# Automatic Diagnosis Code Assignment with KB-BERT

## ICD Classification Using Swedish Discharge Summaries

Sonja Remmer

Department of Computer and Systems Sciences

Degree project 30 credits
Computer and Systems Sciences
Degree project at the master level
Spring term 2021
Supervisor: Hercules Dalianis
Swedish title: Automatisk klassificering av diagnoskoder med KB-BERT


Stockholm University

# Abstract

Systematically keeping track of which diseases patients suffer from is essential, both from a caregiving and a research perspective. To record patients' diseases, health professionals worldwide use a system called the International Classification of Diseases (ICD). Still, manually assigning ICD codes to patients' medical records is both time-consuming and error-prone. In light of this, efforts have been made to automate diagnosis code assignment. Automating clinical coding requires computers to understand human language, and a recent breakthrough within this area called Natural Language Processing (NLP) is the deep learning language model Bidirectional Encoder Representations from Transformers (BERT). BERT has successfully been used in previous studies to assign ICD codes automatically. However, there are no studies until this point using BERT to conduct ICD classification on Swedish patient records. This master's thesis investigated the Swedish version of BERT, KB-BERT, posing the research question *How well does KB-BERT, compared to traditional supervised machine learning models, perform in pairing Swedish gastrointestinal discharge summaries with the correct ICD codes?* The ICD codes were considered at a block level, meaning similar codes were grouped into ten blocks. Performance was represented by the micro-averaged $F_1$-*score* ($F_{micro}$), and KB-BERT's performance was compared to the baseline models Support Vector Machines, Decision Trees, and K-nearest Neighbors. An experiment using 10-fold cross-validation was set up to determine whether there was a difference in classifier performance. Wilcoxon signed-rank tests showed that the KB-BERT was statistically significantly superior to the baseline models, obtaining a final $F_{micro}$ of 0.80. The baseline model with the highest performance score, Support Vector Machines, achieved a final $F_{micro}$ of 0.71. However, the performance differences between the Support Vector Machines and the other two baseline models were not statistically significant. This thesis contributed to the research area of Swedish ICD classification. In turn, this knowledge can prove helpful in the development of a Swedish ICD coding tool.

*Keywords:* Natural Language Processing, Clinical Text Mining, Text Classification, KB-BERT, ICD, Swedish Medical Corpora, Discharge Summaries

# Acknowledgments

First and foremost, I would like to thank Hercules Dalianis for his constant feedback and support, Anastasios Lamproudis for his tireless work with the KB-BERT implementation, and the DSV students and researchers who provided me with valuable comments. I would also like to thank the Ixa research group at the University of the Basque Country, especially Alberto Blanco, for sharing their knowledge within the area of ICD classification. Last but not least, I would like to thank the ClinCode project group for making me feel like a part of the team right from the start.

# Contents

**List of Figures** **i**

**List of Tables** **ii**

**List of Abbreviations** **iii**

**1 Introduction** **1**
1.1 Problem . . . 3
1.2 Research Question . . . 3
1.3 Delimitations . . . 4
1.4 Thesis Structure . . . 4

**2 Extended Background** **5**
2.1 Theoretical Framework . . . 5
2.1.1 ICD Codes . . . 5
2.1.2 Multi-label Text Classification . . . 6
2.1.3 Baseline Models . . . 7
2.1.4 KB-BERT . . . 10
2.2 Related Research . . . 11
2.2.1 Rule-based and Machine Learning Models . . . 11
2.2.2 Deep Learning Approaches . . . 12
2.2.3 Swedish Context . . . 14

**3 Methodology** **15**
3.1 Research Strategy . . . 15
3.2 Data Collection . . . 16
3.2.1 Electronic Patient Records . . . 17
3.2.2 Evaluation Metrics . . . 22
3.3 Data Analysis . . . 23
3.3.1 Experiment Design . . . 23
3.3.2 Model Implementations . . . 26
3.3.3 Statistical Testing . . . 30
3.4 Ethics . . . 32

**4 Results** **34**
4.1 Classifier Comparison . . . . . 34
4.2 Final Evaluation . . . . . 36

**5 Discussion and Conclusion** **38**
5.1 Summary . . . . . 38
5.2 Analysis of Findings . . . . . 39
5.3 Limitations and Research Quality . . . . . 40
5.4 Future Research . . . . . 41
5.5 Research Impact and Final Remarks . . . . . 42

**Bibliography** **42**

# List of Figures


1.1 The structure of ICD codes . . . . . 1

2.1 A toy clinical note with multi-label output . . . . . 7
2.2 An illustration of SVM. Adapted from Shehzadex (2016) . . . . . 8
2.3 An illustration of Decision Trees (Bach 2013) . . . . . 8
2.4 An illustration of KNN. Adapted from Maryamvaez (2019) . . . 9

3.1 Number of notes per number of ICD blocks . . . . . 20
3.2 Number of notes per ICD blocks . . . . . 20
3.3 Number of notes per number of tokens . . . . . 21
3.4 An illustration of the experiment setup and 10-fold cross-validation 25
3.5 KB-BERT performance metrics during the 10-fold cross-validation 30

4.1 $F_{micro}$ scores during the 10-fold cross-validation . . . . . 34

# List of Tables


2.1 ICD Chapter XI code blocks . . . . . 6

3.1 Number of notes during pre-processing steps . . . . . 19
3.2 Description of ICD-10 Corpus . . . . . 19
3.3 Descriptive statistics of ICD-10 Corpus . . . . . 19
3.4 SVM hyper-parameters . . . . . 28
3.5 Decision Trees hyper-parameters . . . . . 28
3.6 KNN hyper-parameters . . . . . 28
3.7 KB-BERT hyper-parameters . . . . . 29

4.1 Performance metrics during the 10-fold cross-validation . . . . . 35
4.2 Statistical test results . . . . . 35
4.3 KB-BERT final evaluation . . . . . 36
4.4 SVM final evaluation . . . . . 37

# List of Abbreviations

BCE – Binary Cross-Entropy
BERT – Bidirectional Encoder Representations from Transformers
BoW – Bag-of-Words
CNN – Convolutional Neural Networks
DT – Decision Trees
EPR – Electronic Patient Records
$F$ – $F_1$-*score*
$F_{macro}$ – Macro-averaged $F_1$-*score*
$F_{micro}$ – Micro-averaged $F_1$-*score*
$FN$ – False Negative
$FP$ – False Positive
GPU – Graphics Processing Unit
ICD – International Classification of Diseases
ICD-10 Corpus – Stockholm EPR Gastro ICD-10 Corpus (second version)
KB – Kungliga Biblioteket (National Library of Sweden)
KNN – K-nearest Neighbors
Max – Maximum
Min – Minimum
MLP – Multi Layer Perceptrons
NaN – Missing value
NER – Named Entity Recognition
NLP – Natural Language Processing
NLTK – Natural Language Toolkit
$P$ – Precision
POS – Part-of-speech
$R$ – Recall
Std – Standard deviation
SVM – Support Vector Machines
tf-idf – Term Frequency-Inverse Document Frequency
$TN$ – True Negative
$TP$ – True Positive
WHO – World Health Organization

# Chapter 1

# Introduction

Modern healthcare is not limited to treating patients; as in most sectors, it is of utmost importance to keep a digital memory of the practice's activities. Within healthcare, this digital memory comes in the form of electronic patient records. The records consist of several data types, ranging from basic information about the patients to in-depth descriptions of their medical conditions. One crucial part of the patient records is tracking the patients' diagnoses, and this process was standardized in the late 19th century to facilitate clinical and research activities. The World Health Organization (WHO) took control of the standardization of diagnoses in the mid 20th century, naming the system ICD (International Classification of Diseases). Today, the ICD system is used by health personnel to represent patients' medical conditions in over 150 countries (WHO 2020). In Figure 1.1, the hierarchical structure of the ICD codes is illustrated. Each code starts with a letter and a two-digit number, together representing which disease category the code belongs to. A decimal point and another digit further specify the disease.


Figure 1.1: The structure of ICD codes

The exemplary ICD code A01.2 in Figure 1.1 represents the disease *Paratyphoid fever B*, which is part of the larger group of diseases A01: *Typhoid and paratyphoid fevers*, which in turn is part of the ICD block A00-A09: *Intestinal infectious diseases*. In turn, this block belongs to ICD Chapter I (A00-B99): *Certain infectious and parasitic diseases*.

While diagnosis coding is vital for clinical and statistical reasons, it can be time-consuming and error-prone (Farkas & Szarvas 2008). For example, a study by Socialstyrelsen (2006) found substantial errors in 20 percent of the codes representing the patient's main diagnosis. It has also been proven that diagnosis codes can be missing. A report by the Swedish National Board of Health and Welfare (Socialstyrelsen) in 2013 showed that 0.9 and 10 percent, respectively, of the medical records from inpatient and outpatient care, are missing a main diagnosis (Jacobsson & Serdén 2013).

One way to improve the efficiency and effectiveness of ICD coding is to automate the process by letting a tool suggest relevant ICD codes based on the health records' information. Such a tool can be developed by learning from existing patient records that already have assigned ICD codes and then using what has been learned to assign codes to unseen patient records without codes. The free-text notes in patient records, that is, *clinical notes*, are key in identifying which ICD codes that should be assigned to the patient records. Being free text, the models used in an automatic clinical coding need to understand human language to use the information in the clinical notes. This means that the development of an ICD classification tool falls within the area of Natural Language Processing (NLP). NLP is the broad field that covers all activities at the intersection of human language and computers. The sub-field of NLP that this thesis is concerned with is text classification, which is the task of automatically pairing human text input with pre-determined categories. More specifically, this thesis covers the research area of ICD classification, which is about pairing free-text clinical notes with ICD codes. For more information about NLP, see Jurafsky & Martin (2020).

The most straightforward solution to text classification problems is to use supervised machine learning models, such as Support Vector Machines, Decision Trees, and K-nearest Neighbors, to detect what makes the input (patient records) be paired with the output classes (ICD codes). However, deep learning models have recently challenged traditional supervised learning approaches to perform classification tasks with text input. More specifically, a deep learning language model called the Bidirectional Encoder Representations from Transformers (BERT) model has since its development in 2018 (Devlin et al.) frequently been used for several types of NLP tasks, including ICD classification.

## 1.1 Problem

The problem that manually assigning ICD codes to clinical notes is time-consuming and error-prone can be solved by automating clinical coding. To solve this practical problem, this thesis aims to address the theoretical problem that not enough is known of which ICD classification approaches perform best in automatically pairing Swedish clinical notes with the correct ICD codes. The latest developments within the area of NLP show great promise in using the BERT model for ICD classification (for example, see Zhang et al. 2020). However, until recently, there has not been a BERT model trained on Swedish texts, and BERT has not yet been used to perform automatic clinical coding using Swedish medical records. Looking at other approaches than BERT, there have been a few previous ICD classification attempts on Swedish clinical notes (Henriksson et al. 2011 and Henriksson & Hassel 2013). Still, these attempts are scarce and do not use the latest developments within text classification using deep learning.

## 1.2 Research Question

This thesis aims to evaluate the effectiveness of using the deep learning language model pre-trained on Swedish text, KB-BERT (Malmsten et al. 2020), for Swedish clinical coding. To put KB-BERT's performance in a context, it is compared to traditional supervised machine learning models. Comparing different ICD classification approaches will add to the existing knowledge of automating Swedish clinical coding. Closing this knowledge gap will, in turn, be a step towards developing a Swedish computer-assisted clinical coding tool. To have a tool that automates the process of assigning medical records with diagnosis codes would decrease the administrative burden for health personnel. If such a tool is effective, it could also help reduce erroneous coding.
The following research question is set up:

*How well does KB-BERT, compared to traditional supervised machine learning models, perform in pairing Swedish gastrointestinal discharge summaries with the correct ICD codes?*

## 1.3 Delimitations

The ICD system that is considered in this thesis is the Swedish version of the latest release of the ICD system called ICD-10-SE. The ICD-10-SE codes used are delimited to those covering gastrointestinal diseases to align the thesis with the ongoing Norwegian research project ClinCode[1]. Codes are considered at the block level, meaning similar codes are grouped into ten blocks. The clinical notes used are summarizing notes written as the patients are discharged, *discharge summaries*, produced between 2007 and 2014 at four gastrointestinal care units at Karolinska University Hospital.

The collection of discharge summaries and their associated codes is the second version of the Stockholm EPR Gastro ICD-10 Corpus (ICD-10 Corpus), which is part of the Swedish Health Record Research Bank (the Health Bank)[2] – a collection of Swedish medical corpora comprised of over two million patients from Karolinska University Hospital (Karolinska Institutet). The supervised machine learning models that serve as benchmark models are Support Vector Machines, Decision Trees, and K-nearest Neighbors. Performance is represented by micro-averaged $F_1$-*score*.

## 1.4 Thesis Structure

The thesis is structured as follows. The next section, Chapter 2, provides an extended background to the research problem and covers a theoretical framework and related research. Chapter 3 describes the thesis's methodology and ethical considerations. Chapter 4 presents the study's results, and in Chapter 5, the results and their implications are discussed.

[1] More information about the ClinCode project is found on the website: https://ehealthresearch.no/en/projects/clincode-computer-assisted-clinical-icd-10-coding-for-improving-efficiency-and-quality-in-healthcare.

[2] For further information about the Health Bank, see https://dsv.su.se/healthbank and Dalianis et al. (2015).

# Chapter 2

# Extended Background

## 2.1 Theoretical Framework

The following sub-sections cover the theory of ICD coding, multi-label text classification, the baseline models, and the KB-BERT model.

### 2.1.1 ICD Codes

As displayed in Figure 1.1, full ICD codes consist of one letter and three digits. The codes are used to represent the patients' diseases and are assigned to the patient records. The system is hierarchical, and full three-digit codes belong to the same group of diseases at a two-digit level. In turn, multiple two-digit codes are grouped into blocks, and multiple blocks are grouped into chapters. The ICD-10 system is comprised of 22 chapters, and each chapter represents a category of diseases.

