Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Emma Valen Rian

Detecting Venous Catheters and Infections through Classification of Norwegian Adverse Event Notes

A Feasibility Study

Master's thesis in Computer Science
Supervisor: Melissa Yan
Co-supervisor: Øystein Nytrø and Lise Tuset Gustad
June 2023

NTNU
Norwegian University of
Science and Technology

Emma Valen Rian

Detecting Venous Catheters and Infections through Classification of Norwegian Adverse Event Notes

A Feasibility Study

Master's thesis in Computer Science
Supervisor: Melissa Yan
Co-supervisor: Øystein Nytrø and Lise Tuset Gustad
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

A peripheral intravenous catheter, used to administer intravenous fluids, medications, and blood transfusions, is one of the most frequent medical devices used worldwide. However, they are associated with many complications, including life-threatening conditions like bloodstream infections and sepsis. This thesis investigates the feasibility of leveraging unstructured Norwegian adverse event notes to detect the occurrence of venous catheters and infections, employing machine learning and natural language processing classification techniques. The study focuses on detecting relevant notes as a crucial initial step toward developing a real-time monitoring system for catheter-related infections and sepsis.

The work includes extensive qualitative research in collaboration with medical professionals to further revise an annotation guideline for labeling adverse event records, in addition to actually labeling the adverse events and creating a data set for classification. Furthermore, the work includes experimental research, proposing an exploratory approach for classifying the adverse event notes into relevant categories: venous catheter and infection.

The results indicate that while the proposed model is a feasible baseline for identifying relevant adverse event notes, it is affected by high bias and limited recall. The classification problem is challenging due to the comprehensive clinical knowledge needed to make assumptions about the presence of a venous catheter or infection. Ultimately, the model should be improved to achieve a higher recall to identify as many relevant notes as possible. Simplifying the classification problem should be considered to reduce the bias and increase model performance.

Sammendrag

Et perifert venekateter, hovedsaklig brukt for å administrere intravenøs væske, intravenøse medisiner og blodoverføringer, er et av verdens mest brukte utstyr innenfor medisinsk behandling. Til tross for dette er perifere venekatetre knyttet til mange alvorlige komplikasjoner, inkludert livstruende tilstander som blodbaneinfeksjoner og sepsis (blodforgiftning). Denne masteroppgaven er en mulighetsstudie som undersøker potensialet for å bruke fritekst-notater som beskriver uønskede medisinske hendelser til å finne tegn på infeksjoner og bruk av venekatetere. Arbeidet benytter seg av maskinlæring og språkbehandling til å klassifiere notatene. Hovedfokuset med studien er å finne relevante notater som et første steg for å utvikle et overvåkningssystem for å identifisere kateterrelaterte infeksjoner og sepsis i sanntid.

Arbeidet inkluderer omfattende kvalitativt arbeid i samarbeid med helsepersonell. Målet for det kvalitative arbeidet var å forbedre retningslinjer for annotering av notater for uønskede hendelser, fulgt av å annotere en mengde notater til bruk for klassifisering. I tillegg innebærer oppgaven eksperimentelt arbeid, i form av et initielt forslag for en klassifiseringsmodell som klassifiserer notater for uønskede hendelser basert på kategoriene venekateter og infeksjon.

Resultatene viser at modellen er egnet som et utgangspunkt for å identifisere relevante notater for uønskede hendelser, men den er begrenset av høy bias og en lav dekningsgrad (recall). Selve klassifiseringsproblemet er svært komplekst, ettersom det er behov for omfattende klinisk kunnskap for å gjøre antakelser om tilstedeværelsen av et venekateter eller en infeksjon. Modellen burde forbedres for å øke dekningsgraden, slik at den returnerer så mange relevante notater som mulig. Det bør også gjøres en fremtidig vurdering om det er gunstig å forenkle selve klassifiseringsproblemet for å minke bias og forbedre modellens resultater.

Preface

This thesis documents the work performed during the final semester of my Master's Degree in Computer Science at the Norwegian University of Science and Technology (NTNU). The thesis project is a feasibility study, aiming to investigate the potential of using multi-label text classification techniques to detect the presence of venous catheters and infections in Norwegian adverse event notes.

The academic year of 2022/2023 has been challenging. I have deep-dived into the computer science fields of Natural Language Processing and text classification and learned a lot. On top of this, I have collaborated with many medical professionals due to the clinical focus of my thesis, leading to an increased interest in the medical field. I would like to thank multiple groups and people who have helped me produce this work.

First, I want to thank my supervisor Melissa Yan, for her continuous guidance and support throughout the year.

Thank you to my co-supervisors, Øystein Nytrø and Lise Tuset Gustad, for many insightful discussions and suggestions.

Thank you to the Gemini Center for Sepsis Research group, who listened to a presentation about my thesis project, gave me valuable feedback, and included me in their meetings throughout the year. In particular, thank you to group member Lise Husby Høivik for taking a special interest in my project and meeting with my supervisors and me several times this year to further develop the Adverse Event Annotation Guideline.

Thank you to HEMIT for providing the adverse event data I used for the project, and to HUNT Cloud for providing and supporting a lab environment where I could work on my experiments.

And lastly, a special thank you to Nurse Lisabet Sorte at Levanger Hospital for spending almost 10 hours in meetings with me after regular work hours to annotate clinical text data for my project. Without these meetings, my project would not have been feasible.

Emma Valen Rian

Trondheim, 12th June 2023

Contents

List of Figures

List of Tables

Acronyms

AAENOTE Annotated Adverse Event NOte TErminology.

AE Adverse Event.

AI Artificial Intelligence.

BoW Bag-of-Words.

BR Binary Relevance.

BRAT Brat rapid annotation tool.

BSI Bloodstream Infection.

CC Classifier Chain.

CIIO Catheter Infection Indications Ontology.

CNB Complement Naïve Bayes.

CoSeM Computational Sepsis Mining and Modelling.

CV Cross Validation.

CVC Central Venous Catheter.

DT Decision Tree.

EHR Electronic Health Record.

GB Gradient Boosting.

HL Hamming Loss.

IAA Inter-annotator Agreement.

ICU Intensive Care Unit.

IDF Inverse Document Frequency.

IV Intravenous.

LG Logistic Regression.

ML Machine Learning.

MLAA Machine Learning of Adverse Event.

MNB Multinomial Naïve Bayes.

NB Naïve Bayes.

NLP Natural Language Processing.

NLTK Natural Language Toolkit.

NSD Norwegian Centre for Research Data.

OBT Oslo-Bergen Tagger.

PIVC Peripheral Intravenous Catheter.

POS Part-of-Speech.

REK Norwegian Regional Committees for Medical and Health Research Ethics.

RF Random Forest.

SVM Support Vector Machine.

TF Term Frequency.

TF-IDF Term Frequency — Inverse Document Frequency.

VC Venous Catheter.

Chapter 1

Introduction

One of the most common clinical procedures across the world is the insertion of a Peripheral Intravenous Catheter (PIVC) [1, 2]. This device is inserted into a peripheral vein and used to administer intravenous (IV) fluids, medications, and blood transfusions. Despite their frequent use, PIVCs are associated with many complications. One prevalent complication is phlebitis, which is a local vein inflammation characterized by redness, swelling, tenderness, pain, warmth, and purulent discharge [2]. More serious complications include a PIVC-associated bloodstream infection (BSI) or sepsis. A BSI is caused by bacteria accessing the bloodstream [3] and can lead to sepsis, a life-threatening organ dysfunction caused by a dysregulated host response to an infection [4].

This thesis explores the feasibility of exploiting free-text Norwegian clinical data sets to identify signs of venous catheters and infections using Machine Learning (ML) and Natural Language Processing (NLP) techniques.

1.1 Background and Motivation

Although PIVCs are very frequently used, they lack satisfactory documentation in clinical records [1]. Additionally, sepsis is poorly documented outside the intensive care unit (ICU), even though the general care wards care for almost half of all patients with severe sepsis [5]. Early identification and rapid treatment of infected patients with early-stage sepsis are crucial objectives in critical care medicine [6]. These objectives introduce the need for electronic surveillance tools to recognize early signs of sepsis and alert clinicians automatically.

This thesis aims to support an extensive collaboration project between the Norwegian University of Science and Technology and St. Olavs Hospital in Trondheim, Norway. The long-term goal of this collaboration project is to utilize clinical text data to develop a real-time infection and sepsis monitoring system, particularly focusing on PIVC-related BSIs [7, 8]. To achieve this, comprehensive knowledge about how to identify and reason about connections between PIVCs and infections is needed.

The primary data source in this project is two sets of free-text adverse event (AE) reports, which contain notes regarding non-favorable events at the hospital. Compared to other types of clinical documentation, these AE notes document failure related to PIVCs more frequently [7–9]. Previous work includes creating a detailed terminology and ontology to represent and reason about PIVC-related BSIs in the AE notes, as presented in Yan et al. [9].

This thesis project explores the potential of using supervised ML and NLP to detect the presence of venous catheters and infections in the AE notes through text classification. This filtering of the reports is necessary to reason for potential catheter-related complications, as the complete AE data set contains an abundance of irrelevant notes. However, correctly identifying relevant notes using supervised learning is challenging, as only a small amount of labeled AE data is available.

1.2 Goals and Research Questions

The primary objective of this project is not to provide a definitive solution for accurately identifying venous catheters and infections in AE notes. Instead, it aims to explore the feasibility of utilizing the available data for this purpose and assess the associated limitations. Several classifiers will be trained and tested to evaluate their performance. However, given the constraints of limited data and potential bias, they are not anticipated to achieve perfect results.

The main goal of this Master's thesis is defined as follows:

Goal *Investigate the potential of using multi-label text classification techniques to detect the presence of venous catheters and infections in Norwegian adverse event notes.*

The thesis further aims to assess if, despite the limitations, this type of classifier is feasible to use for identifying relevant AE notes that can be further evaluated. This further evaluation includes analyzing the notes in more detail to identify potential catheter-related complications, especially between PIVCs and BSI/sepsis. This leads to the following subgoal:

Subgoal *Assess if this multi-label classification approach is feasible as a baseline to filter notes for further identification of potential catheter-related complications.*

Based on these two goals, three research questions were formulated. The research questions form the core of the thesis, guiding the exploration and investigation conducted throughout the study. The conclusion presents a final evaluation and answers to these questions.

Research question 1 *How can the available adverse event data sets be used to train a supervised multi-label classifier to detect the presence of venous catheters and infections?*

Research question 2 *What are the key limitations to consider when performing supervised text classification on a small, labeled data set in a clinical setting, and how can these limitations be alleviated?*

Research question 3 *How feasible is it to use the proposed approach for further identification of catheter-related complications?*

1.3 Research Method

The research methodology used in this thesis is a combination of two main approaches:

1. **Qualitative research strategy:** Interviews and discussions with clinical domain experts were conducted for the purpose of requirements elicitation and validation of an annotation guideline. Furthermore, labeled data were gathered and verified through several meetings with a medical professional.
2. **Experimental research strategy:** Several experiments were conducted to explore the feasibility of using the available data to achieve the clinical goal of identifying venous catheters and infections in AE notes.

1.4 Approvals and Acknowledgements

Due to the use of sensitive data in the project, an ethical approval was required. The project has been approved by the Norwegian Regional Committees for Medical and Health Research Ethics (REK), with approval number 26814.

This project also make use of and process personal annotator data. The Norwegian Centre for Research Data (NSD) has approved the project to collect personal annotator data and use their annotations (NSD reference no. 142683). Annotators themselves have also consented to mentioning their profession and using their annotations.

This thesis project is funded by the Computational Sepsis Mining and Modelling (CoSeM)[1] and Machine Learning of Adverse Events (MLAA) projects. The CoSeM project is funded by the Norwegian University of Science and Technology's Health Strategic Area, while the MLAA project is funded by the Helse Midt-Norge Innovation Funding 2021. The thesis is part of a pipeline in the CoSeM and MLAA projects.

[1] https://www.ntnu.edu/cosem#/view/publications

1.5 Thesis Structure

The thesis consists of eight chapters following this introduction. The structure and purpose of these chapters is as follows:

- Chapter 2 introduces the relevant theoretical concepts and methods necessary to understand the work.
- Chapter 3 presents an overview of related work performed in the field of text classification of clinical narratives.
- Chapter 4 presents an overview of the data and resources used in this project.
- Chapter 5 describes the process of creating the final labeled data set used for classification in detail, including the annotation process.
- Chapter 6 presents an overview of the architecture and methods used for the classification experiments.
- Chapter 7 describes the experimental plan and limitations, followed by the experimental results.
- Chapter 8 provides a detailed discussion based on both the experimental results and the process of creating the labeled data set.
- Chapter 9 concludes the thesis by providing a summary of the findings and suggestions for future work.

Chapter 2

Background Theory

This chapter provides the theoretical background for comprehending the experiments and results presented in this thesis. It establishes a solid foundation for understanding the fundamental concepts and methodologies relevant to the research.

2.1 Machine Learning

Machine Learning (ML) is a field of computer science focusing on using Artificial Intelligence (AI) to develop algorithms that can learn how to represent a set of data in the best way [10]. A data set for ML consists of multiple data points, where each data point is commonly called an instance or example. Each data point has a set of features that can be used to analyze the data. For example, in a data set consisting of a list of employees, each employee is regarded as an instance. The characteristics of each employee, like their name, age, and salary, are the data set's features.

There are four main types of approaches to ML: supervised, unsupervised, semi-supervised, and reinforcement learning [10]. This section presents the two methods focused on learning with labeled training data: supervised and semi-supervised learning.

2.1.1 Supervised Learning

Supervised learning methods use training data with known observations and outputs to learn rules, so they can use these rules to predict the output for new data in the future [11]. A training data set with known outputs is called a labeled dataset, and the process of labeling data is called annotation.

There are two main categories of supervised learning: classification and regression [10]. Classification is the problem of categorizing the instances, while regression involves predicting continuous numerical data.

The basic steps of supervised machine learning are:

1. Obtain a labeled data set and split it into training, validation, and test data sets.

2. Use the training data set to train a model and tune its performance using the validation set.
3. Evaluate the model's final performance by comparing the predicted outcomes to the known outcomes in the unseen test data.

The reliability of a supervised learning method depends on the quality and size of the labeled data set. In cases where labeled data is scarce, it is relevant to consider using a semi-supervised learning method.

2.1.2 Semi-Supervised Learning

Semi-supervised learning is a combination of learning with labeled data (supervised) and unlabeled data (unsupervised). It can be beneficial in scenarios where the data is partially labeled, meaning that all the features are present, but only some of them have known outputs [10]. Semi-supervised learning is highly relevant when annotating the complete data set is time-consuming and requires knowledgeable annotators.

Self-Training algorithm

One algorithm for building a semi-supervised classifier is called Self-Training [12]. The Self-Training algorithm starts with a small amount of labeled data and then iteratively predicts labels of the unlabeled instances. The predictions for the unlabeled instances are ranked based on confidence during each iteration, and the most confident instances are permanently added to the labeled training data. It is recommended to apply confidence measures and thresholds to avoid mislabeling, as including wrongly labeled instances in the training data set can decrease the classifier's performance.

2.2 Natural Language Processing

Natural Language Processing (NLP) is the field of applying computational techniques to learn, understand and produce human language content [13]. NLP can be used for various purposes. One well-known use case of NLP is to support human-human communication, such as machine language translation or text-to-speech systems. Other use cases are information retrieval, information extraction, text summarization, and question-answering [14] — all tasks that help humans access necessary information.

NLP methods are traditionally built on a rule-based or ML approach [15]. Rule-based approaches to NLP are the oldest, relying on domain-specific linguistic rules and knowledge bases. These rule-based approaches are still used today, but they compete with statistical ML approaches, which eliminate the need for handwritten rules. ML approaches to NLP use annotated example data or clustering methods to build knowledge and generate

rules based on probabilities. Unlike rule-based approaches, there are no guarantees that the rules the machine learning models create are correct. However, the ML approaches usually give good results in practice since they learn from a large amount of example data and can therefore utilize the most common cases [16]. The size and representativeness of the data are directly correlated to the performance of the statistical ML approaches to NLP [16].

2.2.1 Introduction to Text Classification

Text classification is the task of categorizing text into predefined groups based on their content. Text classification techniques are today used in numerous fields, such as medicine, social sciences, healthcare, psychology, law, and engineering [17].

The input data to a text classification problem consist of a raw text data set (corpus), where each text segment is regarded as a data point, called a *document* [17]. Text is an example of unstructured data, which can not be easily organized in a database because it lacks a standard format. In order to use mathematical modeling to analyze and categorize text data, the unstructured text sequences much be converted into structured data [17]. This process is commonly called text representation or feature extraction. It includes cleaning the data to remove unnecessary information and splitting the unstructured text into separate *features*. These features are further structured into numerical representations to be interpretable by mathematical models.

After preparing the text data, a classification method can be applied to categorize the data into classes. Previously, rule-based methods were used for text classification, but supervised ML models dominated from the 1960s until the 2010s [18]. In later years, deep learning methods have become an increasingly popular approach to text classification [18]. However, deep learning algorithms generally require a large data set to train the model accurately, which is an issue in many domains, like healthcare [19].

Sections 2.2.2 and 2.2.3 describes common techniques for cleaning (preprocessing) the data and methods for feature extraction, respectively. Section 2.3 further examines the text classification problem, focusing on methods for classifying the same documents into multiple categories.

2.2.2 Preprocessing Methods

This section introduces some of the most common text preprocessing techniques for preparing documents for classification or other NLP tasks.

Tokenization

Tokenization is the process of separating the entire text in a document into a more usable form called tokens [20]. Usually, the document is divided into separate words as a starting point for further cleaning the data. However, it is not always straightforward how to tokenize a document. A white space between words is commonly regarded as a main delimiter for splitting, but the strategy for tokenization should be tailored to the problem.

Standardization and Cleaning

The next step is to standardize and clean each token, to remove noise and make the documents more comparable. This process is typically done using one or more of the following techniques:

- **Lowercase all tokens:** Lowercasing the tokens ensures that identical words with different casing are not considered distinct. Without lowercasing, the words "Coffee" and "coffee" are regarded as different tokens.
- **Removing numbers, special characters, punctuation, symbols, emojis, and emoticons:** It is beneficial to remove all sorts of noise in the data. However, the definition of noise can vary depending on the problem. For instance, if the problem is classifying product reviews into positive and negative categories, emojis and emoticons can be very informative for the analysis.
- **Correcting spelling errors:** Some text documents have many spelling errors, which can lead to misunderstandings or an inability to compare words that should be identical.
- **Formalize slang and abbreviations:** It can be helpful to convert slang and abbreviations in the document into formal language.

Removal of Stopwords

Stopwords are words in a language that are so common that they are irrelevant to the analysis of the document [21]. Therefore, removing these words from the document is helpful for further analysis, as they do not provide any information relevant to the document's content. Stopwords are usually identified by performing a keyword search in the document, using a list of the relevant language's stopwords as keywords.

Text Normalization

The last common step of text preprocessing is normalization. Text normalization is the process of identifying and removing word suffixes that are connected to inflection [21].

This process reduces the number of unique tokens to analyze and helps to compare them, as words with the same root often have a similar meaning. There are two main types of normalization: stemming and lemmatization.

- **Stemming:** Stemming reduces a word to its root (stem) by removing the suffix. Performing stemming does not necessarily produce a meaningful word [21] because stemmers are based on rules to strip a word of its suffix, but there are always exceptions to these rules. An example of this is stemming the word *hospitals*, leading to the stem *hospit*.
- **Lemmatization:** Lemmatization reduces a word to its grammatical base (lemma), using information about its inflection and the Part-of-Speech (POS) of the word (e.g., noun, verb, or adjective). More complex linguistic competence is required to perform lemmatization [21], but it is more accurate than stemming. Lemmatization of the word *hospitals* leads to the lemma *hospital*. Also, a more complex example of lemmatization is using irregular verbs: *am, are,* and *is* becomes *be*.

2.2.3 Text Feature Extraction Methods

In their survey, Kowsari et al. [17] examines various methods of feature extraction for text classification. They discuss two primary approaches: weighted words and word embeddings. Weighted word methods directly count the tokens in a document and apply a scoring system to compare documents. In contrast, word embedding methods learn from sequences of words and consider the words' position (syntax) and meaning (semantics) in the text.

Word Embeddings

Word embedding vectors are trained on a vocabulary and contain information about each word's syntactic and semantic meaning. Using an unsupervised approach, they can be trained directly on a text corpus, hence not requiring manual feature extraction or labeling [22]. Some examples of popular word embedding methods today are Word2Vec [23], Global Vectors for Word Representation (GloVe) [24], and FastText [25].

Many word embedding methods, such as Word2Vec and GloVe, have one substantial limitation: they can not capture out-of-vocabulary words from the corpus [17]. This limitation can be a severe problem for text classification of documents that contain many unique words and abbreviations, like in medical text. This problem is even more severe if the training corpus is small. As word embeddings gain vocabulary directly from a corpus of text, the performance is highly dependent on the training corpus size [17, 22]. One way to overcome this problem is using pre-trained word embeddings trained on a corpus with relevant vocabulary.

Weighted Words

Weighted word approaches lack the syntactic and semantic benefits of word embedding methods. However, they are very intuitive and easy to implement. Since they are based on counting the frequency of tokens in a document, they build their vocabulary during feature extraction. Accordingly, unlike many word embedding methods, weighted word methods are not limited by unknown words [17].

Bag-of-Words: The most straightforward technique of text feature extraction is to count the occurrences of unique tokens and represent them in a vector. This method is called Term Frequency (TF), more commonly known as Bag-of-Words (BoW). An example of BoW is given below:

Document:

> *"These words display one of the many examples of a bag of words approach"*

BoW:

> *{"These", "words", "display", "one", "of", "the" "many", "examples", "of", "a", "bag", "of", "words", "approach"}*

BoW feature count:

> *{"These": 1, "words": 2, "display": 1, "one": 1, "of": 3, "the": 1, "many": 1, "examples": 1, "a": 1, "bag": 1, "approach": 1}*

Term Frequency - Inverse Document Frequency: A problem with BoW is that common words are regarded as impactful features, as they appear in many of the documents in the corpus. However, words that are common across all documents in the corpus are irrelevant for comparing the similarities between the documents. Inverse Document Frequency (IDF) is a measure that lowers the importance of words with a high frequency across the entire corpus. The result of combining BoW with IDF is a weighting scheme where the value increases proportionally to the word's frequency in the document, called TF-IDF [26].

Kowsari et al. [17] define the weight $W(t,d)$ of a term t in a document d by TF-IDF as given in Equation (2.1):

$$W(d,t) = TF(d,t) * log(\frac{N}{df(t)}), \tag{2.1}$$

where N is the number of documents in the entire corpus and $df(t)$ is the number of documents containing the term t in the corpus.

N-Grams

Another common way of dividing words in a document into features is splitting the words into tuples of *n-grams*. An n-gram is a set of n words organized in the order they appear in the text [17]. The standard BoW use unigrams (1-grams), where each word is its distinct feature. Using bigrams (2-grams) or trigrams (3-grams) as features gives more information about the context of words in a document. An example of the difference between unigrams and bigrams is given below:

Document:

> *"This is an example of using n-grams to increase information about word context"*

Unigrams (standard BoW):

> *{"This", "is", "an", "example", "of", "using" "n-grams", "to", "increase", "information", "about", "word", "context"}*

Bigrams:

> *{"This is", "is an", "an example", "example of", "of using", "using n-grams" "n-grams to", "to increase", "increase information", "information about", "about word", "word context"}*

2.2.4 NLP in a Clinical Text Setting

In later years, the healthcare industry has undergone an extensive digitalization process. An example is the increasing use of Electronic Health Records (EHRs), which are digital collections of health or healthcare information related to an individual [27]. EHRs are comprised of both structured data and unstructured free-text narratives [27]. The increased availability of EHRs and other free-text narratives introduces the possibility of applying NLP to analyze this data.

The primary means of communication in healthcare is through clinical narratives [28]. These narratives contain valuable clinical information that is not necessarily included in a structured format, like descriptions of symptoms. There are several types of clinical narratives. Discharge summaries, physician notes, or progress notes are examples of patient-related narratives that are included in a patient's EHR. Other relevant types of clinical narratives are reports relevant to patients' safety [28]. Examples are patient complaints, incident reports, and adverse event reports.

Spasic and Nenadic [28] present a study investigating types of NLP tasks applied to clinical text data and examining key data properties used to train and evaluate clinical ML models. They found that the training data set size is usually relatively small and that most studies focused on text classification using specific types of clinical narratives.

The classification results were commonly used to support phenotyping (identify patient characteristics), prognosis, care improvement, resource management, and surveillance.

Challenges: NLP research in the clinical field has been active since the 1960s, but its progress has been slower compared to other domains [29]. This is primarily due to various barriers that hinder the advancement of NLP in the clinical domain. Chapman et al. [29] analyze the challenges of using NLP for clinical text. Some of the challenges presented are:

- **Lack of access to shared data:** Hospitals and clinics are often reluctant to share clinical data for researchers outside the institutions because of patient privacy.
- **Lack of annotated data sets for training and benchmarking:** Clinical language differs from the general language, introducing the need for domain-specific NLP training and development. This training requires large amounts of annotated clinical data sets.
- **Insufficient common conventions and standards for annotations:** Because of the reluctance to share data sets across institutions, annotated data sets remain small and do not share a standard annotation format.

Processing text from clinical narratives can be challenging due to the lack of formatting and structure. Unlike biomedical text, which is written to present results from medical research, clinical narratives are written under time pressure to share information with other medical professionals [30]. According to Leaman et al. [30], some common challenges in processing text from clinical narratives include handling flexible formatting, atypical grammar, and rich descriptions. Some more detailed examples of this are the use of clinical abbreviations and jargon, the high presence of spelling errors, and the excessive use of punctuation and parenthetical expressions. Additionally, resources are needed to understand the clinical vocabulary.

2.3 Multi-label Classification

In traditional single-label classification, each instance is associated with a single label l from a set of disjoint labels L, where $|L| > 1$ [31]. In the case where $|L| = 2$, the problem is called a binary classification problem. An example of this is predicting if an image contains a fruit (label 1) or does not contain a fruit (label 2). If $|L| > 2$, the problem is called a multi-class classification problem. By expanding the example of predicting fruits, a multi-class classification problem could be predicting if an image of a fruit contains an orange, apple, pear, or banana. The fruit can be either of these, but only one of them.

Multi-label classification differs from single-label classification as each instance is associated with a subset of labels $S \subseteq L$ [32]. Hence, a multi-label classification problem allows an instance to be related to more than one class. For instance, a news article might be

related to several topics simultaneously. A multi-class classifier can only predict separate topics for each article, which gives limited information. On the other hand, a multi-label classifier can assign multiple topics to each article, like science *and* politics.

2.3.1 Problem Transformation Methods

The generality of multi-label classification makes it more complex than single-label classification. A common approach to multi-label classification is to transform a multi-label problem into one or more single-label problems, called problem transformation [32]. This approach makes it possible to use existing single-label classifiers for multi-label problems.

One popular approach to problem transformation is the Binary Relevance (BR) method [32, 33]. This method involves training a set of $|L|$ binary classifiers for each label, $C_1, ..., C_{|L|}$, where each classifier C_j predicts a binary association (0 or 1) for its corresponding label $l_j \in L$ [32]. A significant limitation of BR is that it assumes label independence, as it ignores label correlations in the training data. An alternative to BR that considers label correlations is the Label Powerset (LP) method, which combines all unique label sets into new, combined labels [33]. However, while label correlations are considered, this method poses a considerable risk of extreme class imbalance for learning based on the frequency of the different unique label sets. Also, the complexity of LP is upper bound by $min(n, 2^{|L|})$, where n is the number of data instances, as the number of possible label combinations increases exponentially with the set of labels $|L|$.

