CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

De-identification of Swedish medical chat messages with transformers

Master's thesis in Data Science and AI

David Arvidsson, William Gerle

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2022

Master's thesis 2022

De-identification of Swedish medical chat messages with transformers

David Arvidsson, William Gerle

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2022

De-identification of Swedish medical chat messages with transformers

David Arvidsson, William Gerle

Supervisor: Dana Dannélls, Department of Swedish, multilingualism, language technology
Advisor: Marcus Olivecrona, Visiba Care Sweden AB
Examiner: Marina Axelson-Fisk, Department of Applied Mathematics and Statistics

Master's Thesis 2022
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2022

De-identification of Swedish medical chat messages with transformers

David Arvidsson, William Gerle
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg

Abstract

Healthcare in Sweden is becoming more digital and even though new technology could enable improved healthcare it also presents risks. In this thesis, which is conducted together with Visiba Care Sweden AB, data security and privacy risks are of special interest. Visiba Care offers a virtual care platform, where it is possible for patients and healthcare professionals to chat. If chat messages could be de-identified, they could be stored and used to improve healthcare for their patients.

The de-identification topic is widely studied within machine learning, however the research on Swedish medical corpora is limited, specifically when considering text corpora which consist of chat messages. Using KB-BERT for named entity recognition (NER), this thesis investigated if it was possible to reach equal performance on Swedish medical chat messages as the current state-of-the-art NER model reaches on Swedish electronic patient records. Furthermore, the thesis investigated the importance of training data size within this domain and also if a KB-BERT NER model trained on rule-based annotated data could reach higher performance than the rules it had been trained on.

Data was collected from two of Visiba Cares customers. The annotation process followed strict annotation rules, where firstly a rule-based script annotated the data before a manual review was conducted. KB-BERT was accessed through the open source library Hugging Face and the hyperparameters were tuned using random search to optimize performance. Furthermore, the decision threshold was tuned to improve recall since this metric was considered to be more important than precision in the given domain.

The results showed that it was possible to exceed current state-of-the-art performance and also that using one class for all entities led to further performance increase. Regarding training data size, the results showed that not only size is important, but also the format of the entities. Lastly, we failed to create a KB-BERT model trained on rule-based annotated data which reached higher performance than the rules it had been trained on. A potential explanation to this could be that the rule-based script did not produce annotations of high enough quality.

Keywords: BERT, Named Entity Recognition, de-identification.

Acknowledgements

We would like extend our sincere gratitude to our academic supervisor Dana Dannélls for your guidance and support. Your expertise and engagement has been invaluable for this master thesis. We would also like to thank Visiba Care for providing the resources and opportunity to conduct this master thesis. In particular, we would like to thank our supervisor at Visiba, Marcus Olivecrona. Through this master thesis you have challenged and encouraged us through every step in the process, and we have learned extensively by working together with you.

We would also like to extend our gratitude to Hercules Dalianis for supporting us in exploring the field and previous research.

William Gerle & David Arvidsson, Gothenburg, June 2022

Contents

List of Figures

List of Tables

1
Introduction

The healthcare in Sweden is becoming more digital, which could be crucial in order to handle the aging population and to meet citizens increasing expectations on the healthcare system such as faster and better treatment. The large amounts of data generated with today's technology have the possibility to save lives and reduce suffering by facilitating development of the healthcare system. Whilst new technology could enable improved healthcare it also presents risks [Blix and Levay, 2018]. In this thesis, which is conducted together with Visiba Care Sweden AB, data security and privacy risks are of special interest.

Visiba Care is a mission-driven company aiming to empower healthcare by increasing patient accessibility. They develop and offer a platform for online consultations, where patients and healthcare professionals can communicate. One part of Visiba Care's healthcare platform includes a chat between patients and caregivers [VisibaCare, nd]. The messages from these chats have the potential to improve the healthcare for Visiba Care's patients in several ways. However, due to privacy regulations, the possibility to store this data is limited.

This introductory section intends to define the problem, purpose and limitations as well as to present ethical considerations for the thesis. These topics are considered to be vital in order to give the basic understanding needed to grasp the thesis for the following sections.

1.1 Problem

The chat messages between patients and caregivers in the applications Visiba Care provide contains personal health information (PHI). Therefore, even though they can improve healthcare, currently the information can not be stored due to privacy regulations. However, de-identifying these messages could potentially solve this problem and allow for data storage. In order to de-identify data, clearly defined rules regarding what is considered personal data would be beneficial. Unfortunately, such rules do not exists in Sweden, which complicated the de-identifying process. The de-identification topic is widely studied within machine learning, however the research on Swedish corpora is limited. Specifically when considering text corpora which consist of Swedish chat messages between patients and various caregivers, which is the studied type of corpora in this thesis. Therefore, as healthcare becomes

more digital and online chats more extensively used, addressing the challenge of de-identification on these texts could be significant for gaining new knowledge within this domain, which is the aim of this thesis.

1.2 Purpose

The machine learning model used in this thesis is KB-BERT. The task of identifying personal health information is referred to as named entity recognition (NER). KB-BERT and NER will be described in Section 2. The thesis aims to address the following questions:

- Can a KB-BERT NER model be developed to reach equal performance on Swedish medical chat messages as the current state-of-the-art NER model reach on Swedish electronic patient records?
- How important is the size of training data when fine-tuning KB-BERT for de-identification on medical chat data?
- Can a KB-BERT NER model trained on rule-based annotated data reach higher performance than the rules it has been trained on?

1.3 Limitations

A limitation for this project is the lack of access to data. There is a thorough approval process required to get authorization to use data from Visiba Cares customers, which meant that only data from two customers was used. This put a limitation on the amount as well as variety of the data available. Furthermore, in order to train a model to perform well, the data needs to be annotated. Annotating data and quality checking annotations is time consuming, and therefore the time available for annotation put a limit on the amount of annotated data for the project.

1.4 Delimitations

According to Integritetsskydssmyndigheten (2021), all information which can directly or indirectly be associated with a living person is considered to be PHI. This definition includes a wide array of different types of information, and for this project a delimitation had to be made on what exact information that were considered to be PHI. Therefore, a deliberate decision was made to focus the following types of information: Name (person), location, email, ID number, phone number, age, date and organisation.

1.5 Ethical considerations

The data used for training and testing in this thesis contains PHI. GDPR aims to protect individuals right to protection of such data. The protection of individuals with regard to the processing of PHI is a fundamental right according to GDPR. The

regulation aims to contribute to the upholding of the right to privacy for the inhabitants in the European union [Integritetsmyndigheten, 2022]. As GDPR shows, this thesis involves data that needs to be handled in a correct manner to avoid breach of privacy.

No model will achieve perfect accuracy, therefore, if the model developed for this thesis gets deployed, it will sometimes fail to de-identify entities, which could lead to PHI being exposed to the detriment of those individuals. Moreover, individuals with personal information that rarely or never occur in the training data for the model could be more exposed to these errors. Such an implication, that certain names, addresses etc, would be more susceptible to error would also pose an ethical implication.

2

Theory

This chapter will present the theory related to this thesis in order to give a understanding of the current concepts. The chapter explains the basics of neural networks and the transformer architecture more specifically, followed by NER, F1-score and privacy regulations.

2.1 Neural networks

Neural networks are machine learning algorithms which model biological neurons. A neural network consists of layers of nodes. More specifically, there is an input layer followed by one or multiple hidden layers which are lastly followed by an output layer. The nodes in neighbouring layers are connected by edges. An example of an neural network is shown in Figure 2.1.

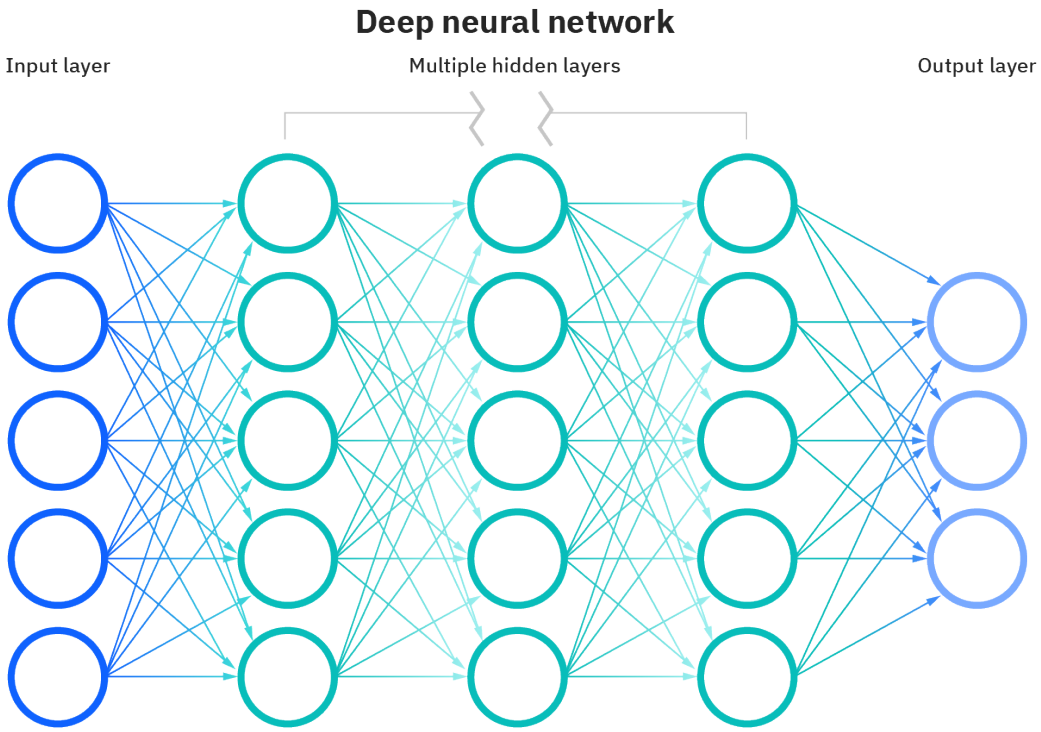

Figure 2.1: Example of an neural network [IBM-Cloud-Education, 2020].

The architecture of a node consists of weights, bias and an activation function as shown in Figure 2.2. Firstly, the input vector is multiplied by the associated weights before being summed. Secondly, a bias is added before the activation function is applied to generate the output. The activation function is an important component in a neural network and controls the information propagation through different layers. The activation function is set by the practitioner and depends on the task at hand. Rectified Linear Units (ReLU) is one of the most common activation functions, it is a partly linear function that will output the input directly if it is positive and output zero if the input is negative. For many neural networks, it is the default activation function because it makes the model easy to train and achieves good performance [Athaiya and Sharma, 2020]. Another activation function, common for the output layer of multilabel classification, such as for instance NER, is the Softmax activation function. The softmax function returns a predicted probability for all possible labels, where the sum of all predictions sum to one[Athaiya and Sharma, 2020]. The neural network starts out with randomized weights and biases for the nodes which are then adjusted when training the network. Training is conducted by processing data through the network and updating the parameters using backpropagation. The neural network has a loss function which computes the error between the actual output and the predicted output. Neural networks also have an optimization function which is used to minimize the errors of the network. Successive adjustments according to this process will make the network more accurate at predicting correct outputs [IBM-Cloud-Education, 2020].

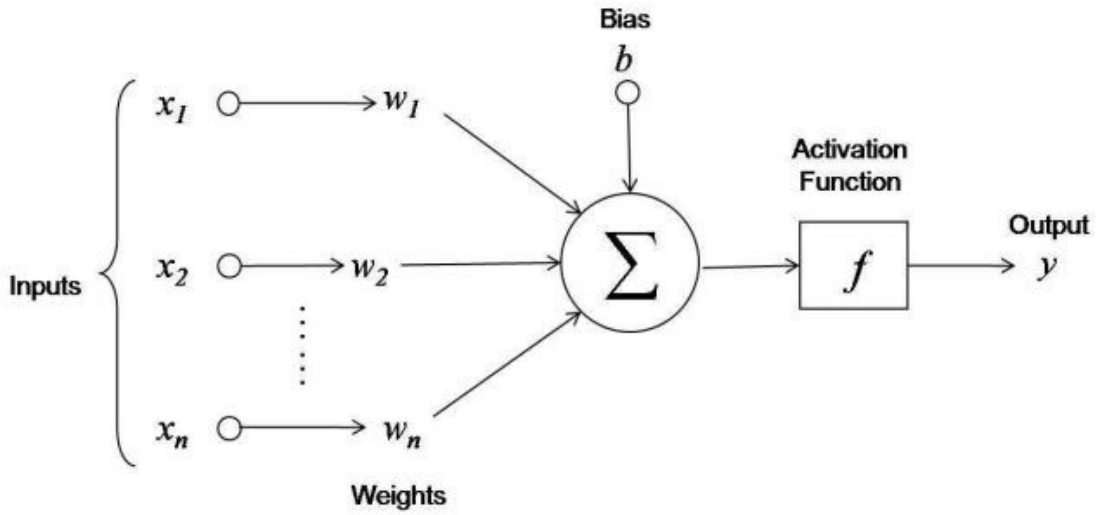

Figure 2.2: Architecture and operations of a neural network node with an input vector, associated weights, bias and output vector [Arnx, 2019]

.

When training a neural network, hyperparameters can be optimized to increase performance, for example learning rates, dropout rates, batch sizes, epochs and warm-up steps. Hyperparameters are set by the practitioner and are not optimized by the model during training. Learning rate is defined as by how much the weights

are updated during training [Goodfellow et al., 2016]. Dropout is a method that randomly "drops" a set amount of neurons from the neural network during training to avoid overfitting [Srivastava et al., 2014]. The batch size is defined as the number of data points that will be propagated in each iteration of training the model. Epochs is defined as the number of times that the model will propagate through the entire dataset [Brownlee, 2019]. Warm-up steps is defined as having a low learning rate for a set number of training steps before applying the defined learning rate [Afrooze, 2018]. Alternating these hyperparameters can have significant affect on performance, and when implementing a model tuning hyperparameters should not be overlooked [Bergstra and Bengio, 2012].