This thesis is delimited to codes from Chapter XI, which covers diseases of the digestive system. In Chapter XI, all codes begin with the letter K. This chapter is divided into ten blocks, where each block represents a subcategory of diseases of the digestive system. There are over 30 000 full codes in the ICD system and over 500 full codes belonging to Chapter XI. The number of codes is increasing as new codes are continuously added. The two-digit codes that the blocks consist of, and descriptions of the blocks are presented in Table 2.1 (WHO 2019).

| ICD Block | Description |
|---|---|
| K00-K14 | Diseases of oral cavity, salivary glands, jaws |
| K20-K31 | Diseases of esophagus, stomach, duodenum |
| K35-K38 | Diseases of appendix |
| K40-K46 | Hernia |
| K50-K52 | Noninfective enteritis, colitis |
| K55-K64 | Other diseases of intestines |
| K65-K67 | Diseases of peritoneum |
| K70-K77 | Diseases of liver |
| K80-K87 | Disorders of gallbladder, biliary tract, pancreas |
| K90-K93 | Other diseases of the digestive system |

Table 2.1: ICD Chapter XI code blocks

### 2.1.2 Multi-label Text Classification

Classification is the task of predicting a categorical output from an input. This is achieved by training a classification model to recognize patterns that disclose what makes an input result in an output. When the model has been trained to predict output from input, it predicts the output of unseen input examples (Alpaydin & Bach 2014). In the context of ICD classification, the clinical notes are the input, and the ICD codes are the output. The goal with ICD classification is to predict the ICD codes of unseen clinical notes correctly. Since patients can have more than one disease at once, one clinical note can be paired with multiple ICD codes. This means that the data, and therefore, the classification task, is of multi-label character. Multi-label classification is different from multi-class classification, where the classes are mutually exclusive (Madjarov et al. 2012). In Figure 2.1, an exemplary clinical note paired with multiple ICD codes is presented. The note is paired with the ICD codes `K44.9` and `K80.4` which correspond to the ICD blocks `K40-K46` and `K80-K87`. Note that the ICD codes are considered at a block level in this thesis.

**Clinical note**

Tidigare helt frisk kvinna med obehag i epigastrium och tilltagande smärta i arcus under 4 dagar. Konstaterat diafragmabråck. Beh för misstänkt gastroenterit utan framgång. CT visade tecken på akut kolecystit och operation genomfördes med framgång. Pat hemskickad med råd att vila i minst 2 v. Fettsnål kost och mindre portioner rekommenderas.

*English translation: Previously completely healthy woman feeling discomfort in epigastrium with increasing pain in arcus for 4 days. Confirmed diaphragmatic hernia. Unsuccessfully treatm for suspected gastroenteritis. CT showed signs of acute cholecystitis. Successful operation. Pat sent home to rest for 2 w min. Low fat diet and smaller portions recommended.*

**ICD codes**

K44.9 – Diaphragmatic hernia without obstruction or gangrene (Block K40-K46)
K80.4 – Acute cholecystitis (Block K80-K87)

Figure 2.1: A toy clinical note with multi-label output

### 2.1.3 Baseline Models

In this section, the theory relating to the traditional supervised machine learning models used as a baseline, Support Vector Machines (SVM), Decision Trees (DT), and K-nearest Neighbors (KNN), are described in further detail. Moreover, the text cleaning techniques and the feature selection for the baseline models are presented. Text cleaning and feature selection are not conducted for KB-BERT since the clinical notes are fed to KB-BERT as they are. SVM, Decision Trees, and KNN are chosen as baseline models since they are well-established basic models used in related studies (see Section 2.2.1).

### Support Vector Machines

Support Vector Machines adopt the idea of solving classification tasks by linearly separating data points. Since much data is not linearly separable, a function (denoted $\phi$) is used to transform the data in its original space into a transformed space in a higher dimension, where a hyperplane can separate the different classes (see Figure 2.2).

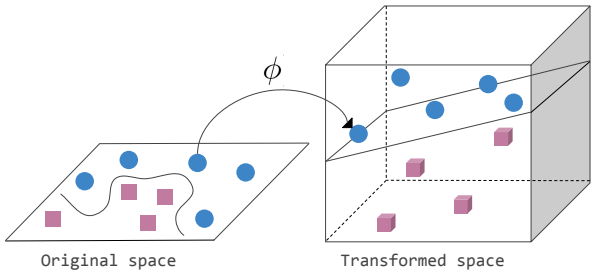


Figure 2.2: An illustration of SVM. Adapted from Shehzadex (2016)

### Decision Trees

Decision Trees learn classification rules in a tree-like structure. When the rules have been learned from training data, unseen examples are taken through the tree's branches until the tree's end node, the leaf, is reached. The leaf node decides the unseen example's class. In Figure 2.3 a Decision Tree model trained to recognize different geometric shapes is displayed. The idea is that once trained, one can feed an unseen geometric shape to the top of the decision tree, and by taking it down the tree, the geometric shape is classified based on if it is a triangle or a square and if it is straight or rotated.

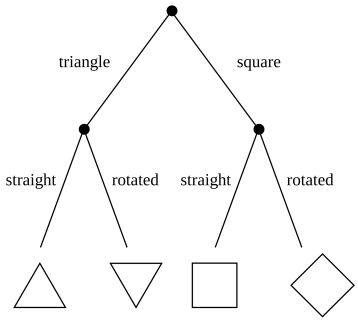


Figure 2.3: An illustration of Decision Trees (Bach 2013)

### K-nearest Neighbors

K-nearest Neighbors assign an unclassified example to the same class as its $K$ closest points (neighbors). In Figure 2.4, the unclassified example (denoted ?) is assigned to the triangle group if $K$ is set to 3 (dotted circle) and to the circle group if $K$ is set to 5 (full drawn circle).

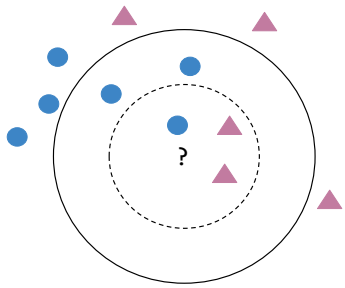


Figure 2.4: An illustration of KNN. Adapted from Maryamvaez (2019)

### Text Cleaning and Feature Selection

When employing supervised learning models such as SVM, Decision Trees, and KNN, it is common to clean the text since it tends to increase classification performance. To clean the text, a common first step is to divide the text into smaller segments and decide what an entity in the text, known as *token*, should be. This process is called *tokenization*. Tokens are usually comprised of words, punctuation, and other symbols.

Two frequently used cleaning steps are to remove punctuation and commonly used words. These common words, known as *stop words*, could either be words that in general often occur in a specific language, like "me" and "by" in English. Stop words could also be defined as words that frequently occur in the study's specific set of documents. When studying medical records, a potential stop word could then, for example, be "patient". The idea is that punctuation and stop words cause noise in the data, distracting the words containing valuable information, and that they, therefore, should be removed (Kowsari et al. 2019).

Two other frequently used cleaning steps are turning all text into lower case letters and grouping inflected versions of a word. These steps are conducted so that two words that look different but have the same or similar meaning, like "Fever" and "fever" or "cough" and "coughed", are interpreted as the same group of words. One common way to group inflected versions of the same word is *stemming*, where all branches of a word are cut off so that only the stem of the word is kept (Kowsari et al. 2019). For example, transforming the words "coughed" and "coughing" into "cough".

Other cleaning techniques that can be used are correcting spelling mistakes and dealing with slang and abbreviations (Kowsari et al. 2019). This could be

particularly effective when handling clinical notes since they are often written in a hurry causing spelling mistakes. Clinical notes are also written by different kinds of health personnel, which can cause the use of different slang and abbreviations. Detecting negated terms could also be vital when analyzing health records, for example, making sure that "no fever" is not interpreted in the same way as "fever" (Dalianis 2018).

After cleaning the data, the text input (clinical notes) needs to be represented as numerical features before it can be fed to the supervised learning models. There are several ways to extract features from the cleaned text. A simple approach is considering each token being a feature and a clinical note's value for a certain feature as the frequency of the token that the feature represents. This approach is known as *Bag-of-Words* (*BoW*). A slightly more sophisticated feature extraction technique is using *Term Frequency-Inverse Document Frequency* (*tf-idf*) weights. With tf-idf weights, instead of using token frequency as the clinical note's value for that feature, a weight representing how important that token is in that clinical note compared to how important it is in all clinical notes is set as the feature value. However, although tf-idf accounts for some of the naivety of *BoW*, it has the same drawback of considering each token independently. To address this issue, an alternative approach is using word embedding techniques, such as *word-to-vector* (*Word2Vec*), where word co-occurrences are accounted for (Kowsari et al. 2019).

### 2.1.4 KB-BERT

The main model evaluated in this thesis is the Swedish version of the deep learning language model BERT. BERT was developed in 2018 by Devlin et al. (2019), and the Swedish version KB-BERT was developed by the National Library of Sweden (KB) in 2020 (Malmsten et al. 2020). BERT is short for Bidirectional Encoder Representations from Transformers. It is named this way since it utilizes the encoder part from the Transformers architecture presented in Vaswani et al. (2017). The Transformer architecture, unlike recurrent neural network designs, uses attention mechanisms. Compared to recurrent designs, attention designs are beneficial since they process all input data at once instead of processing it sequentially. This implies dependencies between input or output entities can be accounted for regardless of the distance between the entities (Vaswani et al. 2017). Unlike many other deep learning models, the Transformers' encoder has a bidirectional design, meaning that it can process text both from left to right and from right to left.

To make the original BERT model a powerful language model, it was pre-trained on around 3.3 billion English words extracted from Wikipedia and a corpus called the BookCorpus (Devlin et al. 2019). The Swedish KB-BERT was pre-trained on about 3.5 billion *Swedish* words withdrawn from sources ranging from newspapers and official reports to social media and the Swedish Wikipedia (Malmsten et al. 2020). Hence, the difference between KB-BERT and the original BERT is that BERT is pre-trained on English text, and KB-BERT is pre-trained on Swedish text, making KB-BERT an expert in Swedish.

The pre-trained model can then be used for many different kinds of NLP tasks, such as text classification, by adding a classification layer.

## 2.2 Related Research

In this section, other ICD classification studies are presented. Firstly, studies using rule-based or traditional machine learning models are discussed. Secondly, research employing deep learning approaches, including the BERT model, is considered.

### 2.2.1 Rule-based and Machine Learning Models

ICD classification has been a popular research area for decades, and there is a myriad of studies conducted within the field. Scholars' interest in the area increased with a shared task called *The 2007 Computational Medicine Challenge* hosted by the Computational Medical Center. During this challenge, pairing radiology reports with the corresponding ICD codes was attempted by over 150 participants. There was both rule-based and traditional machine learning components among the best-achieving contributions of the shared task, resulting in $F_{micro}$ of 0.8-0.9. Common traits of the more successful approaches were that they considered negations as well as synonyms or hypernyms. One of the best-performing systems used Decision Trees (Pestian et al. 2007).

Since 2007, there have been plenty of ICD classification studies using rule-based, traditional machine learning, and deep learning approaches. In 2018, Wang et al. (2018) examined around 250 articles dealing with clinical information extraction published between 2009 and 2016. Out of these papers, 65 percent used rule-based methods, and 23 percent used traditional machine learning algorithms. None of the reviewed articles in Wang et al. (2018) utilized deep learning methods, which is telling of the rapid recent development within the area of deep learning.

One current study employing hand-crafted rules to perform ICD classification in practice is Zhou et al. (2020). In Zhou et al. (2020), Chinese patient records were paired with ICD codes by using combinations of regular expressions representing the different codes. When trying this system in a hospital, it achieved high precision (0.89) but lower recall (0.26) and $F_1$-*score* (0.39) (see Section 3.2.2 for definitions of precision, recall, and $F_1$-*score*). The low recall of this study was due to the high number of false negatives, which the authors of the study defined as being ICD codes that could have been automatically assigned but were not. An ICD code was not assigned automatically if the rules could not match the diagnosis description to an ICD code or if the diagnosis description could fit multiple rules. Moreover, the program that automatically assigns the ICD codes could only be run at specific times, which led to many false negatives since many codes were assigned manually in-between these times. The fact that this study, unlike many other studies, applies an automatic coding

tool in practice and that practical issues were leading to a low recall score makes the results hard to compare with other papers within the area.

Many recent studies have also used traditional machine learning approaches. In the area of unsupervised learning, Kavuluru et al. (2013) used a combination of text summarization, Named Entity Recognition (NER), and graph mining, achieving an average precision of 0.47 and an average recall of 0.42. More specifically, Kavuluru et al. (2013) used NER to map concepts in the summarized clinical notes to concepts in the ICD codes as well as graph mining to make use of note similarity when mapping. Like Kavuluru et al. (2013), Sonabend W et al. (2020), use mappings between concepts in clinical notes and ICD codes, but also applies a word embedding approach to utilize concepts in the clinical notes that cannot directly be mapped to ICD codes. This combination of methods resulted in an average AUC score of 0.92.

However, the most traditional approach for ICD classification, given the labeled data, is to use supervised learning models. Commonly used conventional supervised machine learning models in related studies are SVM, Logistic Regressions, Conditional Random Fields, Decision Trees, Naive Bayes, and K-nearest Neighbors, as well as ensembles of these single models, for example, Random Forests (Wang et al. 2018). The most frequently used traditional machine learning algorithm in the clinical information extraction papers reviewed in Wang et al. (2018) was SVM. An example of an article employing SVM for ICD classification is Koopman et al. (2015), where death certificates were paired with ICD codes belonging to the diseases diabetes, influenza, pneumonia, and HIV, resulting in a $F_1$-*score* of 0.80. When only considering the diseases and not the fine-grained ICD codes, a $F_1$-*score* of 0.96 was achieved.