Classifier Chain

In 2009, Read et al. [32] introduced a new problem transformation method based on BR called Classifier Chain (CC). This method considers label correlations while only including $|L|$ binary classifiers, as in BR. The binary classifiers are linked in a chain where the successive classifiers are extended to include previous classifiers' predictions. As it passes label information between the classifiers in the chain, it exploits label dependence to overcome the label independence problem of BR.

A more detailed description of CC based on the definition by Liu and Tsang [34] is presented below:

> Assume $\mathbf{x}_t \in \mathbb{R}^d$ is a real vector representing an instance for $t \in \{1, ..., n\}$, where n denotes the number of training samples. $Y_t \subseteq \{\lambda_1, \lambda_2, ..., \lambda_q\}$ is the corresponding output (label). $\mathbf{y}_t \in \{0, 1\}^q$ is used to represent the label set Y_t where $\mathbf{y}_t(j) = 1$ if an only if $\lambda_j \in Y_t$.
>
> The CC model trains $|L|$ binary classifiers for each label, $c_j (j \in 1, ..., |L|)$. The classifiers are linked in a chain where each classifier c_j is responsible

for the binary classification problem for label λ_j. The augmented vector $\{\mathbf{x}_t, \mathbf{y}_t(1), ..., \mathbf{y}_t(j)\}_{t=1}^{n}$ is then used as the input for training classifier c_{j+1}.

2.3.2 Relevant Text Classifiers for a Small Dataset

Section 2.2.1 provided an introduction to text classification and discussed that while deep learning algorithms are becoming increasingly popular for text classification tasks, they require a large data set to build an accurate predictive model. Therefore, classic supervised ML algorithms are often a more viable choice when dealing with a small data set for multi-label learning. This section presents some of the most popular classic ML methods for classification tasks that are also widely used for text classification. All the presented classifiers are traditionally single-label classifiers but can be expanded to multi-label learning using a problem transformation approach, as presented in Section 2.3.1.

Linear Classifiers: Logistic Regression and Linear Support Vector Machine

Logistic regression is a classification algorithm that aims to find a relationship between the given features and the probability of a specific outcome [10]. Logistic regression uses a sigmoidal curve to estimate the class probability, determined by the sigmoid function defined in Equation (2.2).

$$y = \frac{1}{1 + e^{-x}} \tag{2.2}$$

The sigmoid function in Equation (2.2) produces an S-shaped curve that converts discrete or continuous numeric features x into a single numerical value y, where $y = \{0, 1\}$.

The logistic regression classifier works well for predicting categorical outcomes, like in text classification. A drawback of the classifier is that the prediction requires that each data point is independent [17].

Support Vector Machine (SVM) classifiers are based on determining the optimal separators (hyperplanes) in the feature space to separate the different classes in the best way [35]. The normal vector to the best hyperplane is the direction in the feature space with maximum discrimination. The SVM is robust to high dimensionality features, which makes it ideal for text classification [35].

SVMs are available as both linear and non-linear classifiers. Linear SVMs are commonly used due to their simplicity and interpretability [35].

Probabilistic Classifiers: Naïve Bayes

Naïve Bayes classifiers are based on computing statistics from a training data set based on a naïve application of Bayes theorem [11]. A high-level description of the Naïve Bayes classifier is presented in Kowsari et al. [17]:

> Let n be the number of documents that fit into k categories where $k \in \{c_1, c_2, ..., c_k\}$, and the predicted class output is $c \in C$. Then, the Naïve Bayes algorithm can be described as Equation (2.3):
>
> $$P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}, \tag{2.3}$$
>
> where d is a document and c is a class.

A variant of the Naïve Bayes algorithm suited for multinomial distributed data is called the *Multinomial Naïve Bayes* (MNB). This model is commonly used for text classification, as it captures information about the number of times a word occurs in a document [36]. One limitation of MNB is that when there is an imbalance in the class labels in the training data, MNB selects poor weights for the decision boundary due to bias [36]. A new approach based on MNB was presented by Rennie et al. [37] to deal with imbalanced training data. This method is called *Complement Naïve Bayes* (CNB). CNB is based on learning the weights for a class by utilizing all the training data that is *not* contained in that class, called the complement class [37].

Tree-based Classifiers: Decision Tree, Random Forest, and Gradient Boosting

A *decision tree* is a supervised learning technique that is commonly used for classification tasks [10]. It is based on splitting the data set in the best way based on a specific feature per split.

Decision trees are rarely used in modern ML tasks in their original form, partly because they tend to overfit [11]. However, decision trees are used as building blocks for other widely popular ML approaches. Two ML classifiers based on decision trees are *random forest* and *gradient boosting*. Both utilize an ensemble of trained decision trees to predict the outcome [11].

The main difference between random forests and gradient boosting approaches is how they build the decision trees [11]. Random forest classifiers are based on a vast amount of deep decision trees, and the results are aggregated at the end based on majority voting [10]. As previously mentioned, these decision trees are likely to be overfitted, but combining the output of many trees reduces this problem. On the other hand, a gradient boosting classifier is based on shallow decision trees, where each new tree is added after the other, aiming to improve the problems with the previous trees (boosting).

Gradient boosting methods has been proven to offer great performance with low computational costs, even without hyperparameter tuning [11]. However, random forests are less prone to overfitting [11].

2.3.3 Evaluation Methods

Unlike binary and multi-class classification, the result from a multi-label classification algorithm contains predictions for a set of labels, meaning that the result can be correct, partially correct, or incorrect [33]. This difference makes it harder to calculate the performance of a multi-label classifier, as the standard evaluation metrics like accuracy, precision, recall, and F_1-measure do not cover the problem of a partially correct prediction. Therefore, more specific evaluation metrics are needed to evaluate multi-label classifiers.

Many different evaluation metrics specific to multi-label classification have been proposed in the literature. Zhang and Zhou [38] and Sorower [33] group these metrics into two main groups: example-based and label-based. Example-based metrics calculate the performance of each example (i.e., instance or document) separately and return the average value over all test examples. On the other hand, label-based metrics calculate the performance of each label independently, followed by returning either the macro- or micro-average value across all labels [33, 38]. This work will present the example-based Hamming Loss metric and the label-based accuracy, precision, recall, and F_β-measure. The definitions of the metrics are based on the definitions given in Zhang and Zhou [38]. They use the following notations:

- $\mathbf{x}_t \in \mathbb{R}^d$ is a real feature vector with q possible class labels $\{y_1, y_2, ..., y_q\}$.
- Y is a label set associated with $\mathbf{x}$.
- $S = \{(\mathbf{x}_i, Y_i) \mid 1 \leq i \leq p\}$ is the test set.
- $h(\cdot)$ is a trained multi-label classifier.

Example-based evaluation metric: Hamming Loss

The Hamming Loss (HL), or Hamming Distance, metric evaluates the fraction of misclassified example/label pairs [38]. It takes into account both the prediction error (an incorrect label is predicted) and the missing error (a correct label is not predicted), normalized over the total number of labels and examples [33]. A well-performing model will have a *low* HL as it reports a fraction of incorrect predictions. Thus, a case where $HL = 0$ describes an ideal situation with no error. Equation (2.4) defines HL:

$$HL(h) = \frac{1}{p}\sum_{i=1}^{p}\frac{1}{q}|h(\mathbf{x}_i)\Delta Y_i|, \tag{2.4}$$

where Δ is the symmetric difference between the two sets.

Label-based evaluation metrics

Label-based evaluation metrics are based on the number of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) test examples with respect to a certain label. These are illustrated in a *confusion matrix* in Figure 2.1, and defined as follows:

- TP: A test result which correctly indicates the presence of a label
- FP: A test result which incorrectly indicates the presence of a label
- TN: A test result which correctly indicate the absence of a label
- FN: A test result which incorrectly indicate the absence of a label.

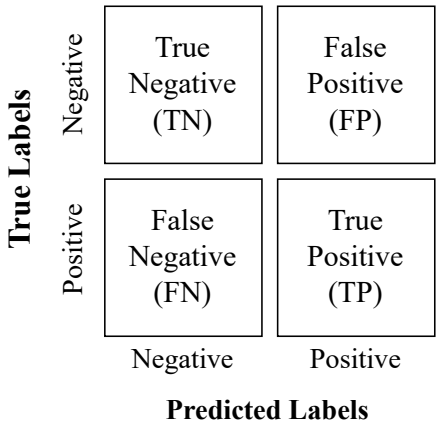


Figure 2.1: Illustration of a confusion matrix based on True Positive, False Positive, True Negative, and False Negative predictions.

The **precision, recall and F_β-score** for the j-th class label y_j are respectively defined in Equations (2.5) to (2.7):

$$Precision(TP_j, FP_j, TN_j, FN_j) = \frac{TP_j}{TP_j + FP_j}, \tag{2.5}$$

$$Recall(TP_j, FP_j, TN_j, FN_j) = \frac{TP_j}{TP_j + FN_j}, \tag{2.6}$$

$$F_\beta(TP_j, FP_j, TN_j, FN_j) = \frac{(1+\beta^2) \cdot TP_j}{(1+\beta^2) \cdot TP_j + \beta^2 \cdot FN_j + FP_j}. \tag{2.7}$$

The precision is the proportion of correctly predicted labels to the total number of true labels. It gives an insight into how many of the instances the model retrieved was actually relevant. On the other hand, the recall is the proportion of correctly predicted labels to

the total number of labels, which gives an insight into how many of the total relevant instances the model managed to retrieve. The F_β score is used to balance recall and precision. The most common choice of β is the value 1, which gives an equal weight for both recall and precision, leading to a harmonic mean between the two measures. The simplified F_1-score is shown in Equation (2.8):

$$F_1(TP_j, FP_j, TN_j, FN_j) = \frac{2 \cdot TP_j}{2TP_j + FP_j + FN_j}. \tag{2.8}$$

Let $B(TP_j, FP_j, TN_j, FN_j)$ represent a binary classification metric $B \in \{Precision, Recall, F_\beta\}$. Then, the label-based classification metrics can be obtained by either macro-averaging or micro-averaging the metrics.

The *macro-averaging* of a metric B is defined in Equation (2.9):

$$B_{macro}(h) = \frac{1}{q} \sum_{j=1}^{q} B(TP_j, FP_j, TN_j, FN_j). \tag{2.9}$$

The *micro-averaging* of a metric B is defined in Equation (2.10):

$$B_{micro}(h) = B\left(\sum_{j=1}^{q} TP_j, \sum_{j=1}^{q} FP_j, \sum_{j=1}^{q} TN_j, \sum_{j=1}^{q} FN_j\right). \tag{2.10}$$

The macro-average score assigns equal weights to each label without taking the frequency of labels into account, leading to a per-label average [17]. On the contrary, the micro-average score assigns equal weights to every instance, leading to a per-instance average.

2.4 Model Selection

Model selection is an essential part of solving a classification problem. There are endless approaches to choose from when training a classifier, and choosing between them can be a challenge. Badillo et al. [11] define the general principle of model selection in their tutorial. They state that when the data size is sufficient, it should be divided into three subsets: training, validation, and test sets. The training set is used to train different models, while the validation set is used to choose the correct algorithm and, if needed, tune the model's hyperparameters. After the model with the best performance on the validation set is selected, the model is tested on the test set. Evaluating the model's performance on the test set gives an insight into the model's ability to generalize to new, unseen data. If the data set used during training is biased, it limits the model's ability to generalize to unseen data [11]. Therefore, validating the model against a completely independent test data set is essential to gain a reliable insight into its performance.

2.4.1 Bias - Variance Tradeoff

When designing a model, having low bias and low variance is desirable.

- **Bias:** The model should predict values in the test set that are close to the observed values in the training set. If it does not manage to do this, the model is *underfitting*, meaning it has a *high bias*.
- **Variance:** The model should be able to generalize to new, unseen data. If a model has a high performance when predicting values in the training set but poor performance when predicting values in the test set, it is *overfitting*, meaning it has a *high variance*.

Finding the correct balance between bias and variance can be challenging. If the model is too simple or the features extracted are not informative enough, it will not be able to learn relevant patterns from the training data (underfitting). Furthermore, if the model is too complex or contains too many features in a small data set, the model will learn patterns that are too relevant to the training data (overfitting). This issue is commonly referred to as the bias/variance trade-off, as increasing a model's complexity decreases bias but reduces variance [11].

2.4.2 Cross Validation

Cross-validation (CV) is a widely employed resampling technique for estimating a model's true prediction error and optimizing model hyperparameters [39]. It can be especially useful when the data set is too small to divide into an additional validation set [11, 40]. A common CV technique is k-fold CV. First, the data should be divided into training and test sets. Then, the training set is divided into k folds, where $k-1$ folds are used to train the model, and the last fold is used to evaluate the model's performance (validation set). The process is repeated k times, followed by averaging the scores over the folds.

In classification tasks, it is common to split the data for the k folds in a stratified manner, which, compared to standard CV, has been proved to improve both bias and variance [40]. Stratification includes splitting a data set to achieve an approximately equal balance of instances related to the same classes in each fold. In multi-label classification, this can be a challenge. Groups can be formed on the different combinations of labels (labelsets). As previously mentioned in Section 2.3.1, the number of distinct label sets increases exponentially with the set of distinct labels. Sechidis et al. [40] present a stratification algorithm for multi-label data that considers each label independently, called iterative stratification. One label is evaluated for each iteration. The algorithm follows a greedy approach that prioritizes rare labels. It selects a suitable subset for distribution based on each instance (x, Y) of this label.

Chapter 3

Related Work

This chapter briefly presents some of the relevant related work in the field of text classification of clinical notes.

In their study, Apostolova and Velez [6] developed a system for detecting signs and symptoms of infection from free-text nursing notes. They observed that whenever a patient has an infection or a suspicion of infection, the nursing note described that the patient is taking or is prescribed antibiotics for infection treatment. Therefore, they aimed to identify nursing notes that described positive mentions of administered antibiotics to identify notes that are relevant to infection. This approach for identifying relevant notes led to 30% of the nursing notes data set being labeled as relevant (suggesting infection). The remaining 70% were used as the negative data set, i.e., not probable infection. Furthermore, they trained a support vector machine classifier using the automatically labeled data set, resulting in a precision of 93.12% and a recall of 99.04%. To assess the effectiveness of their automated labeling approach, the researchers compared its performance against a separate data set comprising 200 manually labeled test notes. These test notes were randomly selected and annotated by a nurse. While the precision remained high at 92.10%, the recall dropped to 68.46%. Apostolova and Velez [6] found that in most cases that were incorrectly considered as irrelevant by the model, the notes indicated a *low* level of suspected infection.

Ehrentraut et al. [41] perform text classification for detecting hospital-acquired infection using support vector machines and gradient tree boosting. Similar to the long-term problem presented in this thesis, their overall aim is to create a surveillance system that can detect patient records that potentially include infections acquired in the hospital. They use Swedish patient records as a data source. Their best model is a gradient tree boosting model, leading to 93.7% recall, 79.7% precision and 85% F_1-score. They highlight that in the process of screening patient records, a high recall is desirable.

Both of the presented studies have a similar goal to the goal of this thesis: screening clinical notes to identify infections. However, they both use a binary approach, investigating whether a note is relevant, instead of performing multi-label classification to assign multiple categories to a note.

Chapter 4

Data and Resources

This section describes the data and resources used to conduct this project. Section 4.1 presents the available data sets, Section 4.2 presents the clinical resources used, Section 4.3 introduces the annotation tool used for labeling the data, and Section 4.4 presents an overview of the required code implementation resources.

4.1 Adverse Event Data Sets

The data source used in this project consists of free-text Norwegian adverse event (AE) reports. Two main data sets were used: the Norwegian AE data set and the Synthetic AE data set. These are presented in the following sections.

4.1.1 Norwegian Adverse Event Data Set

The Norwegian AE data set consists of 18 555 AE reports extracted from the electronic incident reporting system at St. Olavs Hospital in Trondheim, Norway. These reports were recorded between September 30, 2015, and December 31, 2019. This project used a de-identified version of the records, including placeholders for sensitive information.

The AE reports (notes) describe various non-favorable events in the hospital, such as procedural and guideline deviations, misunderstandings, near-miss events that could have harmed patients, resource needs, and circumstances caused by patients acting out that put other patients at risk.

4.1.2 Synthetic Adverse Event Data Set

The Synthetic AE data set consists of 100 synthetic, manually created notes meant to replicate probable AE notes. The notes were either manually created by a nurse or created by combining existing AE records from the Norwegian AE data set and having them quality-checked by a nurse to ensure they were realistic. A synthetic data set guarantees that the data is completely anonymized, which in turn, makes accessibility

easier. The synthetic AE dataset has been annotated by eight medical professionals over several sessions. This process is described in detail in Chapter 5.

4.2 Clinical Resources

This project made use of several clinical resources that were tailored to the data set in order to annotate the data. These resources are presented in this section.

4.2.1 Annotated Adverse Event NOte TErminology

The Annotated Adverse Event NOte TErminology (AAENOTE)[1] is a terminology used for reasoning about labels provided by annotators within an AE note. The terminology supports annotators during labeling and provides insight into the hierarchy of the labels. This hierarchical structure enables users to adjust the level of detail in the labels, which can be useful for users interested in performing downstream analysis. It is a bilingual resource available in both Norwegian and English. The development of the AEENOTE is described in depth in Yan et al. [9].

4.2.2 Catheter Infection Indications Ontology

The Catheter Infection Indications Ontology (CIIO)[2] accompanies the AEENOTE by representing the clinical insight needed to reason about the presence of a catheter or infection in an AE note, particularly related to PIVC-related BSIs. The development of the CIIO is also thoroughly described in Yan et al. [9].

4.2.3 The Adverse Event Annotation Guideline

The AE Annotation Guideline is a guideline for annotators providing information on how to annotate the AE notes properly. The preliminary guideline was introduced by Yan et al. [7], and at the beginning of this project, it had been revised five times. The fifth version of the guideline was during this project revised to create a sixth version based on the project's classification objective. This process is described in detail in Chapter 5. The revised sixth version of the AE annotation guideline is available in full in Appendix B.

[1] Detailed specifications can be found at: https://folk.ntnu.no/melissay/ontology/aaenote/index-en.html
[2] Detailed specifications can be found at: https://folk.ntnu.no/melissay/ontology/ciio/index-en.html

4.3 BRAT Rapid Annotation Tool

The brat rapid annotation tool (BRAT) is a web-based tool for text annotation supported by NLP technology [42]. This tool was used to annotate the AE notes. The final annotations were stored in the BRAT standoff format.

4.4 Code Implementation Resources

This section provides an overview of the coding resources that need to be installed to replicate the experiments in this project. This includes programming languages, libraries, modules, and any other tools necessary for the implementation and development of the models and experiments.

Basic Packages

`Python` (v.3.8.16) was used as the programming language for the experiments in this project. Furthermore, the libraries `Pandas` (v.1.5.3) and `Numpy` (v.1.23.5) were used for data analysis and computing, and the data visualization libraries `Matplotlib` (v.3.7.1) and `Seaborn` (v.0.12.2) were used for plotting.

Scikit-learn

The open source ML library `scikit-learn`[3] (v.1.2.2) was used as the core library for the ML tools used in this project. `scikit-learn` provides support for a wide range of state-of-the-art ML algorithms for both supervised and unsupervised learning problems [43]. It is based on the `Python` programming language.

The Natural Language Toolkit

The Natural Language Toolkit (NLTK)[4] is a suite of open-source Python modules for NLP tasks, providing ready-to-use computational linguistics courseware [44]. NLTK is widely used for NLP today, but it has limited functionality for Norwegian. It can perform Norwegian sentence segmentation and word tokenization using the multilingual `Punkt` package. The `Punkt` package also includes a list of Norwegian stopwords. Furthermore, NLTK supports stemming in Norwegian, using the `SnowballStemmer`. In this project, version 3.8.1 of the NLTK was used for tokenization and identification of stopwords.

[3] Documentation available at: https://scikit-learn.org/stable/
[4] Documentation available at: https://www.nltk.org/

The Oslo-Bergen Tagger

The Oslo-Bergen Tagger (OBT)[5] [45] is a Norwegian morphosyntactic tagger developed at the University of Oslo and Uni Computing in Bergen. It takes raw text as input and performs sentence segmentation and tokenization. Each word is then associated with all its relevant grammatical tags extracted from the Norwegian lexicon *Norsk ordbank*.

The OBT consists of three main modules:

- A preprocessor with a multi-tagger, which works as a tokenizer, morphological analyzer, and compund analyzer.
- A Constraint Grammar tagger for morphological and syntactic disambiguation, based on the compiler `VISL CG-3`[6], developed at the University of Southern Denmark in Odense.
- A statistical module that removes remaining morphological ambiguities, called `OBT-Stat`.

Every module except for the `OBT-Stat` works for both Norwegian written varieties: Bokmål and Nynorsk. The tagger produces comprehensive grammatical information for each token. While some tokens may have more tags than others, they are all assigned a specific POS tag. Some other common tags include gender, singular/plural form, definite/indefinite form, abbreviation indications, and adjective forms. Furthermore, the tagger can identify the base form of the token, which is equal to the token's lemma.

In this project, the OBT was combined with a third-party Python library called `obt`[7] (v.0.1.0), which parses the output to a Python-friendly format. The `obt` was used to identify the tokens' lemmas.

iterative-stratification

The python module `iterative-stratification`[8] was used to perform stratified cross validation with multi-label data.

The module provides cross validators with stratification for multi-label data, compatible with `scikit-learn`'s cross validators. They are based on the algorithm presented by Sechidis et al. [40], which was presented in Section 2.4.2.

[5] Available at: https://github.com/noklesta/The-Oslo-Bergen-Tagger, with additional documentation at http://www.tekstlab.uio.no/obt-ny/english/index.html
[6] Available at: https://github.com/GrammarSoft/cg3
[7] Available at: https://github.com/draperunner/obt
[8] Available at: https://github.com/trent-b/iterative-stratification

mendelai-brat-parser

A tool called `mendelai-brat-parser`[9] (v.0.0.11) was used to read and parse annotated files in the BRAT standoff format efficiently in Python. The `mendelai-brat-parser` splits the annotated files into dictionaries of entities, relations, attributes, and groups.

[9] Available at: https://pypi.org/project/mendelai-brat-parser/

Chapter 5

Creating the Classification Data Set: The Annotation Process

The records in the Norwegian and Synthetic AE data sets introduced in Chapter 4 were used as the foundation for the data in this project. However, both data sets were unsuitable for performing supervised classification of venous catheters and infections. The Norwegian AE data set consists of pure text notes with no annotations, and the Synthetic AE data set contains detailed annotations on a word- and phrase-level basis. Also, the Synthetic AE data set includes many duplicated notes, as multiple annotators worked with the same raw notes. Therefore, the first part of this work was to process and annotate these two data sets further so they could be used as training and test sets for supervised and semi-supervised classification. This process was done in six steps:

1. **Update Annotation Guideline:** The first step was to create a set of note-level target labels by expanding the annotation guideline used to annotate the Synthetic AE data set. This process is discussed in Section 5.1.2.
2. **Annotate Gold Standard Synthetic Notes:** The second step was to go through all the annotated synthetic notes manually, choose a gold standard for each unique note, and assign note-level target labels to the gold notes. This step is described in detail in Section 5.2.1.
3. **Choose and Annotate Subset of Real Training Notes:** After annotating the synthetic notes, 100 notes were available for supervised training. The amount of labeled training data needed to be expanded, so two meetings were held to annotate note-level target labels for raw, real notes picked from the Norwegian AE data set. These meetings and the note selection process are described in Section 5.2.2.
4. **Choose Subset of Unlabeled Real Training Notes:** In addition to the labeled training data, a subset of unlabeled notes was selected from the Norwegian AE data set. This subset was used as an additional training set for semi-supervised classification. The selection of these notes is described in Section 5.2.3.
5. **Choose and Annotate Subset of Real Test Notes:** A final meeting was held to annotate AE notes from the Norwegian AE data set to be used as a test set. The creation of the test set is described in Section 5.2.4.

6. **Extracting and Merging Target Labels:** Lastly, all annotations were processed to create a final data set for classification containing the raw text and classification target labels. This final step is presented in Section 5.2.5.

The final data set used in this project consisted of 100 synthetic labeled notes, 247 real labeled notes, and 500 real unlabeled notes. 100 of the real labeled notes were used as a test set, while the rest of the data was used for training.

5.1 The Adverse Event Annotation Guideline

The AE annotation guideline was introduced by Yan et al. [7] to guide medical professionals in annotating the synthetic AE data set. It defines the different concepts available for annotation and when they should be used. Also, it presents relationships for linking two concepts together. The original annotation guideline was revised several times. At the beginning of this project, it had reached version five. This section describes the original annotation guideline, the completed annotation sessions, and the development of version six of the guideline, which is further used to annotate the notes for this project.

5.1.1 Previous Work

The preliminary annotation guideline was developed by Yan et al. [7], and this guideline was later used as a framework to develop the accompanying ontology [9]. The preliminary guideline was based on the following domain-specific questions of interest [7]:

- What are the different signs of infections, specifically for BSIs, sepsis, or infected PIVCs?
- What are the signs for different types of catheters?
- Where are the anatomical insertion sites of catheters?
- What events can be related to catheter use?

These domain-specific questions were answered by medical professionals and then categorized, leading to four main categories of interest: **Sign**, **Location**, **Device** and **Procedure**. Additionally, three categories were added to ensure anonymization of the data, to identify actions related to a person, and to have a note-level category for something that is relevant for the whole adverse event. This led to the following seven categories [7]:

1. **Sign (Tegn)**: infection signs.
2. **Location (Plassering)**: anatomical insertion sites.
3. **Device (Enhet)**: signs of catheter types.

4. **Procedure (Prosedyre)**: procedures, interventions, or activities related to catheters.
5. **Sensitivity (Sensitiv)**: protected health information.
6. **Person (Person)**: individuals (i.e., patient, clinician, or relative).
7. **Whole (Hel)**: note-level label indicating whether the note contains infection, BSI, sepsis, faulty device malfunctioning, catheter, PIVC, or sensitive information.

Figure 5.1 [7] shows an overview of the development of the annotation guideline and the annotation process. After developing the preliminary guideline, a nurse generated and validated the synthetic AE data set described in Section 4.1.2. Using the guideline, eight annotators each annotated the 100 synthetic notes over five sessions. One annotator dropped out during session five, so the final data set after five sessions consisted of 770 annotated synthetic AE notes. All notes were annotated using the BRAT rapid annotation tool (BRAT) [42] and stored in the BRAT standoff format.

The guideline was revised after each session based on evaluating the annotations using the inter-annotator agreement (IAA), the F_1-score, and assessing whether the relevant clinical information was captured [7]. All ambiguities and annotator feedback were discussed with medical professionals and incorporated into the revisions [7].

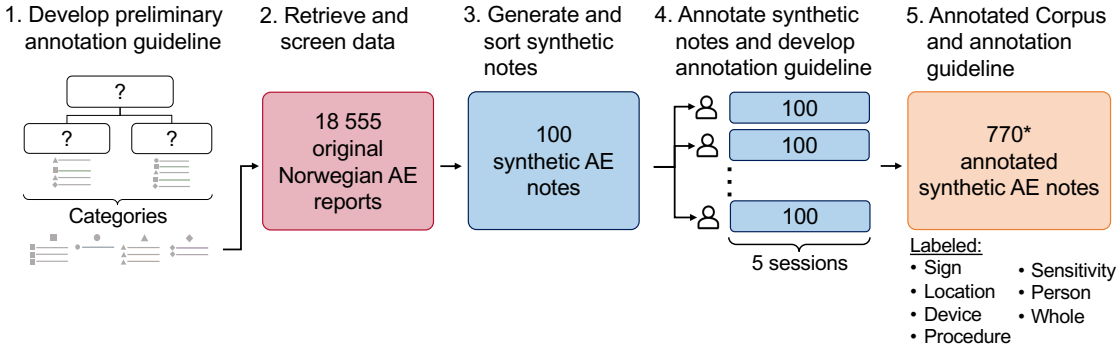


Figure 5.1: Overview of the annotation process during annotation sessions 1 to 5. © 2021 IEEE. Reprinted with permission from Yan et al. [7] and modified to include session 5. *Missing 30 notes because an annotator dropped out.