There are also several different approaches to training a machine learning model such as a neural network, among these are supervised and unsupervised learning. Supervised learning is based on training a model with annotated data with correct labels. For example training a model for a classification task where the training data is already annotated with the correct classes for each data point. Therefore, in order to train a model in a supervised fashion, access to annotated data is required. On the contrary, unsupervised training means to train the model to identify hidden patterns in unlabelled input data. Unsupervised learning therefore refers to the ability of the model to learn and organize information without providing an error signal to evaluate the potential solution[Sathya and Abraham, 2013].

2.1.1 Transfer learning

According to Ruder et al. (2019), the traditional supervised machine learning paradigm is based on learning in isolation, training a single predictive model on a specific dataset for each isolated task. Supervised machine learning models require a large number of labeled training examples and achieves the highest performance for well-defined and narrow tasks. Transfer learning is a set of methods that extend the traditional approach by leveraging data from additional domains or tasks to train a model with improved generalization properties. Since 2017, several transfer learning methods and architectures have emerged within Natural Language Processing (NLP) which outperformed previous state-of-the-art models on a wide range of NLP tasks [Ruder et al., 2019]. Currently, the most promising area within transfer learning is sequential transfer learning, where tasks are learned in sequence. Sequential transfer learning consist of two stages, a pretraining phase and a fine-tuning phase. During the pretraining phase, general representations are learned on a source task or domain and during the fine-tuning phase this learned knowledge is applied to a target task or domain, such as for example NER [Ruder et al., 2019]. When fine tuning models, the pretrained models weights are used as a starting point. There are three main fine-tuning techniques, namely to train the entire architecture, to train some layers while freezing others and finally to freeze the entire architecture. When training the entire architecture, the entire pretrained model is trained on the new dataset with a new layer as output layer. With this approach, training back-propagates through the entire architecture and the pretrained weights of the model are updated. To train some layers while freezing others means that some

of initial layers are kept frozen, meaning that training back-propagates only on the higher layers of the architecture, where the specific amount of layers to train can be determined. When freezing the entire architecture, all layers and weights are kept constant, and additional layers are added on top of the previous architecture [Ruder et al., 2019].

2.1.2 Attention

The attention mechanism is a part of a neural network architecture. The general idea of attention is to learn a weight between each input item and each output item, where more relevant elements receive higher values. For example, the input items could be a sentence in one language and the output items could be the sentence translated into another. In NLP these items are typically a sequence of words. This enables the network to dynamically sort relevant elements of the input data as shown in Figure 2.3.

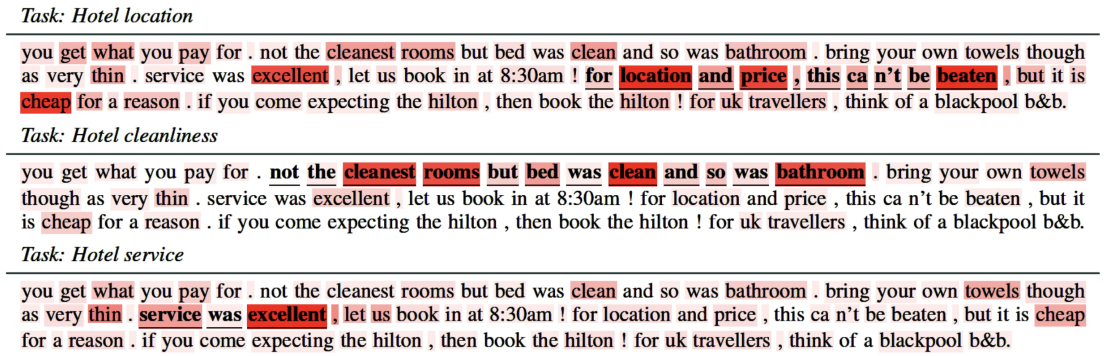

Figure 2.3: Example of attention to determine sentiment for a hotel review. The words are highlighted according to their attention scores and the bold phrases are ones considered relevant [Galassi et al., 2021].

Self-attention is similar, however, instead of looking at the relationship between items in the input sequence and the output sequence, the mechanism looks at the relationship between each item in the input sequence and every other item in the input sequence. Thereby, attention transforms the default embeddings by analyzing the whole sequence of items, which makes the values more representative of the item they represent in the context of the sentence [Galassi et al., 2021]. In the example shown in Figure 2.4, a Transformer model has identified two words, animal and street, that the word *it* could possibly refer to given that particular sentence, and gave these two words more attention based on the context [Uszkoreit, 2017].

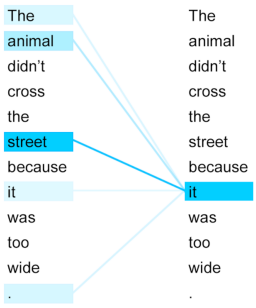

Figure 2.4: Example of self-attention distribution for the word *it* from a Transformer model [Uszkoreit, 2017].

2.1.3 Transformers

The Transformer is a neural network architecture based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. The entire architecture is shown in Figure 2.5, whose different parts will be described in this section. Firstly, the architecture consists of positional encodings, which can vary between models. The positional encoding is used at token level to retrieve positional information and is added to the embeddings which have the same dimension as the positional encoding. Thereafter, the Transformer has an encoder-decoder structure with stacked self-attention and point-wise, fully connected layers as shown in Figure 2.5.

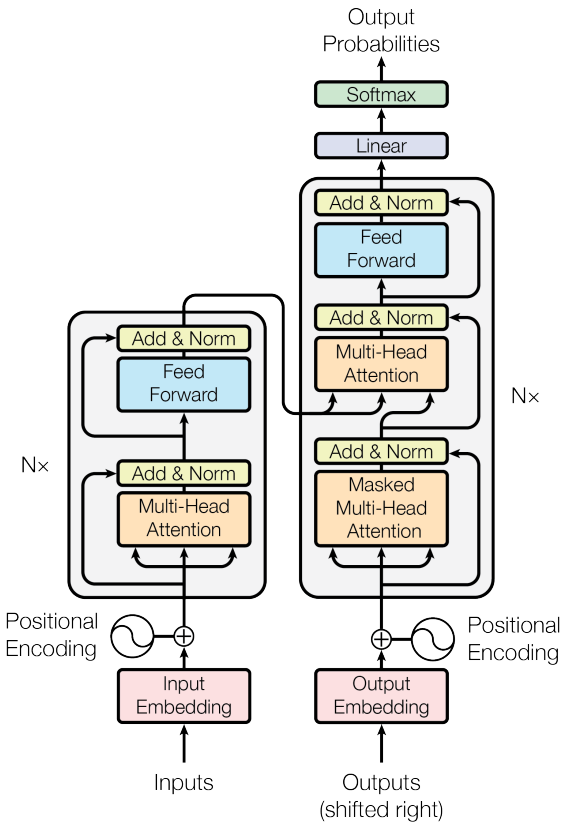

Figure 2.5: Model architecture of the Transformer.

At each time step, the encoder maps an input sequence of symbol representations to a sequence of continuous representations and then the decoder generates an output sequence of symbols. Inside the encoder and decoder there are multi-head attention functions as shown in Figure 2.6.

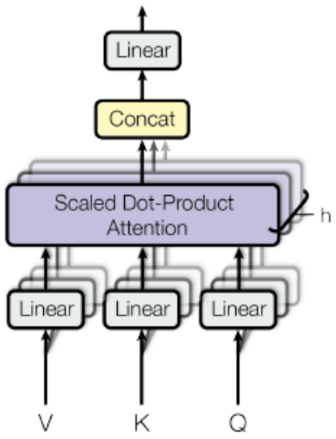

Figure 2.6: Multi-Head Attention function which internally uses the scaled dot-product self attention.

By dividing the input and using several attention heads in a multi-head attention function, where each attention head is applied on a respective input-subset, it is possible for the model to simultaneously attend to information from different representation subspaces at different positions of the sentence. The attention mechanism used in the multi-head attention functions is self-attention where the output is scaled down, i.e. scaled dot-product attention. In the decoder, there is a masked multi-head attention function, which is similar to the multi-head attention function. However, unlike the multi-head attention function where attention is calculated for the entire sequence at once, the masked multi-head attention function works by only applying attention on tokens up to the current position to be predicted. In other words, while decoding, in the masked multi-head attention function, latter tokens are masked while predicting the current token in order to prevent latter tokens to be part of the attention. The independent attention outputs of each head are concatenated before entering the feed-forward layer. Each feed-forward layer consists of two linear transformations with a ReLU activation applied in between. In the Transformer architecture there are also several Add & Norm layers. This layer sums the outputs from the previous layer and normalizes the result [Vaswani et al., 2017].

2.1.3.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a technique based on the Transformer architecture described in Section 2.1.3. Shown in Figure 2.7 are the two separate steps for developing the BERT, which are pretraining and fine-tuning. In Figure 2.7 and 2.8, E represents input embedding, T_{i} represents the contextual representation of token i, [CLS] is a symbol used for classification output, and [SEP] is a symbol used to separate non-consecutive token sequences.

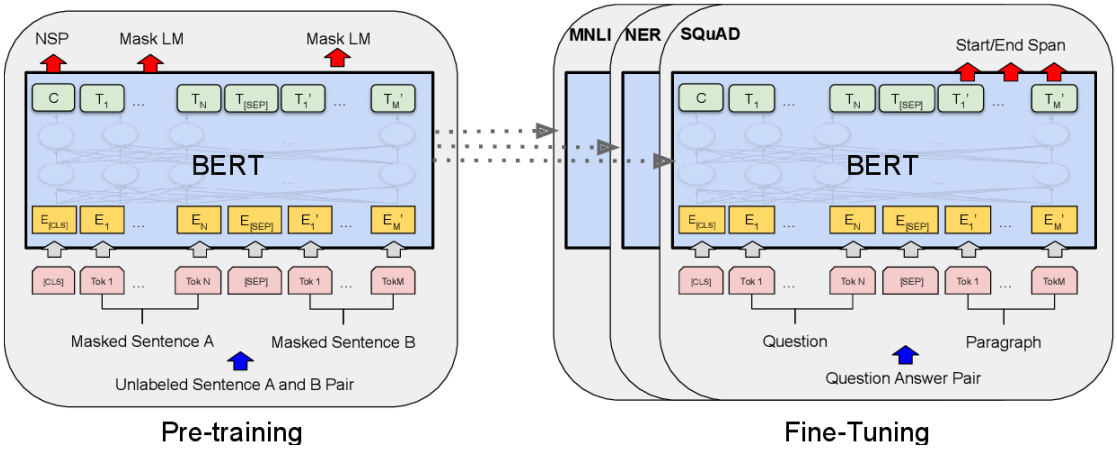

Figure 2.7: Pretraining and fine-tuning procedures for BERT.

The pretraining of BERT is done using two unsupervised tasks which are Masked Language Modelling (MLM) and Next Sentence Prediction (NSP). MLM involves masking some percentage of the input at random, and then predicting those masked tokens. By doing this task, the model learns relationships between words. The model is trained on NSP simply by predicting for a pair of sentences if the former is followed by the latter. Therefore, it is important to train on a document-level corpus rather than a shuffled sentence-level corpus. By doing NSP the model learns to understand longer-term dependencies across sentences. Fine-tuning is straightforward since the Transformer architecture allows BERT to model several downstream tasks. When fine-tuning BERT, the output layer is changed and all parameters are fine-tuned. The same pretrained model parameters can be used to fine-tune models for different tasks. For each specific task, the output layer is changed and the task-specific inputs and outputs are plugged into BERT and then the parameters are simply fine-tuned end to end. Figure 2.8 illustrates the fine-tuning of BERT on four different Natural Language Understanding (NLU) tasks. Task (a) and (b) are sequence-level tasks, and (c) and (d) are token-level tasks [Devlin et al., 2018].

