Zero-Shot Approach to Redacting Personally Identifiable Information in Medical Reports Using Large Language Models

University of Turku
Department of Computing
Master of Science (Tech) Thesis
Health Technology
June 2025
Amanda Myntti

Supervisors:
Assoc. Prof. Antti Airola
Docent Laura-Maria Peltonen

UNIVERSITY OF TURKU
Department of Computing

Amanda Myntti: Zero-Shot Approach to Redacting Personally Identifiable Information in Medical Reports Using Large Language Models

Master of Science (Tech) Thesis, 60 p.
June 2025

The presence of personally identifiable information in large data collections poses a significant barrier to their effective use. Specifically in the health care domain, a myriad of text documents filled with sensitive information – such as clinical notes and electronic health records – are created each day. While these document collections are valuable for machine learning, whether as the target of analysis or as training data for models, their usage is profoundly limited by the information they contain.

Natural Language Processing, and specifically Large Language Models, are at the forefront of current tools best suited for sensitive data redaction. In this thesis, a language model-based zero-shot approach to redacting personally identifiable information is implemented and applied to a synthetic medical corpus. The study investigates the effectiveness of this method across clinical text data in English, Finnish, and Spanish. The results show that despite clear differences in the capabilities of tested models, the zero-shot redaction method is incapable of reliably detecting personal information in electronic health records.

Keywords: Natural Language Processing, Large Language Models, Sensitive Data, Personally Identifiable Information, PII Redaction, Clinical Records, Electronic Health Records

Table of Contents

1 Introduction

Personally identifiable information, or *PII*, is any information present in the data that can be used to link the data instance to a real-life person. Nowadays, many datasets used to train machine learning models are massive and crawled from the web, and are thus filled with personal information. Especially in the medical and healthcare domain, data containing PII is created daily, which could be used to develop systems that make the healthcare system more reliable and the databases easier to use. However, the abundance of PII in these documents is of the most injurious type; leakage of personal health records can lead to harm for individuals as well as institutions. The health care system is also heavily regulated when it comes to personal information. Redacting – in other words, censoring – PII in these datasets is therefore a crucial task. These days, the size of the datasets often necessitates the use of machine learning methods to redact PII instead of manual redaction.

Natural Language Processing (NLP) is a field of study at the cross-section of computer science and linguistics. NLP uses machine learning and data-analysis techniques for problems such as text classification, text generation, semantic analysis, question answering, and many more language-related tasks [1]. The field is

rapidly evolving, and many of the most commonly used research techniques and tools have emerged in the last 10 years. NLP methods have been successfully used to redact PII [2]–[5]; however, the task remains complicated for three reasons in particular: firstly, the redaction models need to have close to perfect performance. Preferably, all PII needs to be detected and redacted, but simultaneously, the models have to redact as little as possible of the surrounding text not to make the data unusable for following downstream tasks. Secondly, PII often has a contextual nature: depending on the context, the same passage of text can or cannot be linked to a person or some other entity. Thirdly, in many cases, it is difficult to know beforehand which types or categories of PII are present in the dataset, which is oftentimes required knowledge for training a PII redaction model. One solution for the last problem is using a zero-shot approach.

In NLP and more broadly in the machine learning context, a zero-shot approach is often described as using a model to do a task it has not been specifically trained to do beforehand [6]. This can mean multiple things depending on the model architecture and the tasks it has been trained for. For example, a model trained for text generation can be used for sentiment classification by formatting the query as *input text* + "*The preceding text is negative or positive:* " and letting the generative model fill in the most likely answer. In the context of computer vision, a commonly used example is that of an image classification model: if a model is trained to identify horses and striped fabric (among other things), we may infer that an image predicted as both may contain a zebra. In the NLP context, an analogous situation could be a text classifier labelling a single piece of text as both theatre and humour, thus most likely containing a comedic play. Another approach

to zero-shot methods is the one covered in this thesis: using intermediate features of a machine learning model to infer some secondary qualities of the input. For example, language models map each input text to a vector space as a part of their prediction process, and these vector embeddings can be used to infer attributes, such as similarity through vector distance or classification by clustering.

Chat or instruction finetuned Large Language Models (*LLM*s) can also be used for PII redaction in an analogous setting [7]. With these types of models, there is no need to know the types of PII present in the data in advance: they can handle contextual PII, have great performance, and their usage is generally easy to implement. Still, these models may not be usable without an internet connection, which is crucial for PII redaction in sensitive data, and those that are available locally may be too resource-hungry to be utilised in secure data environments where PII data is stored. Although these issues are actively being tackled and resource-abundant yet sensitive data storage services are created[1], more resource-efficient solutions are still sought after in certain conditions.

In this thesis, a zero-shot approach to PII redaction using a transformer-based on a language model is presented and tested on health data in English, Finnish, and Spanish. The topic of this thesis is motivated by a PII redaction method first described by Albanese et al. [2]. This zero-shot method uses the language model's internal predicted probability distribution to evaluate which words in a given passage are the most informative, thus likely containing PII. This zero-shot method is originally evaluated in English with non-healthcare domain data. This raises the question of possible multilinguality – specifically Finnish – for this thesis.

[1] `https://csc.fi/osaamisemme/arkaluonteinen-data/` (in Finnish)

Being a synthetic language (using inflections like verb and noun conjugations to convey meaning), Finnish differs to a great extent from analytical English (relying on word order and prepositions), and in many instances, methods developed for English require some modification to work on Finnish. Similarly, health data has specific features unique to it: especially in the case of PII redaction, some PII redaction models trained to redact names might redact valuable medical information, such as diseases named after a person, like *Andersen disease*. Hence, the research questions of this thesis can be defined as

- Is it feasible to use a zero-shot redaction method in electronic health records?
- What problems may be encountered in the redaction pipeline of Albanese et al. [2] when changing the language context from analytical English to synthetic languages such as Finnish?

Experiments will also be conducted on Spanish, as the data source used in this study is originally in Spanish and is translated to English and Finnish as a part of the experiments.

The structure of this thesis is as follows: First, I define central concepts in NLP for the needs of this thesis. Secondly, definitions and key notions of PII in the context of NLP and health care are presented. Chapter 4 introduces different methods of PII detection with relevant examples in the field of NLP. Chapters 5 and 6 cover the experiments of this thesis and answer the research questions given above. Lastly, I discuss the implications of the results.

2 On Natural Language Processing

In this chapter, I briefly introduce the machine learning and, specifically, Natural Language Processing (NLP) concepts required to cover the topics in this thesis.

2.1 General concepts and vocabulary

2.1.1 Lemmatisation

Lemmatisation, somewhat interchangeably known as stemming, refers to changing a word from an inflected form to the normal form. One of the first lemmatisation algorithms was introduced by Lovins [8] in 1968. For instance, the segment "I was sleeping" can be lemmatised as ["I", "be", "sleep"]. Lemmatisation was long a standard preprocessing step for NLP tasks and, for instance, has been used in information retrieval [9]. As lemmatisation removes grammatical information and the relationships between the words, it is rarely used as a preprocessing step with modern LLMs.

2.1.2 Tokenisation

Using non-numerical data as input to a machine learning model requires converting the data into a numerical format while preserving as much of the information in the data as possible. In the case of images, a natural approach is to represent the data as a tensor (a multidimensional matrix), with each value corresponding to a pixel and its colour. Working with textual data involves mapping units of text – words, punctuation, and whitespace – to a defined set of *tokens*, which can then be associated with numerical indices. Two obvious strategies may come to mind: using words or using characters. Both approaches have their drawbacks. With word-based tokenisation, the number of required tokens is large, especially in non-analytic languages. Character-based tokenisation reduces the token set size, but results in much longer sequences for the same text, leading to unnecessary computation and limitations on the length of segments the model can process at once. It also strips away more linguistic information from the tokens, increasing the learning burden during training. As a result, various alternative tokenisation algorithms have been proposed. All algorithms presented below require training data to optimise a set of final tokens. This optimisation leads to the final set of tokens containing both characters and full words, but most importantly, partial words. These partial words are optimised to be used in multiple words, and in cases where a word is divided into multiple tokens by the tokeniser, they are referred to as *subword tokens* in this thesis.

Sennrich et al. [10] popularised the use of the Byte-Pair Encoding (*BPE*) algorithm for tokenisation in NLP, while originally it was designed for text compression. The BPE algorithm is used in many language models, such as the GPT-family [11]. The

BPE algorithm begins building tokens from the set of all characters in the training set. Then, the model learns *merges*, rules which are optimised to merge the most commonly seen combinations of tokens. This means that for a training dataset comprising words "hug", and "bug", the first merge would be to combine characters u and g to a token "ug". These merges are applied until the maximal vocabulary size, a preset value, is reached. Wu et al. [12] introduced the WordPiece tokeniser, which uses a similar method to BPE, starting with the smallest units of text, and iteratively building up a set of final tokens. Compared to BPE, WordPiece uses a different heuristic than frequency for building merge rules, leading to a different optimisation result. Although BPE and WordPiece are very similar algorithms, in practice, they differ in their implementations, specifically how they handle white space. Lastly, Kudo and Richardson [13] define the SentencePiece algorithm, sometimes referred to as Unigram tokenisation. This tokenisation method begins by forming a large vocabulary, and, conversely to BPE and WordPiece, iteratively breaks tokens into smaller pieces using probabilities of the tokens in the training set, until a final vocabulary size is reached.

2.1.3 Metrics

This section is based on two books [14], [15] and an article [16], and covers the metrics needed in the experiments of this thesis.

Metrics familiar from general machine learning are often used with NLP tasks. In classification, common metrics are precision, recall, and F1-score. The results of a binary classifier model can be divided into 4 categories: True Positive (TP), a correct classification as true, True Negative (TN), a correct classification as false,

False Positive (FP), an incorrect classification as true, and False Negative (FN), an incorrect classification as false. In multiclass classification, these categories can be defined analogously, considering a prediction to the wrong class as negative. Precision and recall can be calculated as

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN}.$$

Precision and recall are often opposing forces in classification: increasing the recall usually reduces precision and vice versa. This is known as the *precision-recall trade-off*. Due to this, it is common to report the F1-score, a metric that accounts for both precision and recall. F1-score can be calculated as

$$\text{F1} = \frac{1}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}} = 2\,\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FN + FP},$$

equaling the harmonic mean of precision and recall. The value of a good F1-score depends on the classifier: for a random, uniform binary variable, the F1-value is 0.5. However, if the label distribution is not uniform, the random baseline for the F1-value is smaller.

Other common metrics to report in a binary classification setting are the ROC-AUC score and the PR-AUC score. The ROC-curve (Receiver-operating characteristic curve) is calculated for a classifier by plotting the true positive rate as a function of false positive rate for the classifier, and AUC (area under curve) is calculated from this plot as the integral between 0 and 1. The PR-AUC metric is calculated the same way but with the PR-curve (Precision-Recall curve), by plotting the precision as a function of recall for different classification thresholds.

PR-AUC is considered a preferable metric in datasets with unbalanced label distribution [17], while ROC-AUC is invariant to the label distribution: ROC-AUC scores a completely random classifier 0.5, which can be considered a baseline for the metric, with values close to 1 indicating a perfect classifier and values close to 0 indicating a classifier so poor, it would be perfect if the labels were reversed. The baseline value of PR-AUC, similar to the F1-score, is dependent on the distribution of labels in the data, but similarly to ROC-AUC, values close to 0 indicate low performance while values close to 1 indicate perfect performance.

Finally, this thesis will lightly touch on cosine similarity [18]. In a vector space where the distance between vectors indicates similarity, the angle between two vectors can be used as a metric. A common way to map an angle to a score of similarity is to use the cosine function:

$$\text{Cosine similarity}(\bar{x}, \bar{y}) = \cos(\measuredangle \bar{x}, \bar{y}) = \frac{\bar{x} \cdot \bar{y}}{||\bar{x}|| \cdot ||\bar{y}||},$$

with $\cdot$ indicating dot product in the numerator and regular multiplication in the denominator, and $|| \cdot ||$ indicating the Euclidean norm. If the vectors $\bar{x}$ and $\bar{y}$ are normalised to unit length, the formula simplifies to $\bar{x} \cdot \bar{y}$. Cosine similarity is a well-established metric used in the field of NLP, but has faced critique on the actual capabilities of measuring similarity [18], [19].

2.2 Transformer architecture and language models

Language models are machine learning models trained on language, such as text or speech, to perform language-related tasks. In recent years, they have played an

integral role in advancing the field of NLP. One of the key developments has been the substantial increase in model size, meaning the number of trainable parameters, which has led to striking performance improvements and even emergent abilities. This growth in scale is reflected in the term large language models (LLMs) [20], [21]. The transformer architecture, the basis of almost all LLMs, was introduced by Vasvani et al. [22]. This architecture is based almost solely on *attention*, compared to older neural architectures for textual or general sequential data [23], [24]. The attention mechanism is a feature in the model architecture which allows the model to direct more focus on some parts of the input by assigning varying levels of importance to each part. In the transformer architecture, multiple concurrent attention modules, *heads*, are used to focus on different aspects of the input, which is known as multi-head attention. This attention is also applied to the input itself, known as self-attention, which allows transformer models to create context-aware representations.

The original transformer architecture consisted of two parts, each in turn comprising multiple attention and feed-forward network blocks. These parts are known as the encoder and decoder, referring to the original purpose of the transformer, machine translation. The motivation of this divide was to let the encoder encode the input text to a semantically meaningful vector representation, and the decoder to decode this representation to return the output, a translation matching the original input text. The encoder and decoder were also connected with residual connections. Despite this, the following work has used them separately, with the encoder architecture used in models such as BERT [25] and XLM-RoBERTa [26], and the decoder in models such as the GPT family [27].

As a general rule, encoder models are used for creating semantically meaningful and context-aware representations of text, while decoder models are used for text generation. The full encoder-decoder architecture is also used, like in the T5 model [28]. For additional explanation on the inner workings of a transformer, see Geva et al. [29].

2.2.1 Training a large language model

The goal of LLMs is to create a probability distribution over their vocabulary, i.e. tokens, given the context of surrounding text, and models may undergo multiple different training states and procedures to reach this goal [30]. Generally, decoder models are trained using a method called Next Token Prediction, where the models' weights are optimised towards correctly predicting the next token in a stream of words. Encoder models are mainly trained using Masked Language Modelling (MLM), which entails masking a token in an input and optimising model weights to predict the masked token, or additionally Next Sentence Prediction (NSP), where weights are optimized to predict if a given sentence follows another in a natural text span [25]. All of these procedures require massive amounts of good-quality text data, and the need for training data scales with the model size [20].

Training a language model with a large, general-purpose dataset to acquire language capabilities is referred to as pretraining. A pretrained model can be further *finetuned*, meaning the model can be trained for a more specific task using a smaller dataset of labelled data. This way, the general language knowledge learned in pretraining can be used to jump-start the supervised learning step. The model can also be further trained, known as *continued pretraining*. In this case, pretraining

techniques are used, but the continuation training dataset is somehow limited, for example, it can be of a specific domain. This way, a general healthcare LLM does not have to be trained from scratch with healthcare domain texts, but a pretrained model can be continually pretrained to understand healthcare domain vocabulary [31], [32].

Another perspective on the in-or-out-of-domain question is the language of the training data. Typically, a multilingual model is trained on a large collection of data comprising multiple languages. Similarly to the above mentioned healthcare domain adaption using continued pretraining, multilinguality with respect to a specific tasks can be built in the model in a somewhat similar setup: if training data for a task exists only in one language, it can be used to finetune a multilingual model while preserving the multilingual capabilities, which can even be beneficial in some cases [33]. For instance, a multilingual model trained for text classification on English only will be able to do classification on the other languages as well [34]. Zhao et al. [35] hypothesise that LLMs predominantly trained on English with smaller amounts of data from other languages understand the reasoning needed to complete the tasks first in English and translate the output to the desired language on the later layers of the model. Chang et al. [36] conduct studies on 88 languages and visualise the internal representations created by `XLM-RoBERTa` [26] and show that some dimensions of these representations are language-independent, meaning they are shared between the languages, while other dimensions are distinctively connected to a single language.