5.1.2 Development of Adverse Event Annotation Guideline Version 6

Version six of the adverse event annotation guideline was developed by the author and the main thesis supervisor, with feedback from two medical professionals. The main point of improvement was to evaluate the text more thoroughly on a note-level basis. The classifier developed in this thesis (see Chapter 6 and Chapter 7) aims to classify adverse event notes into infection and venous catheter categories. Therefore, the training data must have clear labels indicating whether a complete note is related to these categories. As of version five, there existed a note-level label called **Annotated Note** (previously

called **Whole** in the preliminary guideline). The annotators were instructed to mark the first word of each note with this label and, if relevant, check items listed as entity attributes, which applied to the entire note. The entity attributes were as follows:

- Has identifier (Har identifikator)
- Has medical device malfunction (Har medisinsk utstyrsfeil)
- Is catheter-related (Er kateterrelatert)
- Is PIVC-related (Er PVK relatert)
- Is infection-related (Er infeksjonsrelatert)
- Is BSI-related (Er BSI relatert)
- Is sepsis-related (Er sepsisrelatert)

The entity attributes "Is catheter-related," "Is PIVC-related," "Is infection-related," "Is BSI-related," and "Is sepsis-related" are relevant for training the classifier. However, the results from the five previous sessions show several cases where one annotator checked several attributes while others did not. This irregularity suggests that the annotators often overlooked this step, meaning that an unchecked entity attribute does not necessarily negate the attribute. Also, many annotators only checked the attributes if they were completely evident from the text without making further clinical assumptions.

When developing annotation guideline version six, the primary focus was to improve the **Annotated Note** label to make it very explicit if a note was related to venous catheters and infections. First, six new attributes were introduced and formatted as drop-down menus instead of checkboxes. The updated guideline obliged the annotators to pick one option from each menu, eliminating the doubt of an annotator forgetting to check a box. The new drop-down menu entity attributes are presented in Table 5.1.

The annotated data needed to be clearly labeled as infection-related and venous catheter-related to be used as training data for the classifier. However, as the annotated data is intended to be used for other projects in the future, it is beneficial to keep the labels more detailed to avoid losing important information. Therefore, the annotators are asked about the note's relation to both venous catheters and PIVCs, even though PIVCs are a subcategory of venous catheters. The last two entity attributes in Table 5.1 are connected to the relation between a catheter and infection/phlebitis. This information was added for future projects.

All the entity attributes shown in Table 5.1 have an option that indicates that the note is *probably* related to the topic in question. Including a separate option for when the annotators were unsure instead of only binary options (yes/no) was discussed with two medical professionals. The *probably*-option might be considered a "safe" choice, stopping the annotators from making fair assumptions about the note. However, through observations and discussions, it has been noticed that when annotators are uncertain and can only provide a yes or no answer, they tend to select "no" more frequently. Including an option for uncertainty will catch the edge cases and can be interesting for further

Table 5.1: Overview of the new entity attributes and their selection options.

Drop-down 'entity attribute'	**Selection Options**
Is the note venous catheter-related? (Er notatet venekateterrelatert?)	• No venous catheter (Ikke venekateter) • Probably venous catheter (Sannsynligvis venekateter) • Venous catheter (Venekateter)
Is the note PIVC-related? (Er notatet PVK relatert?)	• Not PIVC (Ikke PVK) • Probably PIVC (Sannsynligvis PVK) • PIVC (PVK)
Is the note phlebitis-related? (Er notatet flebittrelatert?)	• No phlebitis (Ikke flebitt) • Probably phlebitis (Sannsynligvis flebitt) • Phlebitis (Flebitt)
Is the note infection-related? (Er notatet infeksjonsrelatert?)	• No infection (Ikke infeksjon) • Probably infection (Sannsynligvis infeksjon) • Infection (Infeksjon)
Is the note sepsis-related? (Er notatet sepsisrelatert?)	• No sepsis (Ikke sepsis) • Probably sepsis (Sannsynligvis sepsis) • Sepsis (Sepsis)
Does the note suggest a catheter-related infection? (Er det tegn på en kateterrelatert infeksjon?)	• No catheter-related infection (Ingen kateterrelatert infeksjon) • Catheter and infection present, but not related (Kateter og infeksjon tilstede, men ingen relasjon) • Is probably catheter-related infection (Er sannsynligvis kateterrelatert infeksjon) • Is catheter-related infection (Er kateterrelatert infeksjon)
Does the note suggest a catheter-related phlebitis? (Er det tegn på en kateterrelatert flebitt?)	• No catheter-related phlebitis (Ingen kateterrelatert flebitt) • Catheter and phlebitis present, but not related (Kateter og flebitt tilstede, men ingen relasjon) • Is probably catheter-related phlebitis (Er sannsynligvis kateterrelatert flebitt) • Is catheter-related phlebitis (Er kateterrelatert flebitt)

analysis. Nevertheless, the annotators were encouraged to make assumptions and choose the confident option to the greatest extent.

Excluding the annotation guideline version five checkboxes that were entirely replaced by the new drop-down menus, the remaining four checkboxes (i.e., "Has identifier", "Has medical device malfunction", "Is catheter related", and "Is BSI-related") were kept. Additionally, two checkboxes were added based on feedback from the annotators in session five. The two new checkboxes describe a case of human error or a patient fall, which often occurred in the 100 synthetic notes. Figure 5.2 shows a preview of the entity attribute checkboxes and drop-down menus the annotator sees when placing a **Annotated Note** label in BRAT using annotation guideline version six.

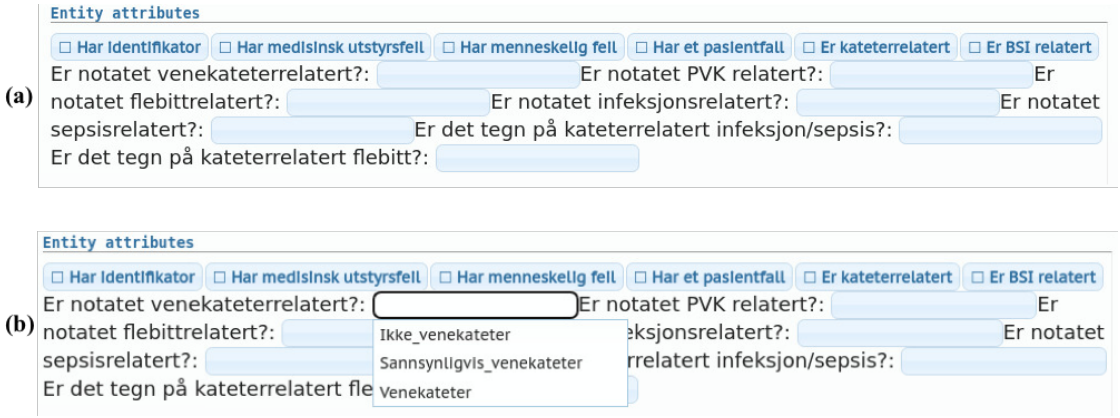


Figure 5.2: Overview of the new **Annotated Note** entity attributes in BRAT. **(a)** Shows the full overview of checkboxes and drop-down menues. **(b)** Shows how the options are presented in an open drop-down menu.

Lastly, additional changes were added to the other label categories based on annotator feedback and irregularities from annotation session five. However, these are irrelevant to this project, so they will not be discussed here. An overview of all guideline changes can be found in Appendix B.

5.2 Creating and Annotating Training and Test Sets

After making an updated annotation guideline that fits the classification problem, the final training and test sets for classification could be created. This section describes the process of selecting and annotating both synthetic and real notes for the data sets.

5.2.1 Selecting Gold Standard Synthetic Annotations

During annotation sessions one to five, eight annotators labeled the same 100 synthetic notes multiple times, resulting in many duplicates. These annotations are slightly

different but use the same note as a base. Having multiple annotations of the same note is beneficial to ensure that the result is unbiased and clinically correct. However, for identical notes with multiple differing annotations, the annotations must be consolidated into a gold standard note to train a classification model because the model should not receive conflicting information. Therefore, the first step of preparing training data for the classification model was to review all existing annotations and choose a gold standard for each text note.

The notes annotated in session five were based on the most recent annotation guideline and, therefore, the best starting point for choosing a gold standard. However, not all 100 notes were annotated in session five. Session five only contained 60 annotated unique notes. Therefore, the remaining latest annotated notes were taken from previous sessions; 30 in session four and 10 in session three. After this initial filtering, each unique note had at least two and at most four annotations to evaluate further. Each annotation was carefully screened and compared before choosing a gold standard. In some cases, a combination of multiple annotations was selected as the gold version.

After identifying the gold standard, all 100 notes were annotated from scratch based on their gold version(s) and the updated sixth version of the annotation guideline. A nurse then fact-checked and revised these annotations to ensure that the annotations were still clinically correct. The final 100 annotated notes were added to the training set for the classification model.

5.2.2 Increasing the Training Data Set

The small size of the synthetic, annotated data set makes it challenging to train a reliable classifier. Therefore, to increase the training data, two annotation meetings were conducted to annotate real adverse event notes from the Norwegian AE data set in collaboration with a nurse. Due to time constraints, the focus of the meetings was only to assign **Annotated Note** labels, including the entity attributes presented in Table 5.1, as these note-level labels contain the information used as target classes for the classifier.

The first annotation meeting was held on the 12th of April, 2023. During this meeting, 100 AE notes were assigned **Annotated Note** labels by the author, following explicit directions from a nurse. The notes to annotate during the meeting were picked randomly from a relevant subset of the Norwegian AE data set. This subset was created by performing a keyword search to filter the original data set, leading to 3309 AE notes. The keywords used to find relevant notes are listed in Listing 5.1.

The keywords in Listing 5.1 include common terms to describe PIVCs (*pvk*, *venflon*, *veneflon*, *venekanyle*), other terms related to venous catheters (*cvk*, *kateter*, *pumpe*, *plaster*, *bandasje*, *seponer*), terms related to infusions (*iv*, *infusjon*, *innstikksted*), common signs of complications around wounds or insertion sites (*hoven*, *rød*, *puss*, *varm*, *smerte*, *siv*), terms related to vital signs (*febril*, *feber*, *frost*), and words to describe infections of interest (*infeksjon*, *sepsis*, *flebitt*).

```
keywords = ["pvk", "venflon", "veneflon", "venekanyle", "cvk",
"infeksjon", "sepsis", "flebitt", "$iv$", "hoven", "rød", "puss",
"febril", "varm", "smerte", "pumpe", "infusjon", "$siv$", "frost",
"plaster", "bandasje", "feber", "seponer", "innstikksted",
"kateter"]
```

Listing 5.1: List of keywords used to filter through the 18 555 adverse event notes in the Norwegian AE data set before the first annotation meeting. Any note that contained any of these words as either a word or substring was included in the subset of filtered notes. The words enclosed in dollar signs ($) limit the search to an exact word match, which was necessary as they are common substrings in Norwegian words. The search resulted in a subset of 3309 notes.

The second annotation meeting took place on the 25th of April, 2023. In this session, 47 new AE notes were assigned an **Annotated Note** label in collaboration with the same nurse as in the first meeting. At this stage, getting more training data related to infections was necessary. The keyword search from the first annotation meeting resulted in many notes related to venous catheters. Thus, a new subset of relevant notes was generated before this meeting, using keywords with only infection-related terms. The keywords used are presented in Listing 5.2. The result was a subset of 1393 relevant notes, excluding the 100 notes annotated in the previous meeting to avoid duplicates. The final 47 notes were picked randomly from this subset of potential infection-related data.

```
keywords = ["infeksjon", "sepsis", "flebitt", "puss", "$pus$",
"febril", "varm", "$siv$","frost","feber", "crp", "blodkultur",
"$bt$", "$sat$", "$rr$", "$resp$", "$sirk$","$rf$", "$temp$",
"puls", "gcs", "$tp$", "$hr$", "blodtrykk", "saturasjon",
"respirasjon","sirkulasjon", "temperatur", "antibiotika", "$ab$",
"benzylpenicillin", "gentamicin", "cefotaksim", "ampicillin",
"metronidazol", "pneunomi", "klindamycin", "cloxacillin",
"kloksacillin", "$uvi$", "stafylokokker", "staph", "aureus",
"epidermidis", "staphylococcus", "streptokokk", "sirs",
"piperacillin", "tazobaktam", "septisk", "news"]
```

Listing 5.2: List of keywords used to filter through the 18 555 adverse event notes in the Norwegian AE data set before the second annotation meeting. Any note that contained any of these words as either a word or substring was included in the subset of filtered notes. The words enclosed in dollar signs ($) limit the search to an exact word match, which was necessary as they are common substrings in Norwegian words. The search resulted in a subset of 1393 notes, excluding the 100 notes that were annotated in the first meeting.

The keywords in Listing 5.2 include all infection-related terms from Listing 5.1 and many new terms. The keywords include:

- Descriptions and abbreviations of vital signs: *febril*, *varm*, *feber*, *frost*, *puls*/*hr*,

blodtrykk/bt, *saturasjon/sat*, *respirasjon/resp/rf/rr*, *sirkulasjon/sirk*, *temperatur/temp/tp*, *sirs*

- Scaling systems used to evaluate patients' vital functions: *gcs*, *news*
- Laboratory results related to infections: *crp*, *blodkultur*
- Signs of infections around wounds or insertion sites: *puss*, *pus*
- Words directly describing infections or inflammations that could be of interest: *infeksjon*, *sepsis/septisk*, *flebitt*, *uvi*, *pneunomi*
- Bacterias relevant to infections: *streptokokk*, *stafylokokker*, *staphylococcus*, *staph*, *aureus*, *epidermidis*
- Antibiotics used for treating sepsis and many other infections (fetched from Helsedirektoratet [46]): *antibiotika/ab*, *benzylpenicillin*, *gentamicin*, *cefotaksim*, *ampicillin*, *metronidazol*, *klindamycin*, *cloxacillin*, *kloksacillin*, *piperacillin*, *tazobaktam*

5.2.3 Choosing an Unlabeled Training Set

In addition to the labeled training notes created in Section 5.2.1 and Section 5.2.2, a subset of raw, unlabeled AE notes were picked from the Norwegian AE data set to be used for semi-supervised training. The subset was generated by combining the keywords in Listing 5.1 and Listing 5.2 to find notes possibly relevant to venous catheters and infections. All notes in the labeled training data set were removed from the subset. This filtering led to 3799 relevant notes, and a random selection of 500 of these was chosen as the unlabeled training set.

5.2.4 Creating the Test Set

A third and last annotation meeting was held on the 24th of May, 2023. In this meeting, 100 new AE notes were annotated in the same manner as in the previous two sessions. However, these notes form the test set used to evaluate the classification model.

The selection process of the notes to include in the test set was more extensive than in the previous annotation meetings. The unlabeled training notes were first removed from the subset used in Section 5.2.3, leading to 3299 notes that were possibly relevant to the classification task and fully unseen by the training data. At this stage, the supervised classification model had been trained on all the labeled training data. The fitted supervised model was used to predict labels for all the 3299 unseen notes. Then, 100 test notes were sampled using a fraction of 25% per label combination. The possible label combinations are:

- Venous catheter / Infection
- Not venous catheter / Infection

- Venous catheter / Not infection
- Not venous catheter / Not infection

This selection led to a test set with a more balanced distribution of target classes than randomly picking notes from the set of 3299, which can be interesting for evaluation. However, some of the classifier's predictions are likely wrong, meaning that this is not a completely balanced data set. The comparison of results from the predicted classification labels and the actual annotated labels from the test set is presented in Chapter 7 and discussed in Chapter 8.

5.2.5 Extracting and Merging the Target Labels

After choosing the final set of labeled training and test notes, the target labels for classification needed to be extracted from the annotations. Only the entity attributes from the **Annotated Note** label were used (see Table 5.1), and the relevant attributes were merged to create two target labels: **Venous Catheter** and **Infection**. Figure 5.3 shows a flowchart of this process for **(a)** venous catheters and **(b)** infections.

To prepare for the merging, the `mendelai-brat-parser` was used to read the annotated files stored in the BRAT standoff format. Then, all the relevant drop-down menu entity attributes were evaluated. A note was assigned the label **Venous Catheter** if it was related or probably related to venous catheters or PIVCs. Similarly, a note was labeled **Infection** if it was related or probably related to sepsis, infection, or phlebitis. This process is shown in Figure 5.3. A note could be assigned to both labels, one or none.

The merging of notes related to venous catheters and PIVCs is clear, as PIVCs are a subcategory of venous catheters. However, merging notes related to sepsis, infection, or phlebitis leads to a broader group. Sepsis is a life-threatening response to an infection [47], directly relevant to the **Infection** label. On the other hand, phlebitis is an inflammation of a vein that can be either infectious, chemical, or mechanical [48, 49]. As stated in the CIIO [9], clinical knowledge is needed to differentiate between the different categories of infusion phlebitis. Still, phlebitis is documented similarly and can be either a catheter-related infection or complication [9]. Cases of phlebitis are therefore regarded as an infection in this classification task.

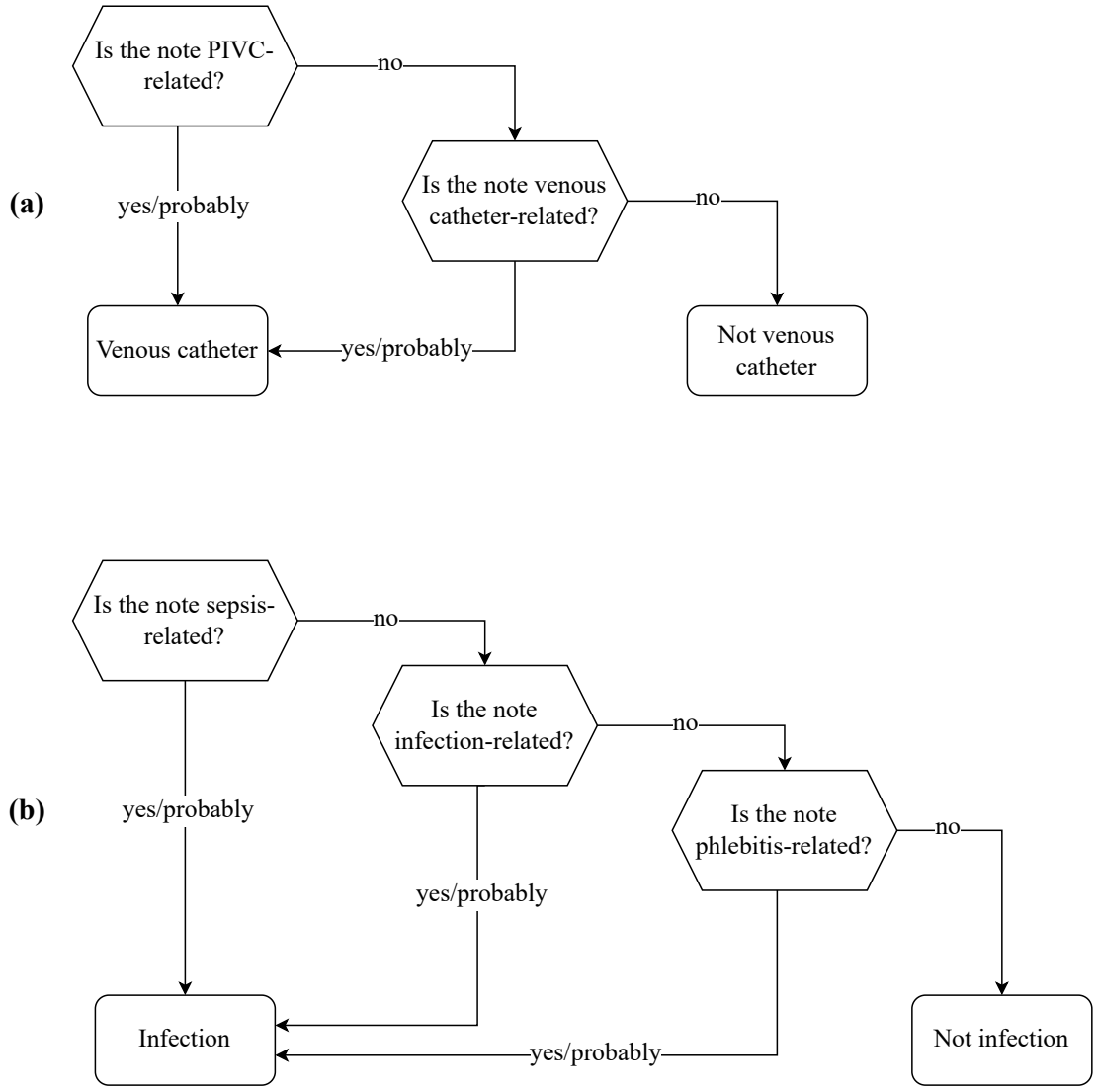


Figure 5.3: Flowchart showing the process of merging target labels for **(a)** venous catheters and **(b)** infections. A note was assigned the label in question as long as one of the relevant target attribute questions had the answer "yes" or "probably".

Chapter 6

Implementation and Architecture

This chapter presents the model architecture and a closer look at the individual steps of the model. The data, referred to as *notes* in Chapter 5, will, from here on, be referred to as *documents*, as they are used as data instances for text classification. Figure 6.1 contains a diagram of the complete model used in this project. The raw input data includes 347 labeled and 500 unlabeled text documents, and the output is the predicted labels for the test data. Sections 6.1 and 6.2 explain the model's pipelines in detail.

6.1 Preprocessing Pipeline

This data is first sent through a preprocessing pipeline, to prepare each text document for classification. The preprocessing is visualized in the "Preprocessing Pipeline" component in Figure 6.1. The first steps were to convert all characters into lowercase and tokenize each document into a list of words. The tokenization was done using the `nltk.word_tokenize()` function, with a parameter specifying that the text was Norwegian.

Next, three text-cleaning steps were performed: cleaning punctuation, cleaning digits, and removing stopwords. The punctuation was cleaned by splitting all tokens containing a punctuation mark into multiple words and removing the punctuation. For example, the token "infeksjon/sepsis" is divided into two tokens, "infeksjon" and "sepsis", and the punctuation mark "/" is removed. However, one exception for the period punctuation mark was added to keep abbreviations, like "i.v.". All tokens containing any digits were completely removed. Lastly, all tokens equal to a stopword were removed. A list of Norwegian stopwords was used as a foundation; some were added and removed to fit the classification goal and medical subject (see Section 8.4). The words "ikke", "ikkje", and "ingen" were removed from the stopword list since they can be relevant to deciding if a venous catheter or infection is present. For example, the phrase "ikke infeksjon" means an infection is not present, while the phrase "infeksjon" alone indicates an infection. The complete list of stopwords is available in Appendix C. The first word of each labeled document was "Hele_Notater", purely used to store the note-level annotations. This word was used to extract the target labels during the data set creation but was removed from labeled documents during this stage, as the word itself was irrelevant to the content.

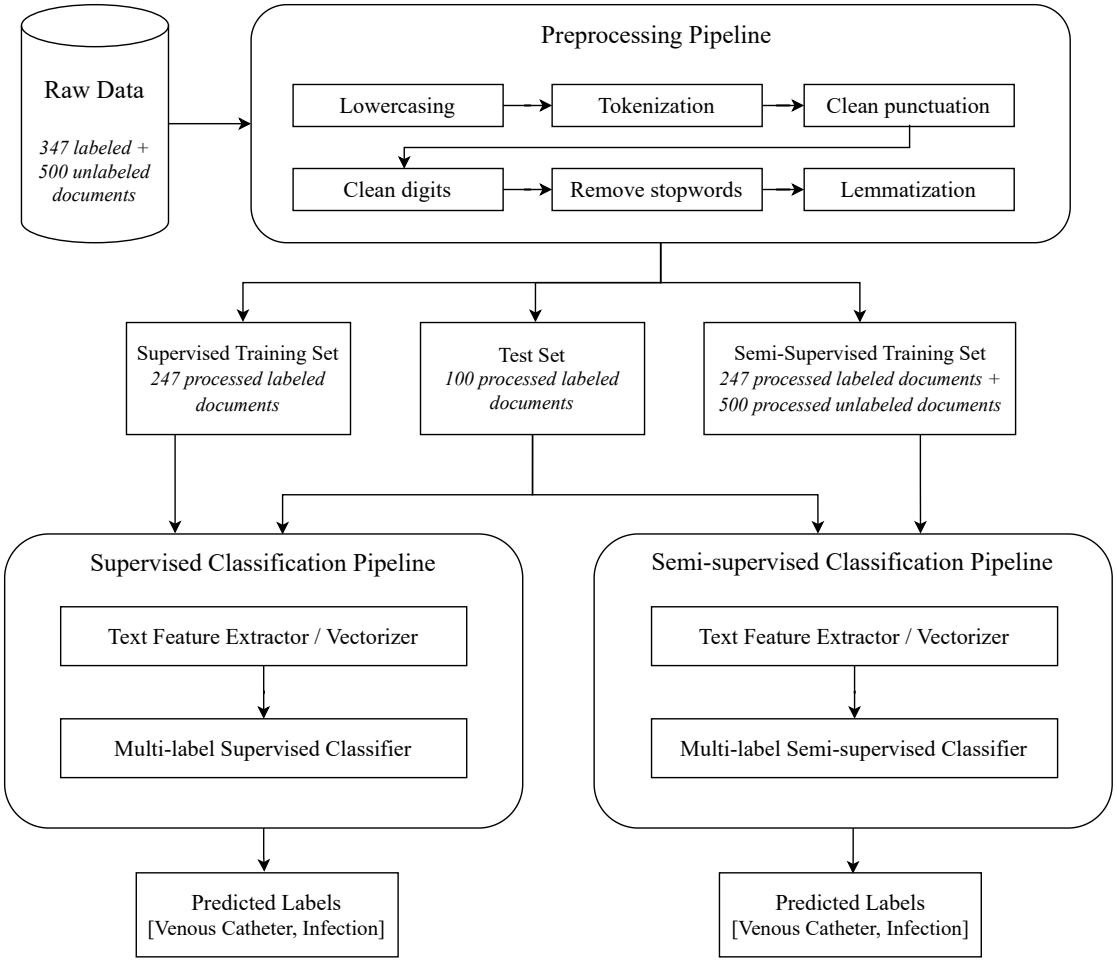


Figure 6.1: High-level overview of whole model, from raw data to predicted labels. The raw text data is first sent through a preprocessing pipeline to prepare for text classification. Then the data is split into training- and test sets and sent through a classification pipeline. Two classification pipelines were used: a supervised and a semi-supervised pipeline, both predicting labels for the same test set.

The last preprocessing step was to perform lemmatization. All lemmas were identified using the `obt.tag_bm()` function, which takes a string of Norwegian text as input, and returns a list of tokenized words with accompanying morphosyntactic tags, including the token's lemma. For each document, the list of tokens from previous preprocessing was combined into a string and sent as input to `obt.tag_bm()`. Then, each token's lemma was extracted from the result and added to a final document containing the processed text. Table 6.1 shows an example of the final preprocessing results from some of the raw synthetic training documents.