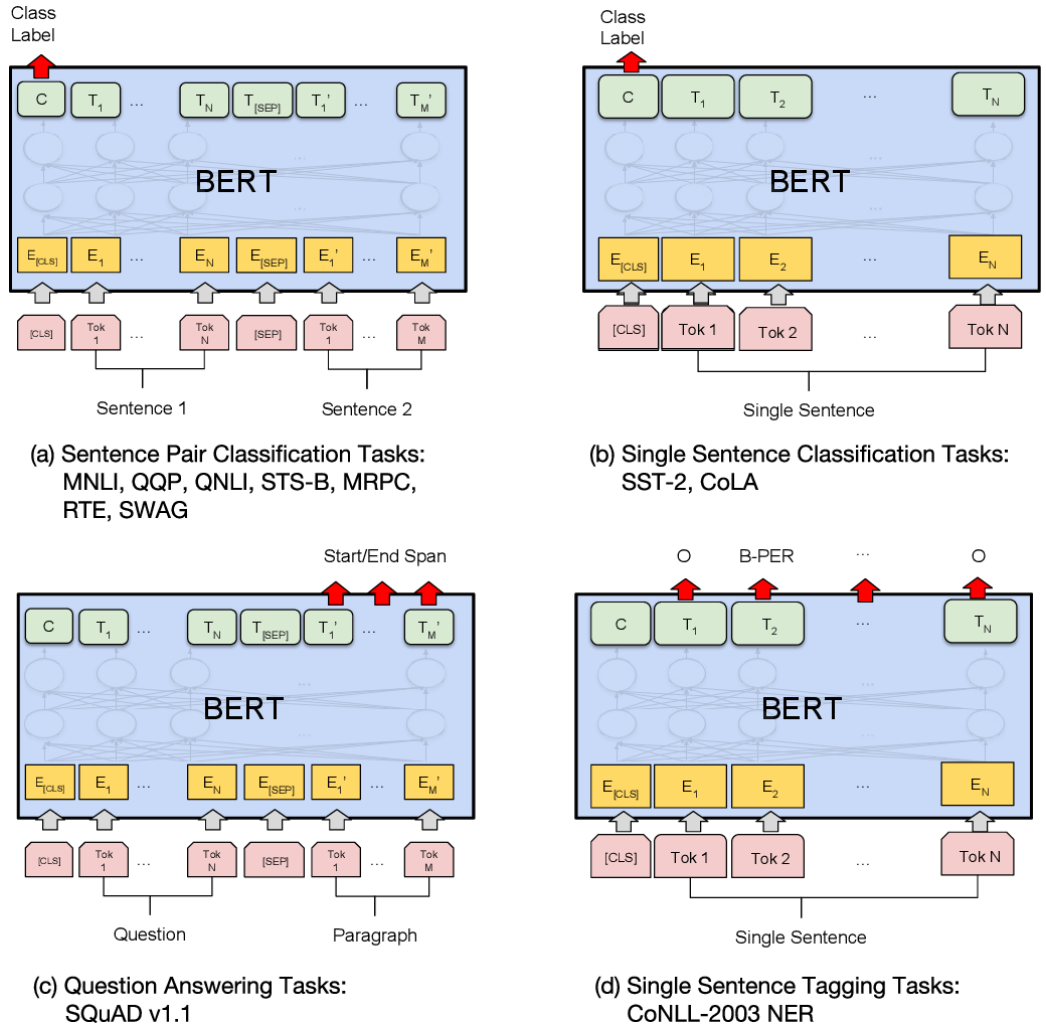

Figure 2.8: Procedures of Fine-tuning BERT on different NLU tasks.

2.2 Named Entity Recognition

Named entity recognition (NER) is the task of finding as well as categorizing entities from text, for example person, organization or place. It is widely used for solving various NLP problems and useful for mining important information from text. Accurate NER tagging is crucial in areas such as question answering, summarization systems and information retrieval systems[Mansouri et al., 2008]. An example output of a sentence being processed by a NER system developed to identify locations and persons is shown in Figure 2.9.

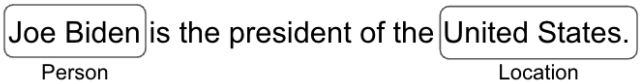

Figure 2.9: Output example of sentence being processed by a NER system.

According to Gudivada (2018), the three major approaches to NER are lexicon-based, rule-based and machine learning models. Lexicon-based approaches utilize lexicons constructed from external knowledge sources to match chunks of the text with entity names. Rule-based approaches consist of rules which have been developed manually or automatically and use them for entity recognition. Machine learning approaches consist of supervised models which require large annotated training data in order to understand the entities and their inherent characteristics. Most modern systems are based on machine learning, but some NER systems combine the three approaches [Gudivada, 2018].

2.3 F1-score

In order to evaluate the performance of different NER models, the metric F1-score is widely used. F1-score considers both precision and recall, where precision represents the proportion of items that the system returns which are actually correct. Therefore, it premiums careful selection and gives a low score to systems that return too many positives. In order to achieve high precision, the system should discard anything that it might be uncertain of [Derczynski, 2016]. Precision is defined as:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

On the contrary, recall indicates the proportion of the true entities found by the system. Therefore, this metric premiums comprehensiveness and in order to achieve high recall, it is better to include entities that the system is uncertain about [Derczynski, 2016]. Recall is defined as:

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

The achieved scores for both precision and recall are between 0 and 1, where 1 is the best possible score for both metrics. On their own, neither of these metrics tells the whole story. It is possible for a system to obtain perfect precision, i.e. by classifying all entities as true, but at the cost of obtaining a poor score for recall. F1-score takes both of these metrics into account by calculating the harmonic mean of the two metrics[Derczynski, 2016]. F1-score is calculated by:

$$F1score = \frac{2 * Precision * Recall}{Precision + Recall}$$

As for precision and recall, the score obtained is between 0 and 1 where 1 constitutes a perfect F1-score.

2.4 De-identification & Privacy regulations

De-identification is the process of removing identifying information from data so that information cannot be linked with specific individuals. De-identification can reduce the risk for breach of privacy of individuals which is associated with collecting, processing, archiving or publishing information. Thus, de-identification balances the contradictory goals of using and potentially sharing personal information while protecting privacy. Many different kinds of data can be de-identified, including structured information, free format text, multimedia and medical imagery. De-identification creates a new dataset with no personal data, which can be internally used by an organization instead of the original dataset to decrease the risk of breach of privacy [Garfinkel, 2015]. Examples of personal data is name, address, phone number and social security number [Integritetsmyndigheten, 2021].

All organizations within the European Union that handle personal data must comply with the General Data Protection Regulation (GDPR). Training an NLP-model on patient health data, for example chat messages, includes extensive processing of personal data and therefore needs to comply with GDPR. According to GDPR, personal data is data which refers to an identified or identifiable physical person. A physical person is identifiable if it can be identified directly or indirectly. Accordingly, data which can not be linked to an identified or identifiable person is not considered to be personal data. Therefore, data which is anonymized in such manner in which it can not longer be linked to physical persons is no longer personal data and is therefore not covered by GDPR [Techlaw, 2021].

In order to accurately anonymize personal data, it requires that a physical person can no longer be identified with the help of the data and the means of all aids which can reasonably be used. In practice, one of the biggest challenges with de-identifying data is that there is no absolute criteria in which the data can be said to be de-identified or not. Therefore, data protection authorities have developed a risk-based approach for anonymization which aims to set the risk of a potential re-identification to an acceptable level. The level of risk which is considered to be acceptable is decided in each individual case, as well as in the light of the most recent technical development. Because of this, the probability and risk of anonymized data to become re-identified increases over time due to technological developments. One approach towards anonymization is de-identification [Techlaw, 2021].

Patientdatalagen (PDL)[SFS 2008:355] also regulate personal data and limits what means personal data may be processed towards within Swedish healthcare. PDL aims to ensure patient safety and to safeguard the integrity and privacy of the patients. According to PDL, personal data may be processed with several different aims, one of which is to develop and ensure the quality in the current operation [SFS 2008:355, 2021]. In order to comply with GDPR and PDL, removal of all personal data is required.

3
Related work

This chapter will present previous conducted related research, both conducted on Swedish corpora as well as non-Swedish corpora.

3.1 Swedish corpora

Previous research has been conducted to explore the importance of domain specific data when training a machine learning model for de-identification of medical data in Swedish. In 2020, Chomutare et al. showed that training a model both on domain specific text and general text could enhance performance for de-identification of medical data. In their experiment, they first trained a RNN model on the Stockholm EPR PHI Pseuedo Corpus which resulted in an F1 score of 0.656 on de-identification. The Stockholm EPR PHI Pseudo Corpus contains pseudonymised electronic patient records (EPR) from clinical units from Karolinska University Hospital during the years 2006-2014 [Dalianis et al., 2015]. In total, the corpora contains 6 217 PHI entities. When training the model on a combination of the Stockholm EPR PHI Psuedo Corpus as well as general medical scientific text the F1 score increased to 0.775. Furthermore, when training the model on the Stockholm EPR PHI Psuedo Corpus and general Wikipedia text, the F1 score increased even more to 0.861. This result suggests that non-sensitive resources from the general domain can be useful for de-identification tasks on medical data [Budrionis et al., 2020].

In 2020, the National Library of Sweden (KBLab) developed and released a pre-trained language models based on BERT, named KB-BERT. The model has been trained on text from multiple resources, i.e. books, news, governmental publications and Swedish Wikipedia with the aim to provide a representative Transformer model for Swedish text. Altogether the training data constitute approximately 15-20 GB of text, corresponding to roughly 3 497M tokens. To pre-train KB-BERT, the same method and hyperparameters as proposed in the original BERT paper were used [Devlin et al., 2018]. All checkpoints are publicly available to download and utilize through Huggingface [Malmsten et al., 2020]. In order to evaluate the model and its relative performance, Malmsten et al. used NER- and POS-tagging as downstream tasks for KB-BERT. For this master thesis, the result obtained for NER-tagging is of special interest and will therefore be further explained. In order to evaluate KB-BERT for NER-tagging, the model was fine-tuned on the Stockholm-Umeå-Corpus 3.0(SUC 3.0), where the corpus were split into a training, test and evaluation set.

With 70% used for training, 20% for test and 10% for evaluation. SUC 3.0 is a collection of Swedish texts and consists of texts from press, scientific writing and prose from the 1990s. SUC 3.0 consists of a total of 1M tokens, all tokens have been tagged with word class, morphological analysis including lemma, as well as some additional structural and functional information. The results obtained showed that KB-BERT outperformed previous existing BERT's either trained for multilingual understanding by Google as well as a model trained specifically for Swedish by Arbetsförmedlingen. The model achieved an average F1 score of 0.927 compared to the second best model M-BERT of 0.906 on the SUC 3.0 corpus, where all models were fined-tuned to identify person name, location, time, organization and events. The high accuracy obtained by Malmsten et al. for NER-tagging shows usability for KB-BERT as a base model for de-identification[Malmsten et al., 2020].

In May 2021, Dalianis and Grancharova fine-tuned KB-BERT on Swedish electronic patient records for NER. The data used was the Stockholm EPR PHI Corpus as well as the Stockholm EPR PHI Pseudo Corpus, where PHIs have been replaced by surrogates. The weighted average F1-score and precision for the original model were both 0.922. The weighted average F1-score and precision for the Pseudo model was 0.882 and 0.883. The recall of the models trained on these different corpora for respective PHI class is shown in Table 3.1. The test set constituted of 20% of the Stockholm EPR PHI Corpus. To avoid confusion, the PHI class PERSON in this thesis could be compared to the classes *First Name* and *Last Name* by Grancharova and Dalianis (2021). The same applies the PHI class DATE in this thesis and *Full Date* and *Date Part*. *Health Care Unit* have been included in ORGANISATION.

PHI class	**Recall Original EPR**	**Recall Pseudo EPR**
First Name	0.939	0.913
Last Name	0.953	0.920
Phone Number	0.905	0.810
Age	1.000	1.000
Full Date	0.952	0.940
Date Part	0.985	0.970
Health Care Unit	0.874	0.803
Location	0.790	0.684
Organisation	0.500	0.600
WEIGHTED AVERAGE	0.922	0.882

Table 3.1: Recall for all the PHI-classes on the test set for the model fine-tuned on the original set and the model fine-tuned on the pseudonymised set [Grancharova and Dalianis, 2021].

According to the authors, no other released model performed as well on this corpus at the time the article was released. The authors also concluded that there is potential in using the model fine-tuned on pseudonymised data in a shareable de-identification system for Swedish clinical text since this model performed relatively well [Grancharova and Dalianis, 2021].

In September 2021, Lamproudis et al. showed that continuous pretraining of a generic BERT model with in-domain data could enhance performance for downstream tasks. By continuous pretraining of KB-BERT on in domain data within the clinical field, namely on 17.8 GB of clincal text from the research infrastrucutre Health Bank - Swedish Health Record Research Bank, Lamproudis et al. developed what they named as *Clinical KB-BERT*. Clinical KB-BERT were then fine-tuned to conduct three downstream tasks, including NER. For NER, Lamproudis et al. fine-tuned Clinical KB-BERT on The Stockholm EPR PHI Corpus, and later also fine-tuned the regular KB-BERT on the same dataset to conduct the same task. The results obtained show that Clinical KB-BERT outperformed KB-BERT on all three downstream tasks, and for the NER task it increased the F1-score from 0.910 to 0.925 which shows that a performance increase can be gained by in-domain continuous pretraining of generic BERT models. Furthermore, Clinical KB-BERT outperformed the original KB-BERT on all three downstream tasks after using only 20% of the in-domain data, corresponding to roughly 3.5 GB of raw text, which indicates that adaption of general language models into domain-specific may be worthwhile even in the absence of large amounts of in-domain data [Lamproudis et al., 2021]. To the best of our knowledge, there exist no previous work on KB-BERT model's performance for de-identification of medical chat data.

3.2 Non-Swedish corpora

In 2016, Dernoncourt et al. showcased a new architecture for de-identification. The presented architecture is based on a combination of recurrent neural networks (RNN) and Conditional Random Fields (CRF). More specifically, the type of RNN used was a bidirectional Long Short-Term Memory (LSTM). This system outperformed the best systems available at that point in time. It achieved an F1-score of 0.979 on the i2b2 2014 dataset[i2b2, 2022], and an F1-score of 0.992 on the MIMIC dataset [Johnson et al., 2021]. Both datasets contain medical data where PHI has been annotated. More specifically, the i2b2 dataset contains circa 1 million tokens and the MIMIC dataset circa 3 million tokens [Dernoncourt et al., 2017].

In 2020 W.Johson et.al. achieved state-of-the-art performance on de-identification of English medical data with an approach based on bidirectional transformer models. More specifically, several models based on the BERT architecture were implemented. The models utilized transfer learning and were initialized with pretrained weights from different BERT models. The pretrained models were fine-tuned by training on publicly available corpora which comprised collections of free-text notes written during routine clinical practice, namely the i2b2 2006, i2b2 2014, PhysioNet and Dernoncourt-Lee corpora. One model initialized with BERT_{base} weights achieved state of the art performance and demonstrated comparable accuracy on the i2b2 2014 corpus as the RNN model released by Dernoncourt et al. described above. Furthermore, the results obtained by W.Johson et al. showcased that the application of pretrained models built using scientific or biomedical corpora did not improve performance. By this, it appears that improved contextual representation does not

translate to improved performance in the de-identification of medical journals, possibly because entities such as names, addresses and personal identification numbers frequently appear in common language as well [Johnson et al., 2020].