3 Personally Identifiable Information

In this chapter, Personally Identifiable Information, or *PII*, is defined for the needs of this thesis. I also cover different kinds of attacks that can be performed on machine learning models trained on sensitive data, and the different PII substitution schemes that are commonly used. In this thesis, as the data used comprises electronic health records with predetermined PII annotations, PII is defined as names, sex markers, emails, IDs, locations, and professions. In reality, the definition of PII and its redaction are multifaceted concepts.

3.1 Risks associated with personal information

PII can be viewed as a spectrum. Depending on the given context and additional information available, one piece of personal information can or cannot be linked to a real-life person or entity. For instance, a PII-redacted passage like " `<NAME>` (*the president of Finland*)" can straightaway be linked to only a couple of people, and further to only one person if additional context is given. On the other hand, another sentence that has not been redacted, "*John Brown enjoyed his coffee before*

leaving to office", can only be linked to a person with a lot of additional context and information, as the name, office work, and enjoying coffee are all common things and may refer to multiple people and situations. Thus, a risk continuum exists for PII, one end of which contains text with no risk of identification and the other information that can be explicitly linked to a person [37], [38].

PII can also be characterised by the legislation around data protection, such as the HIPAA (Health Insurance Portability and Accountability Act of 1996, US Public Law 104 - 191) in the US and the GDPR (General Data Protection Regulation, Regulation (EU) 2016/679) in the EU. The GDPR sets the principles for handling personal data, e.g. the data minimisation principle (collecting and retaining only the necessary amount of records that are required for operation), and data security measures. In Finland, *Tietosuojalaki* (1050/2018) [39] (eng. *Data protection act/law*) complements the GDPR with special exceptions related to healthcare, supervisory powers, and sanctions. Similar legislative definitions around the world are often behind their time, and they may leave some categories, like emails, out of their definitions of PII, despite them being generally considered personal information these days [40].

Many data sources valuable to different parties cannot be accessed due to them containing identifiable information and the subsequent massive task of PII redaction [41] [42]. This is particularly true in the medical field [43]. Thus, whenever feasible, rule-based algorithms or machine learning approaches are commonly used in PII redaction. Like any automated method, they are also prone to demographic bias and may lead to subpar results in some populations [44]. Currently, the state-of-the-art results are achieved with LLMs. Although powerful tools, their usage

also holds the prospect of accidentally exposing PII.

LLMs can leak unwanted information in multiple ways [45]. Carlini et al. [46] investigated the possibility of gaining access to a language model's training data from the model's output probability distribution. They used the GPT-2 [27] model in their experiments, trained on crawled web data, and were able to access training data examples containing PII, like email addresses, with simple queries. They assert that the larger the model is, the more likely it is to memorise its training data despite no clear signal of model overfitting. They also recommend measures to mitigate these risks. Li et al. [47] propose a multi-step prompting procedure to make ChatGPT reveal PII present in the model's training data, even though ChatGPT is finetuned to not reveal any personal information, and refuses to generate PII with direct prompts. Overriding this quality is done by first giving the direct prompt to access PII, then using a second prompt while posing as the model to set the model in "jailbreak mode", and finally prompting the model again to reveal PII. Their results show that while the success rate for these attacks is not high, it can cause serious privacy threats.

Use of data containing PII in Artificial Intelligence (AI) applications has been noted in lawmaking. The European Union's AI Act (Regulation (EU) 2024/1689 [48]) categorises AI systems based on the risks they pose to users, with more risks equating to more regulations. The European Data Protection Board (EDPB) has stated in an opinion piece [49] that while the privacy regulations, like the GDPR, protect the right to privacy and encourage responsible innovation, they emphasise that an AI model trained on sensitive data cannot be considered anonymous without substantial scrutiny to ensure the probability of PII leakage is insignificant.

3.2 Personal information in the medical field

In healthcare, almost all data is sensitive by design. Specifically, electronic health records contain a massive number of PII, up to 20% of words in discharge summaries [3]. Healthcare records containing PII are also of a very injurious type if leaked, as seen in the Finnish psychotherapy provider Vastaamo's data breach in the late 2010s.

Record keeping in medicine serves the important purpose of maintaining continuity of care and is thus required of the service provider in Finland. The Finnish law defines multiple aspects of this record keeping, such as the contents and authors of these records, data retention periods, and the permissions to store and access these records (Laki sosiaali- ja terveydenhuollon asiakastietojen käsittelystä 703/2023 [50]), as well as regulations on the file formats, obligatory data fields, data retention and deletion plans, and classification of records warranting special level of protection, such as genetic information (Sosiaali- ja terveysministeriön asetus sosiaali- ja terveydenhuollon asiakastietojen käsittelystä 457/2024 [51]).

Healthcare workers are required to record, retain, and protect health records by law (Laki terveydenhuollon ammattihenkilöistä 559/1994, §16 [52]). Laws and regulations also ensure the rights of the patient to accurate and timely records, with the possibility to check their records (Laki potilaan asemasta ja oikeuksista 785/1992, §12-§13 [53]). Despite the laws surrounding personal information being restrictive by nature, the Finnish law allows for the secondary use of medical and health data in research, population statistics, innovation, education, and supervision by authorities, as long as privacy is upheld (Laki sosiaali- ja terveystietojen

toissijaisesta käytöstä, 552/2019 [54]). The EU legislation requires this data to be anonymised and distributed in secure remote environments (Regulation (EU) 2025/327 [55]).

The EU AI Act (Regulation (EU) 2024/1689 [48]) considers AI applications concerning health data in the high-risk category. The European Union's Data Act (Regulation (EU) 2023/2854 [56]) requires providers to share data only after all personally identifiable information has been removed. Other regulations surrounding the usage of health data in Finland include legislation on social welfare and health care records (Laki julkisen hallinnon tiedonhallinnasta [57]).

3.3 Redaction schemes

Like PII, ways of redacting PII can also be viewed as a spectrum. The most important distinction between these methods concerns the differences between anonymisation and pseudonymisation. While pseudonymised data can still be linked to a person, full anonymisation means no possibility of recognition. This level of anonymisation can be achieved by aggregating results or by representing the results as population-level statistics [58]. The WP29, the EU working party for article 29 which has been since replaced by European Data Protection Board, has also released an opinion (Opinion 05/2014 on Anonymisation Techniques [59]) relating to PII, which defines categorisation of PII classes, a risk-continuum of PII, and different pseudonymisation techniques, which they emphasise are not the same as full anonymisation.

Different methods of redacting – or finding and pseudonymising – PII are com-

monly specified in the literature. For the purposes of this thesis, the most important schemes are the ones used by Vakilini et al. [3] and Berg et al. [4]. First of these is "blacking out" or censoring the text segment containing PII, by, for example, replacing the segment with some other symbol. The second method can be called tagging, where each type of PII is replaced with a tag, like `<PHONE-NUMBER>`. This differs essentially from blacking out, as information about the type of PII is preserved, which may increase the usability of the data. The last method is substitution, sometimes also called pseudonymisation. In this method, a word with a similar semantic meaning is used to replace the word, like a phone number being replaced with another possible phone number, or a name being replaced by another name. Substitution has the added benefit of alleviating the problem with false negatives: for example, if an adequate percentage of PII has been successfully substituted with a believable replacement, it is incredibly hard for an attacker to figure out which PII has and which has not been redacted, called a Hiding-in-Plain-Sight approach to PII redaction. With the two other methods, false negatives are immediately evident for an attacker [60].

Substitutes can be generated, for example, from a knowledge base, as described by Sanches et al. [61], or by an LLM, as in the work of Albanese et al. [2]. Despite protecting unredacted PII better than censoring or tagging redaction, both of these methods may be vulnerable if an attacker has the information on the knowledge base or LLM used for substitution generation, and can thus gain some estimates on which PII is redacted or unredacted.

Baudart et al. [62] also describe additional redaction schemes, such as encryption, in addition to tagging and providing a human-readable redacted version of the

PII passage in the context of chatbots. In cases where a chatbot needs to access personal information to do the given task, their solution limits the amount of compute using the actual PII so that an attacker targeting the chatbot cannot access it in unredacted format.

Berg et al. [4] report that low precision in PII detection and following redaction leads to decreased downstream performance, which can be somewhat mitigated using a substitution scheme, and note that redacting PII always contains a trade-off between redaction ability and data usability. Lothritz et al. [5] investigate the effect PII redaction scheme has on two Transformer-based LLMs' performance in 9 different tasks. The models they test are BERT [25] and ERNIE [63]. Their findings show that, on average across models and tasks, redacting PII has a negative impact on LLM performance, with some tasks showing slight improvements and some losses, the losses being more substantial. They also report that the substitution method of PII redaction leads to smaller losses in the performance of downstream tasks. This is corroborated by Vakili et al. [3] in the context of health care related documents; however, their results showed better downstream performance in almost all tasks with substitution redacted training data, and they note that there is no adverse effect in substitution redaction. Substitution redaction in health records presents the evident risk of modifying the medical subject matter, like changing diseases to another condition or modifying the results of the test run on the patient. This may cause unique issues for the usability of the data after the PII redaction.

4 Methods of PII Detection

Before PII can be redacted, it needs to be detected in the data reliably. As discussed in Chapter 3, different redaction schemes are susceptible to varying levels of security threats, and this is, of course, amplified by the robustness of the detection system. In this chapter, four different ways of detecting PII are presented. For each, the advantages and disadvantages are covered. For the main focus of this thesis, the implemented zero-shot method, an additional substitution method is also presented. I also introduce recent, real-life examples of PII redaction from the field of NLP.

4.1 Rule-based systems

Rule-based systems are, on the surface, the easiest methods to implement for PII redaction. As the name suggests, this method uses some hard-coded rules to detect and redact PII. Despite being a simple technique, it is used in many very recent LLM training efforts [64], [65], [66], all of which use regular expression-based PII detection pipelines.

The main benefit of rule-based systems is that they can be incredibly fast to run on massive amounts of data [66]. They are also sufficient for PII categories that

are defined by some agreed format, like phone numbers and email addresses [65]. However, in web crawled datasets, it is not uncommon for people to misspell or simply choose to write in an unconventional format, which leads to PII evading detection. Rule-based methods also cannot take the surrounding context into account, which renders them ineffective for other types of PII.

As an example, Allal et al. [65] describe their PII removal in the following way. Their model, *SantaCoder*, is a programming language and problem-solving focused LLM. Their training data consists predominantly of code; most PII in their data is API and SSH keys, together with email and IP addresses, all of which follow established formats. Hence, as discussed above, a rule-based approach is deemed appropriate for redacting the training data. The authors evaluate their rules by manually constructing an annotated testset, and reach precision and recall in the range of 80-90% for all targeted PII categories.

4.2 Named entity recognition

Named Entity Recognition (NER) is a token classification task where a language model, usually an encoder model, assigns a tag from a pre-determined set of tags to each token of a text [67] (originally by [68]). These tags may feature categories such as person, location, organisation, and naturally a "none" token for no prediction. NER is commonly used in information extraction [69], question answering [70], and PII detection [38].

Using a NER tagger for PII detection is especially good in cases where the PII is particularly context-dependent, in which case using a rule-based system is error-

prone, or the number of rules the system needs to operate grows too large. NER-type detection is specifically beneficial in cases where PII is critical to maintaining the performance of the models trained on it, or in cases where the data contains instances that on the surface look like PII but in reality are not.

As a downside to using NER-based systems, the model requires PII-specific training and thus, a lot of annotated data is needed. This usually requires the efforts of manual annotators, as the NER approach is used to specifically target more contextual types of PII that cannot be found and thus cannot be generated using rule-based systems. Manual annotations can be and often are crowd-sourced [71], [72]. Current trends in the field of NLP also point towards training data generation using LLMs [73].

An example of using NER for PII detection in recent LLM training is *StarCoder*, a generative model trained on multiple programming languages [38]. The authors initially implemented a NER tagger with 6 tags, including entities commonly found in code, such as email addresses and usernames. In this case, the PII present in the data is contextual, as it may contain licenses where names should not be altered, or placeholder names that are not actual PII and redacting them may affect the executability of the code. Training the NER tagger required manually annotated data produced by paid human annotators. Even with this annotation process, the results of NER tagging did not reach the needed performance, so additional measures were employed. The training data was synthetically augmented with a pseudo-labelling technique: they created more training examples using an ensemble of two taggers trained on separate training samples. Lastly, post-processing by removing tagged entities which violate some set of simple rules was done.

NER PII detection has been successfully applied to healthcare-related documents. Vakili et al. [3] use a NER model for PII detection on Swedish electronic health records. They use a large PII annotated health record corpus for training the model, encompassing 9 PII categories, and reach precisions in the range of 47% – 98%, and recalls in the range of 35% – 97%, depending on PII category. They show that detection of numerical data, such as dates and IDs, proved to be harder than detecting PII categories such as names, which were detected very reliably.

As the above examples show, using NER for PII detection requires either a lot of effort to create adequate amounts of training data or a ready-to-use large corpus specifically applicable to the task at hand. Nonetheless, NER is often the best solution due to the contextual nature of PII present in the data at hand.

4.3 Zero-shot approach using a large language model

A novel and somewhat unconventional technique for using a large language model to detect PII was introduced by Albanese et al. [2]. In this article, the authors present a method for using an MLM-trained encoder model's output probabilities to infer which tokens contain information that should be redacted. This method iteratively masks each word of the input and analyses the model's prediction of the masked word. The underlying hypothesis of this method is as follows: a model that is trained to create contextual representations will create less confident predictions for a token that cannot be inferred from the context. This means the information content of this token is large, and redacting these tokens will decrease the overall information content of the document. This corresponds to the set-up for finding

PII, which are the most information dense segments of a text: a language model usually gives confident predictions for a masked token if it can be inferred from the sentence structure, but the predictions are unsure if it is a name, phone number, address, or other PII. The underlying principle is illustrated in Figure 4.1. In this example, the model struggles to predict the PII but can give confident predictions for the word "name". This is analogous to how humans usually interpret and approach this problem.

As discussed in 2.2.1, for a masked token in a sentence, an encoder model produces a probability distribution over the whole vocabulary of the model. By masking each token, one by one, we get the conditional probability distribution

$$\mathbb{P}(w_i \mid s \setminus \{w_i\}) = f_i(s \setminus \{w_i\})$$

where $\setminus$ indicates masking token w_i in sentence s, and f_i is the output for ith token. In practice, the output for a sentence can be simplified to a vector containing probability values corresponding to each token in the model's vocabulary. This usually allows us to choose the most probable token as the final prediction, but in this method, we instead look at the probability of the correct token: if the probability of the correct token falls below some threshold, then the token is hard to infer from the context. This corresponds to the likelihood of the token containing PII, and consequently, tokens with scores falling under some threshold can be deemed as redactable.

Results of this PII detection scheme show a significant precision-recall trade-off, as the threshold of redaction varies. High precision is obtained with low thresholds,

Figure 4.1: Illustration of the zero-shot PII redaction method. The model can easily predict the masked word "name"; however, it struggles to predict the word "Peter", resulting in weak guesses overall. The probability value of Peter is low, which means that "Peter" cannot be inferred from the context and is likely PII.

while high recall is obtained with high thresholds. Overall, the highest F1-score obtained in the study is 0.4. Recall, which measures the number of false negatives, is in some sense the more important metric in PII redaction. However, maximising recall leads to the full redaction of the documents. To combat this problem and the issue of false positives with increased recall, the authors also introduce a substitution redaction method working in parallel with their detection system. As the probability distribution is calculated for each token to see if it needs to be redacted or not, this distribution can be used to calculate a substitute directly. This substitute is selected by computing the cosine similarity of candidate tokens and selecting a random substitute within the k best matches. They show that while their system produces false positives, the selected semantically similar substitutes mitigate the effects on downstream tasks.

As a benefit of this method, no additional fine-tuning data is needed, unlike with the NER approach. This means that no additional PII annotated dataset is required, and this method can be employed directly as long as there exists a language

model with adequate language modelling capabilities for the target language. In this thesis, this claim is challenged by analysing this setup for PII redaction in the context of electronic health records, which greatly differ from the data used by Albanese et al. [2].

