HIPAAway: developing software for de-identification and exploring bias in name detection

by

Shulammite Lim

S.B. in Computer Science and Molecular Biology and in Music
Massachusetts Institute of Technology (2022)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Molecular Biology

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by:	Shulammite Lim Department of Electrical Engineering and Computer Science May 19, 2023
Certified by:	Tom Pollard Research Scientist Thesis Supervisor
Certified by:	Roger Mark Distinguished Professor Thesis Supervisor
Accepted by:	Katrina LaCurts Chair, Master of Engineering Thesis Committee

HIPAAway: developing software for de-identification and exploring bias in name detection

by

Shulammite Lim

Submitted to the Department of Electrical Engineering and Computer Science
on May 19, 2023, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Molecular Biology

Abstract

De-identification, the process of removing identifiers, is a crucial step in the preparation of clinical data for use in biomedical research. Advances in natural language processing have increased interest in developing an accurate and adaptable automatic de-identification system for clinical text. Models for de-identification have been found successful but are largely unavailable for public use due to a lack of provided code and a cost associated with using commercial models. A lack of transparency in de-identification model training may bias the models against certain demographic groups, which are hidden in overall performance metrics and need to be evaluated due to the disproportionate potential harm to marginalized communities. In this thesis, we review current de-identification methods, present a new de-identification dataset, audit demographic biases in existing de-identification approaches, and develop an easy-to-use, open-source de-identification software package. This package would make clinical text de-identification more accessible to researchers and clinicians, alleviating the bottleneck of de-identification to free up more data for biomedical research. This would help make future research more robust and beneficial to not only the medical community, but also people around the world.

Thesis Supervisor: Tom Pollard
Title: Research Scientist

Thesis Supervisor: Roger Mark
Title: Distinguished Professor

Acknowledgments

I am deeply indebted to my direct thesis supervisor, Tom Pollard, for his invaluable patience, feedback, and ideas throughout the course of not only this thesis, but also the majority of my undergraduate years. He introduced me to both web development and natural language processing in the context of health data, now both strong interests I hope to continue to pursue in the future. I also would like to express my deepest gratitude to my thesis advisor, Professor Roger Mark, who kindly offered me the opportunity to join the Laboratory for Computational Physiology (LCP) three years ago and has been instrumental in guiding and supporting my work. Additionally, I am extremely grateful to Professor Leo Anthony Celi for his mentorship, encouragement, and numerous opportunities to learn about the intersection of medicine and computer science as a teaching assistant for the HST.936 course. This thesis work would not have been possible without generous financial support from Professor Mark, Professor Celi, and LCP.

Special thanks to Yuxin Xiao for his extensive collaboration on the bias audit and to Professor Marzyeh Ghassemi for her substantial guidance and support. Many thanks to all the members of LCP for their support and advice, especially to Alistair Johnson, Dana Moukheiber, Lama Moukheiber, and Mira Moukheiber for their crucial help with the de-identification dataset. I am also thankful to Eric Lehman and the Huggingface team for their technical help.

I am grateful to MIT and particularly the EECS department for providing me with the opportunity to pursue this thesis work and for the many resources and opportunities they have provided me with over the past five years. I am also thankful to my friends and fellow MEng students for their camaraderie and moral support.

Lastly, with deep appreciation I'd like to acknowledge my family. Their belief in me and unwavering support have kept my happiness and motivation high during this process.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Aims

In this thesis, I aim to (1) review the state of the art in de-identification of clinical text; (2) carry out an audit of existing de-identification tools with a particular focus on algorithmic fairness; and (3) develop easy-to-use, open-source software for detection and removal of protected health information from clinical text.

1.2 De-identification: a bottleneck in data sharing

Electronic health records (EHRs) contain a wealth of information that is of strong interest to medical researchers for developing insights and algorithms to assess and improve patient care. In particular, free-text medical records have been found to hold key insights absent from their more rigid, structured counterparts [55, 48]. To conduct meaningful research beyond the bedside, sharing patient data between clinicians and researchers is crucial. To preserve patient privacy, this sharing of data is often contingent on de-identification, which is the process of removing identifiers such as names, contact information, and dates to protect patient privacy. If the data has not undergone de-identification, it is severely limited in terms of how it may be shared and reused [53].