In 2020, Boros et al. evaluated the effect of training data size when fine-tuning the BERT$_{multilingual-uncased}$ model. BERT $_{multilingual-uncased}$ is a pretrained model on the 102 languages with the largest Wikipedia base[HuggingFace, 2022]. The downstream task was multilingual text classification in the context of the epidemiological field. Based on their experiments, there is generally a positive trend for F1 score performance when trained on increasingly large datasets. The training set in the experiment held approximately 3,3M tokens and what they found was that model performance plateaued at roughly 30% of the training set. Furthermore, training the model on only 5% of the training data achieved an F1 score of 0.640 which were considered to be a significant performance for a minimal amount of data [Mutuvi et al., 2020].

4
Methods

This chapter presents how the aim of the thesis will be reached. The chapter have been divided into four separate sections, namely data collection, data annotation, model implementation and evaluation.

4.1 Name entities

Through discussions with our supervisor at Visiba Care and in order to enable comparison to other NER models, the entities that were chosen to be tagged for this task were: personal identification number, phone number, location, organisation, date, age and email. In this section, all entities will be described further in two separate subsections since some entities have a gold standard while some do not.

4.1.1 Entities followed by gold standard

In 2020, the Common Language Resources and Technology Infrastructure (CLARIN) released a gold standard for annotating named entities in Swedish text.The gold standard includes guidelines for how tokens should be tagged. The covered entities include Persons(PRS), Locations(LOC) and Organisations (ORG)[L. Ahrenberg, 2020].

4.1.1.1 Person

Persons include both real and fictional people, as well as gods and mythical characters. Animals and other creatures are not included [L. Ahrenberg, 2020]. According to the gold standard the following tokens are considered to be Persons:

- Proper names referring to a person, by itself or part of a longer sequence. For example: *David*, *William Gerle*.
- Plural references: *Johanssons*, *familjen Svensson* (the Svensson family).
- Titles and epithets preceding a proper name, or other attributes that classifies the referent should all be included. *apotekare Lundin* (pharmacist Lundin), *medborgare Vreeswijk* (resident Vreeswijk), *mamma Annika* (mother Annika).
- Initals and prepositions are included as part of a name: *John A Ericson*, *Björn af Kleen*.
- Nick-names are treated as proper names. For example, *Olle "Bagarn" Larsson* and *Olle Larsson, called "Bagarn"* should both be tagged as persons.
- Names listed as part of a group that have accomplished something together, should be marked as one instance. For example, *Hansson & Karlsson* as one instance.

The following tokens should not be tagged as Persons:

- References based on family roles. *mamma, pappa, bror* (mother, father, brother).
- Common references based on an attribute of a person, for example *älskling, lillen* (love, the little one) unless it is clearly established as a nick-name.
- Prepositions preceeding name reference, for example *"till"* (to) in *till William* (to William).

4.1.1.2 Location

Locations include geographical locations of all sorts, real or fictional. Big and small geographical locations are tagged, including continents, countries, regions, cities, villages, areas, parks, streets, mountains, rivers, etc [L. Ahrenberg, 2020]. The following tokens should be marked as Locations:

- Proper nouns referring to a location. For example, *Göteborg, Slottskogen, Kungsgatan 24, Europa*
- In cases when a proper name is preceded by an article or possessive pronoun, it should also be marked. For example, *mitt Göteborg* (my Gothenburg).
- Common nouns which refer to locations that have developed the character of a standard, namelike reference. For example *Gamla stan, Norrlands inland*.

The following tokens should not be tagged as Locations:

- Locations which are part of a name expression for another category, for example *Frölunda* in *Frölunda Hockey Club*.
- References using adverbs or common nouns such as *hemma* (home), *utlandet* (abroad).
- Prepositions preceding a location name.
- Common names of rooms, *köket, vardagsummet* (the kitchen, the livingroom).
- Common websites such as Instagram are often linguistically treated as locations, for example *Jag var på Instagram hela dagen* (I were on Instagram all day) but should not be marked as locations.
- URLs.

4.1.1.3 Organisation

Organisations include companies, government, political parties, NGOs, public bodies, sport clubs, schools, healthcare units as well as anything with a legal status in society [L. Ahrenberg, 2020]. According to the gold standard the following tokens are considered to be Organisations:

- Proper nouns and acronyms referring to entities of organisations. For example: *Telenor*, *Volvo*.
- Common nouns that have been established as names. For example: *Moderaterna*, *Svenska akademin* (the Swedish academy).
- Common nouns or abbreviations for societal institutions. For example: *vårdcentralen* (the health care center), *Riksdagen* (the parliment).
- If a name and an abbreviation are followed by each other they should both be marked, but as separate references. For example: *Frölunda Hockey Club (FHC)*.

- If a proper noun to an organisation is preceded or followed by a common noun, all attributes should be a part of the name. For example: *Apoteket Stjärnan* (the Pharmacy the Star), *Östras sjukhus* (East hospital).
- If a company is named as a URL. For example: *flygresor.se*

The following tokens should not be tagged as Organisations:

- If a location is not clearly part of the name for the organisation the tokens should be marked separately: For example *Maxi i Göteborg* (Maxi in Gothenburg).
- Prepositions preceding the name of an organisation should not be marked.
- URLs should not be considered organisations if it is not the name of the company.
- Projects are not considered organisations.

4.1.2 Entities without a gold standard

In this section, the annotation rules for the entities without a gold standard, email, age, phone number, date and personal identification number will be described.

4.1.2.1 Email

Email addresses include all tokens which have the following format *text@text.text* since this format is considered to be the standard for emails.

4.1.2.2 Age

Age include all tokens which refer to the age of a person as described in Section refer to Section 4.1.1.1. According to the standard chosen for this task the following tokens are considered to be Age:

- Tokens that are followed by and refer to a numerical age token. For example: *55-årig* (55 year old), *55 år* (55 years).
- Tokens that refer to age, either as a number or as letters. For example: *55 år* (55 years), *femtiofem år* (55 years).

The following tokens should not be tagged as Age:

- Tokens that are refer to the age of an organisation. For example: *Sjukhuset är 25 år gammalt* (the hospital is 25 years old).
- Tokens that refer to the age of a product. For example:*Medicinen är 2 veckor gammal* (the medicine is 2 weeks old).
- Tokens that refer to the age of a location. For example:*Torget är 2 år gammalt* (the square is 2 years old).

4.1.2.3 Phone number

Phone number includes all tokens which follow the structure described by The Swedish Post and Telecom Authority [The-Swedish-Post-&-Telecom-Authority, 2021]. According to this structure, an international phone number contains a international prefix, a country code, a national destination code and a subscriber number. The phone number is allowed to be a maximum of 15 numbers excluding the international

prefix. A national phone number contains a national prefix, national destination code and a subscriber number. The national prefix in Sweden is a zero. National phone number can be either geographical or non-geographical. Sweden is divided into 264 different destination code areas which are two or three number numbers long. The subscriber number for geographical phone numbers could be between five and eight numbers long. For mobile phone numbers the subscriber number is always seven numbers long.

4.1.2.4 Date

All tokens which describe dates, either in text or digits will be tagged as date. This includes both part of dates and whole dates. The following tokens will be tagged as date:

- Tokens which describe a full date or part of date in numerical format. For example: The full date *2022-03-11*, or a partial date such as *03/11*.
- Tokens that refer to dates in a text format, or combined text and numerical. For example: *22 mars* (22 of march), *tredje mars* (the third of march).
- Week numbers and months. For example: *December*, Week *25*.

4.1.2.5 Personal identification number

All Swedish citizens have a personal identification number(PID) issued by Skatteverket. The personal identification code is a ten digit code and is constructed by three different parts. The first part is six digits which indicates a person's birth time, for example *640823* for a person born on the 23 of August in 1964. The second part is a three digit birth number, where the third digit is odd if a person's legal gender is male and even if the legal gender is female. Between the birth time and birth number is a hyphen, which switches to a plus sign the year a person turns 100 years old. Lastly, a control number which is calculated by the first nine digits and can be used to control if a 10 digit number is a valid personal identification number [Skatteverket, 2021]. All tokens which follow the format outlined by Skatteverket will be tagged as PID. Furthermore, tokens which describe a PID with twelve tokens, where the birth year is stated with four digits instead of two, will also be considered to be PID.

4.2 Data collection

The corpora was collected through Visiba Care and two of their customers. The data consisted of chat messages between patients and various caregivers. As a consequence of the data being collected from chats of only two customer, the variety of entities are not considered to be as wide as if data from other customers were available as well. The total number of tokens in the corpus from one of the customers is 1 606 426. Hereafter, when referring to this corpus, the name Corpus A will be used. The total number of tokens in the other corpus is 14 277 and this corpus will only be used for testing. Hereafter, this corpus will be referred to as Corpus B.

An example of a sentence in corpus A is shown in Figure 4.5. For comparison, Figure 4.1 shows an example of a pseudonymised electronic patient record from The Stockholm EPR PHI Pseudo Corpus, which has been used in previous research on Swedish NER, as mentioned in Section 3.1. As the figures shows, the language in the chat data is more unstructured compared to data from electronic patient records.

Planeringsansvarig: SSK Tjänstgörande
Patientansvarig läkare: <First_Name>Mohamed</First_Name>
<Last_Name>Åström</Last_Name>
Kontaktorsak: Ramlat i hemmet <Full_Date>10/5-2006</Full_Date> och krampat
<Date_Part>12/5</Date_Part>.
Hade inte ätit eller druckit på 4 dygn.
Hälsohistoria: vårderf. Se läkare anteckningar.
Närstående: Dotter <First_Name>Jessica</First_Name><Last_Name>Fredriksson</Last_
Name> tel: <Phone_Number>0715-463920</Phone_Number>,
tel hem <Phone_Number>92 35 45</Phone_Number> <Last_Name>Fredriksson</Last_Name>
tel. <Phone_Number>0392-857461</Phone_Number>
Social bakgrund: Bor på gruppboende, <Health_Care_Unit>Lärkan</Health_Care_Unit>
på <Location>Ladugårdsgärdet</Location>.

Figure 4.1: Example of a pseudonymised EPR [Grancharova and Dalianis, 2021]

4.3 Data annotation

Supervised machine learning models require annotated data in order to learn patterns and make accurate predictions. Annotating data manually is time consuming and costly since machine learning model need large amounts of training data [Schreiner et al., 2006]. During the data annotation the rules described in Section 4.1 were followed. The annotation process was divided into two parts, which were automatic annotation and manual annotation, both will be described in this section.

4.3.1 Automatic annotation

In order to speed up the annotation process we wrote a rule-based annotation script in Python. The script was constituted by rules which tagged words if they met certain criteria. The criteria for each specific PHI class are described in the following subsections.

4.3.1.1 Person

Statistikmyndigheten (SCB) have published lists with the 100 most common first names for women and men in Sweden, as well as the 100 most common last names [Statistikmyndigheten, 2021]. From SCB these lists were downloaded as comma separated values files (CSV), and if a token in the text was present in these lists it was tagged as *Person*.

4.3.1.2 Location

SCB provides lists with all postal codes as well as their respective city. We considered a token to be a Location if it was present in the list, either as a postal code or as a city. However, the list with cities included tokens such as "Vara" and "Bara", which are tokens that could refer to Swedish cities but also have other meaning. Therefore, such tokens were excluded from the list. Furthermore, postal codes are often divided into two tokens, therefore if two succeeding words formed a postal code both tokens were tagged as Location. Tokens which contained "vägen" were also tagged as Location, however if the word contained *bättrings* (improvement) we did not tag it. This is because the corpora included several instances of the word *bättringsvägen* (the road to improvement) on different forms, which is not a location. Lastly, tokens containing *gatan* (street) with a length above six characters were tagged as Locations. The reason for tagging words including *vägen* and *gatan* was that these suffixes were considered to be common in Swedish street names.

4.3.1.3 Organisation

Since all the chat data came from one of Visiba Cares customers, all tokens which contained the organisation name of this customer were tagged as Organisation.

4.3.1.4 Email

The most common format for emails were considered to be text@text.text. Furthermore, the corpora did not contain any tokens which included "@" which were not an email, therefore all tokens which included the character "@" were tagged as Email.

4.3.1.5 Age

All tokens which contained *-årig* (year old) were tagged as age, as there were several instances of ages written in the chat data on this format. Furthermore, if a token contained *år* (year) and the previous token were a digit, and the second to previous token were *är* (is) both the numeric token as well as the token which contained *år* were tagged as age.

4.3.1.6 Phone number

The most common formats to type Swedish phone numbers were considered to be on 07N-NNN NN NN and +467N-NNN NN NN, where N is replaced by arbitrary digits. Therefore, in our code we combined sequences of tokens representing numbers with the following tokens. If the resulting token included a number with ten or twelve digits and was not already tagged as a PID, all three tokens were tagged as Phone number.

4.3.1.7 Date

We tagged all months and all tokens on the format DD-MM-YYYY, YYYY-MM-DD or DD/MM-YY and which were already not tagged as any other PHI as date.

4.3.1.8 Personal identification number

To check if a ten digit token is a valid Swedish personal identification number, the Luhn algorithm was used. The Luhn algorithm controls if the tenth digit in the Swedish personal identification number is correct by calculations on the first nine digits [Regeringskansliet, 2008]. The algorithm consists of three steps. The first step is to multiply each digit with either 2 or 1, where the first digit is multiplied by 2, the second with 1, the third with 2, the fourth with 1 and so on for all nine digits[Bankgirot, 2016]. Figure 4.2 below shows the first step for the first nine digits of the test Swedish personal identification number 170101-239 provided by Skatteverket [Skatteverket, 2017].