4.4 Instruction tuned large language models

Instruction or chat-tuned LLMs, like ChatGPT, can also be used for PII redaction. They are well-adept at various tasks, and the interest in utilising them for NER-style PII detection-related tasks exists [74], also in the context of health records [75]. It is clear that these models hold the most power regarding robust PII redaction, as their capabilities include the benefits of the contextual NER approach while also not requiring explicit PII category definitions or large fine-tuning datasets. However, these models are massive in size, and computation can be expensive for large datasets. Furthermore, some most readily available models are not open source and are only available online, meaning they cannot be used to handle sensitive data. Open-source alternatives include the Llama-3 model family [76] and DeepSeek-v3 family [77].

Despite the drawbacks in size and required compute, Wiest et al. [7] demonstrate the power of using a large model. They develop an LLM PII redaction pipeline for clinical text and show that using the Llama-3 model with 70 billion parameters results in recall of 99,24% and precision of 97,57%, outperforming rule-based and NER approaches.

5 Experiments

In this chapter, I describe the implementation of the zero-shot PII redaction pipeline, based on the work of Albanese et al. [2]. Encountered problems, their causes and their solutions are covered with examples. Likewise, I introduce metrics used in evaluation.

5.1 Data

The data used in the experiments of this thesis is the MEDDOCAN dataset [78]. This dataset is a high-quality health data collection comprising patient reports in Spanish, with manual annotations for PII included in the metadata. Labels of the PII present in the dataset are in Table 5.1. Each document in MEDDOCAN consists of a list of patient information followed by a short but often multi-paragraph patient record, summarising the examination and possible procedures done for the patient. All data in MEDDOCAN is synthetic, meaning it does not contain any real sensitive information. For the experiment of this thesis, publicly available development and test sets[1] (N=20, both separately) of the dataset were used for threshold optimisation and final result calculation, respectively, and were trans-

[1] `https://github.com/PlanTL-GOB-ES/SPACCC_MEDDOCAN`

lated to English and Finnish using DeepL, which is the at the time of translation the leading translation tool with a free version available[2]. Quality of DeepL was also observed in this experiment, as the quality of the translations was deemed good aside from problems with sex markers (F/M), which were sometimes mistranslated from the Spanish M/H for "mujer" (woman) and "hombre" (man).

Annotations for PII are given in a separate file as a range of characters, e.g. (`NAME, 22, 29`), signalling that characters between indices 22 and 29 of the text contain a name that should be redacted. This causes a problem for machine translation, as translating a text changes the location of these annotations. The solution that worked with both the annotation scheme and DeepL translation tool was to include the annotations as tags in the original text, then translate to English or Finnish, and finally remove the tags and save their indices to get a clean translated document with the same annotation format as before. Multiple tagging systems were explored (i.e. full substitution with a tag, surrounding the annotated segment with markers) and the best results were achieved by encompassing the segment with brackets and preprending the annotation type to the segment (e.g. `[ #NAME <segment>]`). This resulted in almost perfect annotation number and order preservation, with a small number of errors in flagged documents corrected manually. As a downside, in both Finnish and English, this resulted in some unnatural translations, such as "Mies iältään 59 vuotta" (approx. *Man of age 59 years*) as opposed to "59-vuotias mies" (*59-year-old man*). Although these do not reflect the most common language usage, they are grammatically correct and are used in certain situations.

[2]e.g. `https://translatepress.com/deepl-translator-review/`

Type	MEDDOCAN label	Translation
Names	NOMBRE_SUJETO_ASISTENCIA	Patient name
	NOMBRE_PERSONAL_SANITARIO	Healt personnel name
	FAMILIARES_SUJETO_ASISTENCIA	Relative's names
Sex	SEXO_SUJETO_ASISTENCIA	Sex
Profession	PROFESION	Profession
Numerical	FECHAS	Date
	EDAD_SUJETO_ASISTENCIA	Age
	NUMERO_TELEFONO	Phone number
	NUMERO_FAX	Fax number
Online	CORREO_ELECTRONICO	Email
	DIREC_PROT_INTERNET	IP address
	URL_WEB	URL
Location	HOSPITAL	Hospital
	INSTITUCION	Institution
	CALLE	Street
	TERRITORIO	Territory
	PAIS	Country
ID	IDENTIF_BIOMETRICOS	Biometric ID
	NUMERO_IDENTIF	ID (other)
	ID_SUJETO_ASISTENCIA	Patient ID
	ID_CONTACTO_ASISTENCIAL	Healt contact ID
	ID_ASEGURAMIENTO	Insurance ID
	ID_TITULACION_PERSONAL_SANITARIO	Qualification ID
	ID_EMPLEO_PERSONAL_SANITARIO	Employee ID
	IDENTIF_VEHÍCULOS_NRSERIE_PLACAS	Vehicular ID
	IDENTIF_DISPOSITIVOS_NRSERIE	Device ID
	ID_CENTRO_DE_SALUD	Health center ID
Other	OTROS_SUJETO_ASISTENCIA	Other

Table 5.1: Labels of the MEDDOCAN corpus with approximate translations. A full explanation of labels and the annotation process can be found in the MEDDOCAN documentation. Type categorisation is not defined by the authors of MEDDOCAN but is presented here for ease of reading.

5.2 Platform and hardware

All computational resources for this project were provided by CSC – IT Centre for Science. For the experiments, I used the Mahti supercomputer located in Kajaani, Finland. All CSC services use 100% renewable energy with waste heat redirected to district heating. The code for this experiment was written in Python using the PyTorch version of the Transformers library, which offers support for a variety of language models and is a standard library in the field. The pipeline was written to support both GPU and CPU, with GPU used in the experiments.

5.3 Implementation

An overview of the whole PII redaction pipeline is presented in Figure 5.1. As the models take input as tokens, the whole pipeline is conceptualised with respect to the tokenised input paragraph and the indices of each token in it. The pipeline starts by tokenising the input text. Next, masking indices are calculated for each white-space separated unit (English) or individually parsed words (Spanish and Finnish). These indices are used to mask the tokenised input. After masking, I calculate the predicted probability of each masked token. Then, I aggregate the results over subword tokens, and as a result, the pipeline outputs one score per word describing the likelihood of the word being PII. Lastly, words below some threshold are redacted, and this threshold is optimised for each model and language. All steps in the pipeline are further explained in the subsequent sections.

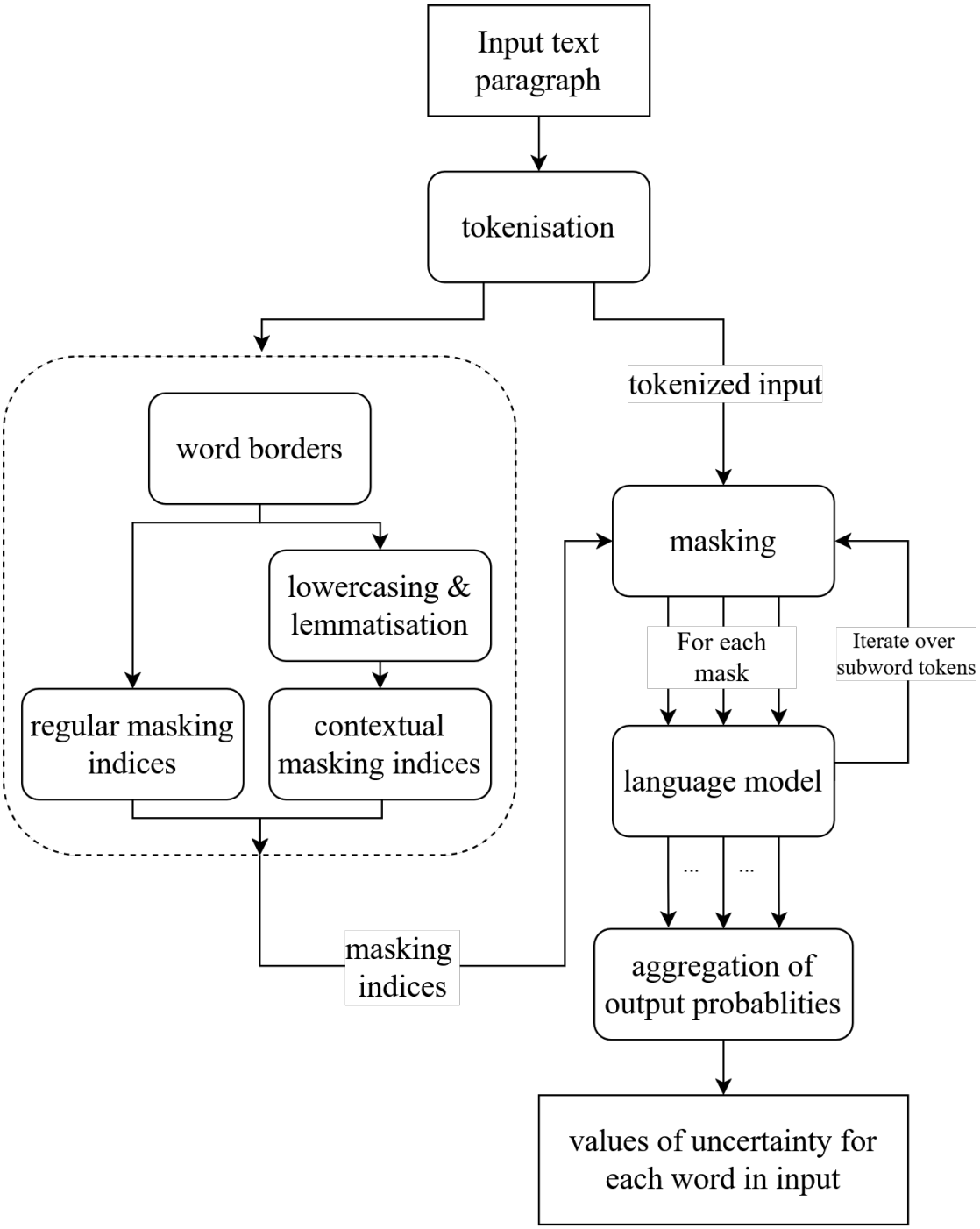

Figure 5.1: Visualisation of the zero-shot redaction pipeline

5.3.1 Models and tokenisation

The models used in this experiment are listed in Table 5.3.1. Model selection was based on the `Huggingface`-library's documentation, equipped with the possibility of filtering language models based on training data types, modality, language, and model architecture. Additionally, two reviews on transformer models in biomedicine [43] and clinical research [79] were consulted to ensure models selected with `Huggingface`'s tool were used in the medical or healthcare domain. Models were chosen to represent both monolingual and multilingual types, as well as models trained on in-domain data with differing setups (continued pretraining vs. full healthcare-domain pretraining), and models not specifically trained on any significant amounts of biomedical or healthcare-related textual data.

Out of the models in the healthcare domain, `SAPBERT-UMLS` and `BioMedRoBERTa` have been trained from general language models using continued pretraining. `SAPBERT-UMLS` model has been trained from `XLM-RoBERTa-large` with the medical UMLS corpus [80], while `BioMedRoBERTa` from `RoBERTa` model using Semantic Scholar bio-medical research articles. Although the UMLS corpus is English, it has been shown that the multilingual abilities of a model used as the basis of continued pretraining or finetuning are preserved in the resulting model, even if the continued pretraining is done on a limited number of languages [33], [34]. Medical and healthcare domain models fully pretrained from scratch are `Dil-BERT`, pretrained from PubMed and medical Wikipedia articles, `BioClin-BERT`, pretrained with the MIMIC III corpus [81], and `PubMedBERT`, pretrained with PubMed. Finally, the general domain language models, `XLM-RoBERTa`(`-base`) and `BERT`, are included to examine the difference in domain shifting. In the following sections, `BERT` is used

Model	Language	Domain	URL	Citation
`BERT`	English	General	link	[25]
`BERT-FI`	Finnish	General	link	[83]
`BERT-ES`	Spanish	General	link	[84]
`BioClin-BERT`	English	Medical (clinical)	link	[85]
`BioMedRoBERTa`	English	Medical* (research)	link	[32]
`Dil-BERT`	English	Medical (disease)	link	[86]
`PubMedBERT`	English	Medical (research)	link	[87]
`SAPBERT-UMLS`	Multilingual	Medical* (research)	link	[88]
`XLM-RoBERTa-base`	Multilingual	General	link	[26]

Table 5.2: Models used in the experiment of this thesis. In the medical/healthcare domain models, asterisk (*) implies continued pretraining as opposed to full pretraining.

to refer to all `BERT` models for simplicity, as the language is specified in each case.

The prevalence of English in NLP and in bio-medical and clinical research is visible in the available models. Models marked as multilingual in Table 5.3.1 feature over 100 languages in their training data, including English, Finnish and Spanish. A model trained specifically on Spanish biomedical and clinical data was considered, `roberta-base-biomedical-clinical-es` [82]. Regardless, due to the pipeline relying heavily on the defined tokeniser types (see Section 5.3.1) and intermediate values of the model, experimenting with this model would have required rewriting the full pipeline, and this was deemed to be out of scope for this work. Leaving this model out of the experiments is unfortunate, as the Spanish documents do not contain any of the biases or translation errors that might have been caused by using DeelL in the Finnish and English data. However, the quality of DeelL translations was deemed good in the manual analysis explained in Section 5.1.

Tokeniser types

The models chosen for this experiment use two different tokeniser types, namely WordPiece in models based on the `BERT` architecture and Byte-Pair Encoding (BPE) for models with `XLM-RoBERTa` architecture. This means all functions dealing with tokenisation, masking, and scoring had to be written for both tokeniser types separately. This was easily the most time-consuming part of implementing the pipeline. To further clarify the difference between the tokeniser types, a Wordpiece tokeniser marks continuation of the same word with a "##", while BPE signals white spaces with a special underscore character "_". As an example, a WordPiece tokeniser tokenises the sentence "I'll call you tomorrow" as

`[“I”, “##’ll”, “call”, “you”, “to” “##morrow”]`,

while a BPE tokeniser outputs

`[“I”, “’ll”, “_call”, “_you”, “_to” “morrow”]`.

This difference requires separate handling for subword tokens in almost all parts of the pipeline, even in the model prediction extraction.

Model sequence length

Despite most documents in the MEDDOCAN dataset being relatively short, most do fall over model sequence lengths, generally 512 tokens. To combat this issue and get results for all tokens in a given document, a sliding window approach was implemented. First, initial k tokens are selected for model inputs, and scores are extracted as normal, where k is the model sequence length. The sliding window is defined as the length of overlap, $m = \lfloor \frac{k}{2} \rfloor - 1$, with 1 subtracted to facilitate a new

[CLS] token added to each window. Then, windows are iterated with given overlap m until scores for all the tokens are predicted and their scores are extracted. See section 5.3.3 for score calculation details.

5.3.2 Masking

As stated earlier, especially with Finnish, tokenisers may and often do split words into multiple tokens. This means all subword tokens in a word need to be masked at the same time, so as not to predispose the model towards the correct answer. The pipeline calculates the indices of tokens corresponding to each word and masks them before the model is used to predict the probabilities. To mask words before predicting probabilities with the language models, I use the <MASK> token available in the models. The <MASK> token is included in all models by design, as they have been trained with MLM.

Context problem

In many cases, a paragraph discussing a topic contains the same words in multiple sentences. When using an LLM that generates contextual representations of words, as is the case with any modern language model, the model is able to infer the word from other parts of the text. This makes masking futile and the analysis of probability values unreliable. The problem is visualised in Figure 5.2. The physician's name is mentioned twice in the document, which means masking only the first instance results in the model outputting high probability values for the name, leading to a false negative in the PII redaction process.

