvement of the current product, which can be used to create new stories.

Reflection on the research process during the previous sprint also takes place at the end of each sprint. Based on this reflection, changes to the process as described here may be made if deemed necessary.

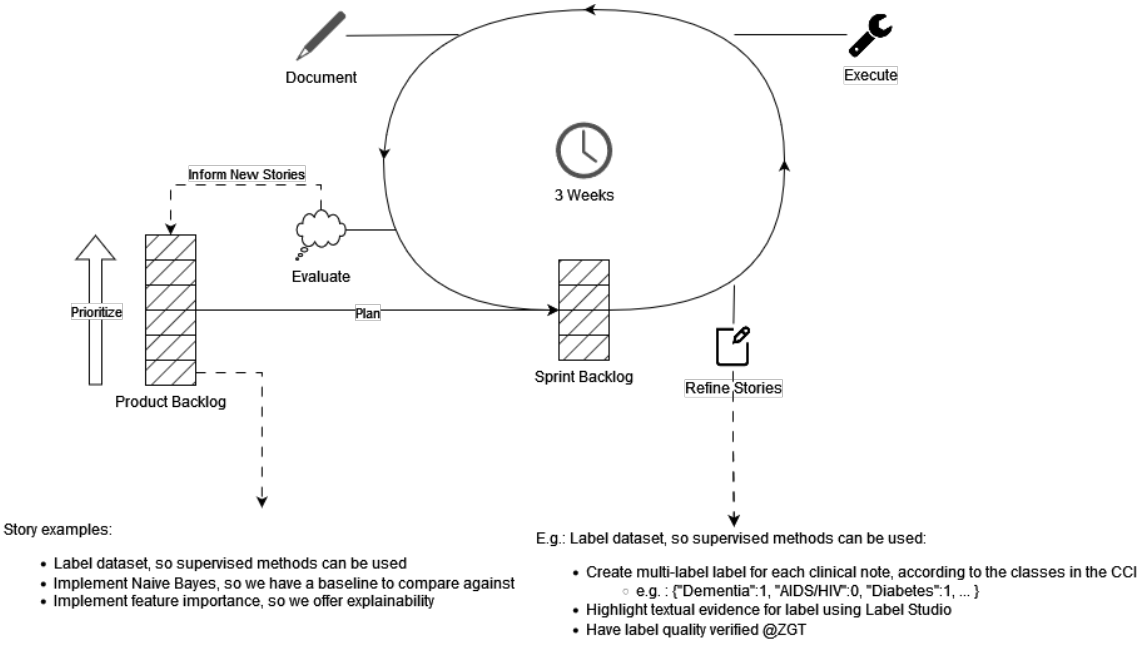


FIGURE F.4: Sprint Process

F.4.8 Agile Core Principles

Along with the Agile manifesto, twelve core Agile principles were defined, these principles form the base on which all Agile frameworks (XP, Scrum, etc.) are built [5]. The twelve original principles are geared towards software development but we can adapt them to fit research projects, we will attempt to adhere to these as best possible:[1]

1. **Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.**
   This principle remains the same, but research is the product and the customer is the party for which the research is performed. This customer party may for example be an examination board, supervisor, journal editor, research chair, third-party company or the researchers themselves.

2. **Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.**
   Welcome changes in direction, based on critique and critical review, even later into the research process. This may improve the final product.

3. **Working software is the primary measure of progress.**
   Evolving drafts are the primary measure of progress.

4. **Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.**
   Deliver "publishable" drafts, associated deliverables regularly, where the work is considered publishable in quality but not necessarily in scope, i.e. features can be missing but not incomplete. Associated deliverables may include data, code and documentation.

5. **Business people and developers must work together daily throughout the project.**
   Work closely with other disciplines, advisors and domain experts. Don't be afraid to ask questions.

6. **Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.**
   This principle remains the same for team-based research activities.

7. **The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.**
   Remains the same, where "development team" is the team of researchers including advisors and supervisors .

8. **Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.**
   Maintain pace through sustainable research and use some system (e.g. Kanban) to manage and track WIP. Finish tasks and avoid *"research debt"*, i.e. increased future workload due to active tasks not being handled appropriately, where possible.

9. **Continuous attention to technical excellence and good design enhances agility.**

[1] Based on the adaptations from [27]

This principle remains the same: think of excellence in experimental setup, reproducability, code quality and writing quality.

10. **Simplicity–the art of maximizing the amount of work not done–is essential.**
    This principle remains largely the same: focus on the research goal, don't overcomplicate.

11. **The best architectures, requirements, and designs emerge from self-organizing teams.**
    Discuss methods and methodology frequently. An "ideal" approach can not be prescribed, thus pivots may be necessary.