	1	7	0	1	0	1	2	3	9
*	2	1	2	1	2	1	2	1	2
	2	7	0	1	0	1	4	3	18

Figure 4.2: Example of step 1 of the Luhn algorithm.

The second step is then to add the resulting numbers, where numbers above 10 is split into two separate digits, for example the number 18 is split into 1 and 8 [Bankgirot, 2016]. Figure 4.3 shows the second step for the same example test personal identification number as Figure 4.2 above.

$$2 + 7 + 0 + 1 + 0 + 1 + 4 + 3 + 1 + 8 = 27$$

Figure 4.3: Example of step 2 of the Luhn algorithm.

The final step is to subtract the closest number dividable by ten from the resulting number [Bankgirot, 2016]. In the example, 30 is subtracted from 27 which leads to the resulting digit of 3. Therefore, the correct tenth digit of the test personal identification number is 3.

We added the Luhn algorithm to the Python script and if a token fulfilled the algorithm's criteria, the token was considered to be a PID. The algorithm was implemented to find social security numbers written in all possible formats, by only checking for digits in the token. For example, the tokens "201701012393", "170101-2393" and "Personnummer:170101-2393" would all be tagged as PID.

4.3.1.9 Automatic annotation results

The frequency of each PHI class tagged from the rule-based script in Corpus A, without any manual review, is shown in Table 4.1.

PHI Class	Uniqueness	Frequency
PERSON	1.6%	16 028
ORGANISATION	0.1%	10 008
PID	95.3%	402
PHONE	29.2%	2 380
LOCATION	6.2%	3 086
DATE	11.1%	998
AGE	32.2%	115
EMAIL	97.1%	35
Total entities	5.2%	33 052

Table 4.1: The frequency of all PHI classes within Corpus A after automatic annotation.

4.3.2 Manual annotation

Since the automatic annotation was not considered to find and tag all entities as described in Section 4.1, manual annotation was also needed to review the annotations and complete the process. However, since manual annotation is a time consuming process, the decision was made to only use a ten percent share of Corpus A. Hereafter, this share of the corpus will be referred to as Corpus AS. Corpus AS contained 20 000 sentences and 179 332 tokens. Essentially, we conducted the manual annotation by going through all sentences and correcting the errors of the automatic annotation and tagging the entities the automatic annotation had missed. In conclusion, Corpus AS is a subset of Corpus A, where Corpus AS have undergone an additional manual annotation process which Corpus A has not. In order to avoid confusion, the three corpora used in the thesis are shown in Figure 4.4.

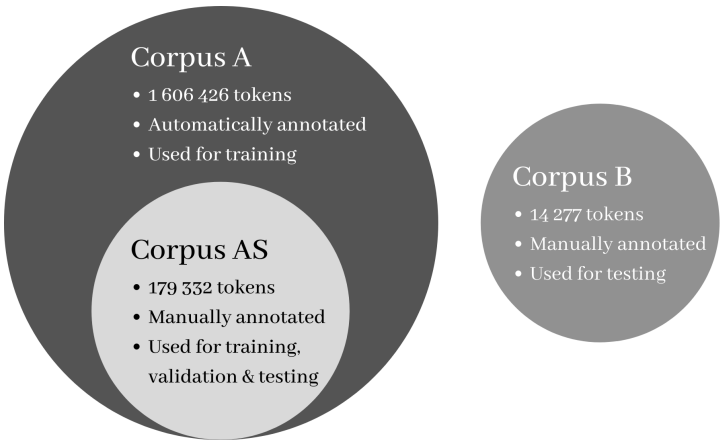

Figure 4.4: Visualisation of the different corpora used in the thesis.

In the following sections we describe the annotation tool, the method used for manual annotation as well as annotation results.

4.3.2.1 Annotation interface

To speed up this process, we developed a program using FastAPI and Uvicorn. FastAPI is a web framework for building APIs with Python and Uvicorn is a web server implementation for Python [FastAPI, nd] [Uvicorn, nd]. The user interface (UI) of the program is shown in Figure 4.5.

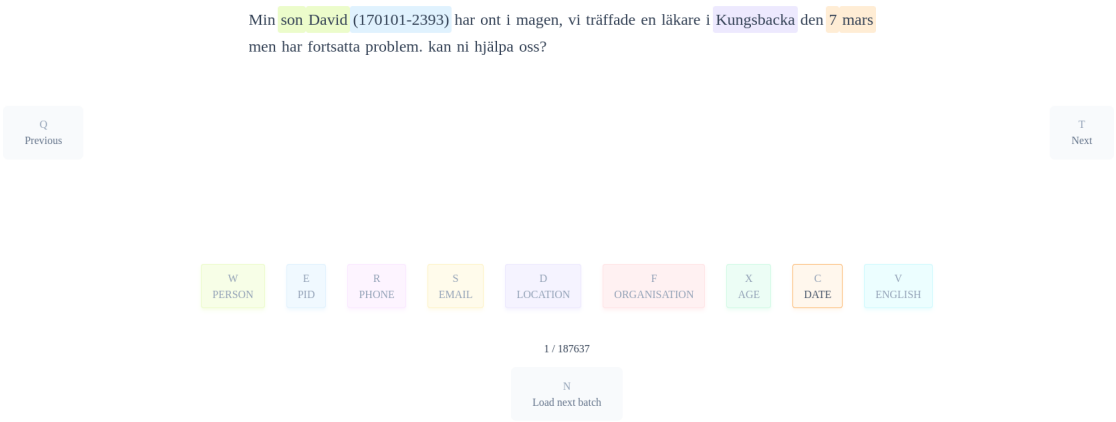

Figure 4.5: The developed user interface with an annotated example sentence. The sentence is fictional, but is characteristic of the type of sentences which were part of the actual corpus.

As shown in Figure 4.5, one sentence is shown at a time with each token colored according to its current PHI class. Below the sentence, the different tags are shown on buttons with their respective color. To tag a word, firstly the correct tag is chosen, followed by choosing the word to be tagged. There are also two buttons on each side of the sentence to switch between sentences in the current batch of sentences. To load a new batch of sentences, there is a button at the bottom of the UI. Every button on the UI is connected to a key on the keyboard, which allows for faster annotation.

4.3.2.2 Inter-annotator agreement

Considering that annotating entities is not entirely an objective task, we decided to ensure compliance regarding how to interpret the annotation rules in practice. This was done by firstly, separately annotating a set of 500 identical sentences from corpus A and secondly comparing the annotations using Cohens kappa. The formula for Cohens kappa is shown in Equation 4.1, where p_0 is the observed agreement and p_e is the probability of an agreement by chance [Artstein, 2017].

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \tag{4.1}$$

The observed agreement p_0 is simply the number of tokens in agreement divided by the total number of tokens. The probability of agreement by chance p_e, is the hypothetical probability of chance agreement. Equation 4.2 presents the equation for calculation of p_e, where k is the number of different labels, N is the number of data points and n_{ki} the number of times annotator i labeled category k.

$$p_e = \frac{1}{N^2} \cdot \sum_{n=i}^{I} n_{ki} \tag{4.2}$$

The 500 sentences, in total 4876 tokens, contained 255 entities, evenly spread between the categories as shown in Table 4.2.

PHI Class	**Frequency**
PERSON	134
ORGANISATION	77
PID	4
PHONE	5
LOCATION	10
DATE	19
AGE	6
EMAIL	5
Total entities	255

Table 4.2: The frequency of all PHI classes within the set used to check annotator agreement.

The kappa score ranges from 1 to +1, where 1 represents perfect agreement. Values lower or equal to zero indicate no agreement, values between 0.01-0.20 slight agreement, values between 0.41-0.61 moderate agreement, 0.61-0.80 substantial agreement, and 0.81 almost perfect [McHugh, 2012]. On the test set of 500 sentences we achieved a score of 0.82. The few tokens that were annotated differently were discussed, after which the remaining part of the corpus was annotated since the results showed sufficient inter-annotator agreement.

4.3.2.3 Manual annotation results

The entire manual annotation-process took 15 hours and was conducted by taking turns annotating. The frequency of entities after Corpus AS and Corpus B had been manually reviewed is shown in Table 4.3 and Table 4.4 respectively. Corpus AS were divided into a training, validation and test set where 80% of entities were included in the training set, 10 in the validation set and 10% in the test set. The test set contained approximately 10% of tokens for each tag, which enabled testing model performance for each tag separately as well as combined performance for each tag. In order to ensure that all sentences in the test were accurately tagged, we manually revised each word and tag in the test set and corrected all errors together. Since annotation agreement was set, the errors were mostly words that had been missed when annotating. For example, *FK*, short for *Försäkringskassan* (the Swedish Social Insurance Agency), had been overlooked.

PHI Class	Uniqueness	Corpus AS	Train	Validation	Test
PERSON	20.6%	7030	5434	816	780
ORGANISATION	6.7%	4064	3063	494	507
PID	95.5%	400	315	42	43
PHONE	60.9%	294	212	47	35
LOCATION	45.9%	294	209	42	43
DATE	79.3%	276	210	33	33
AGE	47.4%	78	55	10	13
EMAIL	97.1%	35	25	6	4
Total entities	21.7%	12 471	9523	1490	1458

Table 4.3: The frequency of all PHI classes after Corpus AS has been manually annotated.

PHI Class	Uniqueness	Corpus B
PERSON	47.4%	694
PHONE	87.0%	23
AGE	44.4%	9
LOCATION	41.2%	68
DATE	64.4%	250
ORGANISATION	43.0%	79
PID	96.8%	31
EMAIL	66.7%	3
Total entities	52.0%	1157

Table 4.4: The frequency of all PHI classes after Corpus B has been manually annotated.

4.4 Model implementation

This section will describe how the model used for the thesis was implemented as well as how its parameters were tuned in order to optimize its performance.

4.4.1 Model

The transformer architecture for this de-identification task is available through the Huggingface library.[1] The Huggingface library provides a suitable model, KB-BERT, which could be easily downloaded and implemented. In order to modify this model for NER, it was fine-tuned using the previously mentioned annotated data, which contained examples of each PHI class. KB-BERT is backed by three deep learning libraries, which are Jax, PyTorch and Tensorflow, with a seamless integration between them. It is possible to train models with one of these before loading for inference with another [HuggingFace, nd]. As a basis for this task, PyTorch was used. KB-BERT was accessed through NERDA which is a python package that offers an

[1] https://huggingface.co/KB/bert-base-swedish-cased (Accessed 2022-03-30)

interface for fine-tuning pretrained transformers for NER [Kjeldgaard et al., nd].

4.4.2 Hyperparameter tuning

To optimize the performance of the model, the hyperparameters were tuned using a random search method. Random search is considered to be time efficient and to achieve good performance, in comparison to other common methods such as grid search and manual search [Bergstra and Bengio, 2012]. The process of tuning hyperparameters with random search includes defining a range of values for each parameter, and then randomly choosing a specific value for each parameter from this range. The model is then trained on the chosen set of hyperparameters and it's train and validation loss recorded. This process is iterated a set amount of times, and the instance of hyperparameters which achieves the lowest validation loss is used for the final model implementation. The chosen ranges for the different hyperparameters in the model are described and motivated below.

- **Learning rate:** The range for learning rate were set to be in between 10^{-4} and 10^{-6}. As initial experiments showed that learning rates above or below this range resulted in poor performance.
- **Dropout:** In order to explore the full range of dropouts, the range for dropout was set to be in between 0 and 1.
- **Epochs:** The number of epochs were set to an integer value between 3 and 8. The lower bound were set to 3 due to an observed decrease of validation loss as epochs increased, and the upper bound were set to 8 as the number of epochs has a significant effect on training time.
- **Batch size:** The batch size were set to one of the values 8, 16 or 32. The lower bound were 8 since batch sizes of 2 and 4 were concluded to result in poor performance from initial experiments. Memory allocation set an upper bound on possible batch sizes, which caused the upper bound to 32.
- **Warmup steps:** The number of total steps in each epoch for this particular model varied between approximately 560 and 2 250 depending on the current batch size. In order to set the number of warmup steps below a full epoch, the range were set to be in between 0 and 560.

4.5 Evaluation of research questions

In order to answer the research questions of the thesis, there are mainly three experiments which have been conducted and evaluated. Fundamentally, the F1-score has been used as evaluation metric for all three experiments which are further described in this section.

4.5.1 Development of state-of-the-art model

In order to evaluate if a transformer NER model can be developed to reach equal performance on Swedish medical chat messages as the current state-of-the-art model

reaches on Swedish electronic patient records, a KB-BERT model was trained and optimized on the train and validation set from Corpus AS. This model was implemented with the set of hyperparameters which performed best on the validation set was thereafter tested on the held out test set from Corpus AS. The achieved precision, recall and F1-scores in this thesis were compared to the model developed by [Lamproudis et al., 2021] to compare the developed model's performance to the current state-of-the-art on Swedish electronic patient records. Furthermore, an experiment where all entities will be set to the same PHI class will be conducted to explore the potential effects on achieved performance of the model. By only focusing on if a token is a PHI class or not, the model avoids errors where it predicts the wrong PHI class, and will therefore potentially reach higher performance.

4.5.2 Importance of training data size

In order to evaluate the importance of the size of training data, one model was trained on the full annotated training set, one model on 75% of the available training data as well as one model on 50% of the available training data. After training, the F1 score on the held out test set was computed for all models and the obtained results compared. For this research question, both Corpus AS and Corpus B was used.