De-identification presents many challenges, however, due to scarce high-quality annotated data for developing models, as well as the highly heterogeneous nature of target identifiers such as patient names. To add to the difficulty, models that achieve anything less than perfect sensitivity may be considered a failure, as any missed private health information could put individuals at risk of harm [20]. This results in the frequent withholding of patient data when sharing would benefit research. Even past work in de-identification has involved training models on datasets that remain internal, exemplifying a lack of transparency that has proven to be common across other areas of health research [7].

In addition to the above challenges, de-identification can also be prone to bias. For example, if a model is trained on a dataset that is not representative of the population, it may be more likely to miss identifiers that are more common in underrepresented groups. This could lead to a higher risk of data disclosure for these groups. Easy-to-use, transparent, and fair de-identification would thus be a useful tool in the research arsenal to make biomedical data more widely available. Increasing the amount of data available through de-identification would power more research and eventually benefit both healthcare providers and patients [23].

1.3 Health Insurance Portability and Accountability Act

Patient health data is protected under different standards depending on the jurisdiction: GDPR (General Data Protection Regulation) in the European Union, PHIPA (Personal Health Information Protection Act) in Canada, and Health Insurance Portability and Accountability Act (HIPAA) in the United States. Some countries, such as Australia, do not have a clear standard for "de-identified" health data, though multiple health-specific privacy laws are in place [44]. This has led to efforts to de-identify Australian health data acording to HIPAA [29]. Many de-identification methods found in literature follow the HIPAA standard, often because of the availability of datasets that have been de-identified according to HIPAA, as well as the organized challenges that invite de-identification solutions [57, 56, 42].

The HIPAA Privacy Rule provides a set of rules for de-identification of protected health information (PHI). Additionally, the Privacy Rule provides two methods of de-identifying health information: Expert Determination and Safe Harbor. Under the Expert Determination method, a person with appropriate knowledge of generally accepted methods for rendering information not individually identifiable applies such methods to the information and determines that the risk of re-identification by an anticipated recipient would be very small.[1] Under the Safe Harbor provision, covered entities need to remove identifiers of the individual or of relatives, employers, or household members of the individual. The Safe Harbor method lists 18 categories of identifiers for removal, including names, all geographic subdividisions smaller than a state, and all elements of dates (except year) directly related to the individual (Table 1.1). This project focuses on de-identification under Safe Harbor, as existing de-identification methods focus on automated approaches to processing text.

1.4 Regulations are not technical specifications

The ambiguity surrounding the definition of PHI poses challenges due to the lack of clearly defined technical specifications. While HIPAA regulations provide guidance on what constitutes PHI, they do not offer specific technical criteria, and thus HIPAA Safe Harbor cannot be considered a technical specification for de-identification. As a result, it may be uncertain whether de-identification methods comply with HIPAA requirements; or, more seriously, data thought to be fully de-identified under the HIPAA requirements retain other information that puts individuals at the risk of re-identification. This uncertainty can hinder the development of standardized de-identification techniques.

Another factor contributing to the complexity of de-identification is the existence of varying annotation schema used by dataset annotators. This variability not only

[1] https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html

Table 1.1: List of HIPAA-defined protected health identifiers (PHI) for removal under Safe Harbor method. Definitions are taken from the U.S. Department of Health and Human Services.

The following identifiers of the individual or of relatives, employers, or household members of the individual, are removed:
(A) Names
(B) All geographic subdivisions smaller than a state, including street address, city, county, precinct, ZIP code, and their equivalent geocodes, except for the initial three digits of the ZIP code if, according to the current publicly available data from the Bureau of the Census: (1) The geographic unit formed by combining all ZIP codes with the same three initial digits contains more than 20,000 people; and (2) The initial three digits of a ZIP code for all such geographic units containing 20,000 or fewer people is changed to 000
(C) All elements of dates (except year) for dates that are directly related to an individual, including birth date, admission date, discharge date, death date, and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older
(D) Telephone numbers
(E) Fax numbers
(F) Email addresses
(G) Social security numbers
(H) Medical record numbers
(I) Health plan beneficiary numbers
(J) Account numbers
(K) Certificate/license numbers
(L) Vehicle identifiers and serial numbers, including license plate numbers
(M) Device identifiers and serial numbers
(N) Web Universal Resource Locators (URLs)
(O) Internet Protocol (IP) addresses
(P) Biometric identifiers, including finger and voice prints
(Q) Full-face photographs and any comparable images
(R) Any other unique identifying number, characteristic, or code

Table 1.2: Different possible interpretations of HIPAA PHI categories. Interpretation #1 presents one option, and Interpretation #2 presents a stricter option. Adapted from [10].