### 2.2.2 Deep Learning Approaches

Recent developments within the deep learning area have impacted the field of ICD classification. An example of a study comparing SVM to a deep learning approach for ICD classification is Li et al. (2019). In Li et al. (2019) Convolutional Neural Networks (CNN) is combined with a word embedding method applied at a document level (document to vector) into something they call the DeepLabler. The DeepLabler achieves $F_{micro}$ of 0.3-0.4, which was superior to the performance of both flat and hierarchical SVM (Li et al. 2019). Another study focusing on deep learning models for ICD classification is Blanco et al. (2020) comparing recurrent and non-recurrent models, as well as different word embedding approaches. Blanco et al. (2020) concludes that recurrent models outperform non-recurrent models and that there are more powerful ways to use word embeddings than the standard setup.

Some previous studies do not only compare a few classifiers but employ multiple of the models previously mentioned. For example, Kaur & Ginige (2018) use both rule-based, traditional machine learning, and deep learning methods as well as simple pattern matching to pair Australian clinical notes with the correct ICD codes. In this study, simple pattern matching and rule-based methods are compared to the single supervised classifiers SVM, Naive

Bayes, Decision Trees, and K-nearest Neighbors, as well as to the ensemble models Random Forests and AdaBoost, and the deep learning model Multi-Layer Perceptrons (MLP). Kaur & Ginige (2018) conclude that AdaBoost and Decision Trees perform the best at a $F_1$-*score* of 0.91 and 0.87, respectively. Moreover, Kaur & Ginige (2018) conclude that the performance of some of the models like MLP and Random Forests may have been negatively affected by the small size of the data and that they, therefore, in future efforts, will use more training data.

Although the comparative study by Kaur & Ginige (2018) was comprehensive at its time, further progress has since been made at the intersection of deep learning and NLP. In 2019, the BERT model was developed by Devlin et al.. The idea of BERT was to feed a model with great amounts of texts, providing it with a deep understanding of human language. This pre-trained model can then be used for various NLP tasks, such as text classification, by adding a classifier layer.

Since the arrival of BERT in 2018, it has been used in several ICD classification studies. For example, Sänger et al. (2019) who used the multi-lingual version of BERT to pair German summaries of animal experiments with the correct ICD codes, achieving a $F_1$-*score* of 0.80. Amin et al. (2019) used the same German animal records, translated them into English, and fed them to a version of BERT pre-trained on biomedical texts by Lee et al. (2020) called BioBERT. Amin et al. (2019) achieved a $F_{micro}$ of 0.73. Zhang et al. (2020) evaluate both BioBERT and a BERT model pre-trained on clinical texts called Clinical BERT (Alsentzer et al. 2019) and compare these models to their version called BERT-XML. BERT-XML is both pre-trained on medical texts and adds a layer accounting for that ICD classification can result in an extreme amount of classes. The BERT-XML achieves an AUC-macro score of 0.93, which can be compared with 0.9 for Clinical BERT and 0.91 for BioBERT.

There are also ICD classification studies using BERT models trained on other languages than English, for example, López Úbeda et al. (2020) using BETO, which is pre-trained on Spanish text. However, the performance of BETO is beaten by a multi-lingual model called XLM, which was developed by Lample & Conneau (2019) and trained to understand connections between different languages. The XLM model performed at a $F_1$-*score* of 0.7 (López Úbeda et al. 2020).

It would be desirable to be able to conclude which the best approach for ICD classification is based on the related research and use that approach henceforth. However, as has become evident from the literature review, the papers' different evaluation methods and metrics make the ICD classification attempts hard to compare fairly. Moreover, as Stanfill et al. (2010) suggest in a review of clinical coding papers, the performance of the coding systems in previous studies is hard to generalize since it is highly context-dependent. Stanfill et al. (2010) claim that the performance is highly interdependent on the complexity of the task, for example, the granularity of the output classes.

### 2.2.3 Swedish Context

There are two previous studies conducting ICD classification using data from the Health Bank. The first paper by Henriksson et al. (2011) employed a word embedding approach using word co-occurrences. More specifically, a word space method called Random Indexing was used to train a model to pair clinical notes with the most semantically correlated tokens matching ICD codes' pattern. This approach yielded the results that the correct ICD code was present among the top 10 suggested codes in 20 percent of the cases. When looking at partial matches, meaning there was a match with a code category on a higher abstraction level, the correct code appeared in the top 10 suggested codes in 77 percent of the cases.

The paper by Henriksson et al. (2011) was further developed in 2013 by Henriksson & Hassel when it was examined whether dimensionality optimization could improve the results of the 2011 paper. Henriksson & Hassel (2013) reached the conclusion that increasing the dimensionality in the 2011 approach increased the performance of up to 18 percentage points.

As previously described, classification attempts using Swedish corpora are limited to word embedding approaches, and newer deep learning models like the BERT model have not yet been tested to perform ICD classification in a Swedish context. Since there seems to be potential in using the BERT model for NLP tasks, including ICD classification, this is the main subject of evaluation in this thesis. While there is a multi-lingual version of the original BERT, this might not perform well in a minor language as Swedish (Malmsten et al. 2020). Instead, the newly developed Swedish version of BERT called KB-BERT, pretrained by the National Library of Sweden (Kungliga Biblioteket) (Malmsten et al. 2020), is used. To put KB-BERT's performance in a context, it is compared to the performance of traditional supervised learning methods. The benchmark models chosen for this thesis are SVM, Decision Trees, and KNN. These models are chosen as a benchmark because they are well-established models that are commonly used in related studies.

# Chapter 3

# Methodology

## 3.1 Research Strategy

This thesis aims to contribute to the sparse existing knowledge of Swedish ICD classification approaches. Since the thesis focuses on knowledge retrieval, it makes it suitable to consider this study being an empirical research project. If the thesis's goal had been to develop a clinical coding tool, rather than gaining knowledge about the tool's underlying classification models, it could have been seen as a design research project.

An experiment was adopted as the research strategy to answer the research question of how well KB-BERT, compared to the baseline classifiers, performs in paring Swedish discharge summaries with the correct ICD codes. An experiment is about determining the effect of some factor. Determining the effect of some factor is accomplished by changing only that factor while keeping everything else constant (Alpaydin & Bach 2014). The research question in this thesis is about comparing the performance of different classifiers. This research question could be translated to determining the effect of the classifiers on the evaluation metrics while keeping the data constant, meaning that the classifier is the factor changing in this experiment setup. However, it should be noted that unlike experimental studies in other academic fields where the interest lies in estimating the size and significance of the effect of the factor changing, an experiment is used in this study to determine which the best classifier is and estimate the performance of that classifier. An experiment is the standard approach for comparing machine learning algorithms and the standard research strategy used in related studies (see Section 2.2). For more information about machine learning experiments and keeping the data constant to see which impact changing the algorithm has on the performance metrics, see Chapter 19 in Alpaydin & Bach (2014) and Demsar (2006).

An alternative research strategy could have been a case study, qualitatively evaluating the classifiers' performance on a few notes. For example, one could analyze how the classifiers handled certain phrasings in the notes, shedding light

on the classifiers' inner workings. This approach could be seen as comparing classifier performance in terms of the quality of the classifiers' basis for decisions. While such a qualitative comparison of performance would be interesting, it is not the focus of this thesis. Moreover, this kind of analysis could not determine how the classifiers perform on average and if there is a statistically significant difference in average classifier performance. In other words, case studies have weak generalizability (external validity) since their result may not be applicable for other instances than the ones examined. On the contrary, conducting an experiment using a large, representative sample can lead to high external validity. The reliability is also generally higher for quantitative research than for qualitative research since the researcher themselves are not part of producing the results (Denscombe 2014).

The degree of external validity of the experiment depends on how similar the experiment's environment is to the environment that it wishes to study (Johannesson & Perjons 2014). For example, while real clinical notes are used in the experiments in this thesis, they were created ten years ago and may not represent how clinical notes look today. Moreover, the validity of the data could be an issue since it is likely that the clinical notes and their assigned ICD codes are not perfectly aligned.

The overall research strategy of conducting an experiment to address the thesis's research question needs to be accompanied by methods for collecting and analyzing the experiment's data. The following sections describe the data collection methods and data analysis methods.

## 3.2 Data Collection

The data collection of this thesis can be divided into two parts; (i) collecting the electronic patient records (EPR) that are fed to the classifiers, and (ii) collecting the performance metrics of those classifiers. The methods used for collecting these two types of data are (i) *extracting information* from already existing EPR stored in a database, and (ii) *observing* evaluation metrics as they are calculated in the software used. Collecting data from EPR is a way of utilizing information from already existing *documents*, and collecting evaluation metrics as they are calculated is a way of *observing* a phenomenon as it occurs (Johannesson & Perjons 2014). The EPR and the evaluation metrics are described in the following sub-sections.

### 3.2.1 Electronic Patient Records

#### The Data

The raw data used in this thesis are electronic patient records (EPR), which are created through the system TakeCare CGM at Karolinska University Hospital. These EPR are available in the Health Bank, which is a research infrastructure kept at the Department of Computer and Systems Sciences at Stockholm University.[1]

The EPR that were extracted from the Health Bank were chosen based on several criteria. To align this thesis with the ongoing Norwegian project ClinCode[2] that focuses on gastrointestinal clinical notes, the EPR were firstly screened based on care unit. Care units were searched for containing the keywords *gast*, *stom* (*mag* in Swedish), or *abdom* (*buk* in Swedish). Within those care units, clinical note templates were searched for the keyword *discharge summary* (*epikris* in Swedish). EPR belonging to these templates in these units were then filtered to contain ICD codes belonging to diseases of the digestive system (ICD Chapter XI, which contains ICD codes starting with a K).

Moreover, to ensure that the ICD codes belong to the discharge summaries, the time stamp of the clinical note could not be more than 24 hours before or after the time of the discharge. To maximize the learning of the classifiers, it is essential that the clinical notes are aligned with the ICD codes, meaning the ICD codes in some way are referenced in the note. Therefore, discharge summaries were chosen over other clinical notes since they are most likely to contain all information about the patient's symptoms and diseases during the care period.

EPR from care units with many (>1000) EPR fulfilling these criteria were extracted, merged, and used as the raw dataset. This resulted in that data from four care units were used. One of the care units was specialized in gastrointestinal diseases among children, two were labeled as gastrointestinal centers, and one was a gastrointestinal surgical satellite unit. Data from more units could have been collected, however, extracting and merging EPR from units with few discharge summaries would have taken more time without contributing with much more data. The data extraction was conducted using the query language SQL.

Using existing discharge summaries and assigned ICD codes as the raw data can be seen as *extracting information from documents* (EPR) as a data collection method. An alternative method to collect ICD codes would be to let multiple clinicians assign ICD codes to the discharge summaries. While this approach probably would lead to a dataset with fewer coding errors, it requires access to clinicians willing to manually code. Since many clinical notes are needed for classification tasks, manual annotation is not considered feasible for this thesis.

[1] For further information about the Health Bank, see https://dsv.su.se/healthbank and Dalianis et al. (2015).

[2] More information about the ClinCode project is found on the website: https://ehealthresearch.no/en/projects/clincode-computer-assisted-clinical-icd-10-coding-for-improving-efficiency-and-quality-in-healthcare.

### Pre-processing the Data

Before the evaluation metrics could be collected, the medical documents had to be pre-processed. Firstly, all EPR that were missing *note*, *ICD code*, *patient id*, or *care event id* were removed. Secondly, duplicate EPR with the same *note*, *ICD code*, *patient id*, and *care event id* were removed. Then, the ICD codes were grouped into ten classes representing which part of the digestive system the diagnosis belongs to. These classes correspond to the pre-determined blocks of the ICD Chapter XI that are described in Table 2.1 (see WHO (2019) for more information). As Blanco et al. (2020) suggest, multi-label classification is challenging and becomes even more demanding if full codes are considered. Reducing the number of classes to the block level is considered an appropriate measure in this thesis to give the classifiers a fair chance to perform well. If full codes were considered, and all classification models would under-perform, it might not have been possible to examine performance differences, which is the purpose of this study. However, for future studies, training classification models on full ICD codes is of great interest since a clinical coding tool used in practice should propose full codes.

Moreover, to allow each note to be paired with multiple ICD blocks (multi-label classification), all identical notes belonging to the same patient and care event were grouped, allowing each unique combination of *note*, *patient id*, and *care event id* to be assigned multiple ICD blocks. When examining the notes, it was discovered that some notes only contained the name of the clinician or the dates the patient was hospitalized. Since these notes are not associated with the ICD blocks, all notes with less than four tokens were removed. Then, to further improve alignment between notes and ICD blocks, all notes that had the same *patient id*, *care event id*, and *ICD block* were merged. The final dataset that was the result of these pre-processing steps is the second version of the Stockholm EPR Gastro ICD-10 Corpus (ICD-10 Corpus).

The number of discharge summaries (notes) during the pre-processing steps, a description of the ICD-10 Corpus and its characteristics are presented in Table 3.1, 3.2, and 3.3. The distribution of number of ICD blocks, the ICD block distribution, and the distribution of number of tokens are presented in Figure 3.1, 3.2, and 3.3.

| | |
|---|---|
| Notes initially | 41 267 |
| Notes - removing NaNs and duplicates | 33 731 |
| Notes - merging identical notes, patient, care event | 27 224 |
| Notes - removing notes with <4 tokens | 18 008 |
| Notes - merging identical ICD block, patient, care event | 6 062 |

Table 3.1: Number of notes during pre-processing steps

| | |
|---|---|
| Notes after pre-processing | 6 062 |
| Unique patients | 4 985 |
| Tokens | 986 436 |
| Unique tokens (vocabulary) | 48 232 |
| Unique ICD blocks | 10 |

Table 3.2: Description of ICD-10 Corpus

| | **Min** | **Median** | **Mean** | **Max** | **Std** |
|---|---|---|---|---|---|
| Tokens per note | 4 | 134 | 162.7 | 1794 | 120.5 |
| ICD blocks per note | 1 | 1 | 1.2 | 4 | 0.4 |

Table 3.3: Descriptive statistics of ICD-10 Corpus

The pre-processing steps led to that the ICD-10 Corpus consists of 6 062 discharge summaries (notes). These notes come from 4 985 unique patients, and each note is paired with one or multiple of the ten unique ICD blocks. There are 48 232 unique tokens and 986 436 tokens in total in the ICD-10 Corpus. As displayed in Figure 3.1, the most common scenario is that one note is paired with one ICD block. However, there exist instances paired with up to four different ICD blocks, but not more.