Table 6.1: Example results of three synthetic documents before and after going through the preprocessing pipeline.

Raw Document	**Processed Document**
CVK-bandasje ligger særs dårlig på, og det er dermed åpent mellom innstikksted og omgivelser. Langs den ene kanten av den gamle bandasjen er to lag blank tape; et forsøk på å tette igjen uten å skifte CVK-bandasje?	cvk bandasje ligge særs dårlig dermed åpen innstikksted omgivelse langs ene kant gammel bandasje to lag blank tape forsøk tett igjen skifte cvk bandasje
Pas ringte på etter 20 min, og fortalte væske fra infusjonen rant nedover armen og føltes rar.	ringe fortelle væske infusjon renne nedover arm føles rar
Pasient med inf cor inn til PCI, etter 3 dager utvikles feber, frostanfall. Sepsis? Rødt hovent ve arm	inf cor pci dag utvikle feber frostanfall sepsis rød hoven ve arm

6.2 Classification Pipelines

After preprocessing, the data is split into training and test sets and sent through a classification pipeline. Two classification pipelines are used for the experiments, one supervised and one semi-supervised. An overview of the pipelines and their differences is shown in Figure 6.2. Both are trained on the same labeled dataset of 247 documents. However, the semi-supervised pipeline has 500 additional unlabeled training documents. The test set consists of 100 labeled documents and is used to evaluate the performance of both methods.

6.2.1 Feature Extraction

The supervised and semi-supervised approach has the same starting step: text feature extraction. A BoW approach is used through scikit-learn's `CountVectorizer` with the standard 1-grams, which converts a collection of text documents to a matrix of single token counts. This simple vectorization method does not consider relative word positions

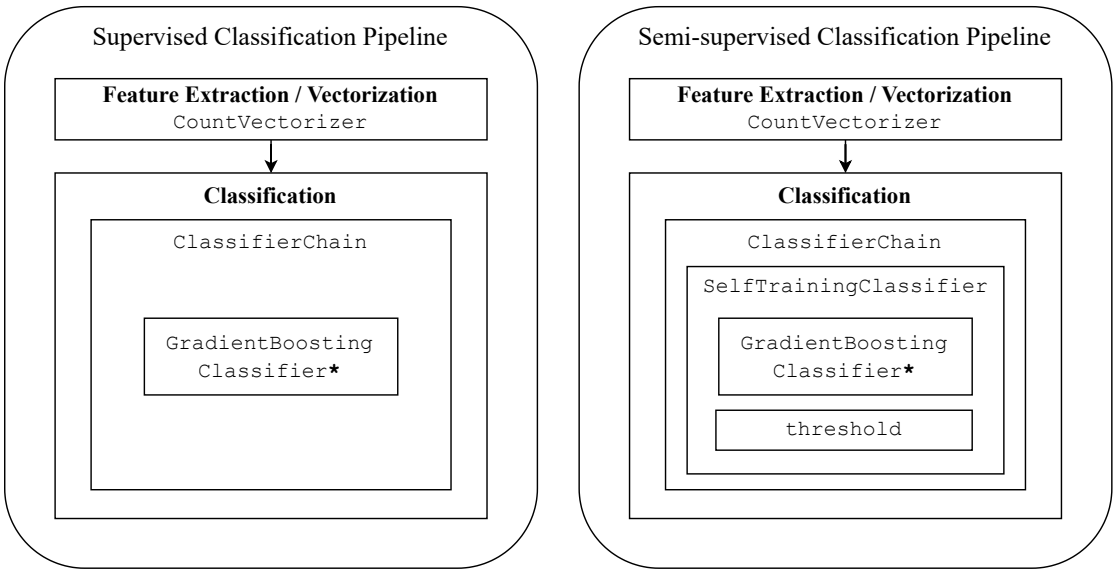


Figure 6.2: Detailed overview of the differences in the supervised and semi-supervised classification pipelines. *Note that the GradientBoostingClassifier was chosen as the final classifier for this model, but several different classifiers were tested during model selection.

or grammar similarities. However, the extensive preprocessing makes this less of a problem. Removing stopwords and other text cleaning removes much of the natural context of the words. Also, as all the words are reduced to their base (lemma), minor grammar differences like different verbal tenses, definite/indefinite articles, and plurals will not impact the word count. For example, the words "armen" and "arm" shown in the two last examples of Table 6.1 will increase the token "arm" count by two instead of counting them as separate tokens, as the base is the same.

Experiments with feature extraction using increased n-grams to maintain some word context and using a TF-IDF vectorizer to score rarer words higher than common words were conducted without any stable performance increase. Therefore, the BoW feature extraction approach was chosen for its simplicity and flexibility.

6.2.2 Classification

After feature extraction, the next step was classification. The target labels for classification were **Infection** and **Venous Catheter**, and this was a multi-label classification problem as a document could be classified as neither, one of them, or both.

As the classifiers used in this model are initially intended for binary classification, problem transformation is necessary. The supervised and semi-supervised approaches both use scikit-learn's `ClassifierChain` as a problem transformation method to transform the multi-label problem into multiple single-label problems. This method was chosen because

it considers label correlations while maintaining the advantages of binary relevance transformation methods, like low memory and runtime complexity.

The classifier chain model wraps around the classifiers, creating a separate chain for each of the two labels. The actual classification per label is performed inside these chains. In the supervised pipeline, Scikit-learn's binary `GradientBoostingClassifier` is directly fitted to the training data and used to predict labels for the test data. However, in the semi-supervised pipeline, an additional step is needed. The meta estimator `SelfTrainingClassifier` by scikit-learn is based on the algorithm presented by Yarowsky [50], and it allows a supervised classifier like `GradientBoostingClassifier` to function as a semi-supervised classifier. It works by iteratively adding pseudo-labels from the unlabeled data set to the labeled data set based on a probability threshold criterion. Only documents with a probability higher than the set threshold for belonging to the label will be included in the training data set. After an iteration, the classifier will be fitted again on the new training data, allowing new unlabeled documents to reach a high probability of belonging to the label and being included in the training data. When either the maximum number of iterations is met, or when there are no more pseudo-labels to add, a final model is fitted on both the original and pseudo-labels and used to predict new labels for the test data.

Part of the experiments described in Chapter 7 includes a model selection phase. In this phase, the base classifier to use in the inner layer of classification is decided by comparing results from multiple classifiers. During the model selection, the `GradientBoostingClassifier` classifier in Figure 6.2 varies between multiple other classifiers, but the other modules remain the same. The other classifiers compared during model selection are scikit-learn's `LogisticRegression`, `MultinomialNB`, `ComplementNB`, `LinearSVC`, `DecisionTreeClassifier`, and `RandomForestClassifier`. The model selection experiments are discussed in Section 7.4.1.

Chapter 7

Experiments and Results

The experiments performed in this project were twofold:

1. **Model selection:** Classification results from seven appropriate supervised classifiers were compared to identify potential issues and choose the best classifier for the problem.
2. **Label prediction:** The best-performing supervised classifier was trained on all available training data and used to predict labels for unseen data. A test set was sampled from the predicted data and manually reviewed by a nurse (see Section 5.2.4). Then, the results from both supervised and semi-supervised classification were evaluated by comparing the predictions to the true, manually reviewed labels.

This chapter describes an exploratory analysis of the data (Section 7.1), the limitations considered prior to the experiments (Section 7.2), the experimental plan (Section 7.3), and the experimental results (Section 7.4).

7.1 Exploratory Analysis

Figures 7.1 and 7.2 provide an overview of the final labeled training data set. Figure 7.1 shows the distribution of label combinations in the data set, indicating an imbalance where the majority of documents are related to venous catheters or have no labels. Figure 7.2 is a scatter plot comparing the label combinations with the document length in tokens. The plot reveals that the average text length across different label combinations is relatively consistent, with most documents being shorter than 200 tokens. However, a few documents stand out as exceptionally long, ranging from 300 to approximately 480 tokens.

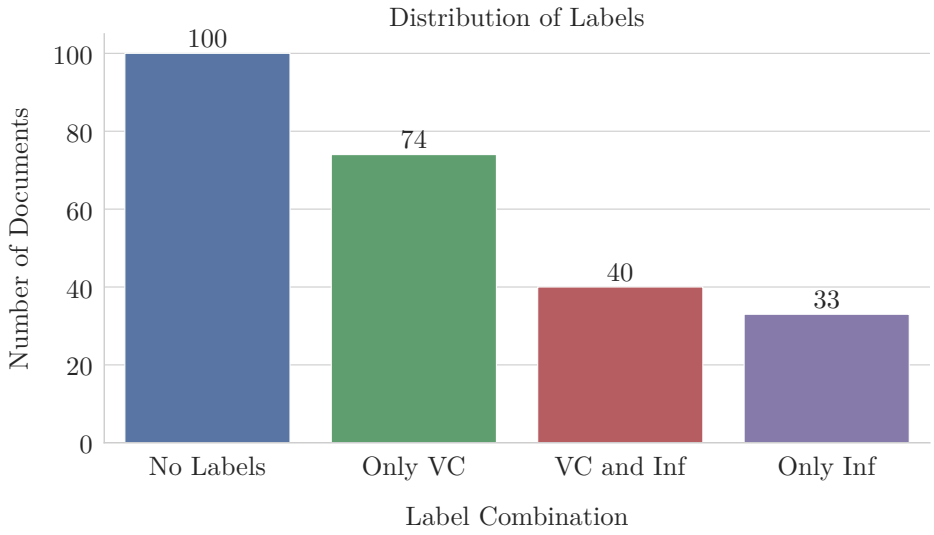


Figure 7.1: Bar plot showing the distribution of label combinations in the final labeled data set. The following abbreviations are used: VC = Venous Catheter, Inf = Infection.

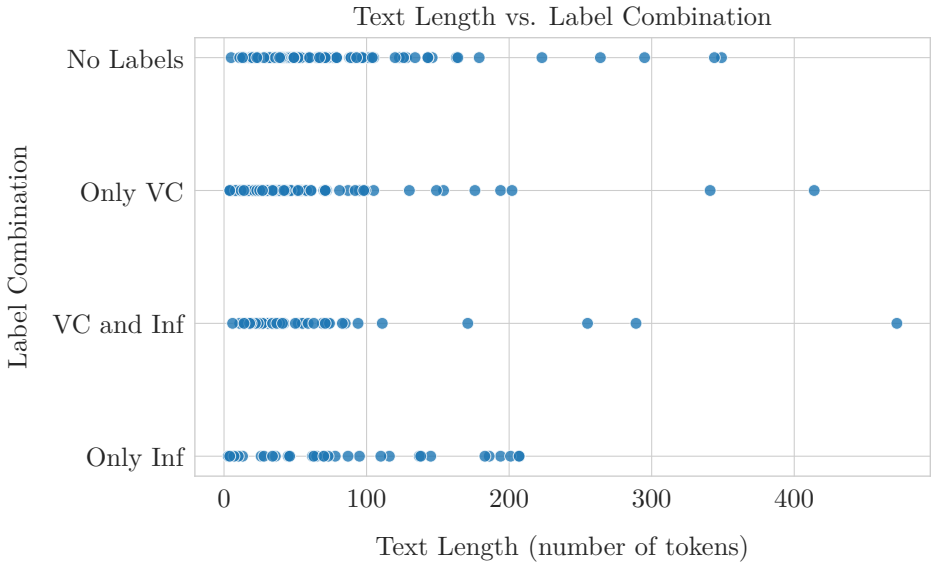


Figure 7.2: Scatter plot comparing the different label combinations with the document length in number of tokens based on NLTK's Norwegian tokenizer. The following abbreviations are used: VC = Venous Catheter, Inf = Infection.

7.2 Experiment Limitations

Before any experiments could be conducted, it was essential to consider the limitations of the data.

One main limitation of performing classification is the small data size. Even with the extensive process of expanding the labeled data, discussed in Chapter 5, the data is very limited in size, with only 347 labeled documents. As mentioned in Section 2.1.1, the performance of a supervised classifier is strongly correlated to the size of the training data.

Another limitation is the complexity of the information contained in the data. Based on experience from the annotation meetings described in Section 5.2.2, even a nurse with a comprehensive clinical background can find it difficult to judge if an AE document is related to venous catheters and infections. In addition to the text contained in the document, the decision is based on clinical knowledge. Some examples of clinical expertise are:

- Understanding the names of different types of catheters to determine if they are venous or, e.g., urinal
- Interpretations of abbreviations
- Differing between antibiotics used for treating infections and antibiotics used as preventive measures
- Understanding if an infusion is venous or subcutaneous based on the specific infusion fluid used

A simple text classification model will learn and increase its knowledge from direct examples in the training data. However, obtaining the clinical knowledge needed purely from the training data is challenging, especially when the labeled training data set is also limited in size.

Furthermore, the challenge of deciding the relevance of each document during annotation leads to some interesting edge cases where one small detail can be enough to place the AE document into a specific category. Another document with very similar information might be assigned a different label. These minor variations make it difficult for the model to learn correctly, especially if the reason for the conflicting labeling is the use of clinical knowledge.

7.3 Experimental Plan

The experimental plan was heavily affected by the limitations of the data set. Initially, the goal was to perform entity-level classification to identify entities related to venous catheters and infections in each document. However, this approach was discarded early

in the project timeline due to the data size and time constraints. The focus was turned to performing note-level classification with two target labels: **Venous Catheter** and **Infection**, based on the information contained in the whole document.

The first step of the experiments was choosing a suitable classifier for the problem, named the model selection experiment. During this experiment, multiple classifiers suited for small data sets were compared. As the data size was limited, the classification results were expected to vary depending on the documents included in the training and test sets. Therefore, all classifiers were trained on multiple splits of the labeled training data set to investigate this hypothesis. The final 100 test data points were not used during model selection as they should not impact the choice of model. Thus, only 247 labeled documents were used at this stage, split into different variations of training and test. This experiment had two main objectives:

- Visualizing how the classification results from a small data set varies with the data set's training and test split.
- Investigating which classifier performs best on average so that it can be used for further experiments.

To further fulfill the last objective of the model selection experiment, 3-fold CV was performed to prevent overfitting and get a more realistic overview of how the different models would perform on unseen data. The results from the model selection experiment are shown in Section 7.4.1.

Initially, the plan also included hyperparameter tuning to identify the best parameters for the classifiers during the model selection experiment. However, this was excluded for two reasons. The small data set made it challenging to divide the already limited training data into training, test, and validation sets needed for hyperparameter tuning. Furthermore, since the results were expected to vary depending on the data split when using a validation set, this could lead to severely overfitted hyperparameters. Ultimately, hyperparameter tuning was not performed, and the default parameters for each classifier were used.

The second experiment planned was to use the best-performing classifier from the model selection experiment to predict venous catheters and infections in an unseen test set. This experiment included training a supervised and multiple semi-supervised models based on the same classifier and comparing the results. The supervised model was trained on all 247 labeled documents, and the semi-supervised models were trained on the same 247 labeled documents and up to 500 unlabeled documents. The test set was chosen during this experiment based on sampling a balanced distribution of label combinations from the supervised model's predictions for 3299 unseen documents. This process has previously been described in Section 5.2.4. This experiment aimed to explore the following:

- **Evaluate how the model performs on unseen data:** Can the model achieve decent results when predicting labels for data it has not seen during the learning process?
- **Compare the results from the supervised and semi-supervised models:** Does including unlabeled data during the training increase the model performance?
- **Compare the choice of thresholds:** How does the selected probability threshold for including pseudo-labels affect the semi-supervised results?
- **Investigate the source of errors:** By taking a closer look at wrongly predicted documents, is it possible to identify some potential challenges for the model?

The results from the label prediction experiment are shown in Section 7.4.2.

7.4 Experimental Results

This section presents the results of all the experiments conducted in this thesis. First, the results from the model selection experiment are presented in Section 7.4.1, intending to identify a suitable base classifier for the problem. Then, the results from applying the best-performing base classifier for label prediction are presented in Section 7.4.2, using both a supervised and a semi-supervised approach.

The tables and figures are introduced and explained in this chapter and further discussed in Chapter 8.

7.4.1 Model Selection Results

The results from the model selection experiment are shown in Figures 7.3 to 7.6. Only the labeled training data (247 documents) was used during this experiment.

All the figures include a selection of "Random States" along the x-axis, ranging from 10 to 100. These numbers indicate the random seed of the data's training and test split and show how the results vary based on the split. Including multiple splits gives a good overview of how sensitive the results are to the split, especially when not performing CV.

The heatmap in Figure 7.3 shows the results from performing supervised classification using seven different classifiers: Logistic Regression (LG), Multinomial Naïve Bayes (MNB), Complement Naïve Bayes (CNB), Linear Support Vector Machine (L-SVM), Decision Tree (DT), Random Forest (RF), and Gradient Boosting (GB). Results from two evaluation metrics, micro-averaged F_1-score and HL, are plotted separately. Each row presents the results from a unique classifier, and each column shows how the classifier's results vary with different training and test splits of the data, using an 80/20 split. A light value indicate a better result for the specific classifier and split.

In Figure 7.4, stratified 3-fold CV is performed for the same classifiers as in Figure 7.3, structured in the same manner. Each cell in the heatmaps in Figure 7.4 presents the *average* score (micro F_1-score or HL) over the three folds. The random states define the random seed used for the stratified 3-fold split, not an 80/20 split as in Figure 7.3.

Figures 7.5 and 7.6 present an in-depth visualization of the results from the best-performing classifier according to the comparison in Figures 7.3 and 7.4, using the same data points. The classifier chosen was the GB, and the reasoning for choosing this classifier is described in Section 8.1.1.

Figure 7.5 highlights the volatility of the supervised classifier's performance depending on the training and test split, and present the mean performance across all the splits.

Figure 7.6 illustrates that the performance is more reliable across the different splits when performing 3-fold CV. Each bar represents the *average* score across the stratified 3-fold split. Figure 7.6 also presents additional information about the standard deviation and mean performance across all the stratified splits.

7.4.2 Label Prediction Results

In the results for the label prediction experiment, GB classifier was used as a base-classifier. The target labels were **Venous Catheter** and **Infection**.

Table 7.1 presents the final results from multi-label text classification using a GB classifier. The table presents results from both a supervised approach and a semi-supervised approach using three different thresholds: 0.6, 0.7, and 0.8. All models were trained on 247 labeled documents, and the semi-supervised classifiers were further trained on up to 500 unlabeled documents.

Figure 7.7 illustrates the confusion matrices for the prediction of the **Venous Catheter** label, while Figure 7.8 displays the confusion matrices for the prediction of the **Infection** label. These matrices are based on the results presented in Table 7.1.

Figure 7.9 showcases the ten most important features for predicting **Venous Catheter** or **Infection** according to the trained *supervised* classifier. The importance assigned to each feature reflects its relevance in predicting the respective label. The sum of importances for all words in the feature space (i.e., all words in the corpus based on the BoW approach) is equal to 1.

Finally, Table 7.2 presents the results from using the same classification approach as in Table 7.1, with the distinction that the models only had the 100 synthetic documents available as labeled training data. The purpose of this table was to demonstrate the advantages gained by augmenting the training data set.

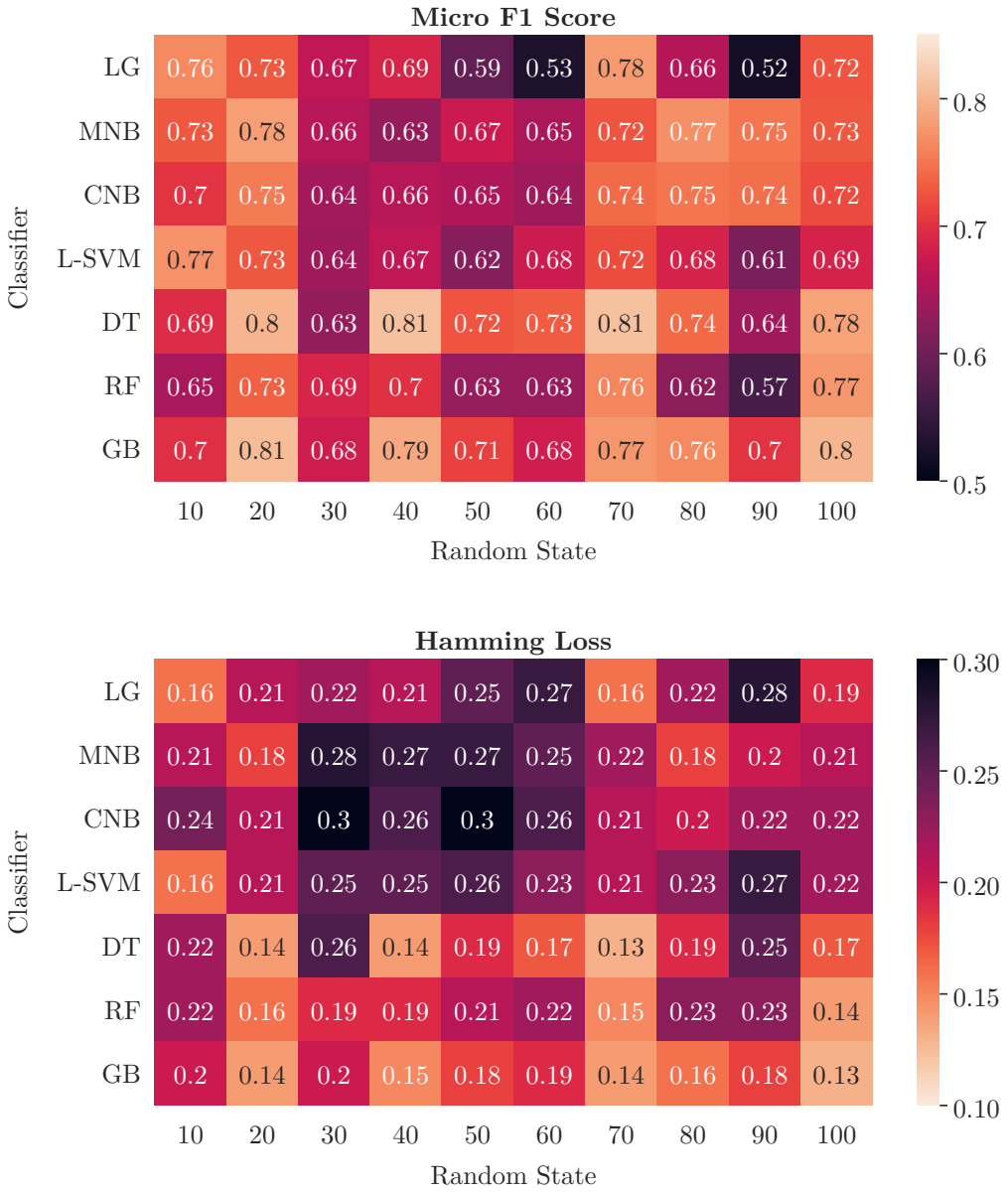


Figure 7.3: Heatmap showing an overview of results from supervised classification using seven different classifiers combined with 10 different 80/20 train-test-splits based on the Random State chosen. Lighter values indicate better performance. Each classifier was trained on 197 labeled documents and tested on 50 labeled documents, but the distribution of the train and test sets vary with each random state. The classifiers compared are Logistic Regression (LG), Multinomial Naïve Bayes (MNB), Complement Naïve Bayes (CNB), Linear SVM (L-SVM), Decision Tree (DT), Random Forest (RF), and Gradient Boosting (GB). Note that the color map is inverted for Hamming Loss to align with the color interpretation for Micro F_1-score.

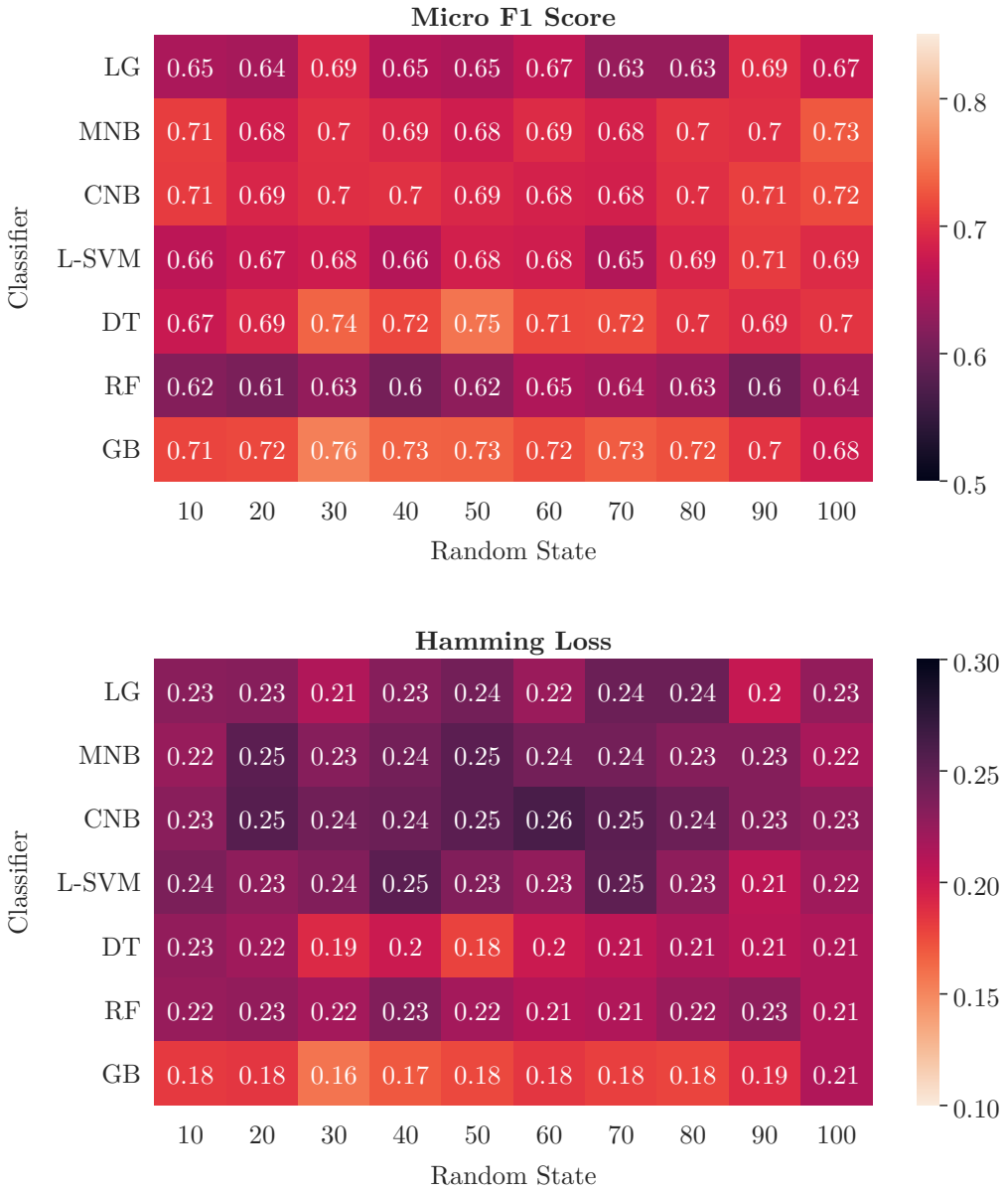


Figure 7.4: Heatmap showing an overview of the average results from performing cross validation on seven different classifiers using 10 different stratified 3-fold splits based on the Random State chosen. Lighter values indicate better performance. Each classifier was trained and tested iteratively over three folds, and the average result from the three folds is shown in the heatmap. The classifiers compared are Logistic Regression (LG), Multinomial Naïve Bayes (MNB), Complement Naïve Bayes (CNB), Linear SVM (L-SVM), Decision Tree (DT), Random Forest (RF), and Gradient Boosting (GB). Note that the color map is inverted for Hamming Loss to align with the color interpretation for Micro F_1-score.