PHI type	**Interpretation #1**	**Interpretation #2 (strict)**
Name	Reason for visit: Mr. **Doe** comes to follow up on his prediabetes.	Reason for visit: **Mr. Doe** comes to follow up on his prediabetes.
Location	She goes to the library twice a week.	She goes to the **library** twice a week.
Profession	She is hoping to intern this summer.	She is hoping to **intern** this summer.

leads to differences in the tools trained on these datasets but also creates difficulties in comparing the performance of de-identification models across different datasets. For instance, HIPAA considers only ages over 89 as PHI, while the 2006 i2b2 de-identification corpus treats all ages as PHI [60]. Models trained to only recognize ages over 89 would not be able to identify ages below 89 as PHI in the i2b2 corpus, though they may have higher performance metrics on their original dataset because of the smaller ranges of ages to identify.

Furthermore, there remains much ambiguity regarding specific elements within the 18 categories of identifiers that should be considered as part of PHI and removed. For example, the inclusion or exclusion of salutations (e.g., "Mr.", "Ms.", "Dr.") or suffixes (e.g., "Jr.", "Sr.") in names can lead to variation in different dataset annotations. This variation in annotation practices extends to other categories of identifiers as well. Table 1.2 presents a reasonable interpretation of HIPAA PHI, as well as a stricter interpretation, adapted from missed PHI in [10]. While the stricter interpretation may be considered safer by removing entities like "grocery store" and "intern," these entities are fairly general. Their removal may be not only unnecessary but also detrimental to the quality of the data, as the text may lose helpful context.

In conclusion, the lack of a technical specification for de-identification has led to ambiguity in defining PHI, which in turn leads to variation in annotation practices and the development of de-identification models.

1.5 Beyond the discovery of HIPAA identifiers

The process of de-identification goes beyond the identification of HIPAA identifiers. Once the PHI have been identified, the next step involves their removal or replacement, commonly known as scrubbing. However, this task is non-trivial and poses several challenges.

The approach employed by the Medical Information Mart for Intensive Care (MIMIC) dataset [22] involves three main methods: redaction, replacement, and perturbation. Redaction refers to the removal of PHI, leaving blank spaces or other markers in their place. Replacement involves substituting the PHI with surrogates, such as randomly generated phone numbers or names. Perturbation, on the other hand, entails modifying the PHI in some way, such as by adding or subtracting a random value to dates or numerical identifiers.

Figure 1-1 in the thesis provides examples of how these scrubbing methods can be applied to different categories of PHI. Replacing the identifiers with surrogates may offer a more privacy-preserving approach, as undetected PHI can "hide" in plain sight, as redaction can potentially reveal the presence of remaining undetected PHI. Care must be exercised when perturbing values to avoid unintentionally disclosing information.

Automating the end-to-end identification process poses additional challenges, as highlighted by Yogarajan et al. (2020) [68]. Surrogate generation and replacement are particularly complex tasks. An example of a challenge in replacement is the generation of a surrogate for a location, such as a zip code. When a zip code is replaced by a random zip code, even if the zip code is from a list of valid zip codes, useful geographic information may be lost. For example, a zip code in a rural area may be replaced by a zip code in an urban area, which may lead to a loss of relevant information, such as expected living conditions, life expectancy, and healthcare access. On the other hand, replacing a zip code with a zip code in the same area may lead to re-identification, especially if there are fewer people living in that area.