To solve this issue, a context-aware masking procedure was implemented. While

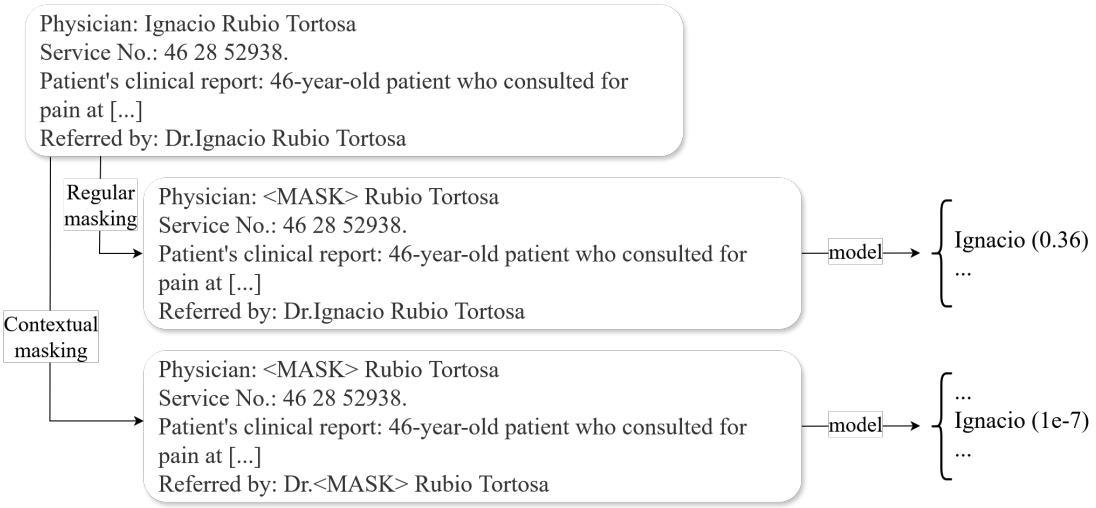

Figure 5.2: Figure illustrating the need for contextual masking as opposed to straightforward, word-by-word regular masking. With regular masking, the language model can infer the physician's name from the end of the document with a high probability value, and thus the name will not be masked. This problem is solved with contextual masking: the probability of "Ignacio" is now small, and the name will be masked. Partial example from translated MEDDOCAN.

iterating over the words of the text and the subword tokens they contain, the pipeline saves additional *context indices* corresponding to identical words in the text. These context indices will be used in the masking step in the same way as the regular masking indices. In the example of Figure 5.2, both instances of "Ignacio" are masked, which leads to a reliable prediction for the name. To mask names correctly within a text span and within other formats, like emails, the pipeline internally lowercases all text for the masking step. The cases are retained elsewhere, especially in the model prediction step, where character case affects the results.

Conjugation problem

Using contextual masking, while mostly solving the problem for English, is lacking for other languages. A further problem is encountered in the redaction of Finnish texts, as well as Spanish texts to a lesser degree, as Finnish and Spanish both feature conjugation. Word conjugations are not a feature of most tokenisers, and thus it is common that the same word in different conjugations consists of completely different tokens compared to its unconjugated form. For example, the word "potilas" (*patient*) and its conjugated form "potilaalle" (*to/for a patient*) are tokenized as `[“pot’’,“il’’,“as’’]` and `[“pot’’, “ila’’, “alle’’]` by the GPT-3 tokenizer [89]. The problem this creates is visualised in Figure 5.3. Conjugated name "Juhan" cannot be found using only contextual masking, as the other occurrence of the name is in its nominative form, "Juha".

This issue was solved by applying lemmatisation – reducing each word to its base form – before calculating contextual masking indices. Instead of looking for perfect token matches in the input paragraph, the pipeline saves the text document as a lemmatised version with mapping to the different original token indices, which are then used to find the matches in the text. Thus, the function masks all instances of the same word, despite conjugated forms. The lemmatiser used for Finnish is `spacy` [90] model `fi-core-news-lg` and `es-core-news-md` for Spanish. Correct lemmatisation-assisted masking is illustrated in the lower half of Figure 5.3. For languages that do not use spaces, a different approach might be needed depending on the used tokeniser. In these cases, a separate model for word limit detection can be used.

Figure 5.3: Figure illustrating the problem with Finnish conjugations. The example translates to "Patient: Juha Heikkilä [...] Juha's wife says that [...]" with the possessive suffix that prevents regular contextual masking highlighted in red. Masking the tokens that make up the word "Juha" might not mask the word "Juhan" as it might not comprise the same tokens as "Juha". With internal lemmatisation, the masking is done correctly.

5.3.3 Scores

Output probabilities, which are used to score tokens in the pipeline, are calculated from the model's last layer outputs before the softmax function is usually applied to get final token predictions. These scores are used as-is for words consisting of one token. However, for words consisting of multiple tokens, the score is aggregated from scores associated with each subword token. The aggregation is calculated as follows: First, the whole word is masked, and the score is extracted from the prediction of the first token from the reading direction. Next, the token processed in the first step is unmasked, and the score for the second token is extracted. This is continued until the end of the word is reached, and finally, the scores corresponding to each subword token are multiplied together. This procedure was

described by [2], as English does feature multitoken words, usually just not as many as Finnish.

A question may arise from the gradual unmaking of the word, namely, if almost the full word is unmasked, would the predictions of the model for the final tokens be biased towards the correct answer? This is exactly what happens, but the multiplication step is used to balance this effect. For example, given a word "Dermatoglyphics" divided in to tokens `[Derma, to, glyph, ics]`, if the probabilities were extracted without gradual unmasking, all tokens would likely be associated with low scores, making the final score – a product of multiplication – minuscule, leading to certain redaction. Contrastingly, by unmasking the start of the word, the probabilities assigned to following tokens are inflated, but this leads to a more balanced aggregated score that reflects the information content more faithfully. Subsequently, the pipeline is not more inclined to give low scores and thus redact long words.

5.3.4 Metrics

In this step, the MEDDOCAN dataset annotations are treated as the gold standard against which the evaluation is done. In the case of labelling multi-token spans in text paragraphs, a common measure to use is the overlap F1-score [70], [72]. This measure accounts for true-positive, false-positive, and false-negative predictions. As described in Section 2.1.3, recall, which emphasises the number of false negatives, is considered a preferable metric when working with sensitive data. However, maximising the recall means fully redacting each document. Thus, recall and precision are both reported in addition to the F1-score. The F1-score is

also used to optimise a redaction threshold for obtaining the final results on the test set.

For the test set, I also report the ROC-AUC score, which uses the true-positive rate and false positive rates, and is commonly used in binary classification, to which the PII redaction can be reduced to by treating redacted words as one class and unredacted as the other. I additionally report the PR-AUC metric, which simplifies to the average precision score over all prediction thresholds in this setting. Multiple metrics are used in this case to show the effects of precision-recall trade-off: high precision indicates the uncertainty of the model's prediction aligns correctly with the likelihood of words being PII, while high recall indicates that the pipeline is able to mask an adequate number of PII in the texts. Metrics are calculated by token, despite the more natural approach of using words. This is due to the multiple tokeniser types that needed separate functions, thus continuing the analysis on the token-level allowed for ease of implementation.

5.3.5 Redaction and threshold optimisation

As the output of the steps described in the above sections, the final redactable list of words and associated scores is obtained. As each model is trained using different procedures, different data, and tokenisers with different-sized vocabularies, they all produce different probability distributions; the threshold for optimal redaction needs to be optimised for all models separately. As a simplified example, the `XLM-RoBERTa` model may output probabilities in the range of $(10^{-1}, 10^{-2})$ for common tokens such as personal pronouns, while the `SAPBERT-UMLS` model may give equivalent tokens probabilities closer to 10^{-4}. Therefore, the development

set of MEDDOCAN is used to optimise a threshold for each model. The optimal threshold is selected by the F1-score. After the thresholds are extracted, they are applied to the test set: all words that score below the optimised threshold will be redacted, while words with higher scores will be preserved. From the test set, final results are calculated with metrics given in Section 5.3.4.

The optimised thresholds are presented in Chapter 6 in Table 6.1, and Figures 6.1 to 6.13 present the metrics as a function of redaction threshold. See Section 6.1.2 for final results on the test set.

5.3.6 Substitution

Substitution generation was implemented as described by Albanese et al [2]. From the predicted probability distribution over the model's vocabulary, the tokens are ranked according to similarity. The similarity is calculated as the cosine similarity of the embeddings produced by the model for each given token in the context of the whole document.

Using this substitution method encountered problems specifically with Finnish multitoken words, as some substituted words became immediately recognisable due to misconjugations in the first tests. Finnish texts required substitutes for words with multiple tokens, requiring multiple rounds of predictions to be ranked by similarity, yet still arriving at a substitute that was not grammatically fitting to its context. This could have been resolved using a Finnish conjugation tool, first detecting the inflectional form of the word in the lemmatisation step, then later asserting that the generated substitution term is in the same form, and if

not, conjugating it with a separate tool. This was nevertheless deemed to be out of scope for this thesis. Secondly, before implementing a full substitution pipeline, it was important to first evaluate the detection prowess of the pipeline. This is due to the nature of electronic health records: if the pipeline predicts medical terminology as PII, and therefore a substitution is generated, the narrative of the text may change drastically, which may render the data unusable. This turned out to be the case, even with the best models tested in this experiment.

6 Analysis of Results

In this section, I cover the results obtained from applying the zero-shot redaction pipeline to the MEDDOCAN dataset. The results are analysed first from a numerical perspective, with threshold optimisation results shown before final results on the test set are covered. Then, qualitative analysis is done by analysing a few examples of documents selected to highlight the advantages and drawbacks of the zero-shot redaction method.

6.1 Numerical evaluation

Numerical evaluation was done in Python using the metrics described in Section 5.3.4. These metrics were calculated using the `sklearn` [91] library for Python, which included functions for all metrics used in the experiment of this thesis.

6.1.1 Threshold optimisation

Results for threshold optimisation, calculated from the development split of the MEDDOCAN dataset, are presented in the next sections, divided by language and model. Models were introduced in Section 5.3.1. These figures present accuracy, F1-score, recall, and precision as a function of the threshold of assigning a word as

redacted or unredacted. In each figure, the lower limit for explored thresholds was selected by checking the lowest value given by the model to any token in the full development set, except for the `SAPBERT-UMLS` model with Finnish data, which scored many tokens as 0. For this model-language pair, 1e-40 was chosen as the minimal value. In all figures, reading from the direction of low threshold values (left to right), the accuracy values are high at first. This is caused by the fact that very low thresholds lead to nothing being redacted, and as PII-annotated words make up a low percentage of all words, redacting nothing results in high accuracy. The same happens, in reverse, to recall: high threshold (close to 1) redacts all words, and thus the number of false negatives drops to 0, making recall equal 1. Precision, which measures the number of false positives, ideally would show high values with low thresholds. This behaviour means the words with the lowest scores, deemed to be most information-dense of all, are also annotated in MEDDOCAN as PII and thus detected correctly. Low precision values overall mean that false positives are detected even with a small threshold, meaning the model's uncertainty in prediction does not align with MEDDOCAN's annotations and does not correspond with PII. The F1-score, taking into account both false negatives and false positives, is used to find the optimal threshold for redacting. In the ideal case, the figures show a peak for the F1-score.

English

Figures 6.1 – 6.7 contain the threshold optimisation results for English. 7 models were used for English data, and most of these models display the precision-recall trade-off – high precision with low threshold values, vice versa for high values and recall – that is visible in the work of Albanese et al. [2]. Notably, this is not the case

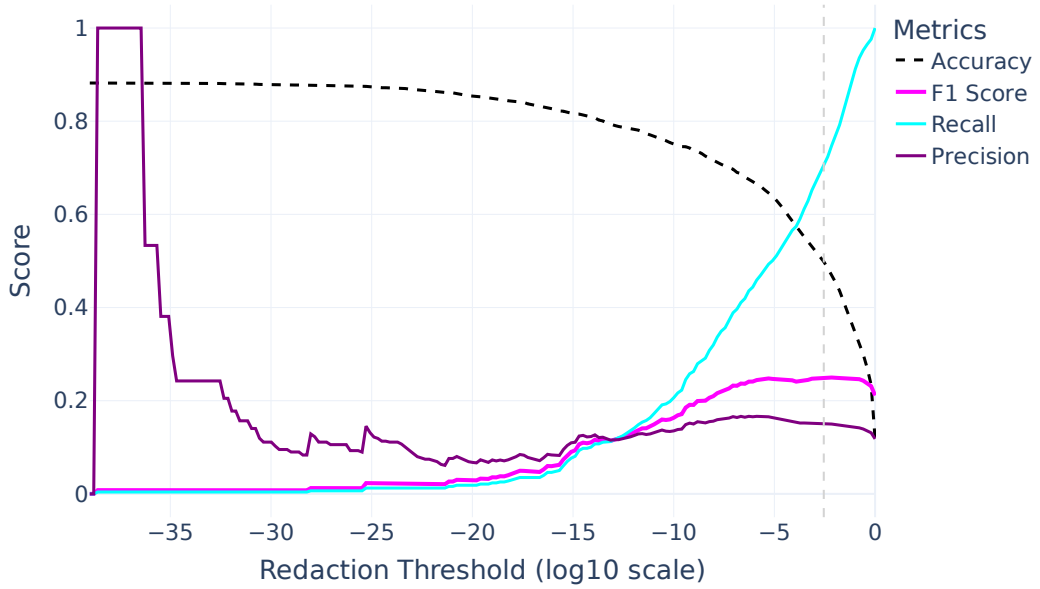

Figure 6.1: `BERT` with English data. The dashed grey line indicates the optimised redaction threshold.

with `XLM-RoBERTa` and `SAPBERT-UMLS`, which do not achieve high precision values at any threshold. This means that the hypothesis of the predicted probability aligning with PII does not hold for these models. `BioMedRoBERTa` model shows precision-recall trade-off but to a smaller extent than other medical or healthcare domain models and `BERT`. `SAPBERT-UMLS` also shows distinction in accuracy as well as recall, showing a "saddle point" in both curves. This model also shows almost no peak in the F1-score, which is used to optimise the threshold. The same is seen in `BERT` and `XLM-RoBERTa`, while the other models show a clear peak. Further analysis of the magnitude of values is left for the test set in Section 6.1.2, as the goal of this section is the selection of thresholds. Selected thresholds are presented in Table 6.1.

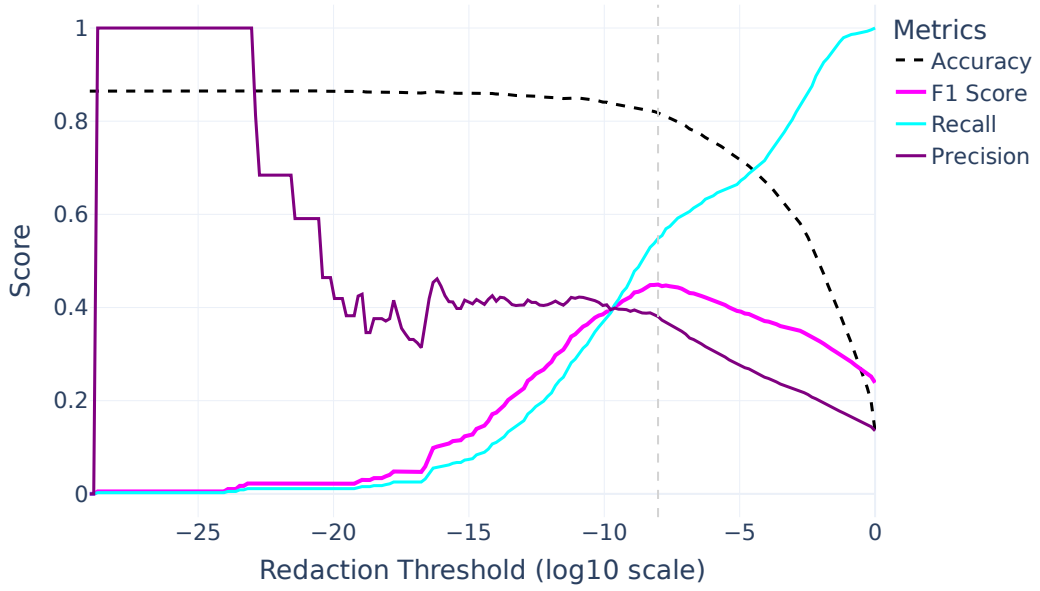

Figure 6.2: `BioClin-BERT` with English data. The dashed grey line indicates the optimised redaction threshold.