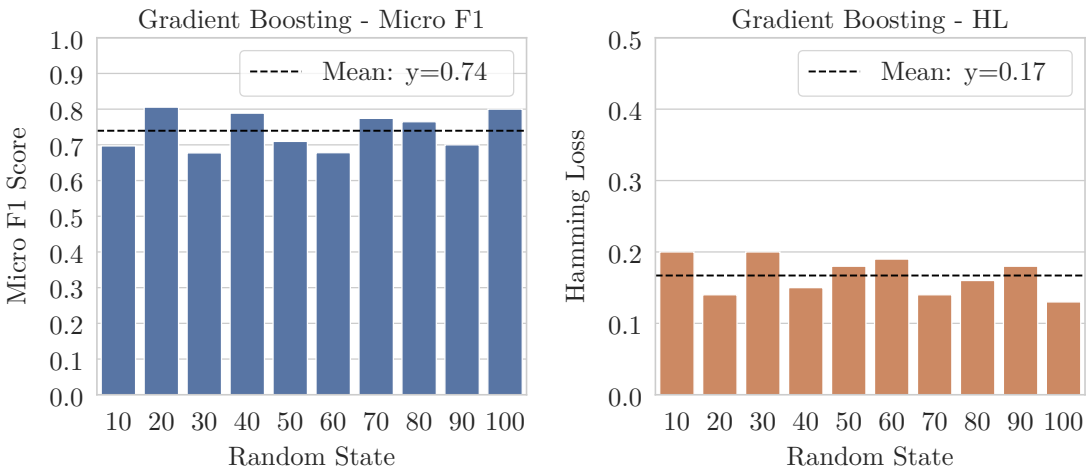


Figure 7.5: In-detail overview of the supervised results from the best performing classifier — Gradient Boosting. These results are identical to the results from Figure 7.3, but only focused on Gradient Boosting. The Micro F_1-score (blue/left) and Hamming Loss (orange/right) are shown in two separate barplots, together with a line showing the mean performance for all 10 random states.

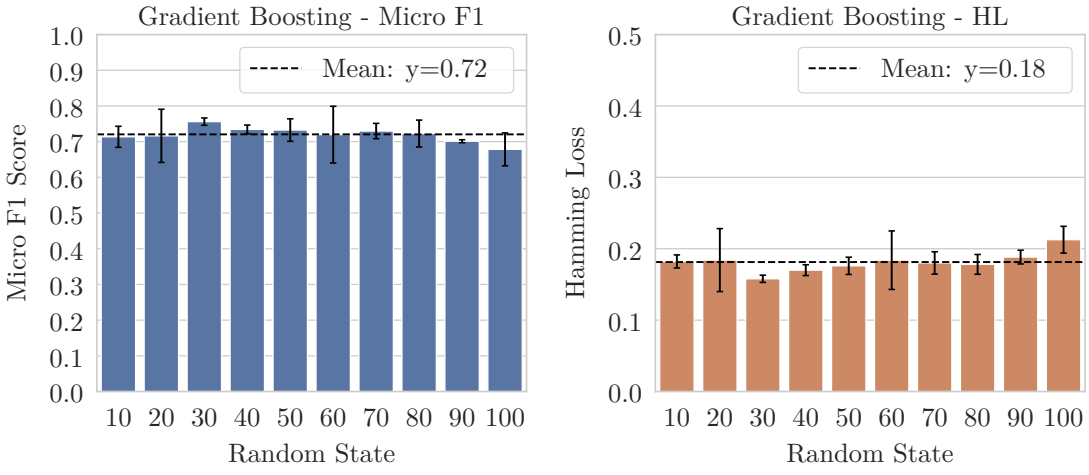


Figure 7.6: In-detail overview of performing 3-fold cross validation using the results from the best performing classifier — Gradient Boosting. These results are identical to the results from Figure 7.4, but only focused on Gradient Boosting. The average Micro F_1-score (blue/left) and average Hamming Loss (orange/right) for the three folds are shown in two separate barplots, along with the standard deviation across the folds and a line showing the mean performance for all 10 random states.

Table 7.1: Results from multi-label text classification of venous catheters and infection, using a Gradient Boosting (GB) classifier as base estimator. The supervised model was trained on 247 labeled documents, while the semi-supervised models were trained on 247 labeled documents and up to a maximum of 500 unlabeled documents. In the semi-supervised models, only the unlabeled documents that had a higher probability than a certain threshold (0.6, 0.7 or 0.8) of belonging to a label were included in the training. All models are applied on the manually annotated test set of 100 AE documents. Abbreviations: GB = Gradient Boosting, HL = Hamming Loss.

Method	**Threshold**	**HL**	**Precision**		**Recall**		**F1-score**	
			Micro	*Macro*	*Micro*	*Macro*	*Micro*	*Macro*
Supervised GB	-	0.180	0.880	0.880	0.786	0.789	0.830	0.830
Semi-Supervised GB	0.6	0.160	0.917	0.918	0.786	0.789	0.846	0.846
	0.7	0.170	0.953	0.953	0.732	0.735	0.828	0.829
	0.8	0.165	0.965	0.965	0.732	0.740	0.832	0.835

Table 7.2: Results from multi-label text classification of venous catheters and infection *using only synthetic data* as labeled training data. A Gradient Boosting (GB) classifier was used as base estimator. The supervised model was trained on 100 synthetic labeled documents, while the semi-supervised models were trained on 100 synthetic labeled documents and up to a maximum of 500 unlabeled documents. In the semi-supervised models, only the unlabeled documents that had a higher probability than a certain threshold (0.6, 0.7 or 0.8) of belonging to a label were included in the training. All models are applied on the manually annotated test set of 100 AE documents. Abbreviations: GB = Gradient Boosting, HL = Hamming Loss.

Method	**Threshold**	**HL**	**Precision**		**Recall**		**F1-score**	
			Micro	*Macro*	*Micro*	*Macro*	*Micro*	*Macro*
Supervised GB (synthetic)	-	0.315	0.930	0.934	0.473	0.463	0.627	0.611
Semi-Supervised GB (synthetic)	0.6	0.350	0.920	0.923	0.411	0.398	0.568	0.540
	0.7	0.320	0.944	0.921	0.455	0.439	0.614	0.577
	0.8	0.295	0.965	0.959	0.491	0.478	0.651	0.835

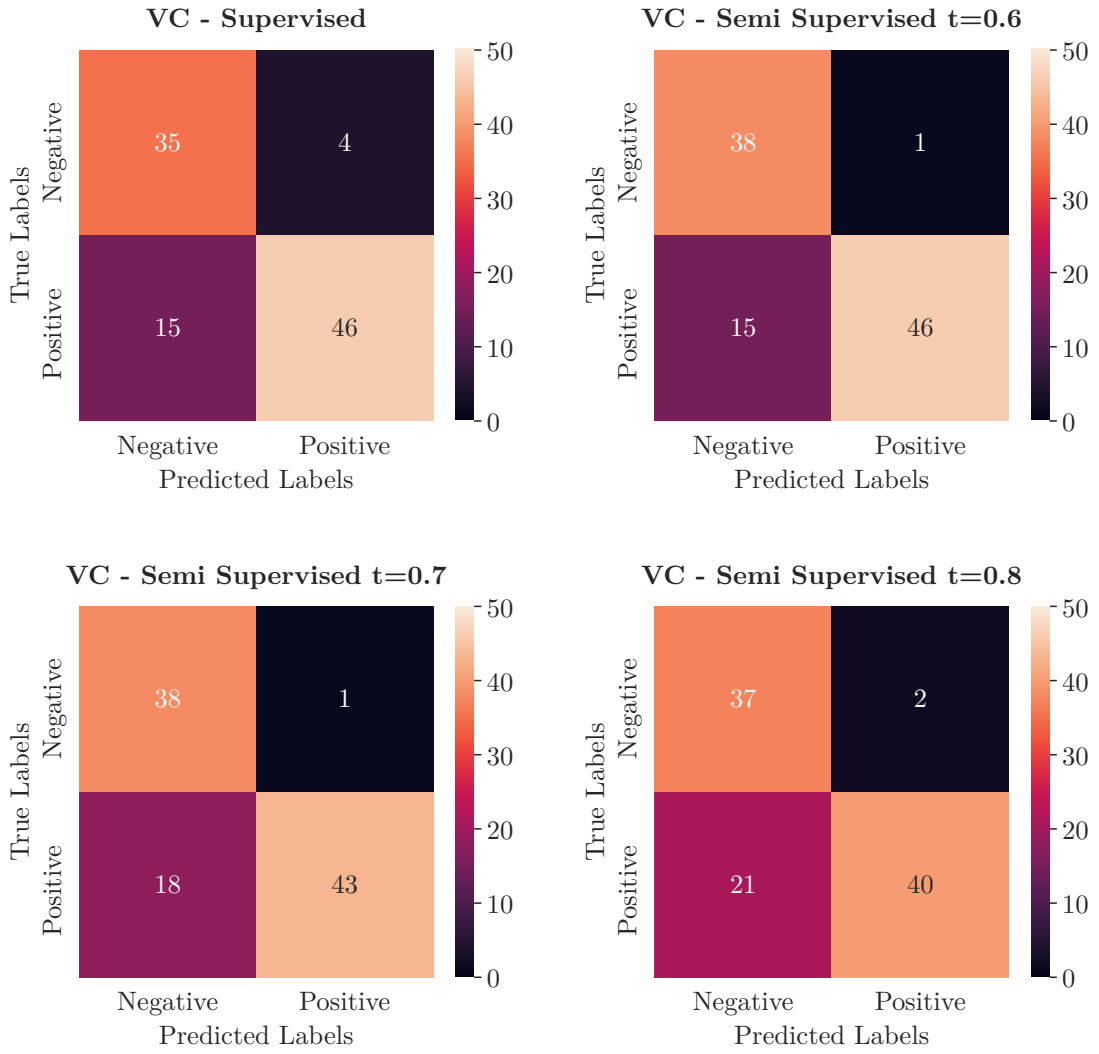


Figure 7.7: Confusion matrices for the **Venous Catheter** (VC) label for all final models in Table 7.1. This includes a supervised Gradient Boosting (GB) classifier trained on 247 labeled documents and three semi-supervised GB classifiers trained on 247 labeled documents and a maximum of 500 unlabeled documents. The threshold (t) decides the probability threshold for including new pseudo-labels from the unlabeled documents.

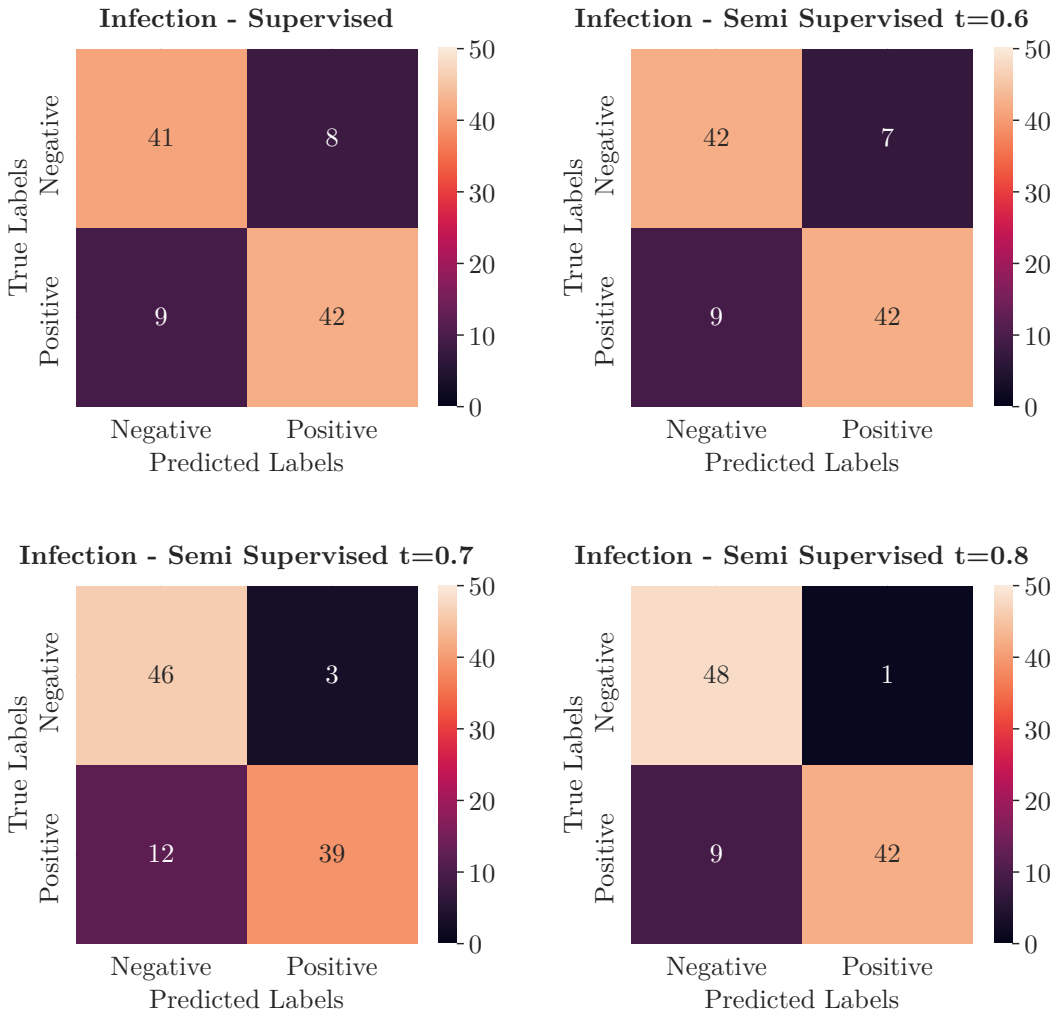


Figure 7.8: Confusion matrices for the **Infection** label for all final models in Table 7.1. This includes a supervised Gradient Boosting (GB) classifier trained on 247 labeled documents and three semi-supervised GB classifiers trained on 247 labeled documents and a maximum of 500 unlabeled documents. The threshold (t) decides the probability threshold for including new pseudo-labels from the unlabeled documents.

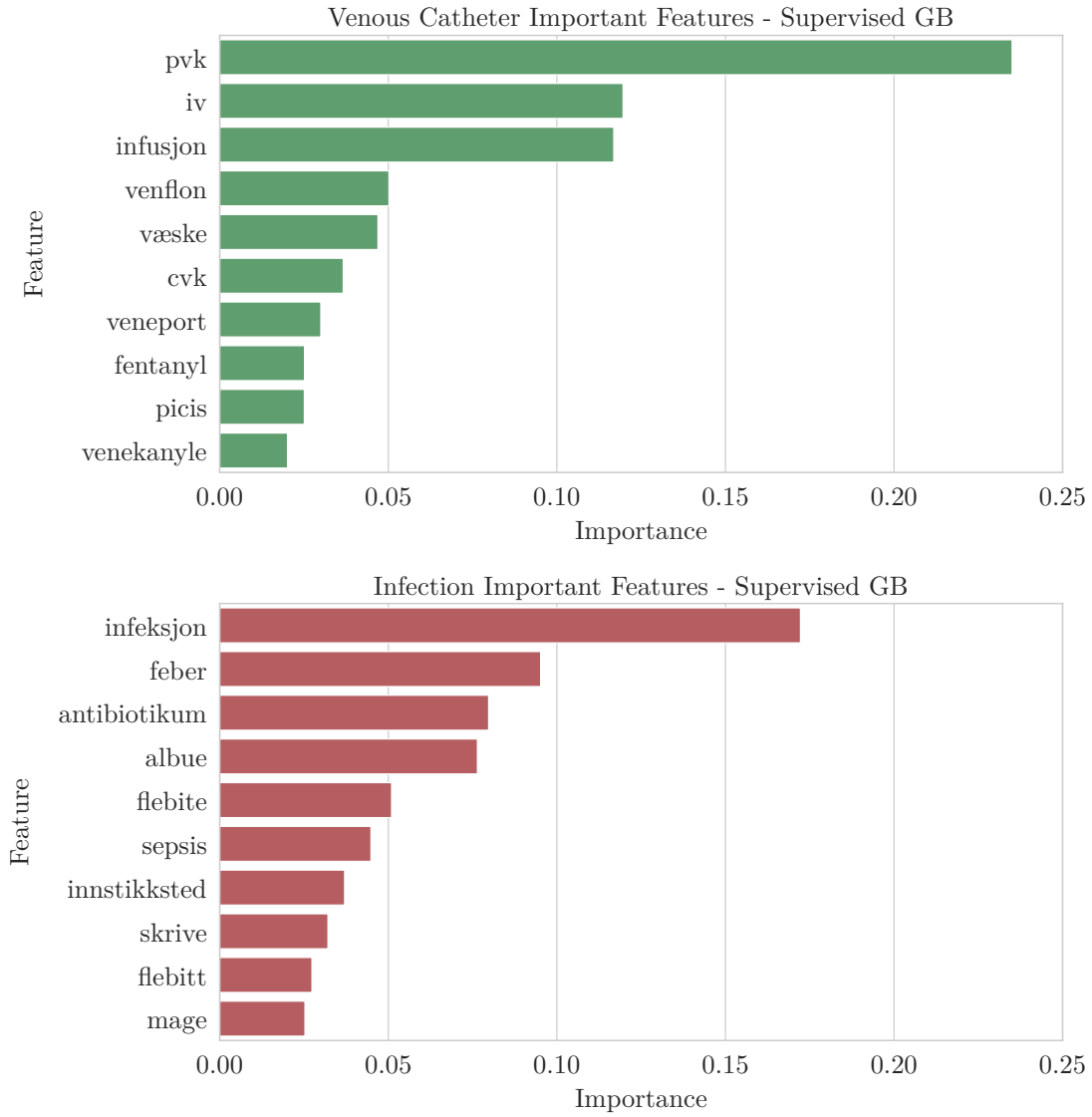


Figure 7.9: Overview of the top 10 most important features for **Venous Catheter** (green/top) and **Infection** (red/bottom) based on the *supervised* GB classifier in Table 7.1. Each unique token in the data set is regarded as a feature (BoW approach), and the importance of all these available features add up to 1. Higher importance means that the feature is more relevant for predicting the label in question.

Chapter 8

Evaluation and Discussion

This chapter provides an evaluation of the results presented in Section 7.4, followed by an in-depth discussion of the limitations and choices made during the experiments. It also explores potential improvements for the created models and the annotation guideline. Analyzing and addressing these aspects gives a comprehensive understanding of the project's feasibility for future work.

8.1 Evaluation of Results

In this section, the results presented in Section 7.4 will be evaluated in detail. Section 8.1.1 discusses the results and reasonings from the model selection experiment, and Section 8.1.2 discusses the results from the label prediction experiment, including an error analysis.

8.1.1 Model Selection Evaluation

The objective of the model selection experiment was to identify a reliable classifier for the final label prediction experiment. This involved testing several classifiers using only the labeled training data to determine their performance and suitability. The classifiers compared in this experiment were Logistic Regression (LG), Multinomial Naïve Bayes (MNB), Complement Naïve Bayes (CNB), Linear SVM (L-SVM), Decision Tree (DT), Random Forest (RF), and Gradient Boosting (GB).

As the training data consists of only 247 labeled documents, the performance of a classifier was expected to vary based on the training and test split of the documents, as discussed in Section 7.3. The heatmap in Figure 7.3 aims to visualize this behavior by showing results from seven different classifiers using different splits of the data sets. Each classifier in this heatmap was trained on 80% of the training data and tested on the remaining 20%. It is clear from the results that the results vary a lot based on the seed (called random state) chosen to split the data. The heatmap also shows that several splits were notably better than others. In the "good splits", most classifiers performed better than average. These good results can be seen in splits with random states equal to 20, 70, and 100.

Furthermore, in "bad splits", the classifiers performed worse than average. Examples of "bad splits" are splits with random states equal to 30, 60, and 90.

When comparing the classifiers in Figure 7.3 against each other, the best classifiers seem to be the DT classifier and the GB classifier. However, the models needed to be evaluated using CV to make a final decision, as it is clear that dividing into a separate validation set here leads to very volatile results due to the small data size. Performing CV leads to better insight into the models' performance as every part of the small data set is used for training and testing.

Figure 7.4 shows a heatmap containing the results of performing 3-fold CV on the seven different classifiers. Each cell in the heatmap displays the average micro-F_1 score or HL over the three folds. It is crucial to note that even though the values of the random states are the same as in Figure 7.3, they do not reflect the same splits. The random states are here used to split the data set into three stratified splits for each model instead of splitting the data into training and test sets with a distribution of 80/20. Performing stratified 3-fold CV led to a more stable performance for each classifier.

As mentioned in Section 2.3.3, the micro-averaged F_1-score is a label-based evaluation metric that considers each label independently. On the other hand, the hamming loss is an example-based metric that considers the correlation between the labels. These metrics can therefore produce different insights into the results for the same model. The CV heatmap in Figure 7.4 shows that the GB classifier performs best when considering both micro F_1-score and HL.

Figures 7.5 and 7.6 provide further insight into the performance of the GB classifier. These figures are based on the same data as in the heatmaps, focusing only on the GB classifier. Figure 7.5 shows a bar plot of the GB results from performing supervised learning. It reveals that while the performance is volatile based on the split, the GB classifier repeatedly produces a high micro F_1-score with a mean of 74% and a low HL with a mean of 0.17. Figure 7.6 shows a bar plot with the results from the stratified 3-fold CV in more detail, including the standard deviation of the measures for each fold. The stratified splits with random states equal to 20 and 60 have a higher standard deviation, but it is relatively low on average. The performance of the GB classifier using CV is still high, with a mean micro F_1-score of 72% and a mean HL of 0.18.

Hyperparameter tuning might have changed the results of the most suitable classifier, and it should have been performed. However, as discussed in Section 7.3, this was not feasible because of the small data size and the possibility of overfitted hyperparameters.

Ultimately, the GB classifier was chosen as the final classifier for further experiments due to its consistent and high performance during the model selection experiments. The high performance was observed across different training and test splits using regular supervised learning and during the 3-fold CV process. Because of the small data set and the lack of hyperparameter tuning, this choice is based on the available resources at the time of the project, and another classifier might be more suitable for a future approach.

8.1.2 Label Prediction Evaluation

The overview in Table 7.1 summarizes the results from predicting venous catheters and infections in the unseen test data set using the augmented labeled training data and up to 500 unlabeled documents. Based on the results, the semi-supervised models outperform the supervised model in terms of HL and micro/macro average precision. This result suggests that compared to the supervised model, the semi-supervised models with thresholds 0.6, 0.7, and 0.8 all reduced the number of incorrectly predicted labels and had a higher frequency of correctly predicted labels. A more detailed overview of this is provided by evaluating the confusion matrices in Figures 7.7 and 7.8, which show that the number of false positives was reduced for both venous catheters and infection in the semi-supervised models. On the other hand, the number of true positives was either equal to the supervised model or slightly reduced.

While the precision increases for the semi-supervised models, the recall decreases for the models with thresholds of 0.7 and 0.8. By evaluating the confusion matrix for the venous catheter label in Figure 7.7, it is clear that the number of false negatives increased in line with a higher threshold for including pseudo-labels. The models with a high threshold will only add new labels to documents if they are very confident that it belongs to the category. These confident documents likely contain highly impactful features. An overview of the most impactful features of the labels **Venous Catheter** and **Infection** for the supervised model is given in Figure 7.9. The top two features of the label **Venous Catheter** are the words "pvk" (PIVC) and "iv" (short for intravenous). These are both undeniable signs of venous catheters. Similarly, the top feature for predicting infections is the word "infeksjon" (infection), which is the exact word of the category to predict. Adding more documents containing these obvious features to the training set will further increase their importance, which could lead to more difficulty in identifying the correct labels of uncertain documents that do not contain them, thus lowering the recall.

Overall, the low recall is a central weakness of all the models, suggesting that they incorrectly assign no labels to documents related to venous catheters, infections, or both. As the data is highly complex and not easy to label, even by a medical professional, many edge cases based on assumptions can be challenging for the models to catch. Something that further extends this problem is that any document labeled as *probably* related to a category was included as relevant in the training data set. This choice was deliberate to avoid only including documents where it was straightforward to assign a category, i.e., in documents that directly mentioned words like "pvk" (PIVC) or "infeksjon" (infection), as discussed earlier. However, it does add complexity to the predictive performance. Examples of two highly uncertain documents are given below. Both documents are presented in their original Norwegian forms but explained in English. They have been entirely de-identified for privacy reasons.

Document 1:

> *U.t hadde vært inne på et annet pasientrom i kveldsstell, oppdaget da h*n kom ut at pasienten ikke lå i sengen og at kateterslangen til pasienten hang fast i senga og strakk seg inn på badet. Der var døren lukket, og da personalet kom inn på badet lå pasienten på gulvet. H*n hadde da reist seg og gått alene uten rullator og falt. Pasienten har i hele dag blitt oppfattet som delirisk, da h*n har hallusinert og vært svært urolig. Dette har også vært tilfellet i kveld.*

Document 2 **(extract)**:

> *Tok over en pasient til natten som var delirisk og motorisk urolig. Planen for natten var å prøve Haldol igjen (noe som ikke hadde hatt effekt natten før), deretter Dexdor om ikke Haldol fungerte. Pasienten fikk 5 mg Haldol, og ble mere urolig. Det ble da bestemt å starte med Dexdor på sengeposten. (...)*

The first document concerns a patient in a delirious state, meaning that he/she is in an acute state of confusion and disorientation, often combined with hallucinations [51]. During the annotation meetings, it was discussed that delirium is often a result of an infection. This patient was assessed as an older person, as it is described that he uses a walking frame. The document also described the patient's catheter tube being stuck in the bed and dragged to the bathroom. This catheter could be either venous or urinary. The nurse concluded that a venous catheter was probable, but it was noted that it could be both. The document was ultimately labeled as probably infection-related and probably venous catheter-related.

The second document also describes a patient in a state of delirium. However, in this case, it was not labeled as infection-related, based on insufficient information to make a judgment. The patient was given the medications Haldol and Dexdor, which are given intravenously, meaning that the document is related to venous catheters.

In summary, these documents have similar topics but are labeled differently due to slight variations in the information provided. If it is tricky for a human medical professional to assign labels to an uncertain document (i.e., not including highly relevant terms like PIVC and infection), it will also be challenging for the models. The lower recall is, therefore, understandable. The problem of bias related to low recall is discussed in Section 8.2.1.

Something to be considered is the importance of precision and recall in this problem. The fact that when the model makes a prediction, it is usually correct (high precision) is beneficial. However, if the model only assigns labels to documents that contain obvious category-specific words like PIVC or infection, the task could have been solved just as well by a keyword search. Correctly identifying the majority of documents relevant to a category (high recall) is just as important, if not more important, than having high precision.

Suppose one also considers that the model is meant to serve as a foundation for identifying catheter-related complications in AE documents in the future. In that case, the need for

high recall is even more prominent. Non-relevant documents would be quickly discarded when performing post-processing reasoning, but documents that have not been identified as relevant will not be available for further evaluation.

Table 7.2 presents the results from predicting venous catheters and infections in the unseen data set when only using the 100 synthetic documents as a labeled training set. This table was added to compare how the performance of the classifiers is affected by the labeled data size. The results show that the model with the larger training set outperforms the model using only synthetic training data in all measures except for precision. The impact of increasing the labeled training data is further discussed in Section 8.2.2.