Looking at Figure 3.2, the most common ICD block is K55-K64 at 2 670 notes, followed by K55-K64 at 1 116 notes. The least common ICD block, K00-K14 only has four assigned notes to it. In Figure 3.3, the number of notes per number of tokens is presented. On average, a note is 163 tokens long. However, the note length varies a lot, and there are both shorter and longer notes than that.

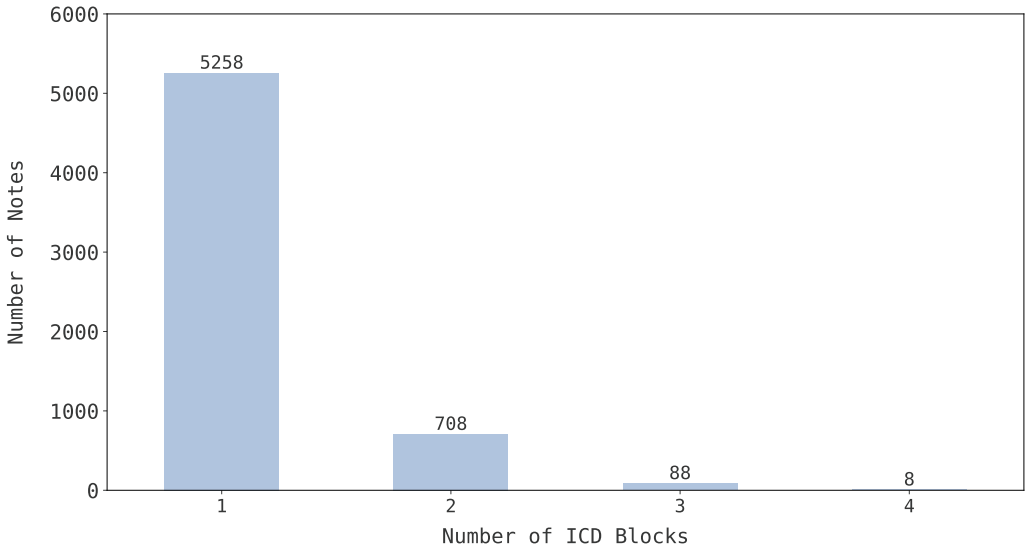


Figure 3.1: Number of notes per number of ICD blocks

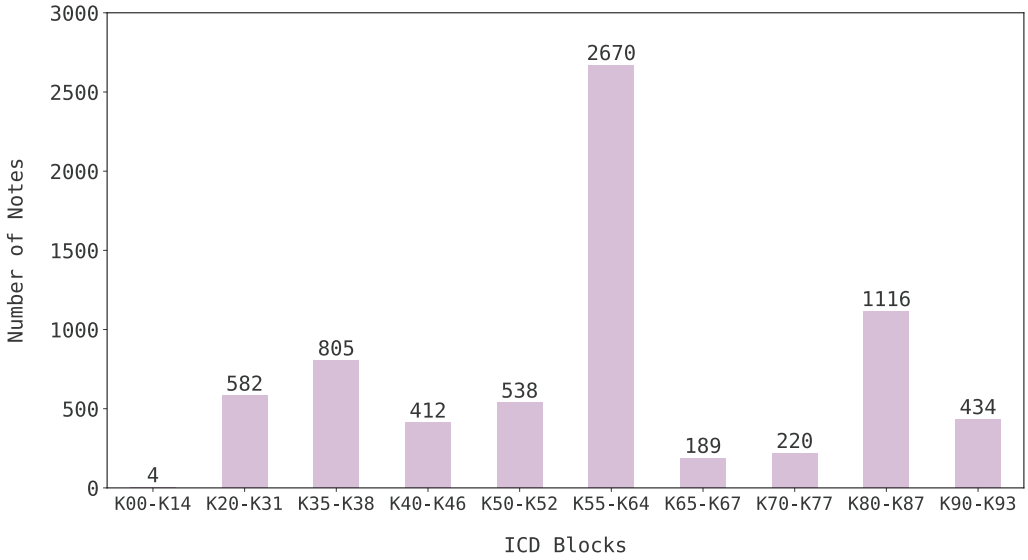


Figure 3.2: Number of notes per ICD blocks

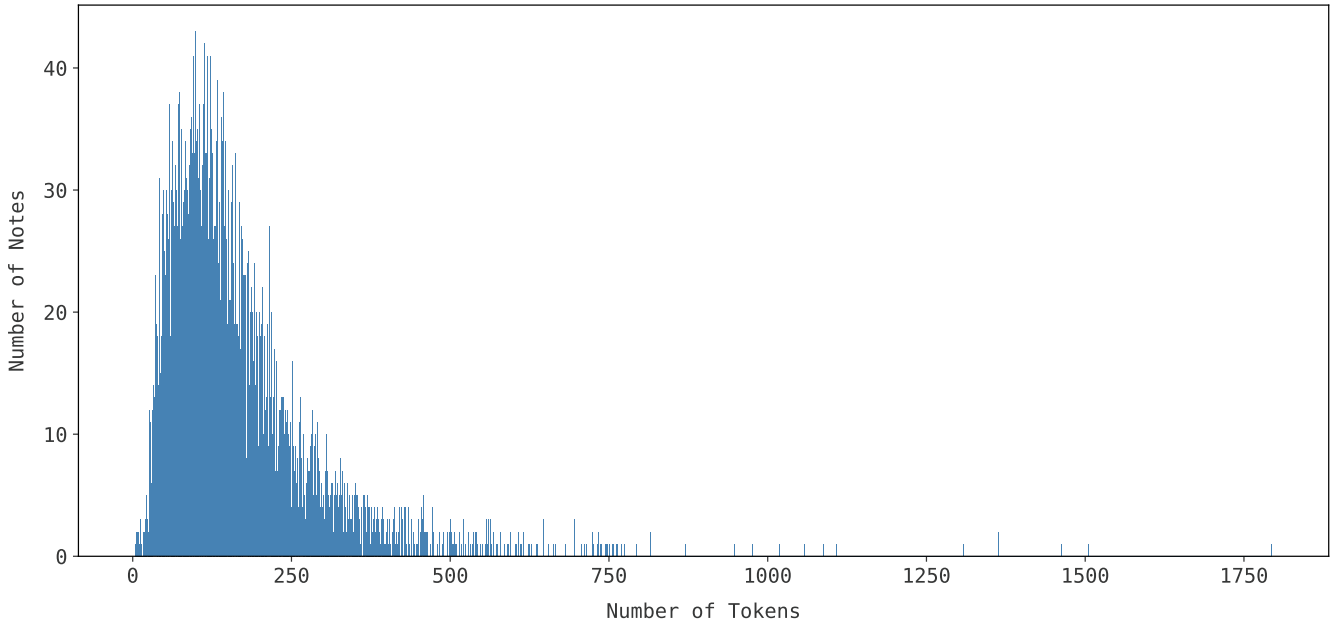


Figure 3.3: Number of notes per number of tokens

### 3.2.2 Evaluation Metrics

It is not enough to analyze the raw EPR data to address this thesis's research question; the classifiers' performance needs to be evaluated to determine how KB-BERT, compared to the baseline classifiers, performs in pairing clinical notes with ICD codes. Evaluation metrics are used to quantify performance, and these metrics were collected by *observing them directly* as they were calculated using Python 3.7.3 and the scikit-learn library (Pedregosa et al. 2011).

The evaluation metrics used in this thesis are precision, recall, and the harmonic mean of precision and recall known as the $F_1$-*score*. These metrics were chosen since they are well-established and commonly used in related studies. If there are $C$ number of classes (ICD codes), and a single class is denoted as $c$, examples that are correctly classified as class $c$ are known as True Positives ($TP_c$). Examples that correctly are not classified as class $c$ are called True Negatives ($TN_c$), and examples that are incorrectly classified as class $c$ are called False Positives ($FP_c$). Examples that are incorrectly not classified as class $c$ are known as False Negatives ($FN_c$) (Tharwat 2020). $TP_c$, $TN_c$, $FP_c$, and $FN_c$ are used to calculate precision and recall, which in turn are used to calculate the $F_1$-*score*. The formulas for precision, recall, and $F_1$-*score* for each class $c$ are given in Equations 3.1, 3.2, and 3.3 Kavuluru et al. (2015).

$$Precision\ (P_c) = \frac{TP_c}{TP_c + FP_c} \tag{3.1}$$

$$Recall\ (R_c) = \frac{TP_c}{TP_c + FN_c} \tag{3.2}$$

$$F_1\text{-}score_c\ (F_c) = \frac{2P_cR_c}{P_c + R_c} \tag{3.3}$$

The average precision and recall across the whole test set are defined differently depending on if micro or macro averaging is used. Macro averaging entails taking the unweighted mean of precision/recall per class, while micro averaging implies taking the mean of the precision/recall per pair of class and example (Yang 1999). While both micro and macro averaging are used in related studies, micro averaging is more commonly used. Some previous studies only report one of the two, and some report both metrics. As Pestian et al. (2007) explains, macro averaging that assigns equal importance to each class would be most relevant to use if it is most important that the classifier is able to correctly predict as many classes (ICD codes) as possible. On the other hand, micro averaging would be more appropriate if it is most important that the classifier successfully can predict as many examples (patient records) as possible (Pestian et al. 2007). To compare the classifiers in this study, the view of Pestian et al. (2007) that micro-averaging is the more reasonable approach for ICD classification is adopted, and micro-averaged $F_1$-*score* ($F_{micro}$) is used as the primary evaluation metric. However, for the sake of transparency, the macro-averaged $F_1$-*score* ($F_{macro}$) is also reported.

Since computational time and the classifier's carbon footprint also is an important aspect of classifier performance, the classifiers' running time was added as an evaluation metric. The time it takes to train and test the classifier was observed using Python's built-in time module.[3]

## 3.3 Data Analysis

An experiment was conducted to determine whether changing the classifier while keeping the data constant has an impact on the evaluation metrics. Within the experiment, multiple evaluation metrics were collected, and this data was analyzed using inferential statistics. *Inferential statistics*, compared to the alternative quantitative data analysis method *descriptive statistics*, is about being able to draw conclusions beyond the specific sample observed (Johannesson & Perjons 2014). Inferential statistics was used as the data analysis method over descriptive statistics since the aim was to be able to draw conclusions about how the classifiers perform on the dataset in general (all possible combinations of training and test sets), rather than describe how the classifiers performed on the particular parts of the data that was used for training and testing. In the following sub-sections, the experiment setup, the implementations of the classification models, and the statistical testing are outlined.

### 3.3.1 Experiment Design

To draw conclusions about which classifier performs the best and to say whether there is a statistically significant difference between the classifiers' performance on the dataset, multiple performance metrics for each classifier need to be collected. Several observations are required for each classifier to ensure the observed difference in classifier performance is not due to pure chance. For example, it might be the case that one classifier seems to outperform another when trained and tested on a specific set of instances, while the opposite is true for all other combinations of training and testing partitions.

Collecting multiple performance metrics per classifier can be accomplished by dividing the data into smaller pieces and collecting one evaluation metric for each classifier and each piece of data. This method is called k-fold cross-validation, where $k$ specifies the number of pieces (folds) the data is divided into (Alpaydin & Bach 2014). $k$, i.e., the sample size, was set to `10` in this thesis. This particular $k$ was chosen since it is a standard choice considered leading to a good trade-off between having a very high $k$ resulting in low bias but high variance estimates and having a very low k resulting in low variance but high bias estimates of the performance (Hastie et al. 2009). The folds were divided randomly, making the sample of data points that the models are tested on random.

[3] https://docs.python.org/3/library/time

The 10-fold cross-validation was conducted on the vast majority of the full dataset. A separate part of the data was set aside to do a final evaluation of the performance of the KB-BERT and the baseline model that performed the best during the 10-fold cross-validation. The former is known as the training set, and the latter is known as the held-out test set. Since deep learning models require large amounts of training data, the great majority, 90 percent of the data, is used for training, and 10 percent of the data is set aside as the held-out test set. This 90/10 partition is a common choice in machine learning experiments. The partitioning of data points into a training set and a held-out test set was conducted using the scikit-learn library[4] (Pedregosa et al. 2011). The test size was set to `0.1`, and the data was shuffled before partitioned. No stratification was used.

The experiment design is illustrated in Figure 3.4. It shows how the full dataset was divided into a training set and a held-out test set (step 1). It also demonstrates how the training part of the dataset was partitioned into ten pieces (folds) to conduct 10-fold cross-validation (step 2). The purpose of the 10-fold cross-validation was to compare the KB-BERT model to the baseline classifiers. Step 2.1–2.10 illustrates that nine folds of the data were used for training, and one fold was used for testing in each of the ten runs. In step 3, the final performance of the main model of interest, the KB-BERT, and the best out of the baseline classifiers were trained on the full training set and tested on the held-out dataset.