While perturbation can address some issues of information loss, it is not uni-

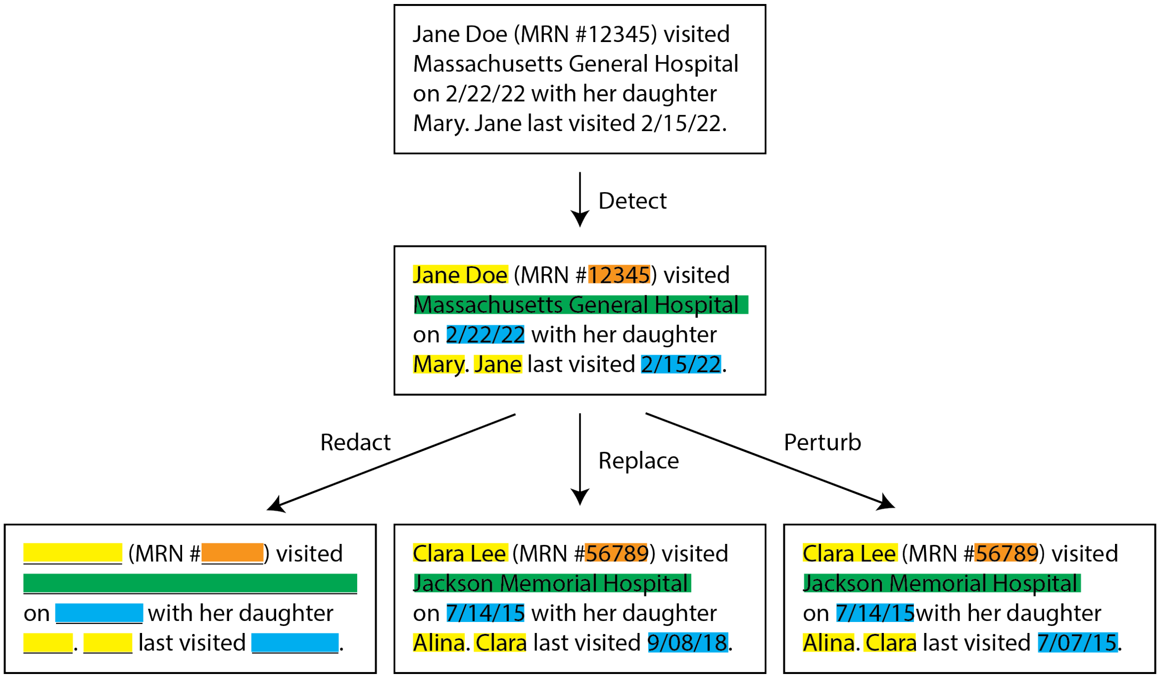

Jane Doe (MRN #12345) visited
Massachusetts General Hospital
on 2/22/22 with her daughter
Mary. Jane last visited 2/15/22.
Detect
Jane Doe (MRN #12345) visited
Massachusetts General Hospital
on 2/22/22 with her daughter
Mary. Jane last visited 2/15/22.
Redact
Replace
Perturb
(MRN #) visited
on with her daughter
. last visited .
Clara Lee (MRN #56789) visited
Jackson Memorial Hospital
on 7/14/15 with her daughter
Alina. Clara last visited 9/08/18.
Clara Lee (MRN #56789) visited
Jackson Memorial Hospital
on 7/14/15with her daughter
Alina. Clara last visited 7/07/15.

Figure 1-1: Different methods of scrubbing PHI. Given a note chunk, the PHI is first detected and then scrubbed via one of the following methods: redact, replace, or perturb. Note: perturbation is not applicable to all PHI categories; here, it is only shown for dates.

versally applicable to all categories of PHI. For example, perturbing names is not a viable option. However, perturbation can be effective for dates, allowing for the preservation of time intervals between events while obscuring the exact dates. Careful consideration is required to define "valid" dates, including considerations such as the treatment of February 29 in non-leap years or whether dates falling on weekends should be deemed valid. These decisions must be made with caution to ensure the integrity of the de-identified data.

1.6 Algorithmic fairness

There is growing concern about the underlying mechanisms by which machine learning models make decisions, particularly in light of the many forms of discrimination present today. One area in which a growing body of research has demonstrated the presence of bias, potentially with negative consequences, is natural language processing. Recent studies have examined how human biases like stereotypes may be reflected in semantic representations (abstract representations of words) like word embeddings, which are word representations for text where words with the same meaning (e.g., "happy" and "cheerful") have similar representations. Researchers have found that applying machine learning to ordinary human language leads to human-like semantic biases. The contexts for these biases range from relatively harmless, like preferences between flowers and insects, to potentially problematic, such as stereotypes for race and gender [6].