Error Analysis

It can be interesting to take a closer look at some of the errors in the models' predictions to evaluate if the error is due to high uncertainty of belonging to a category or an immediate problem in the model architecture. Table 8.1 presents an overview of five hand-picked prediction errors in the *supervised* model with extended training data. Words that are important for the analysis of the error are highlighted. The following paragraphs provide an analysis of these errors in chronological order. The clinical reasoning is based on the discussions with a nurse during the annotation meetings.

Table 8.1, Document 1: This document was related to both venous catheters and infections but was classified as neither. Both Keflin and Diclocil are names of antibiotics used to treat an infection. They are given intravenously, which suggests that a venous catheter is used. The annotator nurse noted that the antibiotics are used pre- and post-surgery in this document, which could mean they are used as a preventive measure. However, they are likely used to treat an infection. This document is an excellent example of a case where it is not directly clear from the document that the patient has a venous catheter or infection, but it is suspected based on clinical knowledge.

Table 8.1, Document 2: In this document, the model has predicted an incorrect case of infection. This prediction is probably due to the redness and irritation (*rødhet/irritasjon*) at an insertion site (*innstikksted*), which often suggests phlebitis. However, in this document, the nurse decided there were not enough signs to label the document as phlebitis. It is an understandable error by the model. This document also contains a de-identification error. The phrase *"PICC-Nordmann_placeholder"* has a placeholder to remove sensitive information. It is likely that this phrase originally was *"PICC-line"*, which is a type of venous catheter. "Line" is a Norwegian first name, so this might have been the reason for the de-identification error. The document is related to venous catheters, but the model was unable to catch this. De-identification errors like this are problematic since they remove essential information from the data set.

Table 8.1: Error analysis for five hand-picked incorrectly predicted documents in the test set by the *supervised* Gradient Boosting classifier. The table presents five raw documents with their accompanying prediction errors. Important phrases for error analysis are highlighted in ***red Bold-Italic***. The labels have been abbreviated for readability: Venous Catheter (VC) and Infection (Inf). Correctly predicted labels for the presented documents are not included in this table.

Raw Adverse Event document (Norwegian)	**Supervised Gradient Boosting Error Results**		
	Predicted, Not True	**True, Not Predicted**	**Type of Error**
Det var forordnet ***Keflin*** på pasientens kurve preoperativt. Pga noe som kunne være en reaksjon på ***Keflin*** ble dette endret til ***Diclocil*** postoperativt. Dette ble ikke ført på kurven, og ***Keflin*** ble heller ikke seponert på kurven. Leste i operasjonsbeskrivelsen at pasienten skulle ha ***Diclocil***, men der sto det ingen dosering. Dosen sto imidlertid på anestesijournalen.	None	VC, Inf	False Negative VC, False Negative Inf
Pasienten kommer til Lungemedisinsk dagenhet for cellegiftbehandling. Har ***PICC-Nordmann_placeholder kateter***. Åpner opp til kateteret, og observerer ***rødhet/irritasjon ved innstikksted*** samt at det ikke ligger på en semipermeabel bandasje (noe som det etter prosedyren skal gjøre).	Inf	VC	False Negative VC, False Positive Inf
U.t. oppdaget på starten av vakten at pasientens kveldsdose med ***i.v.*** antibiotika ikke var blitt gitt.	None	VC	False Negative VC
Dyp ***sårinfeksjon*** etter dekompresjon ve. L5 rot.	None	Inf	False Negative Inf
(extract) Pasient i palliasjon pga ca pulm. Smerteproblematikk, med behov for ***spinalkateter***, samt i tillegg ***sc morfininfusjon***. Har vært fulgt av smerteteamet.Det var planlagt visitt fra smerteteamet på denne pasienten denne dagen. (...) Sykepleiere på sengepost ønsket at vi førte ny hovedkurve mht ***subcutan morfininfusjon***, samt at vi også laget en bestilling på CADD-pumpe for ***spinalkateterinfusjon***. (...) Apoteket blandet det som var bestilt, og i neste feil glemte sykepleiere på post å kontrasjekke innholdet på pumpen mot det som står i hovedkurve (der doseringen er oppført riktig), slik at det i ***infusjonen*** intrathecalt var 10 x mer morfin enn ønskelig. (...)	VC, Inf	None	False Positive VC, False Positive Inf

Table 8.1, Document 3 and 4: Both document 3 and 4 illustrate cases of clear model-related errors. The third document is related to an intravenous infusion because of the abbreviation "i.v.". This is a common abbreviation that the model should have been able to learn. It is even included in the top ten most important features for predicting venous catheters, shown in Figure 7.9. The problem is that most documents include the abbreviation without period marks, leading to the simpler phrase "iv". Even though these phrases are highly similar, they are regarded as entirely different features during analysis because of the BoW feature extraction approach. An error like this suggests that using a feature extraction method that can combine these phrases based on their similar semantic meaning could have helped reduce errors. Document four shows a similar scenario, where the phrase "wound infection" (*sårinfeksjon*) is not understood as an infection, even though the very similar word "infection" (*infeksjon*) is the top important feature for predicting infection according to Figure 7.9.

Table 8.1, Document 5: This is an example of a very long and confusing document where only an extract of the complete document is included. The document mentions different types of infusions several times (*morfininfusjon*, *spinalkateterinfusjon*, *infusjonen*), which is a common feature of venous catheters, according to Figure 7.9. However, these infusions are subcutaneous infusions (*sc*, *subcutan*) and spinal infusions (*spinalkateter*), which are not connected to any veins. The model predicted this document as related to both venous catheters and infections. However, it is related to neither.

8.2 Data Limitations

This section discusses some significant limitations of the data set used to train the models. The issues discussed here are irrelevant to the models' architecture and implementation. Nevertheless, issues with the data set can greatly impact the models' performance.

8.2.1 Data Bias

One of the primary limitations of the final data set used for training the models is its pronounced bias. Two key types of bias that are important for evaluating the data are annotator bias and sample bias. These biases are discussed separately in this section to highlight their respective impacts on the data.

Annotator Bias

Annotator bias is caused by the annotators' subjective understanding and different knowledge levels regarding an annotation task [52]. This type of bias was introduced in

the final data set while selecting the gold-standard annotations and during the annotation meetings.

The synthetic AE data set was initially annotated by eight annotators over multiple sessions, as described in Section 5.1.1. During this thesis project, these annotations were combined into one set of gold-standard annotations to avoid duplicates. With no formal clinical knowledge, the author chose the gold-standard annotations by comparing the most recent ones. This process introduces an annotator bias. A qualified nurse carefully fact-checked the final gold documents to limit this bias.

During the annotation meetings, all documents were labeled by a qualified nurse. However, as previously discussed, it can be challenging to label the documents as the presence of labels is often based on clinical assumptions. An annotator bias is introduced as the nurse's subjective understanding of the document is bound to affect the final labeling of the data. Including an additional medical professional in the meetings' discussions could have reduced this bias. Alternatively, the documents could have been labeled twice by two different medical professionals and compared. If humans with extensive clinical knowledge can not agree on the labeling, the problem might need to be refocused.

Even with only one annotator, the data set suffers from inconsistent labeling for similar documents. This inconsistency arises from the complex nature of the data, making it challenging to achieve accurate and consistent labels. An example of such inconsistent labeling was provided in Section 8.1.2, and the data set contains more instances of this issue. Consequently, this leads to an absence of a reliable ground truth, making it harder for the classifier to generalize accurately. This incompleteness in the ground truth can lead to low recall as the classifier might miss relevant instances that were not consistently labeled.

Sample Bias

All the documents in the final data set are affected by sample bias. Medical professionals specifically designed the synthetic documents to include many mentions of venous catheters and infections to prepare for developing classifiers to identify PIVC-related BSIs. The number of documents directly referring to venous catheters and infections in the synthetic documents is not representative of the distribution in the Norwegian AE data set.

Only a tiny subset of the 18555 documents in the Norwegian AE data set is relevant to venous catheters and infections. Many of the documents are not even related to patient-specific incidents, concerning topics like administrative complaints and staff shortages. Some initial filtering is unavoidable to make the data source more fitting to the task, especially for identifying more training data. If new training samples were fetched randomly from the 18555 Norwegian AE documents, there is a high chance that there would be very few relevant samples, which would not allow the model to learn the positive cases.

Filtering the Norwegian AE data set can be very beneficial for increasing relevant training data and thus identifying new documents of interest. However, the filtering method should be considered carefully to avoid sample bias. During this project, relevant documents were identified using a keyword search. The keywords were based on relevant terms for venous catheters and infections. This filtering method can potentially exclude relevant documents that do not contain these keywords. An alternative approach could have been creating exclusion keywords to identify documents entirely irrelevant to the task, like administrative complaints. Excluding irrelevant documents instead of searching for relevant documents could potentially lead to a more accurate subset for identifying cases of venous catheters and infections where they are not explicit from the text.

The effects of the keyword search can be seen by evaluating the top features for identifying venous catheters and infections for the supervised classifier in Figure 7.9. Most of these top features are words that exactly match the keywords used to find the relevant dataset. In summary, these are the features that are either exact matches or lemmas of the words from the keyword search:

- Six of the top ten features for identifying venous catheters: *pvk*, *iv*, *infusjon*, *venflon*, *cvk*, and *venekanyle*.
- Six of the top ten features for identifying infections: *infeksjon*, *feber*, *antibiotikum*, *flebite*, *sepsis*, and *flebitt*.

This similarity is not troubling, as it means that the keywords used in the search were actually relevant for identifying documents of interest. However, documents that include words that are undeniable identifiers of venous catheters and infections could have been just as easily identified with a keyword search. The most interesting documents for classification are those relevant to venous catheters and infections in an implicit way. The model still catches some of these documents, but the usefulness of the model might be increased by including more implicitly relevant documents in the training.

As the same sampling method was used to identify a relevant subset in both the training and test data, the effects of sampling bias are not particularly visible in the results. The test data is bound to include many of the supervised classifier's important features (see Figure 7.9) as it is retrieved using the same keyword search. This similarity means it will be easier for the model to identify positive cases, as the test data is affected by the same sample bias as the training set. The results from applying the model to unseen test data are overall good, with a consistent average F_1-score of over 80% and HL of 18% and lower. However, if the model were tested on random data from the entire Norwegian AE data set, the performance would likely drop, as the model is not generalized to perform well on a data set with many irrelevant and uncertain documents. This generalization concern is not a big issue, as testing on a completely random subset is not a viable approach to this problem.

8.2.2 Size of the Labeled Training Data Set

The data size is one of the most crucial constraints of this project. As previously mentioned, predicting the presence of venous catheters and infections can be very difficult in cases where they are not explicitly present in the text. Increasing the training data will give the model more patterns to learn from and likely increase the overall performance. An example of this can be found by comparing the supervised classifiers in Tables 7.1 and 7.2. Table 7.1 presents the performance of using the augmented labeled data set with 247 training documents, while Table 7.2 shows the results of using the original synthetic data set with 100 documents for training. In every measure except precision, the model with the larger training set outperforms the model with the small training set by several decimals.

Interestingly, the precision is higher for the model with a small training set. However, the recall is extremely low, with a micro average of 47% and a macro average of 46%. This result suggests that the model only returns results that are confidently related to the categories but avoids labeling uncertain documents. As previously discussed, this is not a desirable result. Increasing the data in Table 7.1 leads to a drastic improvement in recall, with a micro and macro average of 79%. The model will likely identify even more relevant documents by increasing the data size further.

Overall, increasing the amount of manually labeled training documents improved the models' performance, especially in improving recall, which is important for future projects. On the other hand, increasing the labeled training data using semi-supervised learning does not, on average, improve recall. This issue was discussed in Section 8.1.2. More data relevant to handling uncertain cases is needed to improve the recall, which can only be provided by a qualified professional.

8.2.3 Problems with Unseen Test Data

In order to obtain an accurate assessment of a model's performance, it is essential to evaluate the model using unseen test data. When choosing AE documents for the test set, it was carefully ensured that no documents with the same identifier were included in both the training and test sets to keep the test set entirely unseen. However, because several of the synthetic training documents were created by combining different parts of original Norwegian AE documents (see Section 4.1.2), there are some instances of repetition. Specifically, at least two AE documents in the test data exhibit almost identical matches in the synthetic training data. This occurrence is likely because these particular synthetic documents were originally derived from some of the test documents. The similarity of certain training and test documents makes it easier for the model to predict the labels of the matching test documents, as it has seen almost the exact patterns before.

8.3 Missing Clinical Vocabulary

The error analysis in Section 8.1.2 suggested that the models would benefit from including additional clinical vocabulary in the learning process. This issue was also emphasized in the limitation analysis of the experiments outlined in Section 7.2. Currently, the model is only familiar with clinical terms that happen to be present in the training set. However, in unseen data, these detailed terms are likely to differ. Some examples of relevant clinical information for this task are the names of different venous catheter types and the names of antibiotics used to treat infections. Incorporating such clinical terms in the training process would likely enhance the models' performance and improve their ability to process unseen data accurately. This improvement is especially relevant for this project as the existing training data is very limited.

Apostolova and Velez [6] used a list of the 60 most common antibiotics to treat infections in their training process to identify patients with infections from nursing notes. This list was further extended by adding additional antibiotic names, abbreviations, common misspellings, and spelling variations. Introducing a similar approach in the training process of the models used in this project would have been very beneficial for easier identifying documents related to infection. Similarly, a list of the most common types of venous catheters could also be added.

8.4 Choice of Preprocessing Methods

Section 2.2.4 introduced some of the challenges of processing text from clinical narratives. Several of these challenges were encountered during the preprocessing of the AE documents in this project. The AE documents are characterized by flexible formatting, spelling errors, sentence fragments, and abbreviations. Also, they include a lot of punctuation and special characters.

The text-cleaning steps of the AE documents included the removal of digits and cleaning punctuation. Most of the digits in the AE documents are related to time, dates, or measurements for medication. The digits are mostly noise irrelevant for predicting the presence of venous catheters and infections, so tokens containing digits were removed. Cleaning the punctuation was more of a challenge, as some types of punctuation and special characters can be relevant for the analysis. An example of this is the phrase "Sepsis?", which indicates that the writer of the document is considering sepsis as a potential diagnosis, but it is not definite. However, it was decided to remove the punctuation marks from the documents, as they mostly added noise to the data. The tokens that included punctuation marks were split into multiple tokens, and the punctuation was removed. One exception was made for period marks. This choice was due to the many relevant abbreviations in the AE documents, which should not be split into separate tokens.

Initially, tokens containing special characters or punctuation were erroneously removed from the documents during text cleaning. Deleting whole tokens was a huge mistake, as some punctuation marks are heavily used in AE documents to separate essential information, like the marks "-" and "/". An example is the phrase "sepsis/urosepsis", used in a document to inform that the patient was diagnosed with sepsis or urosepsis. This phrase was entirely removed in the initial preprocessing since it contained a punctuation mark. When this error was discovered, the cleaning was changed to split each token containing one or more punctuation marks on the marks and then remove the marks. This change led to splitting the phrase "sepsis/urosepsis" into two tokens without the punctuation mark, namely "sepsis" and "urosepsis". This change significantly impacted the models' performance, as more important information was kept in the document.

The stopwords used were based on a Norwegian list of common stopwords and changed to better fit the problem context. As described in Section 6.1, the negation words "ikke", "ikkje", and "ingen" were removed as stopwords as they could be important for predicting the presence of venous catheters and infection. Also, some additional stopwords were added that were very common in the AE documents but did not provide any relevant information. Examples are words and abbreviations to describe patients (*pasient*, *pasienten*, *pas*) and standard abbreviations to describe time or measurements (*kl*, *ca*, *mg*, *g*, *ml*, *cm*, *kg*, *mm*). Due to the AE documents being de-identified, some specific phrases were also used as placeholders for sensitive information. Examples are the phrases *"Nordmann_placeholder"*, *"Plassen_placeholder"*, and *"Language_placeholder"*. These de-identification phrases were also added as stopwords as they appeared very frequently and did not provide relevant information.

For text normalization, lemmatization was chosen instead of stemming. Using lemmatization was more time-consuming, as a separate complex linguistic resource (the OBT) had to be installed and used to identify the tokens' correct lemmas. Nevertheless, since BoW was used as a feature extraction method, words needed to be reduced to their true base to ensure they were counted as the same features. Stemming does not always result in the same stem for the same word with different inflections, which is a problem when using BoW. However, the OBT produced decent lemmas, even for rare clinical abbreviations and words.

Some of the preprocessing choices were due to limited resources. For instance, checking and correcting spelling errors as part of the preprocessing pipeline would have been beneficial. However, because of the many abbreviations and Norwegian clinical language, no reliable resources for spell correction could be found. Using a spell checker relevant based on the standard Norwegian language would likely have done more damage than improvement. A viable alternative to spell-checking could have been using a feature extraction method that captures semantic and syntactic similarities. The choice of the feature extraction method is discussed in Section 8.5.

8.5 Choice of Feature Extraction Method

The final models used the simplest text feature extraction method, BoW, with no additional weighting using TF-IDF. This choice was defended in Section 6.2.1 based on the observation that using TF-IDF during the initial testing of the models did not, on average, increase the results. However, in retrospect, this was a poor choice. Even with extensive preprocessing and removing common stopwords, the model would most likely benefit from varying the weight of the features based on how common they are across the documents. Using TF-IDF as a feature extraction method should have been considered *after* designing the final model architecture and not only during initial testing.

Using word embeddings as a feature extraction method should also be considered in the future. Word embedding methods were not considered in this project due to the lack of sufficient resources and data. Since the data set has a clinical topic and is in the Norwegian language, a suitable word embedding method likely needs to be trained from scratch. This training requires a large amount of data. Still, using word embeddings could significantly improve the models' ability to make connections between similar words. This problem was highlighted in the error analysis in Section 8.1.2, where the model could not catch the relevance of the feature "sårinfeksjon" despite its semantic similarity to the feature "infeksjon".

8.6 Suggested Improvements for Future Revision of the Annotation Guideline

This section provides some general suggestions for improvement for a potential future revision of the annotation guideline based on experiences from the annotation meetings and using the data for classification. The suggestions refer back to the description and improvement of the Adverse Event Annotation Guideline from Section 5.1, and, therefore, use the terms *note* and *document* interchangeably.

Improve the Annotated Note entity attributes' selection options: The entity attributes for the Annotated Note label were described in Section 5.1.2, where the new drop-down menus were presented in Table 5.1. During the annotation meetings, it was experienced that the selection options for the drop-down menu entity attributes could have been formulated better to capture more information. The selection options "related", "probably related", and "not related" did not capture certain essential cases. One document was labeled as unrelated to sepsis during the annotation meetings, but the nurse pointed out that the patient showed early signs of sepsis that would be followed up on in a hospital setting. This example is a case that would be highly relevant for further analysis. Modifying the categories to include separate options for "early signs of

sepsis" and "early signs of phlebitis" might help to identify cases where the patient is at risk for a diagnosis.

The drop-down menu entity attributes "Does the note suggest a catheter-related infection?" and "Does the note suggest a catheter-related phlebitis?" were not used in this project but were added for future use to analyze relations between catheters and infections in a document. These entity attributes might work better as checkboxes instead of drop-down menus, as they were a source of confusion during the annotation meetings. A simple checkbox to mark if the note suggests a catheter-related infection or phlebitis is less complex and provides all the information needed. The options for "probably related" and "related but not present" are unnecessary as the question already specifies that the note "suggests" a relation.

Directions for defining relation to a subject and handling negations: During the annotation meetings, a lack of proper directions led to uncertainty during labeling. The questions in the entity attributes ask if a note is *related* to different subjects, like phlebitis and infection. There is a need to specify what "related" implies. For example, if an AE document describes an error where a used PIVC was found on the hallway floor, is this related to venous catheters? It is clearly related to the *subject* of venous catheters, but it is not a document of interest as the PIVC is not used on a patient. Another example is a document complaining about needing more staff, where they describe that a sepsis patient did not get proper care due to limited resources. Again, this document is related to sepsis, but the topic of the document is not related to a sepsis patient — it is an administrative complaint. During annotation, all documents that were even slightly related to a category were labeled as the category in question. However, this might not be the best approach for identifying relevant documents. Introducing some constraints and proper definitions for evaluating if a document is related to a subject can be useful for a future annotation guideline.

Another source of confusion during annotation meetings was handling negated relations to a category. One document specified that they did *not* suspect the patient had an infection. In this case, the document itself is related to infection, as they mention it specifically and assess that the patient is not likely to have an infection. Therefore, this document and similar cases were labeled as infection-related, even though they explicitly specify the opposite. Further directions are needed here. All documents discussing infections should be labeled as infection-related, or it should be properly explained in the guideline that only documents that suggest a positive case of infection are regarded as relevant. Which choice to go for should be discussed and decided based on future project goals. However, the guideline directions must be specified to avoid confusion and inconsistent labeling among the annotators.

Introducing an entity attribute for the note's topic: Many documents in the Norwegian AE data set are related to topics that do not concern a patient, like administrative

complaints. It could be beneficial for further analysis to collect an overview of all potential AE topics and add a new entity attribute to the Annotated Note label concerning the note's topic. Alternatively, this entity attribute could simply divide the topic into two groups: patient-related and not patient-related. Adding this entity attribute as a feature to the classification model could help identify more relevant documents.

Revision of the "Phlebitis-related" category: It has previously been discussed in Figure 5.3 that the infection, sepsis, and phlebitis categories were merged into one category for this project. However, in the annotation guideline, they were separate to gain a higher level of information which could be helpful for future work. During the annotation meetings, a document was labeled as phlebitis-related if it described multiple common symptoms of phlebitis, but it was not necessarily labeled as infection-related. However, the document was labeled as both phlebitis-related and infection-related when the word "phlebitis" (*flebitt*) was explicitly mentioned. The use of the word "phlebitis" implies that a diagnosis has been specified, and in that case, the nurse regarded it as infection-related. As previously discussed, introducing a separate category for "early signs of phlebitis" might clarify this. Currently, it is a potential source of confusion and might lead to inconsistent labeling depending on the annotator's understanding.

Chapter 9

Conclusion and Future Work

This chapter concludes the work performed in the thesis. An overview of the thesis contributions is provided in Section 9.1, by presenting a final answer to the research questions based on the findings in the discussion. Section 9.2 presents the final conclusion of the work. Lastly, suggestions for improvement and future work is presented in Section 9.3.

9.1 Contributions

Section 1.2 introduced three research questions which the thesis aimed to investigate. This section summarizes the main findings and contributions, based on the research questions.

Research question 1 *How can the available adverse event data sets be used to train a supervised multi-label classifier to detect the presence of venous catheters and infections?*

The Norwegian and the synthetic AE data set consists of 18 555 unlabeled and 100 unique labeled free-text notes, respectively. Only the 100 synthetic notes were suitable for training a supervised multi-label classifier, which is a very limited data size for classification. Also, the available labels for the synthetic AE notes were very detailed and unsuitable for the classification objective. Therefore, a choice was made to process and augment the labeled training data before performing classification experiments. This phase included revising the existing AE annotation guideline, re-annotating the synthetic notes, and annotating 247 new notes from the Norwegian AE data set through three annotation meetings with a nurse.

Creating a suitable training data set for classification using the available AE data sets involved a quantitative research strategy, where the guideline revisions and final annotations were based on interviews and discussions with clinical domain experts. The process was designed to optimize the information gained with limited time and resources.

Research question 2 *What are the key limitations to consider when performing supervised text classification on a small, labeled data set in a clinical setting, and how can these limitations be alleviated?*

Even with an augmented data set suitable for the classification task, the total data size was small, which makes it challenging to train a reliable supervised classifier. The data size was shown to be too small to extract a reliable validation set for model selection. Instead, CV techniques were used during model selection to gain a more realistic overview of how the model would perform on unseen data. Adding unlabeled data to the training process to increase the data size was tested with mixed results. While slightly improving the average F_1-score and HL of the models, it did not improve the average recall.

During the annotation process, the sampling of notes to annotate and the limitation of only one annotator introduced a problem of data bias. During sampling, the need for relevant data instances made it impossible to select random samples from the Norwegian AE data set for annotation. An alternative approach for sampling should be used in the future to limit the data bias, like excluding potentially irrelevant notes instead of only including potentially relevant notes. Also, including multiple annotators in the annotation process will reduce the possibility of an annotator's subjective understanding influencing the training data.

The AE notes contain advanced clinical terminology, and the model could only learn patterns from the terminology introduced in the training set. This limitation led to many relevant notes being classified as irrelevant, giving the model a low recall. Including clinical knowledge sources in the training process is a method for alleviating this limitation.

Research question 3 *How feasible is it to use the proposed approach for further identification of catheter-related complications?*

The proposed supervised approach identifies venous catheters and infections in an unseen test set with an HL of 0.18, an average precision of 88%, and an average F_1-score of 83%, which is promising. The recall of the model is lower, with an average of 79%. The current model should be improved to increase recall, even at the expense of precision, to identify as many relevant documents as possible. Irrelevant documents can be removed during further evaluation and processing, but it is not possible to evaluate notes that are not identified. It is also essential to consider that sampling and annotator bias affected the test set used in this project and that the model might perform worse on a more general set of AE notes.

Overall, the model has several points for improvement but could be used as a starting point for identifying relevant notes for further evaluation. Regardless of potential model improvement, the problem is challenging, as even highly skilled medical professionals struggle with correctly labeling the notes as venous catheter-related and infection-related. The classification problem might need to be simplified to be feasible for identifying relevant notes in an unbiased data set.

9.2 Conclusion

One significant challenge in achieving the long-term goal of utilizing the Norwegian AE data set to study potential catheter-related complications is the abundance of irrelevant notes. This thesis has proposed an exploratory approach for classifying these AE notes into relevant categories of **Venous Catheter** and **Infection**, enabling further evaluation of the relevant notes. However, it is important to acknowledge that high bias and limited recall affect the model's performance.

The results indicate that the proposed model serves as a viable *starting point* for identifying relevant notes within the Norwegian AE data set. Nevertheless, given the complexity of the classification problem itself, achieving a reliable classifier is challenging. Even experienced medical professionals encounter difficulties in accurately categorizing the notes. A preferable final approach would involve simplifying the classification problem and annotation guideline, enabling human annotators to easily allocate the notes into distinct note-level categories.

9.3 Future Work

As the thesis was designed as a feasibility study to explore potential and limitations, much of the potential future work has already been addressed in previous sections. This section summarizes some of the main points that should be considered to improve the proposed approach. Additionally, it explores the possibility of utilizing the identified AE notes to investigate potential catheter-related complications in more depth.

Improving the Proposed Model and Annotation Guideline

The primary focus for improving the proposed model is increasing its recall, ensuring that more relevant notes are accurately retrieved. The model currently encounters difficulties in classifying uncertain notes, which lack the most prominent features. Several strategies can be employed to address this challenge:

1. **Introducing more training data:** Incorporating additional training data relevant to venous catheters and infections that do not include the most common features can improve the model's performance in handling uncertain notes.
2. **Incorporating clinical vocabulary:** Augmenting the training process with relevant clinical terminology can enhance the model's understanding of medical concepts.
3. **Utilizing word-embedding methods:** Leveraging word-embedding techniques for feature extraction can capture semantic similarities between words and accommodate potential misspellings.

4. **Improving the annotation guideline:** Providing more detailed descriptions and instructions in the annotation guideline can mitigate confusion and reduce errors during the annotation process, leading to improved model performance.

Alternatively, it can be worth considering if the classification problem should be simplified. An alternative approach could involve categorizing the notes into two binary classes: relevant or irrelevant. The current detailed labeling during annotation should be kept for future evaluation purposes. However, a new note-level category could be introduced as a checkbox to determine the overall relevance of a note. Any note deemed relevant to either venous catheters or infections would be considered relevant, simplifying both the annotation process and the problem itself.