[4]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split

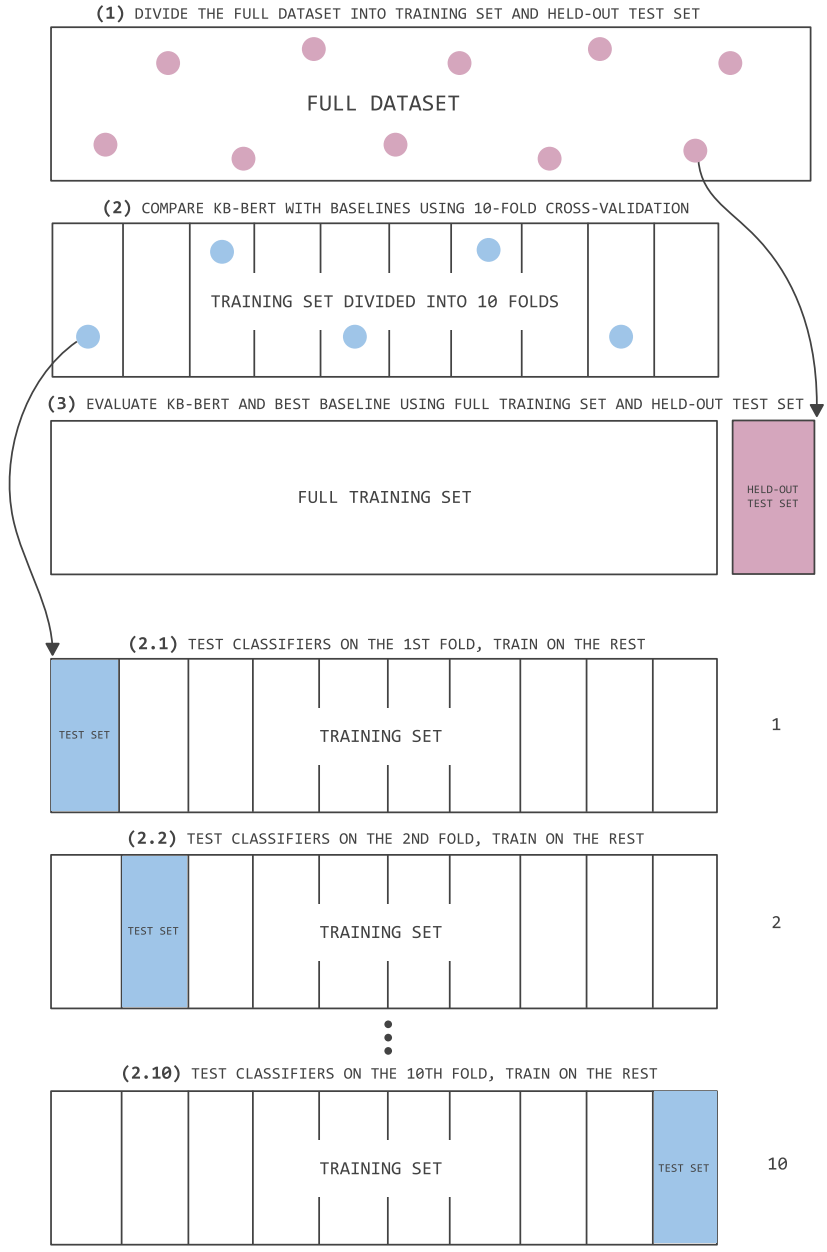


Figure 3.4: An illustration of the experiment setup and 10-fold cross-validation

### 3.3.2 Model Implementations

The main classifier subject to evaluation in this thesis was KB-BERT. The version of KB-BERT used was the Swedish cased $\text{BERT}_{BASE}$[5]. This basic pre-trained model was chosen over fine-tuned versions suitable for other NLP tasks such as Named Entity Recognition (NER) and Part-of-speech (POS) tagging. While KB-BERT has been pre-trained to understand the Swedish language, an additional layer needed to be added to perform classification tasks. The Huggingface Transformers library (Wolf et al. 2020) offers implementations of different NLP tasks that allow for BERT models (including KB-BERT) to be utilized. However, it does not provide an implementation for multi-label classification tasks. Some previous studies employ multi-label classification using BERT. For example, Amin et al. (2019) do multi-label ICD classification using a BERT model and provide the code to their implementation[6], but this is not adapted for Swedish text. Therefore, a customized multi-label classification implementation of KB-BERT constructed by Anastasios Lamproudis[7] using the Huggingface Transformers library (Wolf et al. 2020) and the Pytorch library (Paszke et al. 2019) was used for this thesis.

For the baseline models, the scikit-learn implementation of Decision Trees[8] developed by Pedregosa et al. (2011), and the scikit-multilearn implementation of KNN[9] developed by Szymański & Kajdanowicz (2018) were used. Since the scikit-learn implementation of SVM[10] is not directly suited for multi-label classification, one classifier was trained per label using the scikit-learn implementation of the one-vs-rest method[11]. Basic cleaning steps, including removing punctuation, decapitalization, and removal of stop words, were conducted. The list of stop words used was the one provided by the Natural Language Toolkit (NLTK)[12] created by Bird et al. (2009).

[5] https://huggingface.co/KB/bert-base-swedish-cased
[6] https://github.com/suamin/multilabel-classification-bert-icd10.
[7] anastasios@dsv.su.se
[8] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier
[9] http://scikit.ml/api/skmultilearn.adapt.mlknn
[10] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC
[11] https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier
[12] https://www.nltk.org/

To represent the notes as numerical features before feeding them to the baseline models, tf-idf weights were used using the scikit-learn library[13] (Pedregosa et al. 2011). *tf* stands for term frequency and represents the number of times a certain token is mentioned in a note. *idf* stands for inverse document frequency and is given by taking the log of the ratio of the number of notes ($N$) and the number of notes a token appears in ($df$) (see Equation 3.4). tf-idf is given by multiplying the term frequency ($tf$) with the inverse document frequency ($idf$) as displayed in Equation 3.5 (Kowsari et al. 2019). tf-idf was used over other alternatives since it is commonly used in related studies.

$$idf = log(\frac{N}{df}) \tag{3.4}$$

$$tf\text{-}idf = tf\ x\ idf \tag{3.5}$$

**Hyper-parameters**

It should be noted that the classifiers have hyper-parameters. These hyper-parameters can be seen as the classifiers' settings, and different settings result in slightly different versions of the classifiers. It is considered outside the scope of this thesis to optimize the hyper-parameters. However, it is worth noting that the performances of the models could have benefited from hyper-parameter optimization.

The hyper-parameters selected for the baseline classifiers were the ones set as the default in their implementation. Using the model implementations' default parameters is a common approach in related studies. For example, see Mujtaba et al. (2017) and Hasan et al. (2016). The hyper-parameters of the SVM, Decision Trees, and KNN are presented in Table 3.4, 3.5, and 3.6.

[13] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer

| Hyper-parameter | Value |
|---|---|
| Regularization Parameter (C) | 1 |
| Kernel | Rbf |
| Kernel Coefficient (gamma) | Scale |
| Shrinking | True |
| Tolerance for Stopping Criterion (tol) | 0.001 |
| Size of Kernel Cache (cache_size) | 200 |

Table 3.4: SVM hyper-parameters

| Hyper-parameter | Value |
|---|---|
| Split Quality Criteria (criterion) | Gini |
| Splitting Strategy (splitter) | Best |
| Minimum Samples Required for Split (min_samples_split) | 2 |
| Minimum Samples Required in Leaf Node (min_samples_leaf) | 1 |

Table 3.5: Decision Trees hyper-parameters

| Hyper-parameter | Value |
|---|---|
| Number of Neighbors (k) | 10 |
| Smoothing Parameter (s) | 1 |

Table 3.6: KNN hyper-parameters

For the KB-BERT, the hyper-parameters were aligned with those used in related studies. To be able to use a batch size of 32 as suggested in Devlin et al. (2019) despite having a GPU with limited memory, the batch size was set to 2 and the gradient accumulation was set to 16. The learning rate of 0.00002 suggested in Devlin et al. (2019) was used.

The KB-BERT model outputs a vector of dimension one multiplied by the number of ICD blocks with floating numbers between zero and one. Each number in the vector represents how likely it is that an ICD block belongs to the clinical note – the closer to one, the more likely it is that the ICD block should be assigned to the note. However, since the output is the result of using the Sigmoid function, these numbers should not be interpreted as probabilities. To binarize the vector into zeros (ICD block not assigned) and ones (ICD block assigned), the activation threshold of 0.5 was used, meaning values of 0.5 and above for a particular ICD block resulted in that ICD block being assigned to the note. 0.5 was chosen simply because it is in-between zero and one. The number of warm-up steps was set to 155 because this amount approximately allowed the model to see all the data once during the warm-up phase.

The hyper-parameters of the KB-BERT are presented in Table 3.7. Hyper-parameters with a value equal to zero, none, or false are not presented. The random state and NumPy random seed equal to 123 were used throughout the study.

| **Hyper-parameter** | **Value** |
|---|---|
| Batch Size | 2 |
| Gradient Accumulation | 16 |
| Learning Rate | 0.00002 |
| Optimizer | Adam |
| Number of Warm-up Steps | 155 |
| Activation Threshold | $\geq$ 0.5 |

Table 3.7: KB-BERT hyper-parameters

Devlin et al. (2019) use 2 to 4 epochs for fine-tuning tasks. To allow for learning beyond four epochs, early stopping was used in this study. Early stopping implies stopping training when the validation loss starts increasing. This study follows Devlin et al. (2019) and used binary cross-entropy (BCE) loss. In Figure 3.5, the number of epochs the KB-BERT is fine-tuned for is plotted against BCE loss, $F_{micro}$, Precision, and Recall, respectively when trained on nine out of the ten of the data folds, and tested on the last of the ten folds. Each fold of the data was trained until the BCE loss stops decreasing. In Figure 3.5, the mean values during the ten folds are represented by the solid line, and the shaded area around the line represents the mean plus and minus the standard deviation. During most of the ten folds, the KB-BERT trained for seven epochs before the loss stopped decreasing. Therefore, when the KB-BERT was trained on the full training set and tested on the held-out dataset, it was trained for seven epochs.

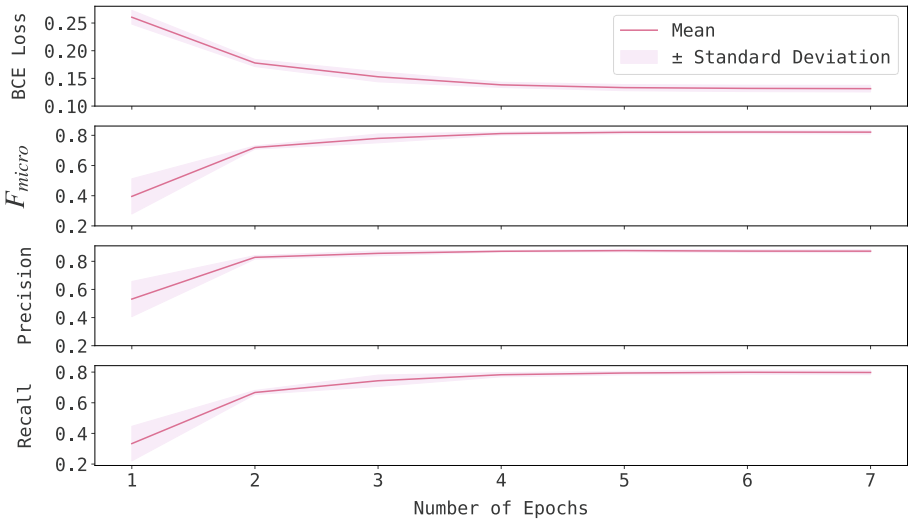


Figure 3.5: KB-BERT performance metrics during the 10-fold cross-validation

### 3.3.3 Statistical Testing

Having $k$ number of evaluation metrics per classifier enables determining the probability of observing the difference in classifier performance in the $k$ number of data folds (the sample) given that there is no true difference in how the classifiers, in general, perform on the dataset. If this probability is lower than a pre-specified threshold known as the significance level, there is evidence that the classifiers' performance on the given dataset indeed differs. A standard significance level that also was used in this thesis is `0.01`. The significance level was set to `0.01` instead of the other typical level `0.05` since it in this project was considered more preferable not to reject a false null-hypothesis (Type II-error) than to reject a true null-hypothesis (Type I-error). In other words, this choice was made since it was considered worse to claim that there is a difference in classifier performance when there actually is not, then there is to claim that such a difference is not supported when there actually exists a difference (see Manderscheid (1965) for a discussion on significance levels and types of errors).

Moreover, having a small sample size comes with the risk of low statistical power. While low power is mostly known for increasing the risk of not rejecting false null-hypotheses, low power also increases the share of falsely rejected null-hypotheses (false positive rate). Therefore, to keep the false positive rate low, choosing a lower significance level is also appropriate for that reason (Benjamin et al. 2018). It was also considered suitable to choose the lower rather than the higher significance level since multiple hypotheses were tested, which increases

the risk of at least one of those hypotheses being falsely rejected. Setting the significance level to `0.01` means that if the probability is less than 1 percent to observe the difference in classifier performance in the sample, given there is no actual difference in classifier performance, the null-hypothesis that there is no true difference in classifier performance is rejected. If the null-hypothesis is rejected, the alternative hypothesis that there is a true difference in classifier performance is trusted.