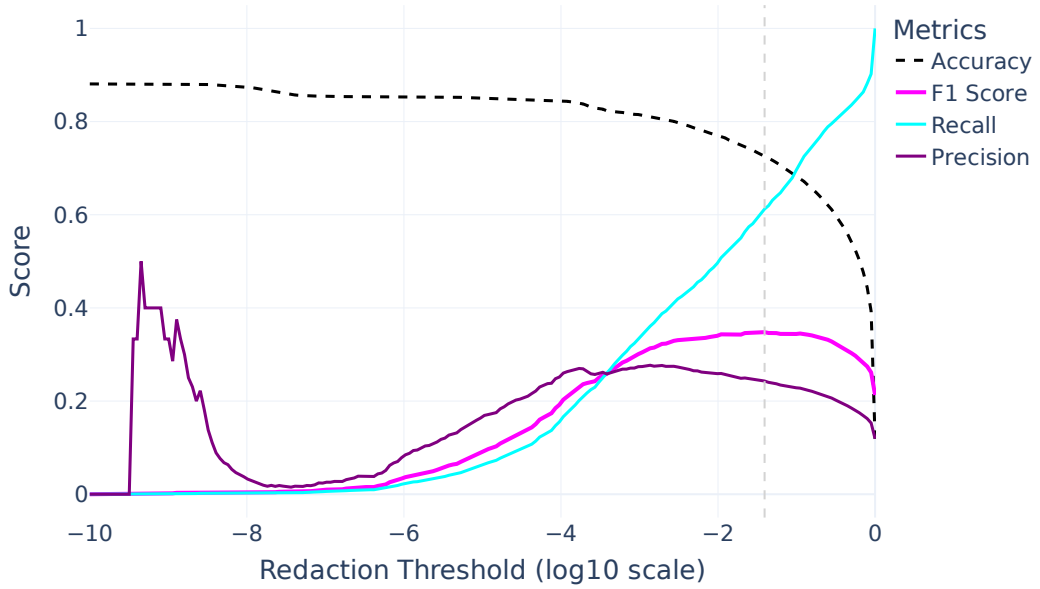

Figure 6.3: `BioMedRoBERTa` with English data. The dashed grey line indicates the optimised redaction threshold.

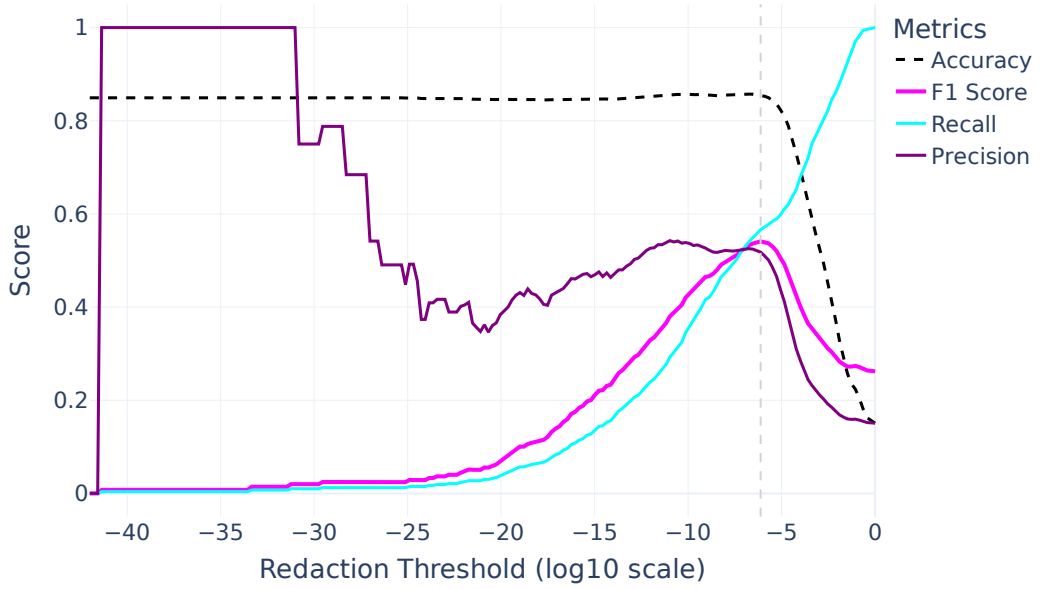

Figure 6.4: `Dil-BERT` with English data. The dashed grey line indicates the optimised redaction threshold.

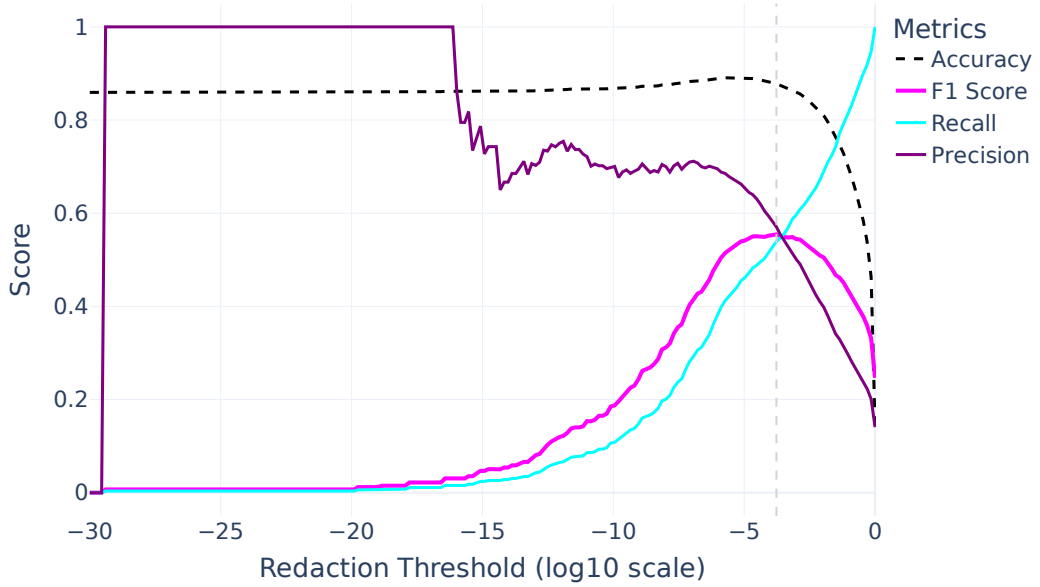

Figure 6.5: `PubMedBERT` with English data. The dashed grey line indicates the optimised redaction threshold.

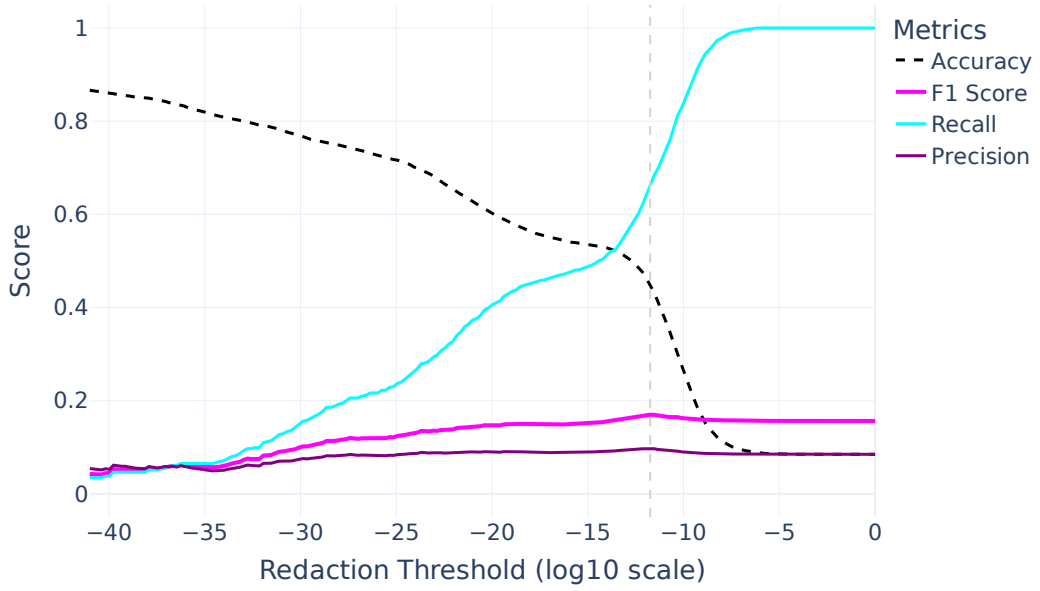

Figure 6.6: `SAPBERT-UMLS` with English data. The dashed grey line indicates the optimised redaction threshold.

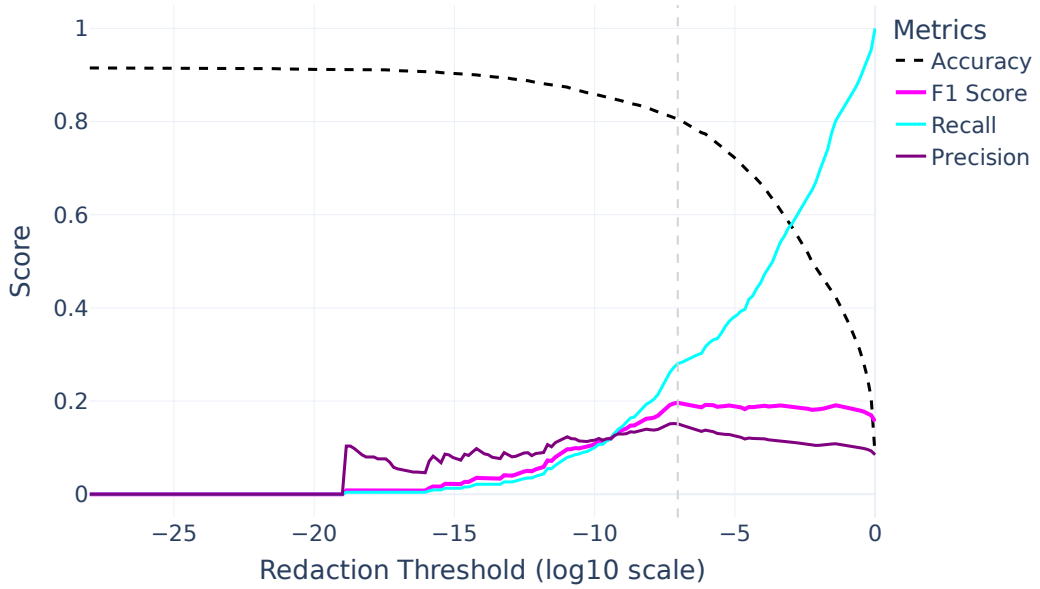

Figure 6.7: `XLM-RoBERTa` with English data. The dashed grey line indicates the optimised redaction threshold.

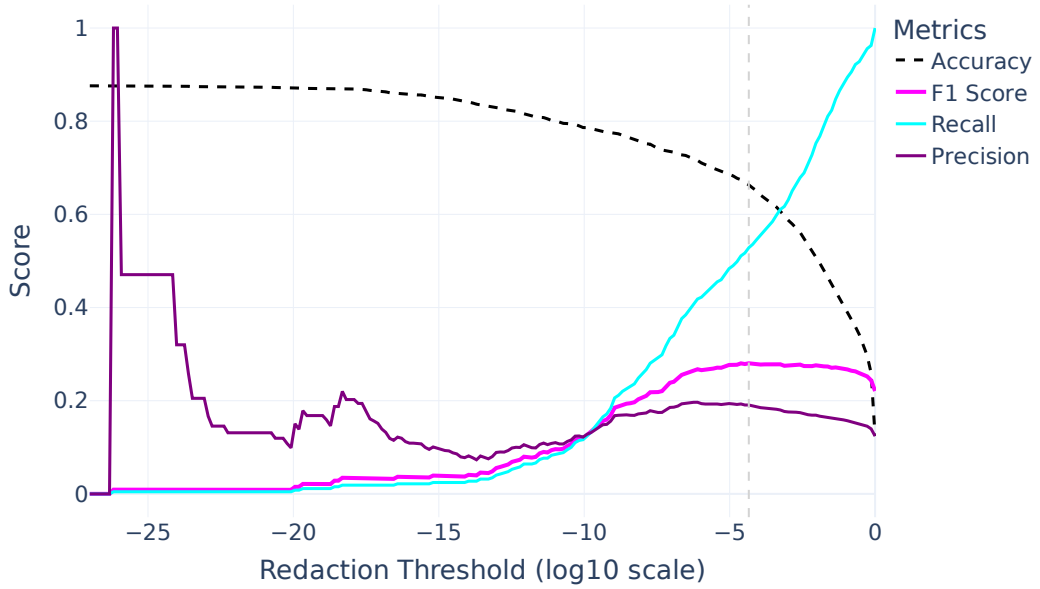

Figure 6.8: `BERT` with Finnish data. The dashed grey line indicates the optimised redaction threshold.

Finnish

Threshold optimisation results are shown in Figures 6.8 to 6.10. High precision is only achieved with the `BERT` model, which shows a clear precision-recall trade-off, with the other models, `XLM-RoBERTa` and `SAPBERT-UMLS`, showing precision values close to 0 at all thresholds. This means that these models' uncertainty does not correlate with the MEDDOCAN annotations, and hints that these models are unsuitable for PII redaction, at least in the context of these health records. `XLM-RoBERTa` and `SAPBERT-UMLS` also show almost no peak in the F1-score. Analogously to English, `SAPBERT-UMLS` shows similar saddle points with Finnish. Table 6.1 shows the selected optimal thresholds, and that the optimal redaction strategy for `SAPBERT-UMLS` was to redact the whole document.

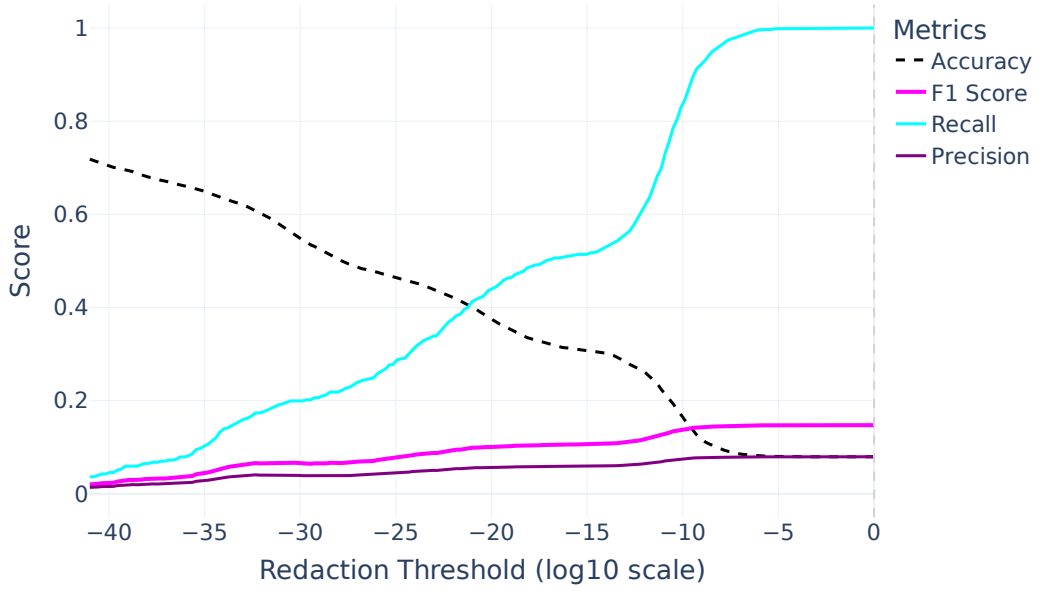

Figure 6.9: `SAPBERT-UMLS` with Finnish data. The dashed grey line indicates the optimised redaction threshold.