Investigating Potential Catheter-related Complications

Another relevant task for future work is further evaluation of the notes identified as relevant. As mentioned in Section 5.1.1, the annotation guideline includes detailed word and phrase-level labels for annotating an AE note, like **Sign**, **Location**, or **Procedure**. These high-level categories are further divided into multiple subcategories, following the hierarchy defined in the AEENOTE. In addition to the note-level **Annotated Note** labels used in this project, the 100 synthetic notes contain detailed labeling, providing valuable information for analysis. These labels could be used for training a model to find relevant entities inside an AE note, such as **Redness**, **Swelling**, or **Pus**. Subsequently, rules based on the clinical knowledge from the CIIO can be used to reason for potential catheter-related complications, like phlebitis or sepsis.

In the long term, this reasoning approach can provide valuable insights into potential catheter-related complications, facilitating early detection and treatment in critical care settings.

Bibliography

[1] Evan Alexandrou, Gillian Ray-Barruel, Peter J. Carr, Steven A. Frost, Sheila Inwood, Niall Higgins, Frances Lin, Laura Alberto, Leonard Mermel, Claire M. Rickard, and OMG Study Group. Use of Short Peripheral Intravenous Catheters: Characteristics, Management, and Outcomes Worldwide. *Journal of Hospital Medicine*, 13(5), May 2018. ISSN 1553-5606. doi:10.12788/jhm.3039.

[2] Walter Zingg and Didier Pittet. Peripheral venous catheters: an under-evaluated problem. *International Journal of Antimicrobial Agents*, 34:S38–S42, January 2009. ISSN 0924-8579. doi:10.1016/S0924-8579(09)70565-5. URL https://www.sciencedirect.com/science/article/pii/S0924857909705655.

[3] Luis E Huerta and Todd W Rice. Pathologic Difference between Sepsis and Bloodstream Infections. *The Journal of Applied Laboratory Medicine*, 3(4):654–663, January 2019. ISSN 2576-9456. doi:10.1373/jalm.2018.026245. URL https://doi.org/10.1373/jalm.2018.026245.

[4] Mervyn Singer, Clifford S. Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R. Bernard, Jean-Daniel Chiche, Craig M. Coopersmith, Richard S. Hotchkiss, Mitchell M. Levy, John C. Marshall, Greg S. Martin, Steven M. Opal, Gordon D. Rubenfeld, Tom van der Poll, Jean-Louis Vincent, and Derek C. Angus. The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). *JAMA*, 315 (8):801–810, February 2016. ISSN 0098-7484. doi:10.1001/jama.2016.0287. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4968574/.

[5] Jeffrey M. Rohde, Andrew J. Odden, Catherine Bonham, Latoya Kuhn, Preeti N. Malani, Lena M. Chen, Scott A. Flanders, and Theodore J. Iwashyna. The Epidemiology of Acute Organ System Dysfunction from Severe Sepsis Outside of the ICU. *Journal of hospital medicine : an official publication of the Society of Hospital Medicine*, 8(5):243–247, May 2013. ISSN 1553-5592. doi:10.1002/jhm.2012. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3650104/.

[6] Emilia Apostolova and Tom Velez. Toward Automated Early Sepsis Alerting: Identifying Infection Patients from Nursing Notes. In *BioNLP 2017*, pages 257–262, Vancouver, Canada„ August 2017. Association for Computational Linguistics. doi:10.18653/v1/W17-2332. URL https://aclanthology.org/W17-2332.

[7] Melissa Y. Yan, Lise Husby Høvik, André Pedersen, Lise Tuset Gustad, and Øystein Nytrø. Preliminary Processing and Analysis of an Adverse Event Dataset for Detecting Sepsis-Related Events. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1605–1610, December 2021. doi:10.1109/BIBM52615.2021.9669410.

[8] Melissa Y Yan, Lise Tuset Gustad, and Øystein Nytrø. Sepsis prediction, early detection, and identification using clinical text for machine learning: a systematic review. *Journal of the American Medical Informatics Association*, 29(3):559–575, March 2022. ISSN 1527-974X. doi:10.1093/jamia/ocab236. URL https://doi.org/10.1093/jamia/ocab236.

[9] Melissa Yan, Lise Gustad, Lise Høvik, and Øystein Nytrø. Terminology and ontology development for semantic annotation: A use case on sepsis and adverse events. *Semantic Web*, 14:1–61, March 2023. doi:10.3233/SW-223226.

[10] Rene Y. Choi, Aaron S. Coyner, Jayashree Kalpathy-Cramer, Michael F. Chiang, and J. Peter Campbell. Introduction to Machine Learning, Neural Networks, and Deep Learning. *Translational Vision Science & Technology*, 9(2):14, February 2020. ISSN 2164-2591. doi:10.1167/tvst.9.2.14. URL https://doi.org/10.1167/tvst.9.2.14.

[11] Solveig Badillo, Balazs Banfai, Fabian Birzele, Iakov I. Davydov, Lucy Hutchinson, Tony Kam-Thong, Juliane Siebourg-Polster, Bernhard Steiert, and Jitao David Zhang. An Introduction to Machine Learning. *Clinical Pharmacology & Therapeutics*, 107(4):871–885, 2020. ISSN 1532-6535. doi:10.1002/cpt.1796. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpt.1796. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpt.1796.

[12] Mohamed Farouk Abdel Hady and Friedhelm Schwenker. Semi-supervised Learning. In Monica Bianchini, Marco Maggini, and Lakhmi C. Jain, editors, *Handbook on Neural Information Processing*, Intelligent Systems Reference Library, pages 215–239. Springer, Berlin, Heidelberg, 2013. ISBN 978-3-642-36657-4. doi:10.1007/978-3-642-36657-4_7. URL https://doi.org/10.1007/978-3-642-36657-4_7.

[13] Julia Hirschberg and Christopher D. Manning. Advances in natural language processing. *Science (New York, N.Y.)*, 349(6245):261–266, July 2015. ISSN 1095-9203. doi:10.1126/science.aaa8685.

[14] K. R. Chowdhary. Natural Language Processing. In K.R. Chowdhary, editor, *Fundamentals of Artificial Intelligence*, pages 603–649. Springer India, New Delhi, 2020. ISBN 978-81-322-3972-7. doi:10.1007/978-81-322-3972-7_19. URL https://doi.org/10.1007/978-81-322-3972-7_19.

[15] Maxim Topaz, Ludmila Murga, Katherine M. Gaddis, Margaret V. McDonald, Ofrit Bar-Bachar, Yoav Goldberg, and Kathryn H. Bowles. Mining fall-related information in clinical notes: Comparison of rule-based and novel word embedding-based machine learning approaches. *Journal of Biomedical Informatics*, 90:103103, February 2019.

ISSN 1532-0464. doi:10.1016/j.jbi.2019.103103. URL https://www.sciencedirect.com/science/article/pii/S1532046419300218.

[16] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, September 2011. ISSN 1067-5027. doi:10.1136/amiajnl-2011-000464. URL https://doi.org/10.1136/amiajnl-2011-000464.

[17] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. Text Classification Algorithms: A Survey. *Information*, 10(4):150, April 2019. ISSN 2078-2489. doi:10.3390/info10040150. URL https://www.mdpi.com/2078-2489/10/4/150. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

[18] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. A Survey on Text Classification: From Traditional to Deep Learning. *ACM Transactions on Intelligent Systems and Technology*, 13(2):31:1–31:41, April 2022. ISSN 2157-6904. doi:10.1145/3495162. URL https://dl.acm.org/doi/10.1145/3495162.

[19] David Chen, Sijia Liu, Paul Kingsbury, Sunghwan Sohn, Curtis B. Storlie, Elizabeth B. Habermann, James M. Naessens, David W. Larson, and Hongfang Liu. Deep learning and alternative learning strategies for retrospective real-world clinical data. *NPJ Digital Medicine*, 2:43, May 2019. ISSN 2398-6352. doi:10.1038/s41746-019-0122-0. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6550223/.

[20] Murugan Anandarajan, Chelsey Hill, and Thomas Nolan. Text Preprocessing. In Murugan Anandarajan, Chelsey Hill, and Thomas Nolan, editors, *Practical Text Analytics: Maximizing the Value of Text Data*, Advances in Analytics and Data Science, pages 45–59. Springer International Publishing, Cham, 2019. ISBN 978-3-319-95663-3. doi:10.1007/978-3-319-95663-3_4. URL https://doi.org/10.1007/978-3-319-95663-3_4.

[21] Stefano Ferilli, Floriana Esposito, and Domenico Grieco. Automatic Learning of Linguistic Resources for Stopword Removal and Stemming from Text. *Procedia Computer Science*, 38:116–123, January 2014. ISSN 1877-0509. doi:10.1016/j.procs.2014.10.019. URL https://www.sciencedirect.com/science/article/pii/S1877050914013799.

[22] Faiza Khan Khattak, Serena Jeblee, Chloé Pou-Prom, Mohamed Abdalla, Christopher Meaney, and Frank Rudzicz. A survey of word embeddings for clinical text. *Journal of Biomedical Informatics*, 100:100057, January 2019. ISSN 1532-0464. doi:10.1016/j.yjbinx.2019.100057. URL https://www.sciencedirect.com/science/article/pii/S2590177X19300563.

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, September 2013. URL http://arxiv.org/abs/1301.3781. arXiv:1301.3781 [cs].

[24] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi:10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162.

[25] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching Word Vectors with Subword Information, June 2017. URL http://arxiv.org/abs/1607.04606. arXiv:1607.04606 [cs].

[26] Hans Christian, Mikhael Pramodana Agus, and Derwin Suhartono. Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF). *ComTech: Computer, Mathematics and Engineering Applications*, 7(4): 285–294, December 2016. ISSN 2476-907X. doi:10.21512/comtech.v7i4.3746. URL https://journal.binus.ac.id/index.php/comtech/article/view/3746. Number: 4.

[27] Theresa A Koleck, Caitlin Dreisbach, Philip E Bourne, and Suzanne Bakken. Natural language processing of symptoms documented in free-text narratives of electronic health records: a systematic review. *Journal of the American Medical Informatics Association*, 26(4):364–379, April 2019. ISSN 1527-974X. doi:10.1093/jamia/ocy173. URL https://doi.org/10.1093/jamia/ocy173.

[28] Irena Spasic and Goran Nenadic. Clinical Text Data in Machine Learning: Systematic Review. *JMIR Medical Informatics*, 8(3):e17984, March 2020. ISSN 2291-9694. doi:10.2196/17984. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7157505/.

[29] Wendy W Chapman, Prakash M Nadkarni, Lynette Hirschman, Leonard W D'Avolio, Guergana K Savova, and Ozlem Uzuner. Overcoming barriers to NLP for clinical text: the role of shared tasks and the need for additional creative solutions. *Journal of the American Medical Informatics Association*, 18(5):540–543, September 2011. ISSN 1067-5027. doi:10.1136/amiajnl-2011-000465. URL https://doi.org/10.1136/amiajnl-2011-000465.

[30] Robert Leaman, Ritu Khare, and Zhiyong Lu. Challenges in clinical natural language processing for automated disorder normalization. *Journal of Biomedical Informatics*, 57:28–37, October 2015. ISSN 1532-0464. doi:10.1016/j.jbi.2015.07.010. URL https://www.sciencedirect.com/science/article/pii/S1532046415001501.

[31] Grigorios Tsoumakas and Ioannis Katakis. Multi-Label Classification: An Overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, September 2009. doi:10.4018/jdwm.2007070101.

[32] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier Chains for Multi-label Classification. In Wray Buntine, Marko Grobelnik, Dunja Mladenić, and John Shawe-Taylor, editors, *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 254–269, Berlin, Heidelberg, 2009. Springer. ISBN 978-3-642-04174-7. doi:10.1007/978-3-642-04174-7_17.

[33] Mohammad S. Sorower. A Literature Survey on Algorithms for Multi-label Learning. 2010. URL https://www.semanticscholar.org/paper/A-Literature-Survey-on-Algorithms-for-Multi-label-Sorower/6b5691db1e3a79af5e3c136d2dd322016a687a0b?p2df.

[34] Weiwei Liu and Ivor Tsang. On the Optimality of Classifier Chain for Multi-label Classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/hash/854d9fca60b4bd07f9bb215d59ef5561-Abstract.html.

[35] Charu C. Aggarwal and ChengXiang Zhai. A Survey of Text Classification Algorithms. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 163–222. Springer US, Boston, MA, 2012. ISBN 978-1-4614-3223-4. doi:10.1007/978-1-4614-3223-4_6. URL https://doi.org/10.1007/978-1-4614-3223-4_6.

[36] Liangxiao Jiang, Zhihua Cai, Harry Zhang, and Dianhong Wang. Naive Bayes text classifiers: a locally weighted learning approach. *Journal of Experimental & Theoretical Artificial Intelligence*, 25(2):273–286, June 2013. ISSN 0952-813X. doi:10.1080/0952813X.2012.721010. URL https://doi.org/10.1080/0952813X.2012.721010. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/0952813X.2012.721010.

[37] Jason D. M. Rennie, L. Shih, J. Teevan, and David R. Karger. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. August 2003. URL https://www.semanticscholar.org/paper/Tackling-the-Poor-Assumptions-of-Naive-Bayes-Text-Rennie-Shih/b7fab2f72ebbf4e98def1daf8c29ffcfe91183bf.

[38] Min-Ling Zhang and Zhi-Hua Zhou. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, August 2014. ISSN 1558-2191. doi:10.1109/TKDE.2013.39. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[39] Daniel Berrar. Cross-Validation. In Shoba Ranganathan, Michael Gribskov, Kenta Nakai, and Christian Schönbach, editors, *Encyclopedia of Bioinformatics and Computational Biology*, pages 542–545. Academic Press, Oxford, January 2019. ISBN 978-0-12-811432-2. doi:10.1016/B978-0-12-809633-8.20349-X. URL https://www.sciencedirect.com/science/article/pii/B978012809633820349X.

[40] Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. On the Stratification of Multi-label Data. In Dimitrios Gunopulos, Thomas Hofmann, Donato

Malerba, and Michalis Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 145–158, Berlin, Heidelberg, 2011. Springer. ISBN 978-3-642-23808-6. doi:10.1007/978-3-642-23808-6_10.

[41] Claudia Ehrentraut, Markus Ekholm, Hideyuki Tanushi, Jörg Tiedemann, and Hercules Dalianis. Detecting hospital-acquired infections: A document classification approach using support vector machines and gradient tree boosting. *Health Informatics Journal*, 24(1):24–42, March 2018. ISSN 1741-2811. doi:10.1177/1460458216656471.

[42] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. brat: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France, April 2012. Association for Computational Linguistics. URL https://aclanthology.org/E12-2021.

[43] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *The Journal of Machine Learning Research*, 12(null):2825–2830, November 2011. ISSN 1532-4435.

[44] Edward Loper and Steven Bird. NLTK: The Natural Language Toolkit, May 2002. URL http://arxiv.org/abs/cs/0205028. arXiv:cs/0205028.

[45] Janne Bondi Johannessen, Kristin Hagen, André Lynum, and Anders Nøklestad. *A combined rule-based and statistical tagger: OBT+stat.* Studies in Corpus Linguistics. John Benjamins Publishing Company, March 2012. ISBN 978-90-272-0354-0 978-90-272-7499-1. doi:10.1075/scl.49.03joh. URL https://benjamins.com/catalog/scl.49.03joh. Pages: 51-66 Publication Title: Exploring Newspaper Language: Using the web to create and investigate a large corpus of modern Norwegian.

[46] Helsedirektoratet. Antibiotika i sykehus - Sepsis. URL https://www.helsedirektoratet.no/retningslinjer/antibiotika-i-sykehus/sepsis. Last updated: 20.12.2022. Date retrieved: 31.05.2023.

[47] Maurizio Cecconi, Laura Evans, Mitchell Levy, and Andrew Rhodes. Sepsis and septic shock. *The Lancet*, 392(10141):75–87, July 2018. ISSN 0140-6736. doi:10.1016/S0140-6736(18)30696-2. URL https://www.sciencedirect.com/science/article/pii/S0140673618306962.

[48] Gillian Ray-Barruel, Denise F. Polit, Jenny E. Murfield, and Claire M. Rickard. Infusion phlebitis assessment measures: a systematic review. *Journal of Evaluation in Clinical Practice*, 20(2):191–202, 2014. ISSN 1365-2753. doi:10.1111/jep.12107. URL

https://onlinelibrary.wiley.com/doi/abs/10.1111/jep.12107. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jep.12107.

[49] Ray Higginson and Andrew Parry. Phlebitis: treatment, care and prevention. *Nursing Times*, 107(36):18–21, September 2011. ISSN 0954-7762.

[50] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, ACL '95, pages 189–196, USA, June 1995. Association for Computational Linguistics. doi:10.3115/981658.981684. URL https://dl.acm.org/doi/10.3115/981658.981684.

[51] MedlinePlus. Delirium. URL https://medlineplus.gov/delirium.html. Publisher: National Library of Medicine. Last updated: 16.06.2023. Date retrieved: 07.06.2023.

[52] Maximilian Wich, Hala Al Kuwatly, and Georg Groh. Investigating Annotator Bias with a Graph-Based Approach. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 191–199, Online, November 2020. Association for Computational Linguistics. doi:10.18653/v1/2020.alw-1.22. URL https://aclanthology.org/2020.alw-1.22.

Appendices

Appendix A

Access To Source Code

The source code for the project is available at:

https://github.com/emmavrian/adverse-event-classification

Appendix B

Adverse Event Annotation Guideline Version 6

The following pages present the complete sixth version of the Adverse Event Annotation Guideline.

Adverse Event Guideline (version 6)

Last Updated: 15. March 2023 at 14:19

Updated Instruction Overview

1. This annotation session has an extra focus on annotating the whole note using the **Annotated note (Annotert notat)** entities. Please follow the Norwegian instructions below:
    - Start med å annotere entiteter og relasjoner i notatet, som i tidligere økter. Deretter, marker ordet “Hele_Notater” som entiteten “Annotert notat” og gjør en vurdering av notatet i sin helhet.
        - Kryss av sjekkboksene som gjelder for notatet
        - Velg en verdi for hvert av spørsmålene med nedtrekksmeny. Disse spørsmålene *må* ha en verdi for at notatet skal regnes som ferdig annotert.
        - Skriv en kommentar som oppsummerer begrunnelsen for valgene dine. Dette er spesielt nyttig for oss dersom du har gjort antakelser for å svare på spørsmålene.
    - Ved annotering av notatet i sin helhet bør du gjøre antakelser basert på klinisk erfaring. Du skal kun vurdere notatet ut i fra informasjonen du har tilgjengelig.
        - Eksempel: Dersom notatet beskriver en infusjon er det logisk å anta at notatet er kateterrelatert, selv om det ikke eksplisitt nevner noe om kateterbruk.
        - Eksempel 2: Dersom notatet beskriver en pasienttilstand/symptomer som indikerer sepsis er det logisk å anta at notatet er sepsisrelatert, selv om sepsis ikke nevnes eksplisitt i notatet.
    - Vi har lagt til usikkerhetskategorier for de obligatoriske nedtrekksmenyene, f.eks. “sannsynligvis infeksjon”. Bruk disse hvis du mener du har for lite informasjon til å støtte utsagnet, men prøv å velg de sikre kategoriene (relatert/ikke relatert) så langt det lar seg gjøre.
    - Annoteringsmenyen for å legge til attributter for entiteten “Annotert notat” er ikke optimal, siden det er mye informasjon på liten plass. Dessverre er dette en begrensning i BRAT programvaren som vi ikke får gjort så mye med. For å få litt bedre oversikt over attributtene kan det hjelpe å utvide vinduet.
2. Relationships should only be used within 1 sentence. Do not link labels between different sentences.
3. Each note’s first word “Hele_Notater” must be annotated with the **Annotated note (Annotert notat)** drop-down menu ‘entity attribute’. If applicable, the checklist ‘entity attributes’ should be used. Additionally, please write us a comment to summarize your reasons for selecting the drop-down menu responses.

- Checklist options (check if true):
  - **Has identifier (Har identifikator)**: This note contains sensitive information.
  - **Has medical device malfunction (Har medisinsk utstyrsfeil)**: This note is about a device malfunction.
  - **Has human error (Har menneskelig feil)** : This note is about a human error.
  - **Has a patient fall (Har et pasientfall)** : This note is about a patient falling.
  - **Is catheter-related (Er kateterrelatert)**: This note is related to catheters.
  - **Is BSI-related (Er BSI relatert)**: This note is related to bloodstream infections (BSI).
- Drop-down menu options must have a response selected:

Drop-down Menu 'Entity Attribute'	**Selection Options**
Is the note venous catheter-related? (Er notatet venekateterrelatert?)	• No venous catheter (Ikke venekateter) • Probably venous catheter (Sannsynligvis venekateter) • Venous catheter (Venekateter)
Is the note PIVC-related? (Er notatet PVK relatert?)	• Not PIVC (Ikke PVK) • Probably PIVC (Sannsynligvis PVK) • PIVC (PVK)
Is the note phlebitis-related? (Er notatet flebittrelatert?)	• No phlebitis (Ikke flebitt) • Probably phlebitis (Sannsynligvis flebitt) • Phlebitis (Flebitt)
Is the note infection-related? (Er notatet infeksjonsrelatert?)	• No infection (Ikke infeksjon) • Probably infection (Sannsynligvis infeksjon) • Infection (Infeksjon)
Is the sepsis-related? (Er notatet sepsisrelatert?)	• No sepsis (Ikke sepsisrelatert) • Probably sepsis (Sannsynligvis sepsis) • Sepsis (Sepsis)
Does the note suggest a catheter-related infection? (Er det tegn på en kateterrelatert infeksjon?)	• No catheter-related infection (Ingen kateterrelatert infeksjon) • Catheter and infection present, but not related (Kateter og infeksjon tilstede, men ingen relasjon) • Is probably catheter-related infection (Er sannsynligvis kateterrelatert infeksjon) • Is catheter-related infection (Er kateterrelatert infeksjon)
Does the note suggest a catheter-related phlebitis? (Er det tegn på kateterrelatert flebitt?)	• No catheter-related phlebitis (Ingen kateterrelatert flebitt) • Catheter and phlebitis present, but not related (Kateter og flebitt tilstede, men ingen relasjon) • Is probably catheter-related phlebitis (Er sannsynligvis kateterrelatert flebitt) • Is catheter-related phlebitis (Er kateterrelatert flebitt)

4. **Observation category (Observasjon)** 'entity attributes' should be included whenever possible to help us determine the severity, signs, symptoms, and diagnoses:

- ‘Entity attributes’:

Subcategories	Entity Attribute	Options
• Respiratory rate (Respirasjons frekvens) • Pulse (Puls) • Blood pressure (Blodtrykk) • C-reactive protein (CRP; C-reaktivt protein)	Has severity level (Har alvorlighetsgrad)	• High severity (Høy alvorlighetsgrad) • Normal severity (Normal alvorlighetsgrad) • Low severity (Lav alvorlighetsgrad)
Triage (Triage)	Triage color-code (Triage fargekode)	• Red (Rød) • Orange (Oransje) • Yellow (Gul)
All from “**Observation category (Observasjon)**”	Observation said by (Observasjon sagt av)	• Health care provider (helsepersonell) • Patient (pasient) • Both (begge)
All from “**Observation category (Observasjon)**”	Observation is diagnosis (Observasjon er diagnose)	checkmark = yes, this is a diagnosis

5. Missing subcategories and corrections are in red below.

   ► Click for update summary

Background

Of the 30 million patients who develop sepsis annually worldwide, over 5 million die. Sepsis a complex disease that can lead to tissue damage, organ failure, and death caused by a dysregulated host response to infection spread through the bloodstream. Bloodstream infections (BSIs) can enter the body through peripheral intravenous catheters (PVKs), which are the most commonly used invasive devices worldwide in hospitals. 80% of patients admitted to hospitals receive one or more PVKs.

Annotation is identifying and marking up text with labels to categorize different concepts. In addition to categorizing concepts, it is possible to link categories based on relationships. The annotated adverse event text samples from this study will be used to automatically identify risk factors and signs of PVK-related BSI to assist in preventing, identifying and treating sepsis.

Instructions

1. Log on to BRAT with your provided username and password in a Chrome browser.
   ► Click for example
2. Annotate Norwegian anonymous synthetic adverse event notes:
   - Annotate the whole note using the checklist and drop-down menu for **Annotated note (Annotert notat)**. Each drop-down menu ‘entity attribute’ **must** be selected. Also, write

a note summarizing your reasons for selecting the drop-down menu responses.
    - Annotate parts of text using the guideline for the 6 categories:
        1. **Identifier (Identifikator**: protected health information
        2. **Person (Person)**: individuals
        3. **Observation (Observasjon)**: observed abnormality
        4. **Anatomical location (Anatomisk lokasjon)**: anatomical location
        5. **Medical device (Medisinsk utstyr)**: different catheter types and parts
        6. **Procedure (Prosedyre)**: procedures, interventions, or activities
            - ▸ Click for example

How to Annotate with BRAT

Using the Brat rapid annotation tool (BRAT) it is possible to:

- Add and delete annotations for categories.
    - ▸ Click for example
- Provide additional attribute information in the ‘entity attributes’ section if applicable.
    - ▸ Click for example
- Add and delete relationships (Add relationships by clicking 1 label and dragging the arrow to another label; Delete by double clicking the relationship).
    - ▸ Click for example
- Include notes in annotations.
    - ▸ Click for example

General Annotation Rules

1. Label all notes, even ones that seem irrelevant to PVK or sepsis.
    - We need to give the machine learning model good examples and bad examples so it can learn “what is pvk and sepsis relevant” vs “what is not relevant”.
2. Only annotate what is explicitly in the text, do not annotate inferences or speculations.
3. Label each word or phrase with only 1 label.
    - ▸ Click for example
4. Include nearby words that describe the labels and add ‘entity attributes’ to give additional label information whenever possible.
    - ▸ Click for example
5. Annotate a category or subcategory each time it is mentioned.
6. If a subcategory cannot be specified, chose the most general category level that describes the term or phrase.
7. Use the ‘entity attribute’ “Negation (Negasjon)” to indicate the opposite of something, failure, or if “it is not” a sign, location, device, or procedure.
    - ▸ Click for example
8. Link relationships between categories when possible to assist the machine in learning what the label is referring to. Only link relationships between categories if both labels are in the same sentence. Possible relationships:
    - Who has the sign, location, device, or procedure? (**Person har**)
    - What device does a procedure use? (**Prosedyre bruker**)
    - What device or procedure is observered with a sign? (**Er observert med**)
    - Where is the sign, device, or procedure located? (**Lokalisert ved/på/i**)

9. Label type of catheter care separately from the catheter type and link them using the relationship “Procedure uses (Prosedyre bruker)”. This makes it possible to identify type of care given to a specific kind of catheter.
   ▸ Click for labels and example

Annotation Guidelines for Peripheral Intravenous Catheters related to Bloodstream Infections

Annotation is identifying and marking up text with labels to categorize different concepts. In addition to categorizing concepts, it is possible to link categories based on relationships. The following sections describe how to annotate the whole note and parts of notes using the six categories and draw relationship links between categories.

Annotation Rules

Annotated note Category (Annotert notat)

The whole note can be annotated by selecting the first word, “Hele_Notater”, and by checking off items listed in the ‘entity attributes’ which are true and by selecting responses from the drop-down menu ‘entity attributes’.