4.5.3 Rule-based trained KB-BERT versus rule-based script

Two models were evaluated on the held out test set in order to evaluate if a transformer model trained on rule-based annotated data can reach higher performance than the rules it has been trained on. Since manual annotation was not relevant to answer this research question, the entire dataset, Corpus A, was used. One of the models will simply be the rule-based script used to annotate the training data. The other model will be KB-BERT, trained on the data annotated by the rule-based script, without any manual review of the annotations.

5
Results

The following chapter will present the results for the conducted experiments, including performance of the developed model on the held out test set, performance reached for varying training sizes as well as performance of the rule-based script versus the model trained on the rule-based annotated data.

5.1 Implementation of state-of-the-art model

The following section presents the optimal hyperparameters, softmax thresholds as well as the achieved performance on the held out test set. Corpus AS has been exclusively used to retrieve all results in this section except from Section 5.1.3 where both Corpus AS and Corpus B has been used.

5.1.1 Hyperparameter results

A total of 81 iterations with a combined running time of 22 hours were conducted. The experiment showed that a learning rate of $5.44*10^{-5}$, warmup steps of 497, 4 epochs, batch size of 32, and dropout of 0.323 resulted in the lowest achieved validation loss. Figure 5.1 below presents the training loss as well as validation loss achieved when training with optimal parameters.

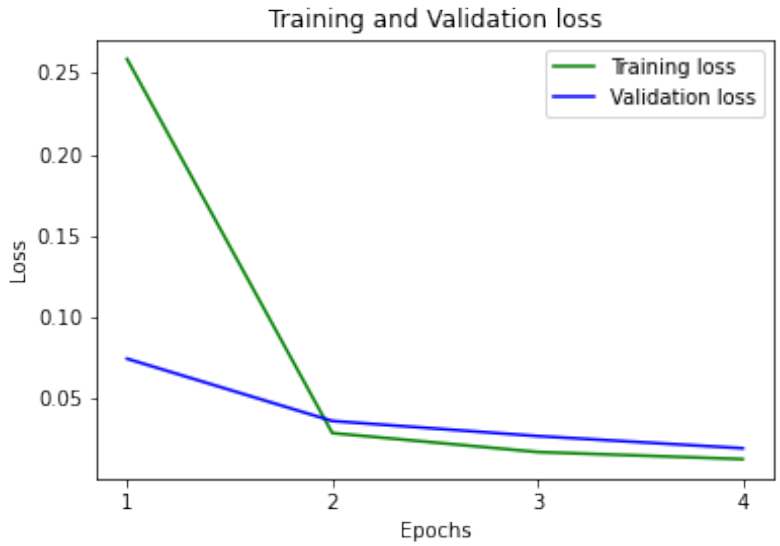

Figure 5.1: Train and validation loss for optimal hyperparameters.

Table 5.1 presents recall, precision and F1 score for each PHI class on the validation set when the model was trained with optimal parameters.

PHI class	F1-score	Precision	Recall
PERSON	0.981	0.982	0.980
ORGANISATION	0.969	0.964	0.974
PID	0.911	0.973	0.857
PHONE	0.940	0.887	1.000
LOCATION	0.930	0.909	0.952
DATE	0.912	0.886	0.939
AGE	0.833	0.714	1.000
EMAIL	0.909	1.000	0.833
AVERAGE	0.923	0.914	0.942
WEIGHTED AVERAGE	0.969	0.965	0.971

Table 5.1: F1-score, precision and recall for all PHI classes on the validation set with optimal hyperparameters.

5.1.2 Softmax threshold

The model was also modified so that it summed the likelihoods of all the PHIs, and if that sum was greater than a certain threshold, the model predicted a token to be the PHI with the greatest likelihood. The F1-score, precision and recall for different thresholds on the validation set are shown in Figure 5.2.

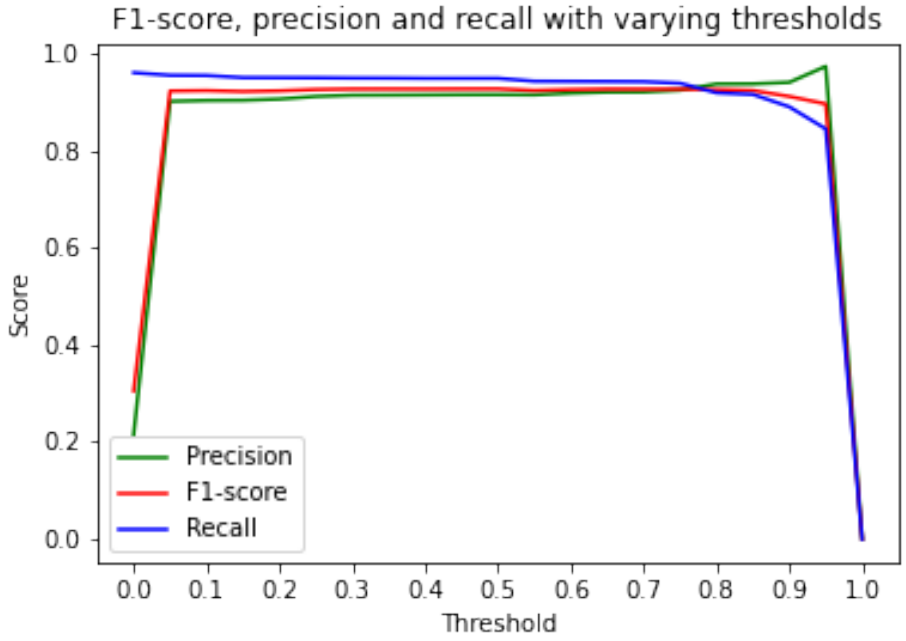

Figure 5.2: F1-score, precision and recall curves for different prediction thresholds on the validation set.

As Figure 5.2 shows, there is a trade-off between precision and recall, where recall is the largest when the threshold is low and the inverse holds for precision. The Figure also shows, the F1-Score increases quickly for thresholds between 0.0 up to around 0.1. After 0.2 the scores are on a steady level up until about 0.7, where recall starts to decrease. Figure 5.3 shows the curves where the thresholds in between 0.05 and 0.3 have been zoomed in on.

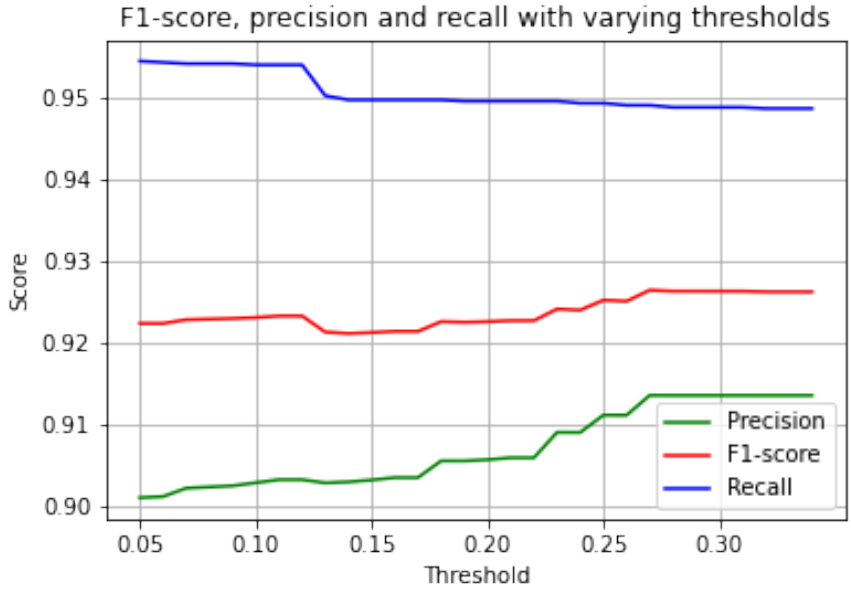

Figure 5.3: F1-score, precision and recall curves for different prediction thresholds on the validation set.

As Figure 5.3 shows, there is a drop in recall at the threshold value of 0.12. Since this value achieved a high recall without compromising the overall F1-score, it was the threshold that was chosen for the final model. The precision, recall and F1-score for this threshold value is shown in Table 5.2.

PHI class	**F1-score**	**Precision**	**Recall**
PERSON	0.978	0.970	0.987
ORGANISATION	0.969	0.953	0.986
PID	0.914	0.949	0.881
PHONE	0.913	0.839	1.000
LOCATION	0.943	0.911	0.976
DATE	0.928	0.889	0.970
AGE	0.833	0.714	1.000
EMAIL	0.909	1.000	0.833
AVERAGE	0.923	0.903	0.954
WEIGHTED AVERAGE	0.966	0.953	0.981

Table 5.2: F1-score, precision and recall for all PHI classes on the validation set, with threshold set to 0.12.

5.1.3 Result on test data

Table 6.1 presents the achieved result on the held out test set from Corpus AS as well as Corpus B after training the model on the chat messages which had been firstly annotated by the rule-based script and secondly manually reviewed. The optimal hyperparameters and a Softmax threshold of 0.12 was used to achieve the presented results.

	Test set Corpus AS			**Corpus B**		
PHI class	F1-Score	Precision	Recall	F1-Score	Precision	Recall
PERSON	0.975	0.959	0.991	0.923	0.871	0.983
ORGANISATION	0.959	0.945	0.974	0.486	0.844	0.342
PID	0.934	0.975	0.907	0.967	1.000	0.935
PHONE	0.986	0.972	1.000	0.750	0.882	0.652
LOCATION	0.864	0.844	0.884	0.566	0.711	0.471
DATE	0.901	0.842	0.970	0.956	0.952	0.960
AGE	0.963	0.929	1.000	1.000	1.000	1.000
EMAIL	1.000	1.000	1.000	0.857	0.750	1.000
AVERAGE	0.948	0.933	0.966	0.813	0.876	0.793
WEIGHTED AVERAGE	0.961	0.947	0.977	0.877	0.882	0.896

Table 5.3: Performance for all PHI classes on the held out test set from Corpus AS as well as on Corpus B.

From Table 6.1 it is clear that the model, on average, performs better on the test set from Corpus AS than on Corpus B. This is expected, as the model is trained on Corpus AS. The performance for PHI classes which have a standard similar format in both sets, i.e. EMAIL, PID, DATE, AGE is comparable and even higher for PID and DATE for Corpus B compared to Corpus AS. However, other PHI classes, which have no standardized format, for example ORGANISATION and LOCATION, sees a relatively significant decrease in performance for Corpus B compared to Corpus AS. These PHI classes have low uniqueness in Corpus AS and Corpus B, but the actual entities vary between the corpora. Therefore, the model which is trained on Corpus AS accurately identifies the locations and organisations which occur in Corpus AS, but perform less accurate when identifying locations and organisations in Corpus B. Corpus B contains several examples where an a persons name is followed by an acronym and a location, where the acronym is for an organisation. This acronym and the following location is inaccurately predicted as PERSON, which results in a relatively low precision for PERSON as well as low recalls for LOCATION and ORGANISATION. In Corpus B, PHONE entities were often written as a composition of a location and a phone number, for example *Gothenburg:0723038454*. PHONE entities never appeared on this format in the train set from Corpus AS, and therefore the model fails to identify these entities as PHONE in Corpus B.

The model was also evaluated on the test set from Corpus AS as well as Corpus B when the training data was modified by changing all the different PHI classes to the same one PHI class, here named as PHI. With this alternation, the model achieved an F1-score of 0.969, precision of 0.958 and recall of 0.981 on Corpus AS. Thus

an improvement to both precision and recall which led to an increased F1-score by 0.008. Furthermore, on Corpus B this alternation lead to a significant performance increase compared to previous results on Corpus B. The model achieved an F1-score of 0.955, a precision of 0.948 and recall och 0.961 on Corpus B. Thus an improvement to both precision and recall which led to an increased F1-score by 0.078.

5.2 Importance of training data size

Figure 5.4 presents the obtained F1-score on the held out test set with different sizes of the training data.

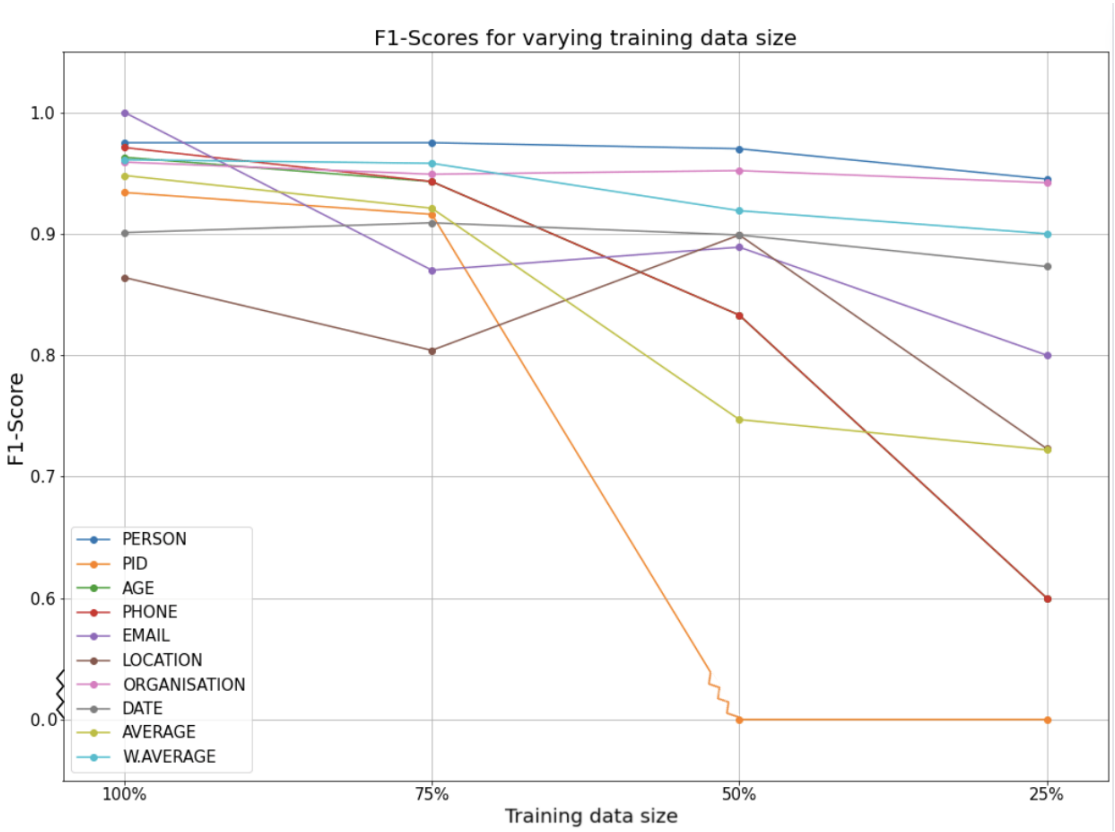

Figure 5.4: Train and validation loss for optimal hyperparameters.