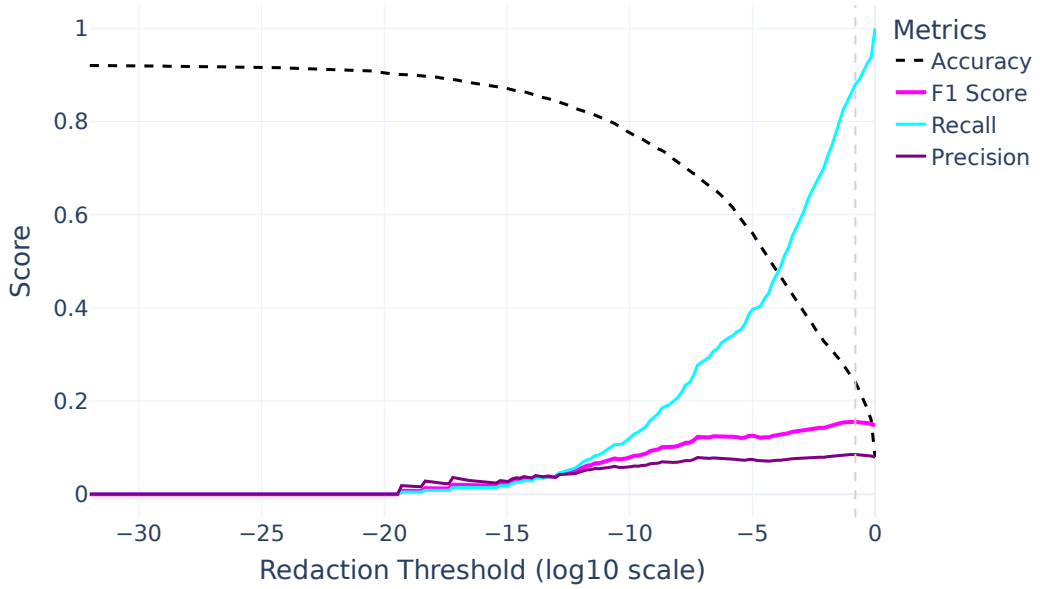

Figure 6.10: `XLM-RoBERTa` with Finnish data. The dashed grey line indicates the optimised redaction threshold.

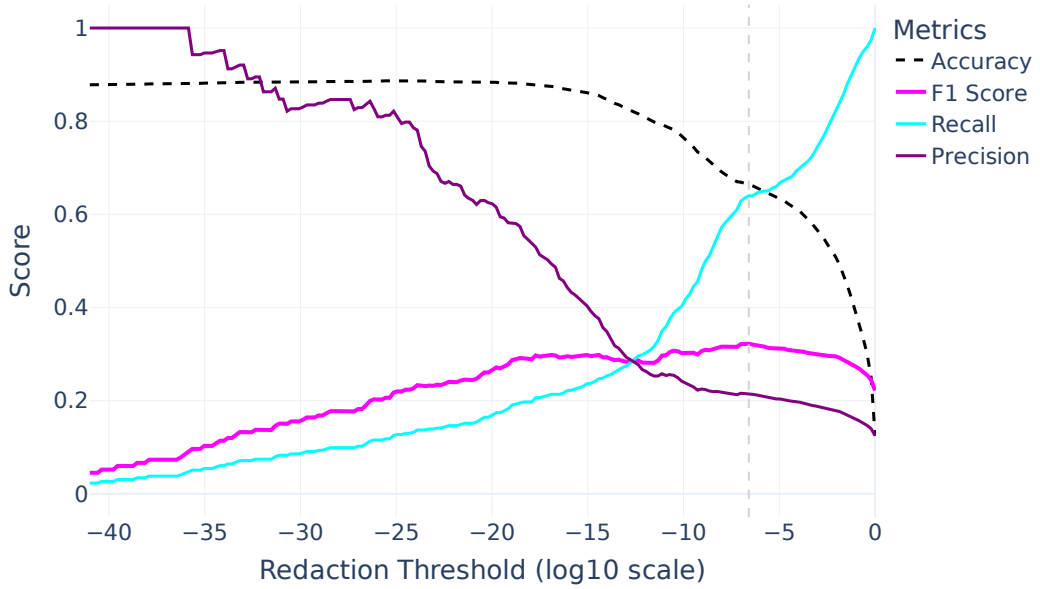

Figure 6.11: `BERT` with Spanish data. The dashed grey line indicates the optimised redaction threshold.

Spanish

The threshold optimisation results for Spanish are presented in Figures 6.11 – 6.13 and Table 6.1. The results mirror those of Finnish, with the `BERT` model being the only one with high precision with low thresholds, while `XLM-RoBERTa` and `SAPBERT-UMLS` show precision close to 0. The general form of the curves follows the ones seen on the Finnish development data, and therefore, the same conclusions apply. The divergent accuracy curve in `SAPBERT-UMLS` is present in Spanish as well.

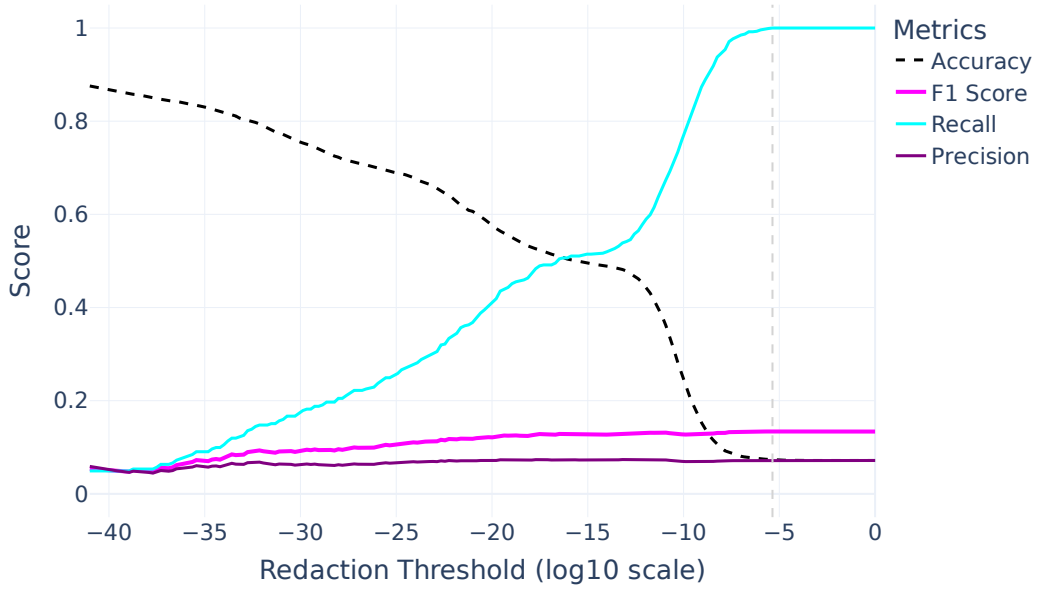

Figure 6.12: `SAPBERT-UMLS` with Spanish data. The dashed grey line indicates the optimised redaction threshold.

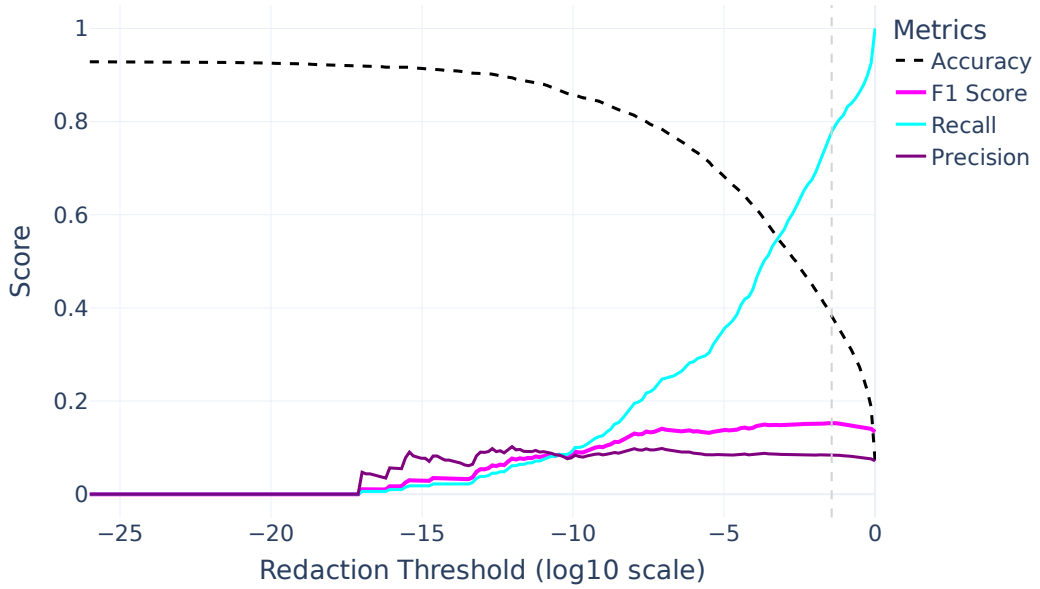

Figure 6.13: `XLM-RoBERTa` with Spanish data. The dashed grey line indicates the optimised redaction threshold.

Language	Model	Threshold
en	BERT	0.0028331
	BioClin-BERT	9.6588e-09
	BioMedRoBERTa	0.039171
	SAPBERT-UMLS	1.8042e-12
	Dil-BERT	7.5753e-07
	PubMedBERT	0.00017028
	XLM-RoBERTa	9.222e-08
fi	BERT	4.5529e-05
	SAPBERT-UMLS	1.0
	XLM-RoBERTa	0.15703
es	BERT	2.5529e-07
	SAPBERT-UMLS	4.3976e-06
	XLM-RoBERTa	0.036544

Table 6.1: Optimised redaction thresholds for language and model combinations.

Lang.	Model	Acc.	F1	Recall	Precision	PR-AUC	ROC-AUC
en	BERT	0.528	0.279	0.733	0.172	0.169	0.648
	BioClin-BERT	0.823	0.470	0.561	0.405	0.381	**0.801**
	BioMedRoBERTa	0.740	0.396	0.684	0.279	0.264	0.763
	SAPBERT-UMLS	0.447	0.174	0.578	0.103	0.108	0.506
	Dil-BERT	**0.865**	**0.569**	0.586	0.554	**0.460**	0.755
	PubMedBERT	**0.885**	**0.585**	0.568	0.602	**0.597**	**0.857**
	XLM-RoBERTa	0.818	0.212	0.243	0.189	0.159	0.653
fi	BERT	**0.674**	**0.303**	0.560	0.208	**0.185**	**0.674**
	SAPBERT-UMLS	0.089	0.163	1.000	0.089	0.077	0.444
	XLM-RoBERTa	0.259	0.174	0.880	0.096	0.089	0.525
es	BERT	**0.659**	**0.326**	0.654	0.217	**0.258**	**0.718**
	SAPBERT-UMLS	0.083	0.152	0.998	0.083	0.083	0.487
	XLM-RoBERTa	0.404	0.180	0.792	0.102	0.106	0.588

Table 6.2: Final results. The best result in each category is highlighted in bold, the two best in English due to the number of models used. Precision and recall are not highlighted, as the goal of the threshold selection was to optimise their combination, not their values separately. AUC-based metrics are calculated from the uncertainty scores, as they do not require threshold selection.

6.1.2 Results

Final results on the test set are presented in Table 6.2. Overall, the obtained results are not satisfactory and show that the pipeline is not able to redact the health records of MEDDOCAN adequately compared to the related work presented in Chapter 4. The ROC-AUC metrics show values under 0.5 for some models, meaning these predictions perform worse than a random guess. Out of the English models, `BioClin-BERT` and `PubMedBERT` receive ROC-AUC scores over 0.8, showing acceptable but not stellar performance. Likewise, the highest F1-scores achieved in this experiment barely manage to show values over 0.5, which is higher than a random-guess baseline for data with this class distribution (PII makes up a small percentage of each document, thus the distribution of labels is uneven); yet, these values cannot be considered indicative of good performance. The same applies to the PR-AUC score, with `PubMedBERT` showing the best performance out of all the models and the second-best results by `Dil-BERT`.

The models do show great variation, and evidently, the health domain models outperform general domain models. The best overall scores are achieved by using `PubMedBERT` on English data. This model achieves an F1-score of 0.585 on the test set, the highest of all models. This model also dominates the PR-AUC and ROC-AUC metrics, and is the only one to break the 0.5 threshold on PR-AUC and has a high precision, meaning the model's uncertainty about a prediction does correlate with the likelihood of the word being PII, which is the ideal result. The second-best model, similarly only available for English, is the `Dil-BERT` model, scoring similarly on the F1-score metric but falling off slightly on the AUC metrics. The second-highest ROC-AUC is obtained using the `BioClin-BERT` model; however,

this model struggles with precision more than Dil-BERT and PubMedBERT, despite being the only model trained on clinical data. On the English test data, the worst out of all models performs SAPBERT-UMLS.

Finnish and Spanish mirrored each other in the development set, and the same can be seen on the test set. The BERT models obtain the highest scores, with the other two models receiving low scores. SAPBERT-UMLS model notably receives a ROC-AUC-score of less than 0.5 for both languages. Unfortunately, as SAPBERT-UMLS was the only health domain model tested for Finnish and Spanish, and it showed low performance on English as well, conclusions about the health record aspect of Finnish and Spanish PII redaction remain somewhat undetermined.

6.2 Manual evaluation

Examples of a redacted test set instance for PubMedBERT, Dil-BERT, BERT, and SAPBERT-UMLS can be found in Figures 6.14 – 6.17. Spaces in the middle of words show token borders. From these figures, it can clearly be seen that PubMedBERT and Dil-BERT perform much better than BERT and SAPBERT-UMLS. Nevertheless, despite the model choice, using the redaction pipeline leads to false predictions: all models result in a number of both false negatives and false positives.

PubMedBERT and Dil-BERT figures feature many fewer false predictions than the other two models, which is to be expected from the numerical analysis. SAPBERT-UMLS model, which had the lowest scores overall, and notably ROC-AUC score below 0.5, has a substantially higher number of false positives than the other models. All models struggle with some numerical types of PII, such as ages and

dates, which coincides with the results by Vakili et al. [3]. Still, long strings of numbers are masked consistently between the models; this is likely due to the score aggregation strategy failing with numbers, as usually, the start of a number string does not imply anything about its end, leading to low scores and thus redaction. The models reliably detect names and emails, with the exception of `SAPBERT-UMLS`. They also struggle to different degrees with the list-like start of the file, with `Dil-BERT` and `SAPBERT-UMLS` redacting the word "surname". All models also display false positive predictions that target healthcare vocabulary, like "haemangioma", which is redacted incorrectly by all models. `PubMedBERT` and `Dil-BERT` redact much less healthcare-related vocabulary than the two other models, which is expected and suggests that healthcare-related training is necessary for this task.

The informational value-based approach of this pipeline is visible on the false negatives, like in the word "Spain", which is unredacted by 3 of the 4 models. Spain, as a country, has a smaller informational value, both in the models' representations based on their training data, and in the sense of personal information: linking sensitive information to a person is much harder if the leaked information contains information about a country, compared to information about an address or a city.

Despite the striking differences between the models, all of them redact essential information. These kinds of errors limit the usability of these documents later. These redacted words are critical to retain, as without them, the value of these documents is lower. A high number of false positives also limits the feasibility of using the substitution method, as it would lead to medical vocabulary being changed, and thus would most likely change the health narrative of the document.