Check off the following checklist ‘entity attributes’ that are true for the note:

- **Has identifier (Har identifikator)**: This note contains sensitive information.
- **Has medical device malfunction (Har medisinsk utstyrsfeil)**: This note is about a device malfunction.
- **Has human error (Har menneskelig feil)** : This note is about a human error.
- **Has a patient fall (Har et pasientfall)** : This note is about a patient falling.
- **Is catheter-related (Er kateterrelatert)**: This note is related to catheters.
- **Is BSI-related (Er BSI relatert)**: This note is related to bloodstream infections (BSI).

The whole note **must** be annotated with all drop-down menu ‘entity attributes’, and with a note written by you that summarizes your reasons for selecting the drop-down menu responses. These are the drop-down menu ‘entity attributes:’

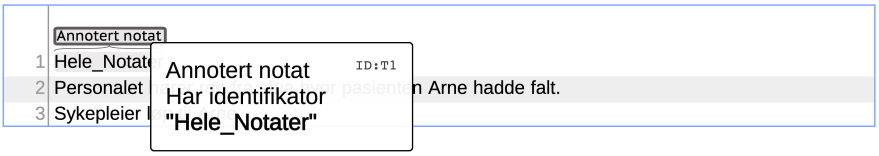
Drop-down Menu ‘Entity Attribute’	Selection Options
Is the note venous catheter-related? (Er notatet venekateterrelatert?)	• No venous catheter (Ikke venekateter) • Probably venous catheter (Sannsynligvis venekateter) • Venous catheter (Venekateter)
Is the note PIVC-related? (Er notatet PVK relatert?)	• Not PIVC (Ikke PVK) • Probably PIVC (Sannsynligvis PVK) • PIVC (PVK)

Drop-down Menu 'Entity Attribute'	Selection Options
Is the note phlebitis-related? (Er notatet flebittrelatert?)	• No phlebitis (Ikke flebitt) • Probably phlebitis (Sannsynligvis flebitt) • Phlebitis (Flebitt)
Is the note infection-related? (Er notatet infeksjonsrelatert?)	• No infection (Ikke infeksjon) • Probably infection (Sannsynligvis infeksjon) • Infection (Infeksjon)
Is the sepsis-related? (Er notatet sepsisrelatert?)	• No sepsis (Ikke sepsisrelatert) • Probably sepsis (Sannsynligvis sepsis) • Sepsis (Sepsis)
Does the note suggest a catheter-related infection? (Er det tegn på en kateterrelatert infeksjon?)	• No catheter-related infection (Ingen kateterrelatert infeksjon) • Catheter and infection present, but not related (Kateter og infeksjon tilstede, men ingen relasjon) • Is probably catheter-related infection (Er sannsynligvis kateterrelatert infeksjon) • Is catheter-related infection (Er kateterrelatert infeksjon)
Does the note suggest a catheter-related phlebitis? (Er det tegn på kateterrelatert flebitt?)	• No catheter-related phlebitis (Ingen kateterrelatert flebitt) • Catheter and phlebitis present, but not related (Kateter og flebitt tilstede, men ingen relasjon) • Is probably catheter-related phlebitis (Er sannsynligvis kateterrelatert flebitt) • Is catheter-related phlebitis (Er kateterrelatert flebitt)

Given the following example:

```
1 Hele_Notater
2 Personalet hører rop fra stua hvor pasienten Arne hadde falt.
3 Sykepleier løp til Arne.
```

The result should be:



Annotated Note (Annotert Notat) Annotation Rules

1. This should only be used to label the word "Hele_Notater" at the top of each note.
    - All 'entity attributes' with a drop-down menu must be selected, and there should be a note commenting the reason for choosing the drop-down menu options.
    - Notes that are true for any of the questions listed under the Annotated Note Category (Annotert Notat) should have the corresponding attribute selected.

Identifier Category (Identifikator)

Protected health information is sensitive data that can identify a patient. These are the following subcategories:

- Time (Tid)
- Numeral identifier (Numerisk identifikator)
  - Sample or test number (Prøve eller testnummer)
  - Norwegian national identity number (Fødselsnummer)
  - Patient identification number in Doculive (PID)
  - Account number (Kontonummer)
  - Telephone number (Telefonnummer)
  - Device number (Utstyrsnummer)
- Geographic location identifier (Geografiske lokasjon identifikator)
- Email (e-post)
- Vehicle identifier (Kjøretøy identifikator)
- Web URL (Nettadresse)
- IP address (IP adresse)
- Biometric identifier (Biometrisk identifikator)

* note: Name is part of the Person Category (Person).

Given the following example:

```
1 Pasienten Kasper Voll ringer på kl ca 05:40, forteller at han har vært uheldig og falt i gulvet.
2 Var her til konsultasjon med Dr. Berg i går søndag 22/11-18 fra Ila, men notat fra besøket mangler.
3 Pas ringer st.
4 Olav 23/11-18 fra 13451123, har sp mål ang sin sykdomstilstand.
5 Han sa send resultatene til KV1965@gmail.com, men vi kan ikke.
6 laboratoriet vet ikke om prøvenummeret 11877619 eller 11877740
```

The result should be:

```
1 Person Tid
  Pasienten Kasper Voll ringer på kl ca 05:40, forteller at han har vært uheldig og falt i gulvet.
2 Helsepersonell Tid Geografiske lokasjon identifikator
  Var her til konsultasjon med Dr. Berg i går 22/11-18 fra Ila, men notat fra besøket mangler.
3 Pasient Tid Telefonnummer
  Pas ringer 23/11-18 fra 13451123, har sp mål ang sin sykdomstilstand.
4 Pasient E-post
  Han sa send resultatene til KV1965@gmail.com, men vi kan ikke.
5 Proeve eller test nummer Proeve eller test nummer
  laboratoriet vet ikke om prøvenummeret 11877619 eller 11877740
```

Identifier Annotation Rules

1. All sensitive data should be annotated.
2. Dates and/or times should be categorized under Time (Tid).
3. Hospital wards/departments and hospital names (ex. St. Olav) should be categorized under the “Geographic location identifier (Geografiske lokasjon identifikator)”.

Person Category (Person)

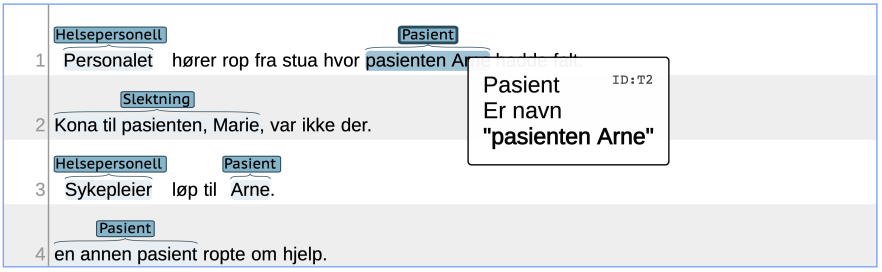
Individuals are people mentioned in the notes. The individuals mentioned in the note should be annotated with a label based on their role in the 3 subcategories:

- Patient (Pasient)
- Health care provider (Helsepersonell)
- Relative of the patient (Slektning)

Given the following example:

```
1 Personalet hører rop fra stua hvor pasienten Arne hadde falt.
2 Kona til pasienten, Marie, var ikke der.
3 Sykepleier løp til Arne.
4 en annen pasient ropte om hjelp.
```

The result should be:



Person Annotation Rules

1. Individual roles should be assigned to one of the subcategories.
2. Include names with their role and check “Is name (Er navn)” in the ‘entity attributes’ section.

Observation (Observasjon)

Certain signs can indicate a patient has an observed abnormal condition (such as an infection) or a device has a malfunction. The subcategories are as follows:

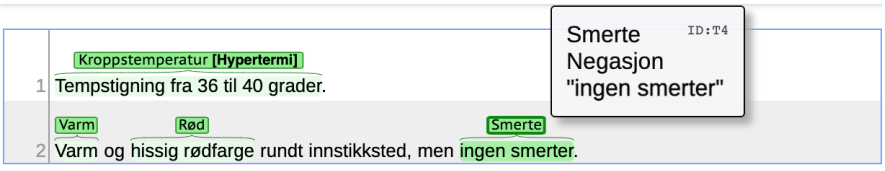
- Descriptive sign or symptom (Beskrivende tegn eller symptom)
  - Bleeding (Blødning)
  - Bloody (Blodig)
  - Phlebitis (Flebitis)
  - Hardness (Hardhet)
  - Headache (Hodepine)
  - Swollen (Hoven)
  - Infection (Infeksjon)
  - Itchy (Kløende)
  - Nausea (Kvalme)
  - Necrosis (Nekrose)
  - Purulence (Puss)
  - Red (Rød)
  - Pain (Smerte)
  - Sepsis (Sepsis)

    - Dizziness (Svimmelhet)
    - Chills (Frostrier)
    - Warm (Varm)
    - Tenderness (Ømhet)
    - Edema (Ødem)
- Vital sign (Vitale livstegn)
    - Blood pressure (Blodtrykk)
    - Body temperature (Kroppstemperatur)
    - Consciousness level (Bevissthetsnivå)
    - Pulse (Puls)
    - Respiratory rate (Respirasjons frekvens)
- Triage (Triage)
- Neurological and physiological (Nevrologisk og fysiologisk)'
    - Mobility impairment (Bevegelseshemming)
    - Psychosis (Psykose)
- Fracture (Brudd)
- Wound (Sår)
    - Laceration (Laserasjon)
    - Abrasion (Skrubbsår)
    - Incision (Snittsår)
    - Puncture (Stikksår)
- Lab result (Laboratorieresultat)
    - Blood culture (Blodkultur)
    - C-reactive protein (CRP; C-reaktivt protein)
    - Leukocyte (Leukocytt)
- Device observation (Utstyrsobservasjon)
    - Moisture (Fukt)
    - Leakage (Lekkasje)
    - Liquid (Væske)'

Given the following example:

```
1 Tempstigning fra 36 til 40 grader.
2 Varm og hissig rødfarge rundt innstikksted, men ingen smerter.
```

The result should be:



Observation (Observasjon) Annotation Rules

1. Annotate each observation separately and include adjectives. Use negation to indicate a sign is not present.
2. Identify the device or procedure observed with an observation using the "Is observed with (Er observert med)" relationship to link the observation to the device or procedure.
3. Include these 'entity attributes':

Subcategories	Entity Attribute	Options
Body temperature (Kroppstemperatur)	Has body temperature range (Har kroppstemperaturområde)	• Hyperthermia (Hypertermi) • Normal (Normal) • Hypothermia (Hypotermi)
• Respiratory rate (Respirasjons frekvens) • Pulse (Puls) • Blood pressure (Blodtrykk) • C-reactive protein (CRP; C-reaktivt protein)	Has severity level (Har alvorlighetsgrad)	• High severity (Høy alvorlighetsgrad) • Normal severity (Normal alvorlighetsgrad) • Low severity (Lav alvorlighetsgrad)
Conciousness level (Bevissthetsnivå)	Has concious state of (Har bevisste tilstand av)	• Alert (Våken) • Confusion (Forvirring) • Verbally responsive (Verbal respons) • Painfully responsive (Smerte respons) • Unresponsive (Bevisstløshet)
Triage (Triage)	Triage color-code (Triage fargekode)	• Red (Rød) • Orange (Oransje) • Yellow (Gul)
All from “Observation category (Observasjon)”	Observation said by (Observasjon sagt av)	• Health care provider (helsepersonell) • Patient (pasient) • Both (begge)
All from “Observation category (Observasjon)”	Observation is diagnosis (Observasjon er diagnose)	checkmark = yes, this is a diagnosis

Anatomical location Category (Anatomisk plassering)

Anatomical locations of devices and observations can identify the source of the issue. Also, anatomical insertion sites of catheters can be indicators of the type of catheter used.

Here is the list of subcategories:

- Body part (Kroppsdel)
  - Limb (Lem)
    - Arm (Arm)
      - Hand (Hånd)
      - Wrist (Håndledd)
      - Forearm (Underarm)
      - Elbow (Albue)
      - Upper arm (Overarm)
    - Leg (Bein)
      - Ankle (Ankel)
      - Foot (Fot)
      - Thigh (Lår)
      - Hip (Hofte)

    - Head (Hode)
    - Neck (Hals)
    - Trunk (Torso)
        - Navel (Navle)
        - Back (Rygg)
- Organ part (Organdel)
    - Skin (Hud)
        - Subcutaneous (Subkutan)
    - Circulartory system (Sirkulasjonssystem)
        - Vein (Ven)
        - Artery (Arterie)
- Injection site (Innstikkssted)

Given the following example:

```
1 Han hadde en PVK på høyre håndbak.
2 Hovnet opp, ble rød og fikk smerter rundt innstikkstedet.
```

The result should be:

```
                  Hånd [Høyre][Bak]
1 Han hadde en PVK på høyre håndbak.

                            Innstikkssted
2 Hovnet opp, ble rød og fikk smerter rundt innstikkstedet.
```

Anatomical location Annotation Rules

1. Whenever it is explicitly stated, always provide additional information about the insertion site by filling in options in the 'entity attributes' section, such as which side of the body and distance:
    - Left (Venstre)
    - Right (Høyre)
    - Front (Foran)
    - Back (Bak)
    - Proximal (Proksimal)
    - Distal (Distal)
    - Medial (Medial)
    - Sideways (Sideveis)
    - Halfway (Halvveis)

Medical device Category (Medisinsk utstyr)

Different catheters and parts of catheters must be identified so that machine learning classifiers can distinguish peripheral intravenous catheters (PVKs) from other catheters. Names and parts of catheter are provided here:

- Catheter (Kateter)
    - Venous catheter (Venekateter)
        - Peripheral intravenous catheter (PIVC) (Perifert venekateter (PVK))
        - Central venous catheter (Sentralt venekateter)
        - Hemodialysis catheter (Hemodialysekateter)

        - Midline catheter (Midline kateter)
        - Peripherally inserted central catheter (PICC) (Perifert innsatt sentralt kateter (PICC))
        - Hickman catheter (Hickman kateter)
        - Vascath catheter (VAS) (VAS-kateter (VAS))
    - Arterial catheter (Arteriekateter)
    - Urinary catheter (Blærekateter)
    - Epidural catheter (Epidural kateter)
    - Intraosseous cannula (Intraosseøs kanyle)
    - Subcutaneous catheter (Subkutant kateter)
- ~~Pump (Pumpe)~~ Medical pump (Medisinsk pumpe)
- Tubes (Rør)
- Valve (Ventil)
    - 3-way valve (Treveiskran)

Given the following example:

```
1 Har morgen tilsyn kl 04.30 og finner pasienten liggende på gulvet ved sengen med
  seponert blærekateter og pvk'er.
2 Aciklovir gitt på pumpe.
```

The result should be:

```
1 Har morgen tilsyn kl 04.30 og finner pasienten liggende på gulvet ved sengen med
                    Blærekateter      PVK
  seponert blærekateter og pvk'er.

                        Pumpe
2 Aciklovir gitt på pumpe.
```

Medical device Annotation Rules

1. Annotate catheters and other devices from the list.
2. If the observation was observed with a device not listed, then annotate it as the highest category “Medical device (Medisinsk utstyr)”. Be sure to identify the device and draw a “Is observed with (Er observert med)” relationship to link the sign to device.
3. Catheter bandages should have the ‘entity attribute’ “Is bandage (Er bandasje)” selected.
4. For peripheral intravenous catheter (Perifert venekateter), it is possible to select the following colors for ‘entity attribute’:
    - Purple (Lilla)
    - Yellow (Gul)
    - Blue (Blå)
    - Pink (Rosa)
    - Green (Grønn)
    - White (Hvit)
    - Gray (Grå)
    - Brown (Brun)
    - Orange (Oransje)
5. If context indicates the medical device or part has failed, it should have the ‘entity attribute’ “Is medical device failure (Er medisinsk utstyrsfeil)”.

Procedure Category (Prosedyre)

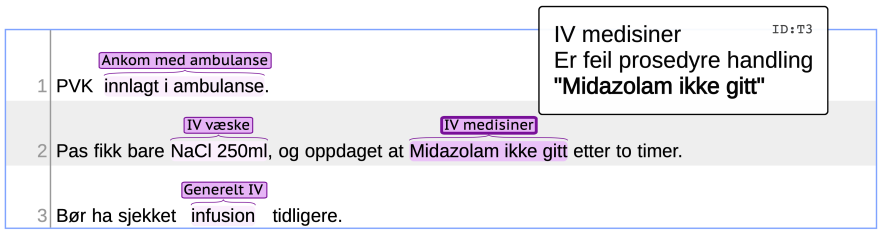
Specific procedures and interventions will require the use of certain catheters even if they are not mentioned. Additionally, activities related to catheters can indicate the type of catheter used. Subcategories for the procedure category are:

- Administration purpose (Administrasjonsformål)
  - General IV (Generelt IV)
    - IV antibiotics (IV antibiotika)
    - IV chemo (IV cellegift)
    - IV medication (IV medisiner)
    - IV nutrient (IV næringsstoff)
    - IV painkiller (IV smertestillende)
    - IV fluid (IV væske)
  - Blood test (Blodprøve)
  - Blood transfusion (Blodverføring)
  - Hemodialysis (Hemodialyse)
  - Oral medication (Oral medisinering)
- Administration way (Administrasjonsvei)
  - Epidural use (Epidural bruk)
  - IV use (IV bruk)
  - Oral use (Oral bruk)
- Arrived by Ambulance (Ankom med ambulanse)
- Catheter procedure (Kateter prosedyre)
  - Catheter replacement (Kateter bytting)
  - Catheter removal (Kateter fjerning)
  - Catheter insertion (Kateter innstikking)
  - Catheter self-removal (Kateter selvseponering)
  - Catheter discontinuation (Kateter seponering)
  - Catheter rinsing (Kateter skylling)
- Surgery (Kirurgi)
- Injection (Injeksjon)
  - Intradermal injection (Intradermalt injeksjon)
  - Intramuscular injection (Intramuskulært injeksjon)
  - Intravenous injection (Intravenøst injeksjon)
  - Self-inflicted injection (Selvpåført injeksjon)
  - Subcutaneous injection (Subkutant injeksjon)
- Infusion (Infusjon)
  - Intraosseous infusion (Intraossøs infusjon)
  - Intravenous infusion (Intravenøst infusjon)
  - Subcutaneous infusion (Subkutant infusjon)
- Radiology (Radiologi)
- Fall (Fall)
- Found on the ground (Funnet på bakken)

Given the following example:

```
1 PVK innlagt i ambulanse.
2 Pas fikk bare NaCl 250ml, og oppdaget at Midazolam ikke gitt etter to timer.
3 Bør ha sjekket infusion tidligere.
```

The result should be:



Procedure Annotation Rules

1. Annotate the procedures, interventions, and activities related to catheters.
2. If the abnormal sign was observed with an action not listed, then annotate it as the highest category “Procedure (Prosedyre)”. Be sure to identify the procedure and draw a “Is observed with (Er observert med)” relationship to link the sign to procedure.
3. Label all IV or oral medication names present.
4. Incorrect procedures should have the ‘entity attribute’ “Is incorrect procedure action (Er feil prosedyre handling)”.

All Relationships

All Relationship Rules

1. Relationships should only be used within 1 sentence. Do not link labels between different sentences.

Person has (Person har)

A person has an observed abnormal sign, anatomical location, medical device, and/or procedure, intervention, or activity.

- Observation (Observasjon)
- Anatomical location (Anatomisk plassering)
- Medical device (Medisinsk utstyr)
- Procedure (Prosedyre)

▸ Click for example

Procedure uses (Prosedyre bruker)

The procedure uses a medical device.

- Medical device (Medisinsk utstyr)

▸ Click for example

Is observed with (Er observert med)

A sign is observed with a medical device and/or procedure, intervention, or activity.

- Medical device (Medisinsk utstyr)
- Procedure (Prosedyre)

▸ Click for example

Located nearby/on/at/in (Lokalisert ved/på/i)

The abnormal sign, device, and/or procedure is located nearby/on/in this anatomical location.

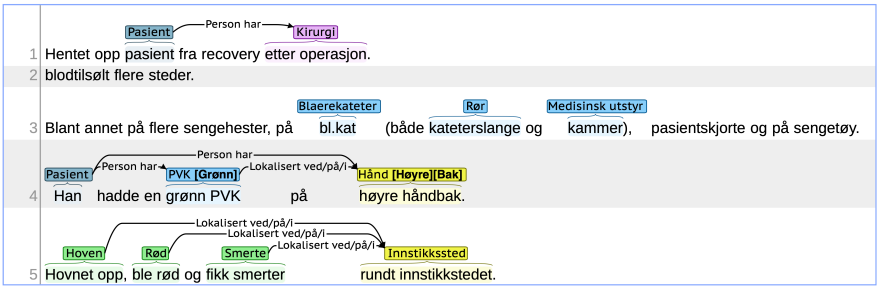
- Observation (Observasjon)
- Medical device (Medisinsk utstyr)
- Procedure (Prosedyre)

▸ Click for example

Given the following example:

```
1 Hentet opp pasient fra recovery etter operasjon.
2 blodtilsølt flere steder.
3 Blant annet på flere sengehester, på bl.kat (både kateterslange og kammer),
  pasientskjorte og på sengetøy.
4 Han hadde en grønn PVK på høyre håndbak.
5 Hovnet opp, ble rød og fikk smerter rundt innstikkstedet.
```

The result should be:



Appendix C

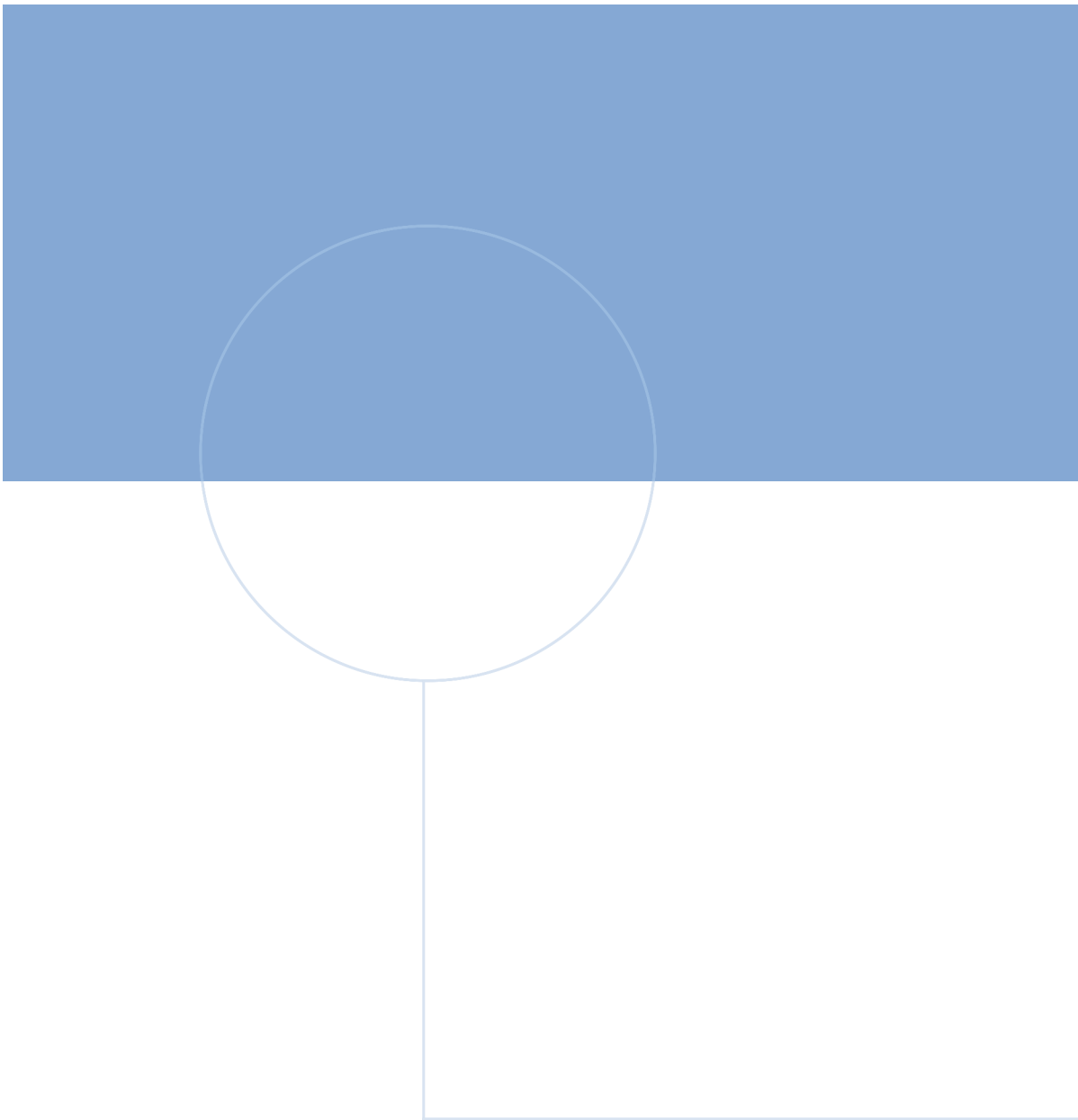
List of Stopwords used in Preprocessing Pipeline

NLTK's list of Norwegian stopwords (except the words 'ikke', 'ikkje', and 'ingen', which were removed from the list in this project):

'var', 'si', 'kvarhelst', 'enn', 'som', 'eit', 'ingi', 'vere', 'eller', 'so', 'dette', 'vi', 'da', 'hver', 'henne', 'no', 'hoe', 'dykk', 'den', 'mykje', 'disse', 'ein', 'ble', 'uten', 'her', 'være', 'korso', 'upp', 'så', 'seg', 'ditt', 'i', 'sidan', 'over', 'hadde', 'også', 'mitt', 'ved', 'nokon', 'en', 'fordi', 'hoss', 'selv', 'samme', 'kvar', 'kan', 'med', 'skal', 'dei', 'inni', 'min', 'ut', 'kven', 'å', 'at', 'vært', 'deira', 'deg', 'sjøl', 'kvifor', 'du', 'hvorfor', 'et', 'elles', 'dykkar', 'deires', 'siden', 'bare', 'somme', 'det', 'sin', 'ho', 'ville', 'blir', 'nå', 'har', 'ned', 'noen', 'kun', 'hvilken', 'mi', 'hennes', 'både', 'di', 'eitt', 'korleis', 'av', 'mange', 'slik', 'hva', 'og', 'er', 'nokor', 'denne', 'vors', 'man', 'ja', 'deres', 'når', 'nokre', 'verte', 'men', 'kunne', 'um', 'der', 'eg', 'sine', 'meget', 'sånn', 'kom', 'somt', 'noko', 'etter', 'sitt', 'då', 'vart', 'skulle', 'vort', 'både', 'kva', 'begge', 'inn', 'varte', 'me', 'han', 'de', 'kvi', 'din', 'hossen', 'mellom', 'jeg', 'vil', 'for', 'mot', 'vår', 'noka', 'hvem', 'til', 'mine', 'medan', 'vore', 'om', 'hun', 'bli', 'meg', 'hvis', 'ett', 'honom', 'sia', 'hans', 'noe', 'fra', 'ha', 'deim', 'inkje', 'på', 'dere', 'hvor', 'hennar', 'alle', 'dem', 'oss', 'blei', 'hvordan', 'hvilke', 'før', 'hjå', 'blitt', 'opp'

Additional stopwords added:

'pasient', 'pasienten', 'pas', 'placeholder', 'nordmann', 'nordmanns', 'plassen', 'month', 'language', 'yrke', 'egennavn', 'ola', 'selskapet', 'kl', 'kl.', 'ca', 'ca.', 'mg', 'g', 'ml', 'cm', 'kg', 'mm', 'm', 'x'



NTNU

Norwegian University of
Science and Technology