The null-hypotheses ($H1_0$-$H3_0$) with their corresponding alternative hypotheses ($H1_A$-$H3_A$) were set up to test if there is a true difference in performance between the KB-BERT, and the baseline models Support Vector Machines (SVM), Decision Trees (DT), and K-nearest Neighbors (KNN). Performance was represented by the $F_{micro}$ during the 10-fold run (denoted $F$ in the hypotheses). More exactly, the null-hypotheses (alternative hypotheses) represent that the distribution of $F_{micro}$ scores are (not) the same for the classifiers. If the null-hypotheses are rejected, there is reason to trust the alternative hypotheses stating that the KB-BERT does not perform the same as the baseline models. The baseline models are also compared in the hypotheses $H4_0$-$H6_0$. All hypotheses reflect two-tailed tests since both positive and negative performance differences are of interest.

$$H1_0 : F_{KBBERT} = F_{SVM} \tag{3.6}$$
$$H1_A : F_{KBBERT} \neq F_{SVM} \tag{3.7}$$

$$H2_0 : F_{KBBERT} = F_{DT} \tag{3.8}$$
$$H2_A : F_{KBBERT} \neq F_{DT} \tag{3.9}$$

$$H3_0 : F_{KBBERT} = F_{KNN} \tag{3.10}$$
$$H3_A : F_{KBBERT} \neq F_{KNN} \tag{3.11}$$

$$H4_0 : F_{SVM} = F_{DT} \tag{3.12}$$
$$H4_A : F_{SVM} \neq F_{DT} \tag{3.13}$$

$$H5_0 : F_{SVM} = F_{KNN} \tag{3.14}$$
$$H5_A : F_{SVM} \neq F_{KNN} \tag{3.15}$$

$$H6_0 : F_{DT} = F_{KNN} \tag{3.16}$$
$$H6_A : F_{DT} \neq F_{KNN} \tag{3.17}$$

The statistical tests that were conducted to test the hypotheses were Wilcoxon signed-rank tests. These non-parametric tests were chosen over similar parametric tests since the assumptions of normality of parametric tests were difficult to assess. The Wilcoxon signed-rank test is used for paired observations, which is the case in this experiment. The observations are paired since when comparing two of the classifiers, one evaluation metric per classifier was calculated for the same pieces (folds) of the data (Demsar 2006). The Wilcoxon test uses the difference in classifier performance and ranks the performance difference, assigning the highest rank to the greatest difference. The rank of the difference ($d$) of the first classifier's performance minus the second classifier's performance in sample $i$ is denoted $rank(d_i)$. The ranks belonging to the differences where classifier 1 was better performing than classifier 2 (positive differences) are summed and denoted $R^+$. Similarly, the ranks belonging to the differences where classifier 2 outperformed classifier 1 (negative differences) are summed and denoted $R^-$. The ranks of zero differences ($d_i = 0$) are evenly split between $R^+$ and $R^-$. The formulas for $R^+$ and $R^-$ are given in Equations 3.18 and 3.19 (Demsar 2006).

$$R^+ = \sum_{i=1}^{k} rank(d_i > 0) + 0.5 \sum_{i=1}^{k} rank(d_i = 0) \tag{3.18}$$

$$R^- = \sum_{i=1}^{k} rank(d_i < 0) + 0.5 \sum_{i=1}^{k} rank(d_i = 0) \tag{3.19}$$

$k$ denotes the sample size, which is equal to the number of folds that the data is divided into during k-fold cross-validation. The Wilcoxon test statistic $T$ is given by the smallest value of $R^+$ and $R^-$, and the null-hypotheses are rejected when $T$ is equal or less than the critical value $T_{crit}$ that is determined by the sample size and the significance level (Demsar 2006).

$$T = min(R^+, R^-) \tag{3.20}$$

The hypothesis tests are conducted using Python `3.7.3` and the Wilcoxon function in the SciPy Stats[14] (Virtanen et al. 2020) module.

## 3.4 Ethics

It is of utmost importance that research does not cause harm to or disrespect the privacy of its participants (Denscombe 2014). This thesis's participants are the patients whose medical records constitute the raw data. Unedited, these medical records contain personal data such as name and social security number alongside sensitive information such as symptom descriptions. It is fair to assume that most patients would be uncomfortable sharing their medical records with other people than their caregivers and loved ones. Thus, to prevent psychological distress among the patients, the EPR available in the Health Bank

[14] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html

are partly de-identified, meaning all structured personal data such as name and social security number has been removed from the EPR. However, there can still be information in the EPR that indirectly can disclose a patient's identity. For example, there can be information in the free-text clinical notes about the patient's address, phone number, or family members that can disclose their identity. To address the sensitive nature of the clinical notes used for training the models in this thesis, the data is treated with caution. The Health Bank containing the full dataset of millions of de-identified patient records, is kept in a secure room only accessed by a few researchers that have signed confidentiality agreements. Whenever smaller subsets of data is extracted from the Health Bank, it is only allowed to be kept in encrypted devices or containers, and the data is never kept or sent over the internet. The patient records are never shared with people outside the research team.[15]

Moreover, as mentioned in Chapter 1, the data used in this thesis most likely contains errors that have occurred when health personnel assigns the wrong codes to clinical notes. Since the classifiers cannot distinguish incorrect codes from correctly classified codes, the connection between attributes in the clinical notes and the incorrectly classified codes is most likely learned by the classifiers. This is a two-folded problem. Firstly, it means the classifiers will not be as well-performing when used in a real clinical coding tool as they could have been, would they have been trained on correctly assigned codes. Secondly, this poor performance will not be reflected in the evaluation metrics; it might look like the classifiers manage to classify unseen examples correctly, but what is labeled as true codes in the data may not be the actual true codes. This second ethical issue is the most relevant in the current study since it focuses on knowledge retrieval. Therefore, it should be considered that the performance metrics observed in this thesis may be overly optimistic. In future studies that would use the classifiers to develop a clinical coding tool to be used in actual health facilities, the first issue of having poorly performing classifiers would be the most pressing ethical issue.

Another crucial ethical aspect to consider when conducting research projects is to remain unbiased towards the results of the study. This implies not, knowingly or accidentally, steering the results in some direction (Denscombe 2014, Alpaydin & Bach 2014). To avoid this, the hypotheses, the significance level, and other research design aspects were determined before conducting the experiment in this thesis. Moreover, all results are reported, not only those in favor of previous beliefs or desires. Besides the impact this research is likely to have on humans, it is reasonable to address its effects on the climate. Training and testing classification models consume electricity, which in its production emits greenhouse gases. Different classifiers demand different amounts of electricity, which implies that some classifiers have a larger carbon footprint than others. Since this thesis focuses on comparing classifiers, this comparison also includes the classifiers' running time.

[15]This research has been approved by the Regional Ethical Review Board in Stockholm under permission no. 2007/1625-31/5.

# Chapter 4

# Results

## 4.1 Classifier Comparison

In Figure 4.1, the $F_{micro}$ obtained for the KB-BERT and the baseline models during each of the ten runs during the 10-fold cross-validation are displayed. Table 4.1 presents the macro and micro averaged precision ($P$), recall ($R$), $F_1$-*score* ($F$) for all ten folds combined. Moreover, the time it took (in minutes) to train and test the KB-BERT and the baseline models using 10-fold cross-validation is also presented in Table 4.1.

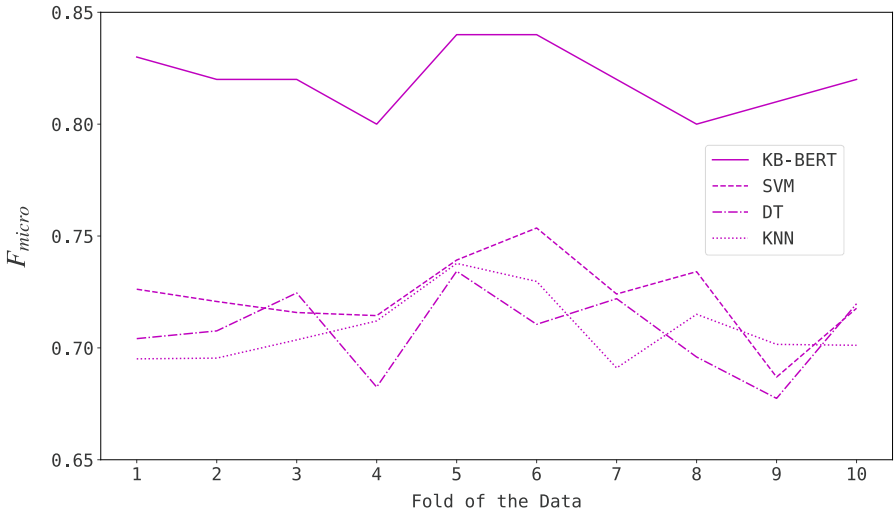


Figure 4.1: $F_{micro}$ scores during the 10-fold cross-validation

| Classifier | $P_{macro}$ | $R_{macro}$ | $F_{macro}$ | $P_{micro}$ | $R_{micro}$ | $F_{micro}$ | Minutes |
|---|---|---|---|---|---|---|---|
| KB-BERT | 0.67 | 0.55 | **0.60** | 0.87 | 0.77 | **0.82** | 300 |
| SVM | 0.76 | 0.33 | **0.41** | 0.90 | 0.61 | **0.72** | 22 |
| DT | 0.54 | 0.50 | **0.52** | 0.72 | 0.69 | **0.71** | 1 |
| KNN | 0.63 | 0.41 | **0.48** | 0.79 | 0.64 | **0.71** | 0.6 |

Table 4.1: Performance metrics during the 10-fold cross-validation

Looking at the main performance metric of interest, $F_{micro}$, Table 4.1 shows that for the ten folds combined, the KB-BERT outperformed all three baseline classifiers. Moreover, it is shown that the SVM performed slightly better than the Decision Trees and the KNN. Significance tests were conducted to test whether these performance differences are likely to be true for the dataset as a whole or only due to chance.

In Table 4.2, the test statistics ($T$) and p-values from the Wilcoxon signed-rank tests are presented. All p-values relating to tests comparing the KB-BERT and the baseline models were smaller than the significance level 0.01, meaning the null-hypotheses $H1_0$–$H3_0$ are rejected, and there is reason to trust the alternative hypotheses $H1_A$–$H3_A$. These results represent that the differences in performance observed between the KB-BERT and the baseline models during the 10-fold cross-validation most likely are not only due to chance but reflect that the KB-BERT and the baseline models actually perform differently on this dataset.

The p-values belonging to the hypotheses comparing the baseline models ($H4_0$–$H6_0$) were not below 0.01, meaning there is no support for the claim that the baseline models performed differently.

| Hypothesis | Compared Classifiers | $T$ | p-value |
|---|---|---|---|
| $H1_0$ | KB-BERT and SVM | 0 | 0.00195 |
| $H2_0$ | KB-BERT and DT | 0 | 0.00195 |
| $H3_0$ | KB-BERT and KNN | 0 | 0.00195 |
| $H4_0$ | SVM and DT | 5 | 0.01953 |
| $H5_0$ | SVM and KNN | 4 | 0.01367 |
| $H6_0$ | DT and KNN | 26 | 0.92188 |

Table 4.2: Statistical test results

The p-values were identical for the tests comparing KB-BERT and the baseline models because the test statistic $T$ was identical for all these tests. No baseline model was superior to the KB-BERT in any of the data folds, meaning the sum of ranks of the baseline classifiers, and, therefore, $T$ was 0 for all pairwise comparisons of KB-BERT and the three baseline classifiers (see Equations 3.18, 3.19, and 3.20 for details of how $T$ is calculated).

## 4.2 Final Evaluation

Since the KB-BERT is the main model of interest in this thesis, its final $F_{micro}$ was estimated by training it on the full training set and testing it on the held-out dataset (see Figure 3.4 for more information about the experiment setup). The final $F_{micro}$ of the best performing baseline classifier during the 10-fold cross-validation, the SVM, was also estimated by training it on the full training set and testing it on the held-out test set. As presented in Section 4.1, it should be noted that SVM was not statistically significantly superior to the Decision Trees and the KNN. However, SVM is used in the final evaluation as a representative of the baseline models to compare the KB-BERT to.

The results of the final evaluation are presented in Table 4.3 and Table 4.4. The $F_{micro}$ obtained was 0.80 for the KB-BERT and 0.71 for the SVM.

| ICD Block | Precision | Recall | $F_1$-score | Support |
|---|---|---|---|---|
| K00-K14 | 0.00 | 0.00 | 0.00 | 0 |
| K20-K31 | 0.74 | 0.50 | 0.60 | 64 |
| K35-K38 | 0.95 | 0.92 | 0.94 | 79 |
| K40-K46 | 0.91 | 0.67 | 0.77 | 48 |
| K50-K52 | 0.88 | 0.70 | 0.78 | 73 |
| K55-K64 | 0.81 | 0.88 | 0.84 | 258 |
| K65-K67 | 0.00 | 0.00 | 0.00 | 22 |
| K70-K77 | 0.73 | 0.42 | 0.54 | 26 |
| K80-K87 | 0.96 | 0.90 | 0.93 | 120 |
| K90-K93 | 0.77 | 0.28 | 0.41 | 36 |
| micro-averaged | 0.86 | 0.75 | **0.80** | 726 |
| macro-averaged | 0.68 | 0.53 | **0.58** | 726 |

Table 4.3: KB-BERT final evaluation

| ICD Block | Precision | Recall | $F_1$-score | Support |
|---|---|---|---|---|
| K00-K14 | 0.00 | 0.00 | 0.00 | 0 |
| K20-K31 | 0.94 | 0.23 | 0.38 | 64 |
| K35-K38 | 0.97 | 0.77 | 0.86 | 79 |
| K40-K46 | 1.00 | 0.23 | 0.37 | 48 |
| K50-K52 | 1.00 | 0.25 | 0.40 | 73 |
| K55-K64 | 0.82 | 0.89 | 0.85 | 258 |
| K65-K67 | 0.00 | 0.00 | 0.00 | 22 |
| K70-K77 | 1.00 | 0.19 | 0.32 | 26 |
| K80-K87 | 0.97 | 0.70 | 0.81 | 120 |
| K90-K93 | 1.00 | 0.14 | 0.24 | 36 |
| micro-averaged | 0.88 | 0.59 | **0.71** | 726 |
| macro-averaged | 0.77 | 0.34 | **0.42** | 726 |

Table 4.4: SVM final evaluation

Table 4.4 and 4.3 display performance results per ICD block. Both models received zero precision and recall for the first group, K00-K14. This is explained by the fact that zero discharge summaries belong to that group in the held-out test set. Moreover, neither of the models managed to correctly label group K65-K67, which received zero precision and recall for both models. The fact that two groups received zero precision and recall explains the discretion between $F_{micro}$ and $F_{macro}$.