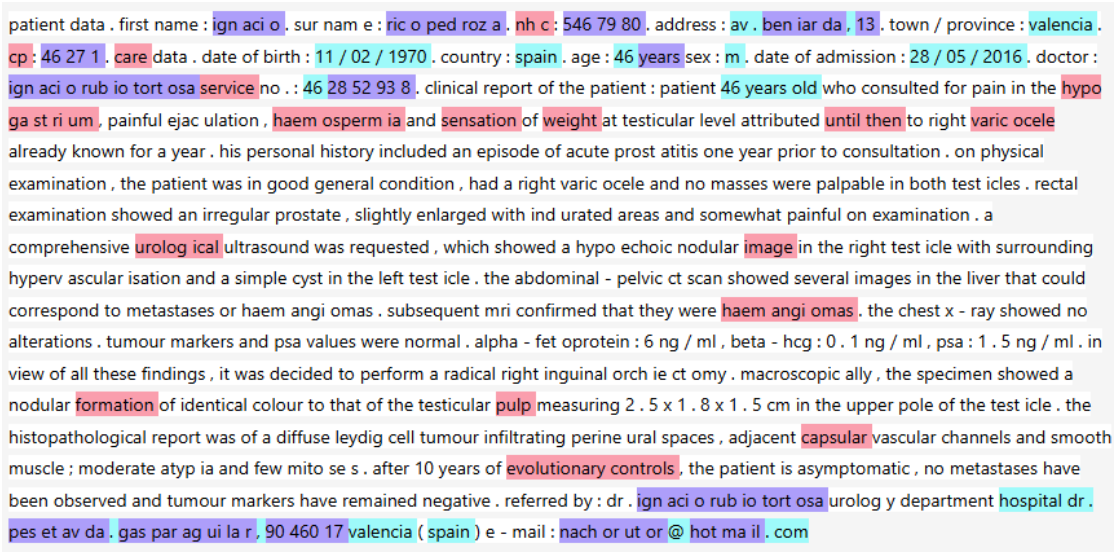

Figure 6.14: Visualisation of the redaction predictions for English and `PubMedBERT`. Red colour indicates false positives, cyan indicates false negatives, and violet indicates true positives.

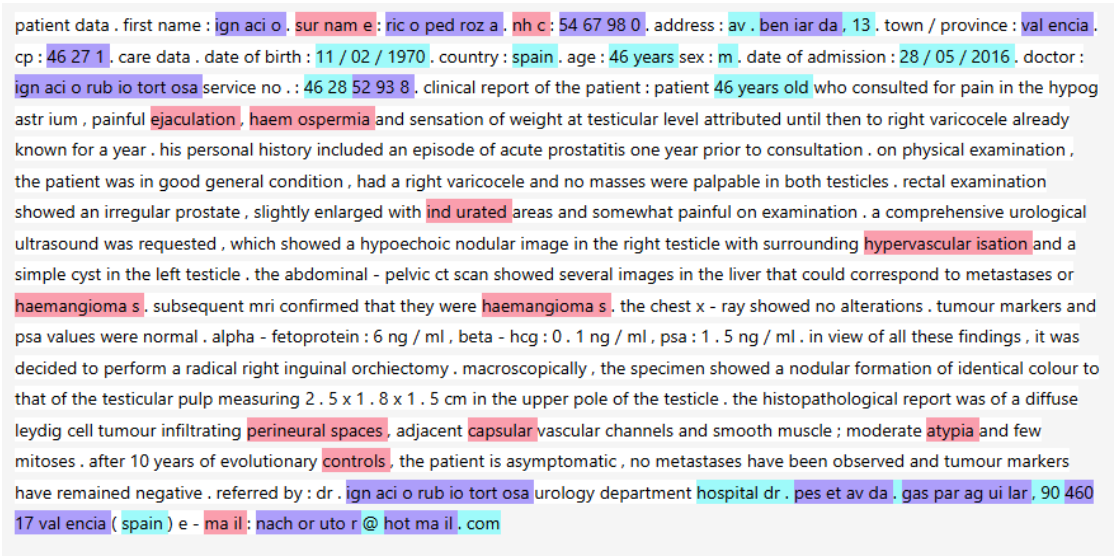

Figure 6.15: Visualisation of the redaction predictions for English and `Dil-BERT`. Red colour indicates false positives, cyan indicates false negatives, and violet indicates true positives.

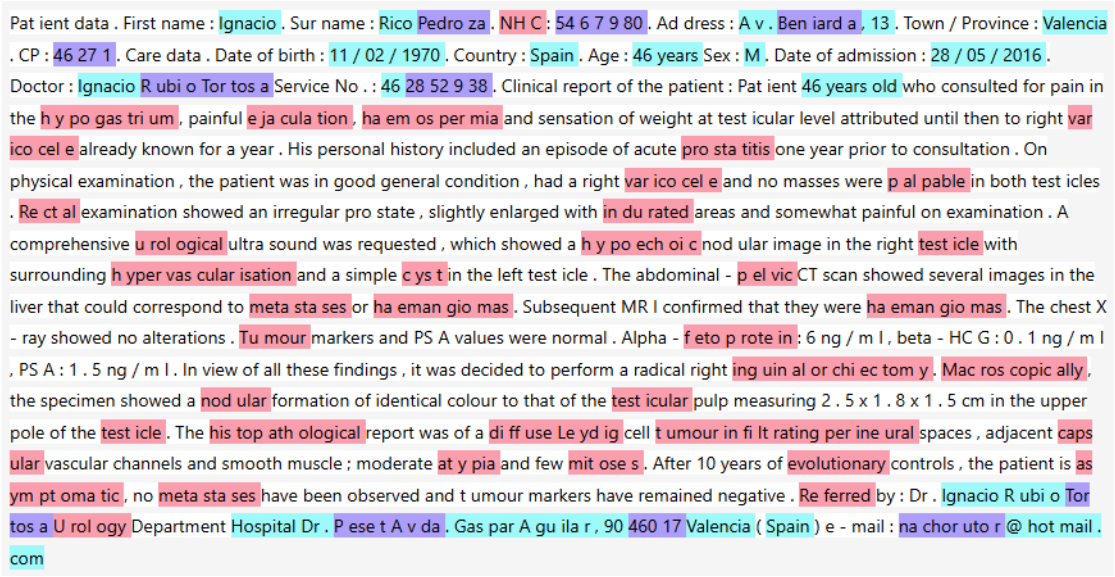

Figure 6.16: Visualisation of the redaction predictions for English and `BERT`. Red colour indicates false positives, cyan indicates false negatives, and violet indicates true positives.

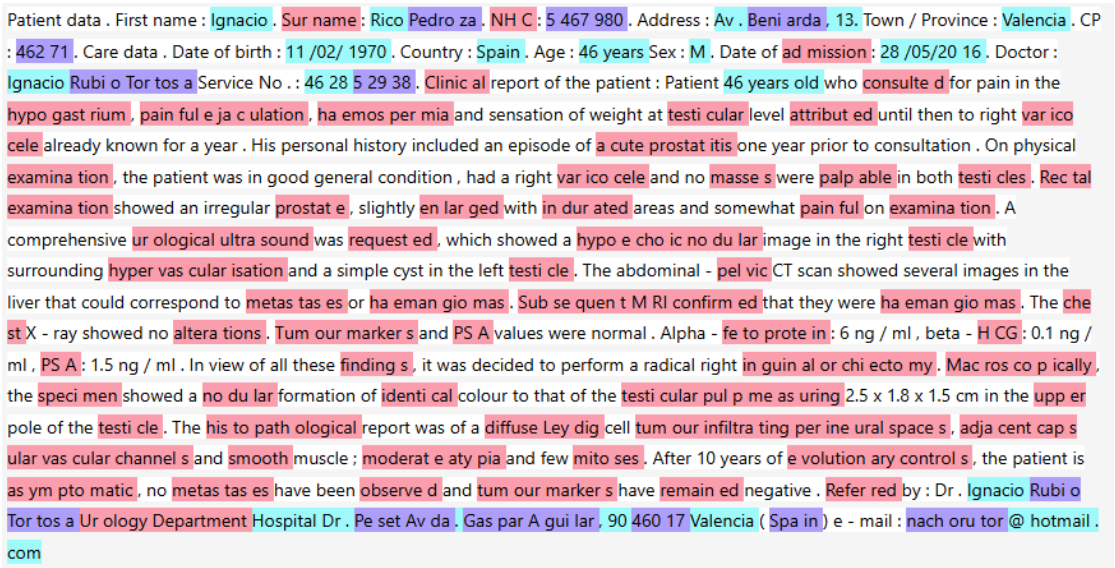

Figure 6.17: Visualisation of the redaction predictions for English and `SAPBERT-UMLS`. Red colour indicates false positives, cyan indicates false negatives, and violet indicates true positives.

7 Discussion and Conclusions

The conducted experiments show that zero-shot redaction of PII in electronic health records is not reliably possible with the tested models using the zero-shot redaction method. Despite this, the models exhibited substantial variation in performance and in general, models with health-related training performed better than those with general-domain training. This means that while the answer to the first research question is that this method cannot be reliably used as-is to redact PII in electronic health records, these experiments show that model selection plays an integral part in the performance, and further research in the model selection can be conducted in the future.

The best models found were the `PubMedBERT` and `Dil-BERT`models, both with full pre-training containing health-related data. The `Dil-BERT`model, trained on disease-related data, slightly underperformed the `PubMedBERT` model, which was trained on medical research data. The `BioClin-BERT` model, trained on clinical data, closer in domain to electronic health records compared to medical research data, performed worse than `PubMedBERT`and arguably `Dil-BERT`. This means that for the zero-shot redaction pipeline, model selection cannot only rely on the knowledge of the domain of the model's training data.

The implementation of the pipeline showed that modifications in the masking step were needed for Finnish and for the pipeline overall, as the context-aware masking was not described by Albanese et al. [2]. Specifically, Finnish required a major change to the masking. This means that the partial answer to the second research question is that synthetic languages did require changes to the setup. Since the limited availability of Finnish, Spanish or multilingual health-domain models affected the experiments, it cannot be said with certainty that the translation of the document did not affect the results. If the experiment were able to be conducted on a native Spanish model and data, there is a chance the experiments would yield different results. Additionally, the untranslated names may affect the results for Finnish, as Finnish textual data rarely contains Spanish names. This problem can affect English to a degree as well, however, as there are prominent Spanish-speaking minorities in the US, English models very likely contain Spanish names in their training data.

At the end of this thesis, the overall results show that he most promising direction in robust PII redaction in health records remains using open-source LLMs in a secure data environment, such as the work of Wiest et al. [7]. While these conditions – open-source models and data security – still partly present as challenges today, progress towards a future with reliable PII redaction is promisingly underway.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning* (Adaptive computation and machine learning). London, England: The MIT Press, 2016, ISBN: 9780262035613.

[2] F. Albanese, D. Ciolek, and N. D'Ippolito, "Text sanitization beyond specific domains: Zero-shot redaction & substitution with large language models", 2023. DOI: https://doi.org/10.48550/arXiv.2311.10785. arXiv: 2311.10785 [cs.CL].

[3] T. Vakili, A. Lamproudis, A. Henriksson, and H. Dalianis, "Downstream task performance of BERT models pre-trained using automatically de-identified clinical data", in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, N. Calzolari, F. Béchet, P. Blache, *et al.*, Eds., Marseille, France: European Language Resources Association, Jun. 2022, pp. 4245–4252. [Online]. Available: https://aclanthology.org/2022.lrec-1.451.

[4] H. Berg, A. Henriksson, and H. Dalianis, "The impact of de-identification on downstream named entity recognition in clinical text", in *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, E. Holderness, A. Jimeno Yepes, A. Lavelli, A.-L. Minard, J.

Pustejovsky, and F. Rinaldi, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 1–11. DOI: 10.18653/v1/2020.louhi-1.1.

[5] C. Lothritz, B. Lebichot, K. Allix, *et al.*, "Evaluating the impact of text de-identification on downstream NLP tasks", in *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, T. Alumäe and M. Fishel, Eds., Tórshavn, Faroe Islands: University of Tartu Library, May 2023, pp. 10–16. [Online]. Available: https://aclanthology.org/2023.nodalida-1.2.

[6] M.-W. Chang, L. Ratinov, D. Roth, and V. Srikumar, "Importance of semantic representation: Dataless classification", in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, ser. AAAI'08, Chicago, Illinois: AAAI Press, 2008, pp. 830–835, ISBN: 9781577353683. [Online]. Available: https://dl.acm.org/doi/10.5555/1620163.1620201.

[7] I. Wiest, M.-E. Leßmann, F. Wolf, *et al.*, "Deidentifying medical documents with local, privacy-preserving large language models: The LLM-Anonymizer", *NEJM AI*, vol. 2, Mar. 2025. DOI: 10.1056/AIdbp2400537.

[8] J. B. Lovins, "Development of a stemming algorithm", *Mechanical Translation and Computational Linguistics*, vol. 11, no. 1-2, pp. 22–31, 1968.

[9] A. Gesmundo and T. Samardžić, "Lemmatisation as a tagging task", in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, H. Li, C.-Y. Lin, M. Osborne, G. G. Lee, and J. C. Park, Eds., Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 368–372. [Online]. Available: https://aclanthology.org/P12-2072/.

[10] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units", in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, K. Erk and N. A. Smith, Eds., Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. DOI: 10.18653/v1/P16-1162.

[11] V. Zouhar, C. Meister, J. Gastaldi, *et al.*, "A formal perspective on byte-pair encoding", in *Findings of the Association for Computational Linguistics: ACL 2023*, A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds., Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 598–614. DOI: 10.18653/v1/2023.findings-acl.38.

[12] Y. Wu, M. Schuster, Z. Chen, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation", 2016. DOI: 10.48550/arXiv.1609.08144. arXiv: 1609.08144 [cs.CL].

[13] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing", in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, E. Blanco and W. Lu, Eds., Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. DOI: 10.18653/v1/D18-2012.

[14] S. Marsland, *Machine learning : An algorithmic perspective*, 2nd ed. CRC Press LLC, 2014.

[15] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT Press, 2012.

[16] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning", *Sci Rep.*, vol. 14(1):6086, 2024. DOI: 10.1038/s41598-024-56706-x.

[17] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves", in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 233–240, ISBN: 1595933832. DOI: 10.1145/1143844.1143874.

[18] H. Steck, C. Ekanadham, and N. Kallus, "Is cosine-similarity of embeddings really about similarity?", in *Companion Proceedings of the ACM Web Conference 2024*, ser. WWW '24, Singapore, Singapore: Association for Computing Machinery, 2024, pp. 887–890, ISBN: 9798400701726. DOI: 10.1145/3589335.3651526.

[19] W. Timkey and M. van Schijndel, "All bark and no bite: Rogue dimensions in transformer language models obscure representational quality", in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds., Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4527–4546. DOI: 10.18653/v1/2021.emnlp-main.372.

[20] J. Hoffmann, S. Borgeaud, A. Mensch, *et al.*, "Training compute-optimal large language models", in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22, New Orleans, LA, USA: Curran Associates Inc., 2022, ISBN: 9781713871088.

[21] J. Wei, Y. Tay, R. Bommasani, *et al.*, "Emergent abilities of large language models", *Transactions on Machine Learning Research*, 2022. DOI: 10.48550/arXiv.2206.07682.

[22] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need", in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[23] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks", *Signal Processing, IEEE Transactions on*, vol. 45, pp. 2673–2681, Dec. 1997. DOI: 10.1109/78.650093.

[24] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks", in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14, Montreal, Canada: MIT Press, 2014, pp. 3104–3112. DOI: 10.48550/arXiv.1409.3215.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding", in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[26] A. Conneau, K. Khandelwal, N. Goyal, *et al.*, "Unsupervised cross-lingual representation learning at scale", in *Proceedings of the 58th Annual Meeting*

of the Association for Computational Linguistics, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747.

[27] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners", 2019. [Online]. Available: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[28] C. Raffel, N. Shazeer, A. Roberts, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer", *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020, ISSN: 1532-4435. [Online]. Available: http://jmlr.org/papers/v21/20-074.html.

[29] M. Geva, A. Caciularu, K. Wang, and Y. Goldberg, "Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space", in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 30–45. DOI: 10.18653/v1/2022.emnlp-main.3.

[30] G. Penedo, H. Kydlíček, L. B. Allal, *et al.*, "The FineWeb datasets: Decanting the web for the finest text data at scale", in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, *et al.*, Eds., vol. 37, Curran Associates, Inc., 2024, pp. 30 811–30 849. DOI: 10.48550/arXiv.2406.17557. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/370df50ccfdf8bde18f8f9c2d9151bda-Paper-Datasets_and_Benchmarks_Track.pdf.