12. **At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.**
    This principle remains the same: reflect on the research process as much as you do on the research results, tune the approach to streamline the process.

F.4.9 Validation of Agile methodology

In order to validate the proposed Agile methodology, the project on extracting comorbidities will be used as a case study. While the scope of this evaluation will be somewhat limited, given the time frame of the project and the low number of people involved it will allow for valuable initial insights to be gained into if the proposed methodology is suitable to the class of research and will lead to a more efficient and effective research process. The evaluation will be based on personal experience of the student and involved supervisor as well as on a qualitative evaluation of the retrospective project timeline. This requires a comprehensive overview of sprint plannings, meeting notes and log entries regarding problems encountered to be kept, wherein care should be taken to also include the feedback and rationale supporting decisions on either the research direction or process. Artifacts that are to be collected thus include:

Sprint Planning: An overview of refined stories that are to be completed in the given sprint, clarifications on the constraints and assumptions for these stories, and any additional acceptance criteria that need to be adopted in addition to the definition of done as given in section F.4.5. If any changes in research direction were made based on feedback or problems previously encountered then these changes and argumentation will also be provided.

Sprint Review: Overview of stories completed during the sprint, if planned stories were not completed a reason should be given. Feedback regarding the current state of the project and intermediate results and relevant updates to the product backlog, for example new stories or a re-prioritization should be included.

Sprint Retrospective: Include a short summary of the research process during the last sprint: What went well? What did not go well? What action can be taken to improve the process for subsequent sprints? Are any changes to the framework necessary?

Log: Any additional issues that arise and research or design decisions that are made during a sprint need to be logged so that they can be taken into account in the review, retrospective and Agile validation.

Collecting this information with the appropriate amount of detail will add some administrative overhead to the project, however this workload should not be large enough as to hinder the main research on clinical note classification as the collection of much of this information gathering is inherent to executing a well-structured research project and taking quality meeting notes. If collected appropriately this information should make it possible to identify patterns and common hurdles and allow for a qualitative evaluation of the methodology, both holistically and on the level of individual components and principles borrowed from Agile methodologies.

F.5 Results & Discussion

The methodology described above was experimented with for 8 sprints in the period from September 2023 until February 2024. Figure F.5 shows an example of the documentation we kept, and includes all elements mentioned in section F.4.9, apart from the log which was kept separately as a part of meeting notes. Figure F.5 was taken from our 4th sprint, and shows the 5th story that was planned for that sprint, with a story-level review paragraph, color coded according to whether it was completed. Also shown are a brief overview of the sprint-level review and retrospective.

F.5.1 Changes made to initial approach

We initially adhered strictly to all the elements and principles described in section F.4, however the process evolved quite significantly based on the retrospectives of early sprints, and settled into a more lightweight approach around sprint 4. The most notable change with respect to what was laid out in section F.4 is that we dropped the requirement to maintain an evolving draft of "publishable" quality and consequently slimmed down our definition of done. We found that maintaining a "publishable" document required an investment of time and effort far beyond what we considered reasonable, as multiple days needed to be set aside towards the end of a sprint simply to maintain the quality of this draft document. Furthermore, early sprint outputs were often exploratory or non-final and therefore unlikely to end up in the final product, documenting these results extensively thus amounted to a lot of unnecessary work that would likely be removed from the draft at a later stage. We therefore changed to a lightweight approach in which we maintaining a structured overview of completed work for a given sprint by compiling all relevant resources, outputs and findings into an organized directory and reporting on the result using a slide deck during sprint review. Towards the end of the project, results that were relevant for the final thesis were picked out of this overview.

Another notable change is the fact that we de-prioritized comprehensive story definition and refinement. While we stuck to stating stories in the form described in section F.4.3 and kept in mind the *INVEST* throughout, it quickly became apparent that nearly all stories reduced to a limited number of tasks with self-evident acceptance criteria, and therefore put less effort into defining these during later stints.

F.5.2 Discussion of Agile elements

We believe that working in time-boxed sprints worked well for our project, especially in conjunction with the use of an immutable sprint backlog. We found that it required us

to make conscious choices of what methods and experiments to include in our research, limiting the time spent of "dead-end" exploration and experimentation. This allowed us to experiment with different models and potential elements for our solution (e.g. augmentation, resampling and pseudo-labeling), and not get stuck in premature or unnecessary optimization. We should note that towards the end of the project, when our solution had calcified, iteration became less valuable and we transitioned to finalization and writing phase that did not involve sprint planning or review.