Both the KB-BERT and the SVM achieved higher precision than recall. While recall and $F_{micro}$ were higher for the KB-BERT than for the SVM, the SVM obtained a slightly higher precision. This implies that while the SVM was not able to correctly identify as many of the true ICD blocks as the KB-BERT, the ICD blocks that the SVM identified are, to a greater extent, the correct ones.

When predicting the samples in the held-out dataset five times, on average, the running time was 10 and 14 seconds for the KB-BERT and the SVM, respectively. It should be noted that the prediction time of the KB-BERT, unlike the prediction time of the SVM, was measured by running the model using a GPU, so there is not enough evidence to conclude that the KB-BERT predicts unseen examples faster than the SVM.

# Chapter 5

# Discussion and Conclusion

## 5.1 Summary

The research question of this thesis was: *How well does KB-BERT, compared to traditional supervised machine learning models, perform in pairing Swedish gastrointestinal discharge summaries with the correct ICD codes?* The discharge summaries were produced between 2007 and 2014 at four gastrointestinal care units at Karolinska University Hospital. The ICD codes were grouped into ten blocks of similar codes (see Table 2.1) and delimited to codes belonging to the digestive system (ICD Chapter XI).

90 percent of the 6 062 discharge summaries was used to compare the performance of the KB-BERT and the baseline models Support Vector Machines (SVM), Decision Trees, and K-nearest Neighbors (KNN). 10-fold cross-validation was conducted, and the $F_{micro}$ of the ten folds combined was higher for the KB-BERT than for the baseline models. These performance differences were statistically significant. The baseline model with the highest $F_{micro}$ during the 10-fold cross-validation was the SVM, which had a slightly higher $F_{micro}$ than the Decision Trees and the KNN. However, neither of the performance differences between the baseline models were statistically significant.

While there is no support for the SVM performing better than the other baseline models, the SVM was used in the final evaluation as a representative of the baseline models to compare to the KB-BERT. When training the KB-BERT and the SVM on all training data and testing it on the held-out test set consisting of 10 percent of the 6 062 discharge summaries, the KB-BERT and the SVM achieved final $F_{micro}$ scores of 0.80 and 0.71, and final $F_{macro}$ scores of 0.58 and 0.42, respectively.

## 5.2 Analysis of Findings

Looking at Table 4.3 and 4.4, it was interesting that both the KB-BERT and the SVM failed to correctly predict group `K00-K14` and `K65-K67`. One possible explanation for this could be that the models do not have a big enough number of instances to train on. However, there could also be other reasons for the poor results for this group. For example, this group may have fewer characterizing features.

It was interesting to note that while the KB-BERT had a higher recall and $F_{micro}$ than the SVM, the SVM had a higher precision than the KB-BERT. When developing a clinical coding tool to be used in health facilities, it is vital to pre-define what is most important; to correctly identify as many of the actual ICD codes or that the ICD codes identified are the actual ones. The former represents a tool with high recall, while the latter represents a tool with high precision. If the tool would automatically assign ICD codes without the supervision of health personnel, it may be reasonable to prefer a classifier with high precision. However, if the tool would be semi-automatic and suggest codes which the health personnel then approves, it might be more reasonable to premier high recall.

Moreover, it is worth noting that the time it took to run the 10-fold cross-validation for the KB-BERT was substantially longer than for the SVM, Decision Trees, and KNN (see the Minutes column in Table 4.1). This implies that the carbon footprint of the KB-BERT is greater than for the baseline models. Therefore, if concluding that the KB-BERT outperforms the baseline models increases the usage of deep learning models over traditional supervised learning models for Swedish ICD classification tasks, this study may negatively impact the climate. Consequently, employing deep learning models like KB-BERT in future ICD classification projects should be followed by considering the models' trade-offs between running time and performance. However, it could be argued that the training time of KB-BERT can be shortened by training it for fewer epochs. While the validation BCE loss on average stopped decreasing at seven epochs, it became evident from Figure 3.5 that there only were marginal improvements to the loss as well as the performance metrics after epoch four. Hence, using early stopping with a stricter stopping criterion could reduce the carbon footprint of training KB-BERT.

Furthermore, it could be argued that training the models is a one-time procedure and that the time that matters the most is the time it takes to predict unseen examples. When examining the time it took to predict the held-out data set once the models were trained on the entire training set, the results were similar for the KB-BERT and the SVM, both taking about 10-15 seconds. Hence, while KB-BERT, compared to the SVM, takes substantially longer to train, the two models' prediction times are comparable.

As discussed in Section 2.2.2, it is difficult to compare the results of this study to other related studies. While the $F_{micro}$ scores of this study are similar to those obtained in other studies, it may not be suitable to compare them since they differ a lot concerning research designs and datasets. However, looking at

this study and one of the few other classification studies conducted on Swedish data, one could argue that considering codes at the block level, as in this study, is somewhat comparable to considering partial code matches such as in Henriksson et al. (2011). Through a word embedding approach, Henriksson et al. (2011) concluded that the correct partial codes were identified in 77 percent of the cases. Still, it is not entirely straightforward how these results should be compared to the KB-BERT obtaining a $F_{micro}$ of 0.80.

Since everything (all data points) was kept equal, and the classifier was the only factor changing in-between observations of $F_{micro}$, the results of this study can be interpreted as that changing classifier affects $F_{micro}$. Note that this relationship is conditional on the specific instantiations of the classifiers and the particular dataset used. However, this study's main objective was not to estimate the effect of the classifiers on performance. Therefore, the performance of the classifiers was reported rather than the size of the effect.

## 5.3 Limitations and Research Quality

It should be kept in mind that the results of this thesis represent how well the classifiers perform on this particular dataset, when the actual interest may lie in how well these models would perform when assigning Chapter XI ICD codes to a random unseen Swedish gastrointestinal clinical note. The notes that the models are trained and tested on may, for several reasons, not be a representative sample of the greater population *Swedish gastrointestinal discharge summaries*. For example, the notes were produced between 2007 and 2014, and it might be the case that notes are written or coded differently today. Moreover, the clinical notes used in the dataset of this thesis were extracted from four care units, and these notes may not represent how clinical notes are written or coded in Swedish gastrointestinal care units in general. With this in mind, it is important to remember that the KB-BERT outperforming the baseline models is not a general truth but true for this thesis's particular dataset and model specifications.

As mentioned in Chapter 1, the data has likely been subject to erroneous coding, meaning the classifiers probably partly will learn incorrect connections between notes and ICD codes. It would have been favorable if the coding quality had been manually assessed to determine how many codes can be assumed to be mislabeled. It would have been even better if all the codes in the data were manually checked by one or multiple manual annotators or clinicians. Since manual annotation is time-consuming and requires medical knowledge, it was not feasible to do it within this project. However, one should bear in mind that the classifiers' performance presented in this thesis may be overestimated due to erroneous coding. Of course, the performance could theoretically also be underestimated, but the fact that the models likely have been trained on erroneously assigned codes is more likely to lead to overestimation.

While there is some uncontrolled randomness when running the KB-BERT, random states are used as seeds when possible and similar results are expected when the same experiment is run multiple times. Moreover, all research design

choices are reported in this thesis. Combining these two factors implies that the repeatability of this study is expected to be very good. Note that for another researcher to reproduce this exact study in practice, they would have to come to the Department of Computer and Systems Sciences at Stockholm University since the data is sensitive and not publicly available. However, trying to replicate the results with a different dataset may not yield the same results, circling back to the discussion above about lacking external validity. The reliability of the performance estimates is ensured by not only running the classifiers once but ten times for randomly partitioned parts of the data.

Furthermore, it is vital to address the fact that the tests used to determine whether the performance differences were statistically significant are based on a small sample of ten observations. While this could result in an underpowered study, a low significance level was set to decrease the false positive rate. Moreover, the Wilcoxon signed-rank tests comparing KB-BERT and the baseline models were conclusive, with intact ranks among the classifiers for all folds of the data, indicating robust performance differences. A larger sample size was not considered feasible since the sample size in this study is equal to the number of folds in k-fold cross-validation, and increasing the $k$ increases the computational time as well as the variance of the expected performance estimates.

## 5.4 Future Research

There are many potential approaches for future related studies. For example, it would be interesting to explore the consequences of fine-tuning the hyper-parameters of the classifiers. Partly, to investigate if this could lead to some of the baseline models outperforming KB-BERT, and partly to examine if fine-tuning hyper-parameters could substantially increase overall performance. Furthermore, the baseline models could have performed better if more text cleaning steps or alternative feature selection methods were used, and including these is suggested in future studies. Since only a few baseline models were considered in this thesis, it would also be relevant to include more classification models to compare KB-BERT to.

Moreover, it is recommended to evaluate KB-BERT's performance compared to traditional supervised machine learning models on other Swedish clinical notes than the particular subset studied in this thesis. It would both be relevant using other gastrointestinal notes, as well as notes from other specialties.

Since this study used ICD blocks, it would also be relevant to consider full ICD codes in future work. It is desirable to consider full codes since health professionals assign full codes, not grouped codes, to the patient records. To still reach satisfactory performance when looking at a larger label set and more finely grained codes, a BERT implementation like the one developed in the forthcoming paper by Blanco et al. (2021) using per-label attention could be of interest. Other BERT models, such as the multi-lingual BERT, could also be a meaningful research design addition. A Swedish BERT model pre-trained on clinical text, similar to the already available English versions, would also be

relevant to develop and test in future studies.

As previously mentioned, when observing the performance results of the classifiers, one should keep in mind that the data used for training most likely contains coding errors. Therefore, it is likely that the performance of the models is overestimated (or, less likely, underestimated). To address this issue, future studies would benefit from re-annotating clinical notes, creating a dataset with higher quality coding.

Last but not least, when enough knowledge is gathered about which classification approaches are appropriate for Swedish ICD classification, it is highly relevant to conduct a design science study to develop a clinical coding tool in close cooperation with its intended end-users. When developing a tool that will be used in practice, it could also be relevant to evaluate the classifiers based on their reasons for decisions. For example, clinicians may prefer a classifier that uses diagnoses and symptoms to assign ICD codes over a classifier with a slightly higher $F_1$-*score* that bases its classification decisions on the names of the patients' doctors or care units. Examining the explanations of classifier decisions can make the users trust the tool more. These explainability mechanisms could also be incorporated in the tool, for example, by highlighting the words in the clinical note that contributed the most to the suggested ICD codes. For an example of how a clinical coding tool could look like, see Montalvo et al. (2018).

## 5.5 Research Impact and Final Remarks

As discussed in Section 3.4, the integrity of the patients whose medical records were used for this study has suffered minimally; there was no explicit personal information in the data, and the data was kept safe at all times. However, using models trained on sensitive data in a clinical coding tool could still cause problems since there is a risk that the tool is hacked and sensitive information is leaked. To address this, a suggestion is to train models used in such a tool on pseudonymized clinical notes. Pseudonymization can be achieved by running the notes through the program HB Deid developed by Dalianis & Berg (2021).

From an environmental perspective, this study has caused some greenhouse gas emissions. Mostly, these emissions are a result of developing and training the KB-BERT, which consumes more computational power than the baseline models. In the long run, this research could also negatively impact the climate if it leads to an increased usage of KB-BERT over the less computationally heavy traditional supervised machine learning models. Therefore, as mentioned in Section 5.2, it is important to consider the trade-off between predictive performance and climate impact when choosing a classifier.

As for other societal consequences, this thesis contributed to the knowledge within the field of Swedish ICD classification. In turn, this could help the development of an effective ICD coding tool. Such a tool would have positive societal consequences by reducing health personnels' administrative burden and improving coding quality.

# Bibliography


Alpaydin, E. & Bach, F. (2014), *Introduction to Machine Learning*, MIT Press, Cambridge, United States.

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jin, D., Naumann, T. & McDermott, M. B. A. (2019), 'Publicly Available Clinical BERT Embeddings', *arXiv:1904.03323 [cs]* . arXiv: 1904.03323 version: 3.
**URL:** *http://arxiv.org/abs/1904.03323*

Amin, S., Neumann, G., Dunfield, K., Vechkaeva, A., Chapman, K. & Wixted, M. (2019), MLT-DFKI at CLEF eHealth 2019: Multi-label classification of ICD-10 codes with BERT, *in* 'CEUR Workshop Proceedings', Vol. 2380, CEUR-WS.

Bach, E. (2013), 'A decision tree for four shapes'. Accessed 2021-03-02.
**URL:** *https://commons.wikimedia.org/wiki/File:Simple_decision_tree.svg*

Benjamin, D. J., Berger, J. O., Johannesson, M., Nosek, B. A., Wagenmakers, E.-J., Berk, R., Bollen, K. A., Brembs, B., Brown, L., Camerer, C., Cesarini, D., Chambers, C. D., Clyde, M., Cook, T. D., De Boeck, P., Dienes, Z., Dreber, A., Easwaran, K., Efferson, C., Fehr, E., Fidler, F., Field, A. P., Forster, M., George, E. I., Gonzalez, R., Goodman, S., Green, E., Green, D. P., Greenwald, A. G., Hadfield, J. D., Hedges, L. V., Held, L., Hua Ho, T., Hoijtink, H., Hruschka, D. J., Imai, K., Imbens, G., Ioannidis, J. P. A., Jeon, M., Jones, J. H., Kirchler, M., Laibson, D., List, J., Little, R., Lupia, A., Machery, E., Maxwell, S. E., McCarthy, M., Moore, D. A., Morgan, S. L., Munafó, M., Nakagawa, S., Nyhan, B., Parker, T. H., Pericchi, L., Perugini, M., Rouder, J., Rousseau, J., Savalei, V., Schönbrodt, F. D., Sellke, T., Sinclair, B., Tingley, D., Van Zandt, T., Vazire, S., Watts, D. J., Winship, C., Wolpert, R. L., Xie, Y., Young, C., Zinman, J. & Johnson, V. E. (2018), 'Redefine statistical significance', *Nature Human Behaviour* **2**(1), 6–10. Number: 1 Publisher: Nature Publishing Group.
**URL:** *https://www.nature.com/articles/s41562-017-0189-z*

Bird, S., Klein, E. & Loper, E. (2009), *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, "O'Reilly Media, Inc.". Google-Books-ID: KGIbfiiP1i4C.