The frequency of each PHI class in the training data for the each size is shown in Table 5.4. As the table shows the share of entities in the training data have been kept around the same percentage when lowering the size. The weighted average in Figure 5.4 shows that the general trend is that as training size decreases, so does the average obtained F1-Score. Furthermore, the decrease in F1-Score as training size decreases is more drastic for entities which are less common in the training data. For instance, entities AGE and EMAIL decrease from 1.00 to 0.762 and 0.750 respectively, as the size of training data decrease from a 100% to 50%. In comparison, the performance on common entities such as PERSON and ORGANISATION decrease only slightly as the size of training data decreases.

PHI class	**100%**	**75%**	**50%**	**25%**
PERSON	5434	4036	2780	1318
ORGANISATION	3063	2248	1578	741
PID	315	224	159	71
PHONE	212	164	100	48
LOCATION	209	155	111	52
DATE	210	165	100	50
AGE	55	40	24	12
PID	315	224	159	71
EMAIL	25	18	13	7
Total entities	8902	7050	4865	2299

Table 5.4: The frequency of all PHI classes for the different training data sizes.

The scores for the different training sizes are also shown in Table 5.5 and 5.6. A big drop is seen for the PHI class PID when going from 75% to 50% of the training data. At 50% training data, out of the 43 PIDs, the model predicts 40 of these tokens as PHONE, 2 as PERSON and 1 as DATE.

	Share of training data					
	100%			**75%**		
PHI class	F1-Score	Precision	Recall	F1-Score	Precision	Recall
PERSON	0.975	0.959	0.991	0.975	0.963	0.987
ORGANISATION	0.959	0.945	0.974	0.949	0.930	0.968
PID	0.934	0.975	0.907	0.916	0.950	0.884
PHONE	0.971	0.971	0.971	0.943	0.943	0.943
LOCATION	0.864	0.844	0.884	0.804	0.755	0.860
DATE	0.901	0.842	0.970	0.909	0.909	0.909
AGE	0.963	0.929	1.000	0.870	1.000	0.769
EMAIL	1.000	1.000	1.000	1.000	1.000	1.000
AVERAGE	0.948	0.933	0.966	0.921	0.931	0.915
WEIGHTED AVERAGE	0.961	0.947	0.977	0.958	0.946	0.971

Table 5.5: F1-score for all PHI classes on the held out test set when trained on 100% and 75% of the training data.

	Share of training data					
	50%			**25%**		
PHI class	F1-Score	Precision	Recall	F1-Score	Precision	Recall
PERSON	0.970	0.955	0.985	0.945	0.908	0.986
ORGANISATION	0.952	0.932	0.972	0.942	0.961	0.924
PID	0.000	0.000	0.000	0.000	0.000	0.000
PHONE	0.585	0.437	0.886	0.600	0.440	0.943
LOCATION	0.847	0.857	0.837	0.723	0.698	0.750
DATE	0.899	0.861	0.939	0.873	0.816	0.939
AGE	0.833	0.909	0.769	0.800	0.833	0.769
EMAIL	0.889	0.800	1.000	0.889	0.800	1.000
AVERAGE	0.747	0.719	0.799	0.722	0.684	0.787
WEIGHTED AVERAGE	0.919	0.901	0.941	0.900	0.880	0.927

Table 5.6: F1-score for all PHI classes on the held out test set when trained on 50% and 25% of the training data.

5.3 Rule-based trained KB-BERT versus rule-based script

To achieve the results presented in this section, the transformer model have been trained on entire Corpus A except from the tokens present in the test set from Corpus AS. The result for the two models that were evaluated on the held out test set from Corpus AS, to evaluate if a KB-BERT model trained on rule-based annotated data can reach higher performance than the rules it has been trained on is shown in Table 5.7.

	KB-BERT			**Rule-based script**		
PHI class	F1-Score	Precision	Recall	F1-Score	Precision	Recall
PERSON	0.457	0.975	0.299	0.425	0.986	0.270
ORGANISATION	0.648	1.000	0.479	0.655	1.000	0.487
PID	0.840	0.895	0.791	0.925	1.000	0.860
PHONE	0.838	0.795	0.886	0.923	1.000	0.857
LOCATION	0.489	0.468	0.512	0.519	0.470	0.556
DATE	0.326	0.700	0.212	0.341	0.875	0.212
AGE	0.267	1.000	0.154	0.555	1.000	0.385
EMAIL	0.667	0.5	1.000	1.000	1.000	1.000
AVERAGE	0.566	0.792	0.542	0.668	0.916	0.578
WEIGHTED AVERAGE	0.541	0.956	0.395	0.535	0.974	0.387

Table 5.7: F1-score for all PHI classes on the held out test set for KB-BERT trained on rule-based annotated data as well as performance for the rule-based script by itself.

As presented in Table 5.7 the KB-BERT trained on rule-based annotated data performs worse than the rule-based script itself. From these results it is clear that KB-BERT were not able to learn patterns not present in the rule-based annotation.

From the results, it is also clear that the precision obtained for both models is far better than the achieved recall, which were also expected. A notable observation is that precision is on average higher than recall.

6
Discussion

This chapter will dive deeper and discuss the results of the thesis. The aim is to analyse the results and use them as a basis to answer the research questions of the thesis.

6.1 Development of state-of-the-art model

The implemented model with optimal hyperparameters as well as a threshold value, achieved a weighted average F1-score of 0.961, with a precision of 0.947 and recall at 0.977. In comparison, the fine-tuned KB-BERT developed by Grancharova and Dalianis (2021), achieved a F1-score of 0.922, precision of 0.922 and recall of 0.922. Based on this, the results obtained in this thesis showed that it is possible to develop a model for medical chat data which achieves performance comparable, and even better, than the performance that current state-of-the-art models achieve on other medical NER-tasks.

However, it is important to note that these results are not possible to compare directly as there are several differences in the ways that the models have been implemented. One such difference is the actual PHI classes, as well as the annotation rules for each PHI class. The fine-tuned KB-BERT developed by Grancharova and Dalianis (2021) has been evaluated on The Stockholm EPR PHI Corpus, for which there are no publicly available annotating guidelines available. Therefore, we have been unable to ensure the actual rules that guided annotation of the corpus, and unable to compare them to the rules adopted for this thesis. Furthermore, both models were evaluated on different corpora, and it is possible that the model developed by Grancharova and Dalianis (2021) would have outperformed the model developed for this thesis on our hold out test set. Unfortunately, it has not been practically possible to evaluate the model developed by Grancharova and Dalianis (2021) on our held out test set. Lastly, another factor that affects the ability to compare the models performance is that the Softmax output layer has been altered in this project. In an effort to increase recall without significant decrease in the overall F1-score, an optimal threshold value for predicting an entity as a PHI class was found on the validation set. This alternation had a small effect on the F1-Score on the validation set, from a weighted average of 0.969 to 0.966. However, no comparison were made on the test set, where it is possible and plausible based on the performance on the validation set, that an ordinary Softmax output layer could have achieved an even higher F1-score on the test set, but at a cost of a lower recall.

Even though the results are not directly comparable, it is interesting to compare the recall achieved by Grancharova and Dalianis (2021) for the different PHI classes. The results have several similarities. Phone numbers and age seem to be easy to identify since it has one of the highest recall scores for both models. The strict format of phone numbers could be an explanation for that entity while it is more difficult to discuss the reason for the age class. Perhaps age is present in recognizable patterns. Additionally, both models have struggled to identify locations correctly. The reason for this could be that variety of what is considered to be a location is wide. As described in Section 4.1.1.2, locations could be anything from continents, cities and villages to parks, mountains and rivers. On the contrary, the largest difference is for organisations. A potential explanation for this could be the low uniqueness of this entity in the dataset used in this thesis, as can be seen in Table 4.3.

The uniqueness level as well as the number of training examples of each PHI class seems to have had an effect on the achieved F1-scores for our model. The two classes with the lowest uniqueness and highest number of training examples, PERSON and ORGANISATION, achieved the highest F1-scores. The likely reason for the high F1-score in both of these classes is that both of these classes have a large number of training examples with low variety, making it easy for the model to spot these entities in the hold out test data. However, other PHI classes with high uniqueness, such as PID and EMAIL, also achieved high F1-scores. Both of these PHI classes contain entities with a very specific format, where all emails contain an @ and all PIDs are a ten digit number. Therefore, the model is more likely to learn these patterns and able to accurately identify entities as either PID or EMAIL even though it has not seen those exact tokens before. Based on the results obtained, it is likely that the more unique and exact the format of a PHI class is, the less training examples are needed. For example, EMAIL only had 25 training examples and still achieved an F1-score of 0.909 compared to LOCATION, which had 209 training examples but only achieved a F1-score of 0.864.

The model developed in this thesis could also be compared to the model developed by Lamproudis et al (2021) which showed that continuous pre-training of a generic BERT model with in domain data could further enhance performance. Their model, Clinical KB-BERT, achieved an F1-score of 0.925, which still is not as high as the F1-score of the model developed in this thesis. However, the aspects regarding lacking possibilities of comparison, as discussed earlier, are also viable when comparing the model in this thesis to Clinical KB-BERT. Furthermore, since Clinical KB-BERT has not only been fine-tuned but also further pre-trained, the comparison is even more difficult to make. However, it shows that it could be interesting to conduct continuous pre-training on the model developed in this thesis to further enhance the performance. If the same percentage increase in F1-score was achieved on this thesis best performing model, when all entities were set as PHI, we would reach a close to perfect F1-score of 0.985.

Without continuous pre-traning, setting all entities as PHI increased the weighted average F1-score on the test set from 0.961 to 0.969. While this increases the F1-score it also decreases readability of the output text. However, this result shows that for NER-task similar to de-identification, an approach where all entities are tagged as the same PHI-Class can be preferable, as the crucial task of de-identification is to remove all sensitive information rather than ensuring perfect readability for the de-identified corpus. This is however, of course a trade-off between readability and performance and the effects of setting all entities as the same PHI-Class needs to be evaluated in each specific implementation.

The F1-scores for varying Softmax threshold values indicate that the implemented model is confident in its predictions. This is shown in the Figure 5.2, where the precision score increases rapidly for small threshold values and thereafter the F1-score stays at a high constant value up until a very high threshold value where recall naturally decreases. This pattern is expected, as the achieved F1-scores indicate that the model can predict most tokens accurately. A lower F1-score would indicate that the model is more uncertain, and the Softmax threshold graph would likely follow a different pattern with flatter increases and decreases of the F1-score. This is naturally a consequence from a solid model with sufficient training data of high quality.

The achieved results on Corpus B in comparison to Corpus AS, shows that evaluating the model on new data from another source, different from the source it had not been trained on decreased the weighted average F1-score significantly by 0.084. That the performance would be lower was expected. However, the model performed significantly higher when setting all entities as PHI on Corpus B, and the difference in performance between the models decreased from 0.084 to 0.014 with this alteration. This indicates that the implemented model can be deployed with high success on other corpora as well. This result also confirms that the alternation to set all entities as the same PHI class is meaningful when implementing a NER model, especially when there might be significant differences between the training set and the data which the model will get exposed to when deployed.

6.2 Importance of training data size

The general trend of decreasing F1-score when decreasing the data size used for training was expected as it has been shown in previous research, for example by Boros et. al in 2020. However, even at 25% of training data the model had a weighted average F1-score of 0.900, but only an average F1-score of 0.722. This is due to the fact that the dataset is imbalanced in regards to the frequency of the different PHI classes. At lower shares of training data, the PHI classes which had a larger share of total entities still performed similar, such as PERSON and ORGANISATION, whilst other PHI classes, such as PHONE and LOCATION, no longer had enough training samples to reach near the original scores. When creating a model for de-identification of medical data, one could argue that in order for the model to be considered for deployment, the F1-score, and the recall specifically, would need

to be close to perfect. The reason for this is that the data contains PHI, and as previously mentioned, thereby is subject to many laws and regulations. Consequently, the results from this thesis show that data size is very important in this context.

Furthermore, an interesting observation is that PID drops from a F1-score of 0.916 to 0 when going from 75% of training data to 50%. Such a drop is not present for any other PHI class. At 50% of training data, there is still 71 entities of PID left, which is the third highest amongst the entities. Therefore, to state that this result only depends on the amount of training data is not sufficient. The decrease in performance on the PID class and the simultaneous decrease in precision for the PHONE class shows that when entities have similar formats, which PHONE and PID has, the number of entities seems to be even more important to ensure when the model does not mix these. This is visible in the result where, at 50% training data, the model predicts almost all PID entities as PHONE.