[31] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification", in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, I. Gurevych and Y. Miyao, Eds., Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 328–339. DOI: 10.18653/v1/P18-1031.

[32] S. Gururangan, A. Marasović, S. Swayamdipta, *et al.*, "Don't stop pretraining: Adapt language models to domains and tasks", in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Online: Association for Computational Linguistics, Jul. 2020. DOI: 10.18653/v1/2020.acl-main.740.

[33] O. Vasilyev, R. Sawaya, and J. Bohannon, "Preserving multilingual quality while tuning query encoder on English only", in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, L. Chiruzzo, A. Ritter, and L. Wang, Eds., Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 321–341, ISBN: 979-8-89176-190-2. DOI: 10.18653/v1/2025.naacl-short.28. [Online]. Available: https://aclanthology.org/2025.naacl-short.28/.

[34] E. Henriksson, A. Myntti, S. Hellström, A. Eskelinen, S. Erten-Johansson, and V. Laippala, *Automatic register identification for the open web using multilingual deep learning*, 2024. DOI: 10.48550/arXiv.2406.19892. arXiv: 2406.19892 [cs.CL].

[35] Y. Zhao, W. Zhang, G. Chen, K. Kawaguchi, and L. Bing, "How do large language models handle multilingualism?", in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, *et al.*, Eds., vol. 37, Curran Associates, Inc., 2024, pp. 15 296–15 319. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/1bd359b32ab8b2a6bbafa1ed2856cf40-Paper-Conference.pdf.

[36] T. Chang, Z. Tu, and B. Bergen, "The geometry of multilingual language model representations", in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 119–136. DOI: 10.18653/v1/2022.emnlp-main.9.

[37] P. M. Schwartz and D. J. Solove, *The PII problem: Privacy and a new concept of personally identifiable information*, UC Berkeley Public Law Research Paper No. 1909366, GWU Legal Studies Research Paper No. 584, GWU Law School Public Law Research Paper No. 584, Dec. 2011. [Online]. Available: https://ssrn.com/abstract=1909366.

[38] R. Li, L. B. allal, Y. Zi, *et al.*, "StarCoder: May the source be with you!", *Transactions on Machine Learning Research*, 2023, ISSN: 2835-8856. DOI: 10.48550/arXiv.2305.06161.

[39] Oikeusministeriö (Ministry of Justice of Finland), *Tietosuojalaki (1050/2018)*, Accessed 23.06.2025. [Online]. Available: https://finlex.fi/fi/lainsaadanto/2018/1050.

[40] A. Narayanan and V. Shmatikov, "Myths and fallacies of 'personally identifiable information'", *Commun. ACM*, vol. 53, no. 6, pp. 24–26, Jun. 2010, ISSN: 0001-0782. DOI: 10.1145/1743546.1743558.

[41] Z. Stein, "Privacy in public archives: Managing personally identifiable information in special collections", *RBM: A Journal of Rare Books, Manuscripts, and Cultural Heritage*, vol. 22, no. 2, p. 85, 2021, ISSN: 2150-668X. DOI: 10.5860/rbm.22.2.85.

[42] J. Yang, X. Zhang, K. Liang, and Y. Liu, "Exploring the application of large language models in detecting and protecting personally identifiable information in archival data: A comprehensive study*", in *2023 IEEE International Conference on Big Data (BigData)*, 2023, pp. 2116–2123. DOI: 10.1109/BigData59044.2023.10386949.

[43] S. Madan, M. Lentzen, J. Brandt, D. Rueckert, M. Hofmann-Apitius, and H. Fröhlich, "Transformer models in biomedicine", *BMC medical informatics and decision making*, vol. 24(1), 214, 2024. DOI: https://doi.org/10.1186/s12911-024-02600-5.

[44] C. Mansfield, A. Paullada, and K. Howell, "Behind the mask: Demographic bias in name detection for PII masking", in *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, B. R. Chakravarthi, B. Bharathi, J. P. McCrae, M. Zarrouk, K. Bali, and P. Buitelaar, Eds., Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 76–89. DOI: 10.18653/v1/2022.ltedi-1.10.

[45] A. Wei, N. Haghtalab, and J. Steinhardt, *Jailbroken: How does LLM safety training fail?*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt,

and S. Levine, Eds., 2023. DOI: 10.48550/arXiv.2307.02483. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/fd6613131889a4b656206c50a8bd7790-Paper-Conference.pdf.

[46] N. Carlini, F. Tramèr, E. Wallace, *et al.*, "Extracting training data from large language models", in *30th USENIX Security Symposium (USENIX Security 21)*, USENIX Association, Aug. 2021, pp. 2633–2650, ISBN: 978-1-939133-24-3. [Online]. Available: https://www.usenix.org/system/files/sec21-carlini-extracting.pdf.

[47] H. Li, D. Guo, W. Fan, *et al.*, "Multi-step jailbreaking privacy attacks on ChatGPT", in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 4138–4153. DOI: 10.18653/v1/2023.findings-emnlp.272.

[48] European Union, *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act)*, Accessed 23.06.2025. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng.

[49] European Data Protection Board, *Opinion 28/2024 on certain data protection aspects related to the processing of personal data in the context of AI models*, Accessed 23.06.2025. [Online]. Available: https://www.edpb.

europa . eu / our - work - tools / our - documents / opinion - board - art - 64/opinion-282024-certain-data-protection-aspects_en.

[50] Oikeusministeriö (Ministry of Justice of Finland), *Laki sosiaali- ja terveydenhuollon asiakastietojen käsittelystä (703/2023)*, Accessed 23.06.2025. [Online]. Available: https://finlex.fi/fi/lainsaadanto/saadoskokoelma/2023/703.

[51] Oikeusministeriö (Ministry of Justice of Finland), *Sosiaali- ja terveysministeriön asetus sosiaali- ja terveydenhuollon asiakastietojen käsittelystä (457/2024)*, Accessed 23.06.2025. [Online]. Available: https://www.finlex.fi/fi/lainsaadanto/saadoskokoelma/2024/457.

[52] Oikeusministeriö (Ministry of Justice of Finland), *Laki terveydenhuollon ammattihenkilöistä (559/1994)*, Accessed 23.06.2025. [Online]. Available: https://www.finlex.fi/fi/lainsaadanto/1994/559.

[53] Oikeusministeriö (Ministry of Justice of Finland), *Laki potilaan asemasta ja oikeuksista (785/1992)*, Accessed 23.06.2025. [Online]. Available: https://finlex.fi/fi/lainsaadanto/1992/785.

[54] Oikeusministeriö (Ministry of Justice of Finland), *Laki sosiaali- ja terveystietojen toissijaisesta käytöstä (552/2019)*, Accessed 23.06.2025. [Online]. Available: https://www.finlex.fi/fi/lainsaadanto/saadoskokoelma/2019/552.

[55] European Union, *Regulation (EU) 2025/327 of the European Parliament and of the Council of 11 February 2025 on the European Health Data Space and amending Directive 2011/24/EU and Regulation (EU) 2024/2847*, Accessed

23.06.2025. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2025/327/oj/eng.

[56] European Union, *Regulation (EU) 2023/2854 of the European Parliament and of the Council of 13 December 2023 on harmonised rules on fair access to and use of data and amending Regulation (EU) 2017/2394 and Directive (EU) 2020/1828 (Data Act)*, Accessed 23.06.2025. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2023/2854/oj/eng.

[57] Oikeusministeriö (Ministry of Justice of Finland), *Laki julkisen hallinnon tiedonhallinnasta (906/2019)*, Accessed 23.06.2025. [Online]. Available: https://www.finlex.fi/fi/lainsaadanto/saadoskokoelma/2019/906.

[58] Tietosuojavaltuutettu, *Pseudonymisoidut ja anonymisoidut tiedot*, Accessed 23.06.2025. [Online]. Available: https://tietosuoja.fi/pseudonymisointi-anonymisointi.

[59] Article 29 Working Party, *Opinion 05/2014 on Anonymisation Techniques*, Accessed 23.06.2025. [Online]. Available: https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/index_en.htm#maincontentSec4.

[60] D. S. Carrell, D. J. Cronkite, M. R. Li, *et al.*, "The machine giveth and the machine taketh away: A parrot attack on clinical text deidentified with hiding in plain sight", *Journal of the American Medical Informatics Association*, vol. 26, no. 12, pp. 1536–1544, Dec. 2019. DOI: 10.1093/jamia/ocz114.

[61] D. Sánchez, M. Batet, and A. Viejo, "Automatic general-purpose sanitization of textual documents", *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 853–862, 2013. DOI: 10.1109/TIFS.2013.2239641.

[62] G. Baudart, J. Dolby, E. Duesterwald, M. Hirzel, and A. Shinnar, "Protecting chatbots from toxic content", in *Proceedings of the 2018 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, ser. Onward! 2018, Boston, MA, USA: Association for Computing Machinery, 2018, pp. 99–110, ISBN: 9781450360319. DOI: 10.1145/3276954.3276958.

[63] Y. Sun, S. Wang, Y. Li, *et al.*, "ERNIE 2.0: A continual pre-training framework for language understanding", in *AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 8968–8975. DOI: 0.1609/aaai.v34i05.6428.

[64] T. L. Scao, A. Fan, C. Akiki, *et al.*, *Bloom: A 176B-parameter open-access multilingual language model*, BigScience Workshop, 2023. DOI: 10.48550/arXiv.2211.05100. arXiv: 2211.05100 [cs.CL].

[65] L. B. Allal, R. Li, D. Kocetkov, *et al.*, *Santacoder: Don't reach for the stars!*, 2023. DOI: 10.48550/arXiv.2301.03988. arXiv: 2301.03988 [cs.SE].

[66] L. Soldaini, R. Kinney, A. Bhagia, *et al.*, *Dolma: An open corpus of three trillion tokens for language model pretraining research*, 2024. DOI: 10.48550/arXiv.2402.00159. arXiv: 2402.00159 [cs.CL].

[67] B. Jehangir, S. Radhakrishnan, and R. Agarwal, "A survey on named entity recognition — datasets, tools, and methodologies", *Natural Language Processing Journal*, vol. 3, p. 100 017, 2023, ISSN: 2949-7191. DOI: https://doi.org/10.1016/j.nlp.2023.100017.

[68] R. Grishman and B. Sundheim, "Message Understanding Conference- 6: A brief history", in *COLING 1996 Volume 1: The 16th International Con-*

ference on Computational Linguistics, 1996. [Online]. Available: https://aclanthology.org/C96-1079/.

[69] L. Weston, V. Tshitoyan, J. Dagdelen, *et al.*, "Named entity recognition and normalization applied to large-scale information extraction from the materials science literature", *Journal of Chemical Information and Modeling*, vol. 59, no. 9, pp. 3692–3702, 2019, PMID: 31361962. DOI: 10.1021/acs.jcim.9b00470. eprint: https://doi.org/10.1021/acs.jcim.9b00470.

[70] A. Eskelinen, A. Myntti, E. Henriksson, S. Pyysalo, and V. Laippala, "Building question-answer data using web register identification", in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, Eds., Torino, Italia: ELRA and ICCL, May 2024, pp. 2595–2611. [Online]. Available: https://aclanthology.org/2024.lrec-main.234/.

[71] V. S. Sheng and J. Zhang, "Machine learning with crowdsourcing: A brief summary of the past research and future directions", *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 9837–9843, Jul. 2019. DOI: 10.1609/aaai.v33i01.33019837.

[72] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, *SQuAD: 100,000+ questions for machine comprehension of text*, 2016. DOI: 10.48550/arXiv.1606.05250. arXiv: 1606.05250 [cs.CL].

[73] B. Ding, C. Qin, R. Zhao, *et al.*, "Data augmentation using LLMs: Data perspectives, learning paradigms and challenges", in *Findings of the Association for Computational Linguistics: ACL 2024*, L.-W. Ku, A. Martins,

and V. Srikumar, Eds., Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 1679–1705. DOI: 10.18653/v1/2024.findings-acl.97.

[74] L. Jaillant and A. Rees, "Applying AI to digital archives: Trust, collaboration and shared professional ethics", *Digital Scholarship in the Humanities*, vol. 38, no. 2, pp. 571–585, Nov. 2022, ISSN: 2055-7671. DOI: 10.1093/llc/fqac073. eprint: https://academic.oup.com/dsh/article-pdf/38/2/571/50488277/fqac073.pdf.

[75] R. Kuo, A. Soltan, C. O'Hanlon, *et al.*, "Comparative evaluation of large-language models and purpose-built software for medical record de-identification", Oct. 2024. DOI: 10.21203/rs.3.rs-4870585/v1.

[76] A. Grattafiori, A. Dubey, A. Jauhri, *et al.*, *The llama 3 herd of models*, 2024. DOI: 10.48550/arXiv.2407.21783. arXiv: 2407.21783 [cs.AI].

[77] DeepSeek-AI, A. Liu, B. Feng, *et al.*, *Deepseek-v3 technical report*, 2025. DOI: 10.48550/arXiv.2412.19437. arXiv: 2412.19437 [cs.CL].

[78] M. Marimon, A. Gonzalez-Agirre, A. Intxaurrondo, *et al.*, "Automatic de-identification of medical texts in Spanish: The MEDDOCAN track, corpus, guidelines, methods and evaluation of results.", in *: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019) @ SEPLN*, 2019, pp. 618–638.

[79] J. A. Omiye, H. Gui, S. J. Rezaei, J. Zou, and R. Daneshjou, "Large language models in medicine: The potentials and pitfalls", *Annals of Internal Medicine*, vol. 177, no. 2, pp. 210–220, 2024, PMID: 38285984. DOI: 10.7326/M23-2772. eprint: https://doi.org/10.7326/M23-2772.

[80] O. Bodenreider, "The unified medical language system (UMLS): Integrating biomedical terminology.", *Nucleic Acids Res.*, D267-70 2004. DOI: 10.1093/nar/gkh061.

[81] A. Johnson, T. Pollard, and R. Mark, "MIMIC-III clinical database (version 1.4)", 2016. DOI: https://doi.org/10.13026/C2XW26.

[82] C. P. Carrino, J. Armengol-Estapé, A. Gutiérrez-Fandiño, *et al.*, *Biomedical and clinical language models for Spanish: On the benefits of domain-specific pretraining in a mid-resource scenario*, 2021. DOI: 10.48550/arXiv.2109.03570. arXiv: 2109.03570 [cs.CL].

[83] A. Virtanen, J. Kanerva, R. Ilo, *et al.*, *Multilingual is not enough: BERT for Finnish*, 2019. DOI: 10.48550/arXiv.1912.07076. [Online]. Available: https://arxiv.org/abs/1912.07076.

[84] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez, "Spanish pre-trained BERT model and evaluation data", in *PML4DC at ICLR 2020*, 2020. DOI: 10.48550/arXiv.2308.02976.

[85] E. Alsentzer, J. Murphy, W. Boag, *et al.*, "Publicly available clinical BERT embeddings", in *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, A. Rumshisky, K. Roberts, S. Bethard, and T. Naumann, Eds., Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 72–78. DOI: 10.18653/v1/W19-1909.

[86] K. Roitero, B. Portelli, M. H. Popescu, and V. Della Mea, "DilBERT: Cheap embeddings for disease related medical NLP", *IEEE Access*, vol. 9, pp. 159 714–159 723, 2021. DOI: 10.1109/ACCESS.2021.3131386.

[87] Y. Gu, R. Tinn, H. Cheng, *et al.*, *Domain-specific language model pretraining for biomedical natural language processing*, Oct. 2021. DOI: 10.1145/3458754.

[88] F. Liu, I. Vulić, A. Korhonen, and N. Collier, "Learning domain-specialised representations for cross-lingual biomedical entity linking", in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., Online: Association for Computational Linguistics, Aug. 2021, pp. 565–574. DOI: 10.18653/v1/2021.acl-short.72.

[89] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners", in *Advances in Neural Information Processing Systems*, H. Larochelle, *et al.*, Ed., vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[90] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength natural language processing in Python", 2020. DOI: 10.5281/zenodo.1212303.

[91] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.