The hybrid sprint review, retrospective and planning sessions were effective, and generally well-received by both the student and involved supervisors. The sessions typically started out with a brief 20-minutes overview of stories completed with relevant results before continuing into a discussion of next steps and required changes to the process. Decisions on how to continue were straightforward based on the results of the completed sprints and suggestions by supervisors. We recommend keeping review, retrospective and planning merged for projects of similar scope, as we were able to complete all three in under 45 minutes in all cases.

One notable complication of the iterative process was that it led to the project consisting of discernible and mostly separate phases. In our case the two phases dealt with full and weak supervision respectively. Many choices and assumptions in the latter stage depended on the results of the former, making it difficult to present the project in a traditional scientific format containing one encompassing methodology. This required us to break up our report and report the phases separately, and only introduce the methodological aspects that apply to both in the prior methodology chapter. All in all we believe it made the writing process more difficult.

As previously mentioned, maintaining a backlog of tasks and outlining an immutable sprint backlog for each iteration worked well. However, we found little added value is detailed tracking of backlog status using tools like Kanban. The number of items in the backlog at any time was limited, and tracking status was not an issue as one person worked on all stories.

We found story definition to be somewhat tedious, and as mentioned in section F.5.1 we moved away from detailing tasks and acceptance criteria. Most stories as we defined them were stand-alone pieces of research or experiments, and did not need complex integration with other completed work, this as opposed to the integration of newly developed features into a larger system in software development. This resulted in most acceptance criteria being steps or criteria that are self-evident for correctly defined scientific experiments (e.g. "Results for model A have been compared to results for model B."), or requirements for documentation. We would suggest simply keeping a backlog of a list of tasks or to-dos rather than fully defined stories, in our experience stories that were large enough to warrant fully defined acceptance criteria, were too large to fit in a single sprint and should be broken up.

Overall we would recommend a sprint-based research approach for explorative and design-focused research projects, but would not recommend the fine-grained Agile artifacts like detailed stories and backlog tracking. Yet it should be taken into account that our qualms with these artifacts may arise from the studied case being a one person project. Larger research projects involving multiple researchers are likely to benefit more from these

artifacts as they are tools for documentation as they are tools for centralized planning, documentation and communication.

F.6 Conclusions

RQ1 How will adopting elements from Agile methodologies impact the research process in terms of effectiveness and efficiency?

A: The high-level concept of time-boxed iteration followed by a review and retrospective was useful in facilitating a flexible yet focused research process, it resulted in effective decision making and limited the time spent on work that would go unused in the final thesis. A downside of the iterative approach is that it resulted in the project going through multiple discernible phases, resulting in a somewhat fragmented body of results that were difficult to compile into a single comprehensive work that adheres to the structure of a scientific report or paper, slowing down the writing process a considerable amount. The incorporation of more fine-grained elements like active backlog status management, story definition and continuous delivery of a "publishable draft" were not successful as these required a significant time investment while adding little value to a research project with one active researcher, though we believe these artifact may be useful in larger projects involving multiple researchers.

5. Explore different BERT variants, so the best base model can be used.

Tasks:

- Collect different variants of BERT that may be applicable for the given problem.
- Reevaluate experiments for each variant.

Acceptance criteria:

- Performances have been evaluated and compared on the same data split.

Review:

Story was completed as desired. Five variants have been compared: MedRoberta (trained on Dutch clinical notes), BERTje (trained on NL books, articles, and Wikipedia), RobBERT (trained on NL crawled web corpus), ClinicalBERT (trained on MIMIC-III), Multilingual BERT (standard model, trained on Wikipedia). Performances varied little between most models, however overall, the initially chosen model (MedRoBERTa.nl) performed best. Multilingual BERT gave a 0 performance, the reason for this is not known.

Output:

- Performance overview histogram (see output file + review .ppt)

Review

Stories were completed as desired, though story 4 is lacking in documentation and scope somewhat. As for next steps: Augmentation and further tuning of a supervised model does not seem like it will improve results much. Weak supervision should be the next step. Notes of non-hip fracture patients (within the same age cohort) may have to be used here.

- Data extraction of additional notes.
- Create improved lexicon.
- Weakly supervised model

[NAME] suggested running documents that are too long for BERT input through a summarizer, this has been added to the backlog.

Retrospective

Review was effective and to the point, supervisors seem to also prefer the new format. "Literature research" stories are difficult to tie to a proper output, as not everything encountered is (immediately) useful for next steps in the project, thus documenting these extensively (i.e. writing a paragraph) is often more effort than it is worth.

Color signifies status at end of sprint: •: Completed •: Missing documentation •: Not completed

FIGURE F.5: Example of documentation of Agile process.