6.3 Rule-based trained KB-BERT versus rule-based script

Both the rule-based script and the implemented KB-BERT model trained on rule-based annotated training data perform poorly compared to the other implemented models. For KB-BERT, it seems like the model has inherited the poor recall showed by the rule-based script, and failed to observe patterns not present in the rules. The rule-based annotated KB-BERT model performs even worse than the script on most measures, which is somewhat surprising, as we expected that a KB-BERT model trained on relatively large amounts of data annotated with high precision would be able to spot patterns and therefore achieve a significant higher recall than the rule-based script. In reality, the weighted average precision is lower and the weighted average recall is only slightly higher compared to the rule-based script. Looking at an individual PHI class such as PERSON, it was expected that KB-BERT would learn patterns and be able to spot names outside of the ones tagged by the rule-based script. However, the KB-BERT model trained on rule-based annotated data does not seem to do so as the achieved recall value only increases slightly compared to the rule-based script. This is surprising, as the number of tagged entities of the rule-based script for this PHI class is high, as well as with high precision which are the circumstances under which we expected the transformer based KB-BERT model to spot patterns and generalize the PHI class to also achieve a high recall.

Furthermore, the rule-based KB-BERT model was trained on significantly larger corpus than the other implemented models. Where Corpus A contained 1 606 426 tokens with 33 052 entities, compared to 179 332 tokens with 12 471 entities in Corpus B. However, from the achieved results it is obvious that the increase in training data size did not lead to an increased performance. It is also probable that the large number of wrongly tagged entities, i.e. the ones that the rule-based model missed, leads the model in the wrong direction.

Furthermore, during this project it has become evident that even though manual annotation is a time consuming process, so is the development of an accurate and efficient rule-based annotation script. We did not measure the exact time spent on developing the script for this project, but considerable time was spent on developing rules, searching for databases with names and locations as well as tweaking each rule to obtain maximal precision and recall. Therefore, any time savings achieved on developing an automatic annotation script were in this case marginal and can not outweigh the substantial decrease in overall performance compared to manual annotation. Considering this, to develop an optimal rule-based script does not seem to be more efficient compared to manual annotation.

	Corpus AS		**Corpus B**	
PHI Class	Uniqueness	Number	Uniqueness	Number
PERSON	20.6%	7030	47.4%	694
ORGANISATION	6.7%	4064	43.0%	79
PID	95.5%	400	96.8%	31
PHONE	60.9%	294	87.0%	23
LOCATION	45.9%	294	41.2%	68
DATE	79.3%	276	64.4%	250
AGE	47.4%	78	44.4%	9
EMAIL	97.1%	35	66.7%	3
Total entities	21.7%	12 471	52.0%	1157

Table 6.1: Performance for all PHI classes on the held out test set from Corpus AS as well as on Corpus B.

7
Conclusion

To conclude, a KB-BERT model can be developed to reach better performance on Swedish medical chat messages than current state-of-the-art NER models reach on Swedish electronic patient records. To further increase performance, changing all the different PHI classes to the same one PHI class is beneficial although it reduces readability. Furthermore, it is possible, by changing the models prediction threshold, to further increase recall and maintain similar F1-score by sacrificing precision. Moreover, although the model performed worse when testing on a corpus from a data source which was different than the corpus it had been trained on, the results show that it is possible to still reach state-of-the-art performance when setting all entities as the same PHI class.

Regarding data size, this thesis confirmed what has been seen in other research related to implementation and training of BERT-based models, namely that as training data size increases, an increase in F1-score can also be expected. The obtained results also indicate that the format of different PHI classes and how entities among PHI classes relates to each other affects performance. When dealing with PHI classes that are similar in format and in how they occur, more examples of each PHI class are necessary.

From the results in the thesis it is also possible to conclude that we were unable to implement a KB-BERT model trained on rule-based scripted annotated data which achieved better performance than the rule-based script itself. The model trained on the automatic annotated data was not able to learn patterns beyond the rules. The results of the experiments made in this thesis indicates that manually annotating a smaller amount of training data is likely a superior approach when preparing to implement and train a KB-BERT based model on Swedish medical data.

The result in this thesis have shown several findings in the area of de-identification on Swedish medical data. To further explore this domain, it could be interesting to use continuous pre-training on in-domain data for the model developed using one PHI class in this thesis, in order to investigate how close to perfection the model would reach.

Bibliography

[Afrooze, 2018] Afrooze, S. (2018). The importance of hyperparameter tuning for scaling deep learning training to multiple GPUs. https://aws.amazon.com/blogs/machine-learning/the-importance-of-hyperparameter-tuning-for-scaling-deep-learning-training-to-multiple-gpus/.

[Arnx, 2019] Arnx, A. (2019). First neural network for beginners explained (with code). https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf.

[Artstein, 2017] Artstein, R. (2017). *Inter-annotator agreement. In Handbook of linguistic annotation*. Springer.

[Athaiya and Sharma, 2020] Athaiya, A. and Sharma, S. (2020). Activation functions in neural networks. *International Journal of Engineering Applied Sciences and Technology*, 4:310–316. https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf.

[Bankgirot, 2016] Bankgirot (2016). Calculate check digits according to the modulus-10 method. https://www.bankgirot.se/globalassets/dokument/anvandarmanualer/10-modul.pdf.

[Bergstra and Bengio, 2012] Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13:281–305. http://dblp.uni-trier.de/db/journals/jmlr/jmlr13.htmlBergstraB12.

[Blix and Levay, 2018] Blix, M. and Levay, C. (2018). Digitalization and health care - a report to the swedish government's expert group on public economics. https://eso.expertgrupp.se/wp-content/uploads/2019/08/Digitalization-and-health-care-2018_6-English-version.pdf.

[Brownlee, 2019] Brownlee, J. (2019). Difference Between a Batch and an Epoch in a Neural Network. https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/.

[Budrionis et al., 2020] Budrionis, A., Chomutare, T., Dalianis, H., Makhlysheva, A., Godtliebsen, F., and Yigzaw, K. Y. (2020). De-identifying swedish ehr text using public resources in the general domain. *Stud Health Technol Inform.*, 270:148–152. https://pubmed.ncbi.nlm.nih.gov/32570364/.

[Dalianis et al., 2015] Dalianis, H., Henriksson, A., Kvist, M., Velupillai, S., and Weegar, R. (2015). Health bank - a workbench for data science applications in healthcare. *Proceedings of the CAiSE-2015 Industry Track co-located*. http://ceur-ws.org/Vol-1381/paper1.pdf.

[Derczynski, 2016] Derczynski, L. (2016). Complementarity, F-score, and NLP evaluation. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 261–266. https://aclanthology.org/L16-1040.

[Dernoncourt et al., 2017] Dernoncourt, F., Lee, J., Uzuner, O., and Szolovits, P. (2017). De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606. https://arxiv.org/abs/1606.03475.

[Devlin et al., 2018] Devlin, J., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805. http://arxiv.org/abs/1810.04805.

[FastAPI, nd] FastAPI (n.d.). https://fastapi.tiangolo.com/.

[Galassi et al., 2021] Galassi, A., Lippi, M., and Torroni, P. (2021). Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4291–4308. https://arxiv.org/abs/1902.02181.

[Garfinkel, 2015] Garfinkel, L. (2015). *De-identification of Personal Information.* U.S. Department of Commerce. https://csrc.nist.gov/publications/detail/nistir/8053/final.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning.* MIT Press.

[Grancharova and Dalianis, 2021] Grancharova, M. and Dalianis, H. (2021). Applying and sharing pre-trained BERT-models for named entity recognition and classification in Swedish electronic patient records. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 231–239. Linköping University Electronic Press. https://aclanthology.org/2021.nodalida-main.23.

[Gudivada, 2018] Gudivada, V. N. (2018). Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications. Handbook of Statistics, Chapter 9.

[HuggingFace, 2022] HuggingFace (2022). BERT-base-multilingual-uncased. https://huggingface.co/bert-base-multilingual-uncased.

[HuggingFace, nd] HuggingFace (nd). Transformers. https://huggingface.co/docs/transformers/index.

[i2b2, 2022] i2b2 (2022). NLP research data sets. https://www.i2b2.org/NLP/DataSets/.

[IBM-Cloud-Education, 2020] IBM-Cloud-Education (2020). Neural Networks. https://www.ibm.com/cloud/learn/neural-networks.

[Integritetsmyndigheten, 2021] Integritetsmyndigheten (2021). Vad är personuppgifter? https://www.imy.se/privatperson/dataskydd/introduktion-till-gdpr/vad-ar-personuppgifter/.

[Integritetsmyndigheten, 2022] Integritetsmyndigheten (2022). Personuppgifter. https://www.imy.se/verksamhet/dataskydd/det-har-galler-enligt-gdpr/introduktion-till-gdpr/personuppgifter/.

[Johnson et al., 2020] Johnson, A., Bulgarelli, L., and Pollard, T. (2020). Deidentification of free-text medical records using pre-trained bidirectional transformers. *CHIL '20: Proceedings of the ACM Conference on Health, Inference, and Learning*, page 214–221. https://doi.org/10.1145/3368555.3384455.

[Johnson et al., 2021] Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L. A., and Mark, R. (2021). MIMIC-IV, Physionet (version 1.0). https://doi.org/10.13026/s6n6-xd98.

[Kjeldgaard et al., nd] Kjeldgaard, L., Albert-Lindqvist, T., Jafari, M., and Enevoldsen, K. (n.d.). Nerda. https://github.com/ebanalyse/NERDA.

[L. Ahrenberg, 2020] L. Ahrenberg, J. Frid, L. O. (2020). A new gold standard for swedish named entity recognition. *SWE-CLARIN*. https://sweclarin.se/sites/sweclarin.se/files/SCR-01-2020.pdf.

[Lamproudis et al., 2021] Lamproudis, A., Henriksson, A., and Dalianis, H. (2021). Developing a clinical language model for swedish: Continued pretraining of generic bert with in-domain data. *In the Proceeding of RANLP 2021: Recent Advances in Natural Language Processing*, page 790–797. http://su.diva-portal.org/smash/get/diva2:1625129/FULLTEXT01.pdf.

[Malmsten et al., 2020] Malmsten, M., Börjeson, L., and Haffenden, C. (2020). Playing with words at the national library of sweden - making a swedish bert. *ArXiv*, abs/2007.01658. https://arxiv.org/abs/2007.01658.

[Mansouri et al., 2008] Mansouri, A., Mamat, A., and Affendey, L. (2008). Named entity recognition approaches. *IJCSNS International Journal of Computer Science and Network Security*, 8(2). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.123.4784.

[McHugh, 2012] McHugh, M. L. (2012). Interrater reliability: the kappa statistic. biochemia medica. *Biochem Med (Zagreb)*, 22(3):276–282. https://pubmed.ncbi.nlm.nih.gov/23092060/.

[Mutuvi et al., 2020] Mutuvi, S., Boros, E., Doucet, A., Jatowt, A., Lejeune, G., and Odeo, M. (2020). Multilingual epidemiological text classification: A comparative study. *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6172–6183. https://aclanthology.org/2020.coling-main.543/.

[Regeringskansliet, 2008] Regeringskansliet (2008). Personnummer och samordningsnummer. https://www.regeringen.se/contentassets/89ce56cd914e42658b9bceca087e5f70/personnummer-och-samordningsnummer-sou-200860.

[Ruder et al., 2019] Ruder, S., Peters, M., Swayamdipta, S., and Wolf, T. (2019). Transfer learning in natural language processing tutorial. *Proceedings of NAACL HLT 2019: Tutorial Abstracts*, page 15–18. https://aclanthology.org/N19-5004.pdf.

[Sathya and Abraham, 2013] Sathya, R. and Abraham, A. (2013). Comparison of supervised and unsupervised learning algorithms for pattern classification. *(IJARAI) International Journal of Advanced Research in Artificial Intelligence,*, 2(2):18–22. https://thesai.org/Publications/ViewPaper?Volume=2Issue=2Code=IJARAISerialNo=6.

[Schreiner et al., 2006] Schreiner, C., Torkkola, K., Gardner, M., and Zhang, K. (2006). Using machine learning techniques to reduce data annotation time. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(22):2438–2442. https://www.researchgate.net/publication/220048712_Using_Machine_Learning_Techniques_to_Reduce_Data_Annotation_Time.

[Skatteverket, 2017] Skatteverket (2017). Testpersonnummer. https://www7.skatteverket.se/portal/apier-och-oppna-data/utvecklarportalen/oppetdata/Test

[Skatteverket, 2021] Skatteverket (2021). Personnummer. https://skatteverket.se/privat/folkbokforing/personnummer.4.3810a01c150939e893f18c29.html.

[Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958. http://jmlr.org/papers/v15/srivastava14a.html.

[Statistikmyndigheten, 2021] Statistikmyndigheten (2021). Namnstatistik. https://www.scb.se/hitta-statistik/statistik-efter-amne/befolkning/amnesovergripande-statistik/namnstatistik/.

[Techlaw, 2021] Techlaw (2021). Ai sweden – utvalda juridiska aspekter rörande personuppgifter vid träning av nlp-modeller med patientjournalstexter, 2021-11-20. https://www.ai.se/sites/default/files/content/bilder/ai-sweden-nlp-artikel$_2$11120.*pdf*.

[The-Swedish-Post-&-Telecom-Authority, 2021] The-Swedish-Post-&-Telecom-Authority (2021). Telefonnummers struktur. https://www.pts.se/sv/bransch/telefoni/nummer-och-adressering/telefoninummerplanen/telefonnummers-struktur/.

[Uszkoreit, 2017] Uszkoreit, J. (2017). Transformer: A Novel Neural Network Architecture for Language Understanding. https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html.

[Uvicorn, nd] Uvicorn (n.d.). https://www.uvicorn.org/.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

[VisibaCare, nd] VisibaCare (n.d.). Om oss. https://www.visibacare.com/sv/om-oss/.