Blanco, A., Perez-de Viñaspre, O., Pérez, A. & Casillas, A. (2020), 'Boosting ICD multi-label classification of health records with contextual embeddings and label-granularity', *Computer Methods and Programs in Biomedicine* **188**, 105264.
**URL:** *https://www.sciencedirect.com/science/article/pii/S0169260719311745*

Blanco, A., Remmer, S., Pérez, A., Dalianis, H. & Casillas, A. (2021), 'On the contribution of per-ICD attention mechanisms to classify health records in languages with fewer resources than English (submitted)'.

Dalianis, H. (2018), *Clinical Text Mining*, Springer International Publishing, Cham.
**URL:** *http://link.springer.com/10.1007/978-3-319-78503-5*

Dalianis, H. & Berg, H. (2021), HB Deid - HB De-identification tool demonstrator, *in* 'Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa), May 31-June 2, 2021'.

Dalianis, H., Henriksson, A., Kvist, M., Velupillai, S. & Weegar, R. (2015), 'HEALTH BANK - A Workbench for Data Science Applications in Healthcare', *CEUR Workshop Proceedings Industry Track Workshop* pp. 1–18. Publisher: CEUR Workshop Proceedings.
**URL:** *http://ceur-ws.org/Vol-1381/paper1.pdf*

Demsar, J. (2006), 'Statistical Comparisons of Classifiers over Multiple Data Sets', *Journal of Machine Learning Research* **7**, 1–30.
**URL:** *https://www.jmlr.org/papers/volume7/demsar06a/demsar06a.pdf*

Denscombe, M. (2014), *The Good Research Guide : For Small-scale Research Projects*, Vol. Fifth edition of *Open UP Study Skills*, McGraw-Hill Education, Maidenhead, Berkshire.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019), 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', *arXiv:1810.04805 [cs]* . arXiv: 1810.04805 version: 2.
**URL:** *http://arxiv.org/abs/1810.04805*

Farkas, R. & Szarvas, G. (2008), 'Automatic construction of rule-based ICD-9-CM coding systems', *BMC Bioinformatics* **9**(3), S10.
**URL:** *https://doi.org/10.1186/1471-2105-9-S3-S10*

Hasan, M., Kotov, A., Idalski Carcone, A., Dong, M., Naar, S. & Brogan Hartlieb, K. (2016), 'A study of the effectiveness of machine learning methods for classification of clinical interview fragments into a large number of categories', *Journal of Biomedical Informatics* **62**, 21–31.
**URL:** *http://www.sciencedirect.com/science/article/pii/S153204641630034X*

Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer New York, New

York, NY.
**URL:** *https://www.springer.com/gp/book/9780387848570*

Henriksson, A. & Hassel, M. (2013), 'Optimizing the Dimensionality of Clinical Term Spaces for Improved Diagnosis Coding Support', *Proceedings of the 4th International Louhi Workshop on Health Document Text Mining and Information Analysis (Louhi 2013)* . Publisher: NICTA.
**URL:** *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.644.7550*

Henriksson, A., Hassel, M. & Kvist, M. (2011), Diagnosis Code Assignment Support Using Random Indexing of Patient Records – A Qualitative Feasibility Study, *in* M. Peleg, N. Lavrač & C. Combi, eds, 'Artificial Intelligence in Medicine', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 348–352.
**URL:** *https://link.springer.com/chapter/10.1007/978-3-642-22218-4_45*

Jacobsson, A. & Serdén, L. (2013), 'Kodningskvalitet i patientregistret (In Swedish)'. Accessed 2021-04-01.
**URL:** *https://www.socialstyrelsen.se/globalassets/sharepoint-dokument/artikelkatalog/statistik/2013-3-10.pdf*

Johannesson, P. & Perjons, E. (2014), *An Introduction to Design Science*, Springer International Publishing.
**URL:** *https://www.springer.com/gp/book/9783319106311*

Jurafsky, D. & Martin, J. (2020), *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (Draft)*, 3 edn.
**URL:** *https://web.stanford.edu/%7Ejurafsky/slp3/*

Kaur, R. & Ginige, J. (2018), 'Comparative Analysis of Algorithmic Approaches for Auto-Coding with ICD-10-AM and ACHI', *Studies in health technology and informatics* **252**, 73–79.
**URL:** *https://doi.org/10.3233/978-1-61499-890-7-73*

Kavuluru, R., Han, S. & Harris, D. (2013), Unsupervised Extraction of Diagnosis Codes from EMRs Using Knowledge-Based and Extractive Text Summarization Techniques, *in* O. R. Zaïane & S. Zilles, eds, 'Advances in Artificial Intelligence', Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 77–88.
**URL:** *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5524149/*

Kavuluru, R., Rios, A. & Lu, Y. (2015), 'An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records', *Artificial Intelligence in Medicine* **65**(2), 155–166.
**URL:** *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4605853/*

Koopman, B., Karimi, S., Nguyen, A., McGuire, R., Muscatello, D., Kemp, M., Truran, D., Zhang, M. & Thackway, S. (2015), 'Automatic classification

of diseases from free-text death certificates for real-time surveillance', *BMC Medical Informatics and Decision Making* **15**(1), 53.
**URL:** *https://doi.org/10.1186/s12911-015-0174-2*

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L. & Brown, D. (2019), 'Text Classification Algorithms: A Survey', *Information* **10**(4), 150. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.
**URL:** *https://www.mdpi.com/2078-2489/10/4/150*

Lample, G. & Conneau, A. (2019), 'Cross-lingual Language Model Pretraining', *arXiv:1901.07291 [cs]* . arXiv: 1901.07291.
**URL:** *http://arxiv.org/abs/1901.07291*

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H. & Kang, J. (2020), 'BioBERT: a pre-trained biomedical language representation model for biomedical text mining', *Bioinformatics* **36**(4), 1234–1240.
**URL:** *https://doi.org/10.1093/bioinformatics/btz682*

Li, M., Fei, Z., Zeng, M., Wu, F.-X., Li, Y., Pan, Y. & Wang, J. (2019), 'Automated ICD-9 Coding via A Deep Learning Approach', *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **16**(04), 1193–1202. Publisher: IEEE Computer Society.
**URL:** *https://doi.org/10.1109/TCBB.2018.2817488*

López Úbeda, P., Díaz-Galiano, M. C., Urena Lopez, L. A., Martin, M., Martín-Noguerol, T. & Luna, A. (2020), Transfer learning applied to text classification in Spanish radiological reports, *in* 'Proceedings of the LREC 2020 Workshop on Multilingual Biomedical Text Processing (MultilingualBIO 2020)', European Language Resources Association, Marseille, France, pp. 29–32.
**URL:** *https://www.aclweb.org/anthology/2020.multilingualbio-1.5*

Madjarov, G., Kocev, D., Gjorgjevikj, D. & Džeroski, S. (2012), 'An extensive experimental comparison of methods for multi-label learning', *Pattern Recognition* **45**(9), 3084–3104.
**URL:** *https://www.sciencedirect.com/science/article/pii/S0031320312001203*

Malmsten, M., Börjeson, L. & Haffenden, C. (2020), 'Playing with Words at the National Library of Sweden – Making a Swedish BERT', *arXiv:2007.01658 [cs]* . arXiv: 2007.01658.
**URL:** *http://arxiv.org/abs/2007.01658*

Manderscheid, L. V. (1965), 'Significance Levels. 0.05, 0.01, or?', *Journal of Farm Economics* **47**(5), 1381–1385. Publisher: [Oxford University Press, Agricultural & Applied Economics Association].
**URL:** *https://www.jstor.org/stable/1236396*

Maryamvaez (2019), 'Example of k-NN classification'. Accessed 2021-04-01.
**URL:** *https://commons.wikimedia.org/wiki/File:Knnclass.png*

Montalvo, S., Almagro, M., Martínez, R., Fresno, V., Lorenzo, S., Morales, M. C., González, B., Álamo, J. & García-Caro, A. (2018), Graphical User Interface for assistance with ICD-10 coding of Hospital Discharge Records, *in* '2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)', pp. 2786–2788.
**URL:** *https://doi.org/10.1109/BIBM.2018.8621420*

Mujtaba, G., Shuib, L., Raj, R. G., Rajandram, R., Shaikh, K. & Al-Garadi, M. A. (2017), 'Automatic ICD-10 multi-class classification of cause of death from plaintext autopsy reports through expert-driven feature selection', *PLOS ONE* **12**(2), e0170242. Publisher: Public Library of Science.
**URL:** *https://doi.org/10.1371/journal.pone.0170242*

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. & Chintala, S. (2019), 'PyTorch: An Imperative Style, High-Performance Deep Learning Library', *arXiv:1912.01703 [cs, stat]* . arXiv: 1912.01703.
**URL:** *http://arxiv.org/abs/1912.01703*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011), 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research* **12**(85), 2825–2830.
**URL:** *http://jmlr.org/papers/v12/pedregosa11a.html*

Pestian, J. P., Brew, C., Matykiewicz, P., Hovermale, D., Johnson, N., Cohen, K. B. & Duch, W. (2007), A shared task involving multi-label classification of clinical free text, *in* 'Biological, translational, and clinical language processing', Association for Computational Linguistics, Prague, Czech Republic, pp. 97–104.
**URL:** *https://www.aclweb.org/anthology/W07-1013*

Shehzadex (2016), 'The process of making linearly separable data in another dimension'. Accessed 2021-03-03.
**URL:** *https://commons.wikimedia.org/wiki/File:Kernel_yontemi_ile_veriyi_daha_fazla_dimensiyonlu_uzaya_tasima_islemi.png*

Socialstyrelsen (2006), 'Diagnosgranskningar utförda i Sverige 1997–2005 samt råd inför diagnosgranskning (in Swedish)'. Accessed 2021-04-01.
**URL:** *https://www.socialstyrelsen.se/globalassets/sharepoint-dokument/dokument-webb/klassifikationer-och-koder/drg-diagnosgranskningar-utforda-sserige-1997-2005-rad-infor-diagnosgranskning.pdf*

Sonabend W, A., Cai, W., Ahuja, Y., Ananthakrishnan, A., Xia, Z., Yu, S. & Hong, C. (2020), 'Automated ICD coding via unsupervised knowledge inte-

gration (UNITE)', *International Journal of Medical Informatics* **139**, 104135.
**URL:** *http://www.sciencedirect.com/science/article/pii/S1386505619313024*

Stanfill, M. H., Williams, M., Fenton, S. H., Jenders, R. A. & Hersh, W. R. (2010), 'A systematic literature review of automated clinical coding and classification systems', *Journal of the American Medical Informatics Association* **17**(6), 646–651. Publisher: Oxford Academic.
**URL:** *https://academic.oup.com/jamia/article/17/6/646/843154*

Szymański, P. & Kajdanowicz, T. (2018), 'A scikit-based Python environment for performing multi-label classification', *arXiv:1702.01460 [cs]* . arXiv: 1702.01460.
**URL:** *http://arxiv.org/abs/1702.01460*

Sänger, M., Weber, L., Kittner, M. & Leser, U. (2019), Classifying German animal experiment summaries with multi-lingual BERT at CLEF eHealth 2019 task 1, *in* 'CEUR Workshop Proceedings', Vol. 2380, CEUR-WS.

Tharwat, A. (2020), 'Classification assessment methods', *Applied Computing and Informatics* **17**(1), 168–192. Publisher: Emerald Publishing Limited.
**URL:** *https://doi.org/10.1016/j.aci.2018.08.003*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017), 'Attention Is All You Need', *arXiv:1706.03762 [cs]* . arXiv: 1706.03762 version: 5.
**URL:** *http://arxiv.org/abs/1706.03762*

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S. J. v. d., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, I., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F. & Mulbregt, P. v. (2020), 'SciPy 1.0: fundamental algorithms for scientific computing in Python', *Nature Methods* **17**(3), 261–272. Number: 3 Publisher: Nature Publishing Group.
**URL:** *https://www.nature.com/articles/s41592-019-0686-2*

Wang, Y., Wang, L., Rastegar-Mojarad, M., Moon, S., Shen, F., Afzal, N., Liu, S., Zeng, Y., Mehrabi, S., Sohn, S. & Liu, H. (2018), 'Clinical information extraction applications: A literature review', *Journal of Biomedical Informatics* **77**, 34–49.
**URL:** *http://www.sciencedirect.com/science/article/pii/S1532046417302563*

WHO (2019), 'ICD-10 Version:2019'. Accessed 2021-02-22.
**URL:** *https://icd.who.int/browse10/2019/en*

WHO (2020), 'Classification of Diseases (ICD)'. Accessed 2021-02-04.
**URL:** *https://www.who.int/standards/classifications/classification-of-diseases*

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q. & Rush, A. M. (2020), 'HuggingFace's Transformers: State-of-the-art Natural Language Processing', *arXiv:1910.03771 [cs]* . arXiv: 1910.03771.
**URL:** *http://arxiv.org/abs/1910.03771*

Yang, Y. (1999), 'An Evaluation of Statistical Approaches to Text Categorization', *Information Retrieval* **1**(1), 69–90.
**URL:** *https://doi.org/10.1023/A:1009982220290*

Zhang, Z., Liu, J. & Razavian, N. (2020), 'BERT-XML: Large Scale Automated ICD Coding Using BERT Pretraining', *arXiv:2006.03685 [cs, stat]* . arXiv: 2006.03685.
**URL:** *http://arxiv.org/abs/2006.03685*

Zhou, L., Cheng, C., Ou, D. & Huang, H. (2020), 'Construction of a semi-automatic ICD-10 coding system', *BMC Medical Informatics and Decision Making* **20**(1), 67.
**URL:** *https://doi.org/10.1186/s12911-020-1085-4*