Recent techniques to facilitate the de-identification process have focused on computational and machine learning approaches, but current de-identification tools in use have variable performance, and they may be prone to bias that reflects the skewed demographics of relatively scarce training data. Bias can occur even with well-designed machine learning architectures, which—-while they can learn well—-are only able to learn what is given to them in the form of training data. If the data given to a machine learning model lacks certain ethnic subgroups of patients, it may lead to uninformative predictions for these subgroups, in effect biasing the model against those groups. This can have dire consequences, as a model that is biased against a particular group may lead to a higher risk of data disclosure for that group, which could lead to a higher risk of harm for that group. Little prior work has been done to evaluate de-identification methods for bias, and part of this thesis aims to address this gap.

1.7 Organization of the thesis

The remainder of this thesis is organized as follows:

- Chapter 2 reviews the state of the art in de-identification models and software.

- Chapter 3 reviews data available for training and evaluating de-identification software and describes the creation of a new clinical notes dataset.

- Chapter 4 presents an audit of existing de-identification software with particular focus on bias.
- Chapter 5 describes the development of HIPAAway, an open-source de-identification package, and evaluates its performance.
- Chapter 6 concludes the thesis and discusses future work.

Chapter 2

Review of existing de-identification software

2.1 Aim

Approaches to de-identification can involve rules (e.g., regular expressions, dictionaries), machine learning, or a combination of the two. This section describes current progress in algorithms for de-identification and available software tools for de-identification.

2.2 Introduction

De-identification approaches aim to identify and remove/replace entities that contain PHI, such as names, dates, and locations. The task of PHI detection can be considered a type of named entity recognition (NER) to identify entities for removal. Each word is a token, and entities can consist of a single token or a string of tokens. In the context of de-identification, entities are PHI (e.g. name, date, location). For example, "Beth Israel Deaconess Medical Center" is a string of five tokens that represents a LOCATION entity. At its core, NER is a two-step process: 1. Detecting a named entity, and 2. Categorizing the entity. While one main goal of de-identification is to detect PHI entities, it is also important to accurately categorize them in order for the appropriate actions to be taken to remove them. For example, NAME entities can be removed or replaced with synthetic names, but they cannot be replaced with a LOCATION entity. As NER approaches become more sophisticated, both detection and classification are improved. This thesis reviews current state-of-the-art NER approaches on de-identification of clinical text.

In order to compare de-identification models, benchmark datasets are needed. The following two datasets are commonly used for evaluating de-identification models and will be discussed in more detail in Chapter 3:

- 2014 Informatics for Integrating Biology & the Bedside (i2b2)/UTHealth dataset
- 2016 Centers of Excellence in Genomic Science (CEGS) and Neuropsychiatric Genome-Scale and RDOC Individualized Domains (N-GRID) de-identification challenge dataset.

2.3 Algorithms for de-identification

2.3.1 Pattern matching algorithms

Initial efforts to facilitate de-identification focused on rule-based approaches, also known as pattern matching. These approaches define rules that characterize whether a token is recognized as sensitive. Rule-based techniques can come in many forms, including regular expressions (sequences of characters that specify a match pattern in text) and lookup tables (hash tables of frequently used terms characterized as sensitive or non-sensitive) [25]. Rule-based systems require minimal training data and can be easily modified to change existing rules or incorporate new ones. In addition, they can help distinguish some ambiguous instances of PHI, such as between a medical record number of "293-46-51-8" and a phone number of "847-156-0392" [31]. However, rules require careful curation by domain experts–people with knowledge of the particular medical domain and with some level of programming skill–and are time-consuming to create. They can also be limited by being too domain-specific for particular note types [31].

MIT Laboratory for Computational Physiology method: *deid* and *pydeid*

Over a decade ago, the MIT Laboratory for Computational Physiology (LCP) developed a Perl-based deidentification software package called *deid*, which contains lists of words and phrases that are or are likely PHI, as well as words and phrases that are not likely to be PHI. After identifying PHI entities, the package then replaces them with their corresponding PHI tag or surrogate PHI-like entities. *deid* achieved a recall of 0.967 and was fine-tuned to deidentify PHI in MIMIC-II nursing notes and discharge summaries [42]. In terms of recall, the software out-performed a single human deidentifier (0.81) and performed at least as well as a consensus of two human deidentifiers (0.84). The algorithm made use of lexical look-up tables, regular expressions, and simple heuristics. One factor of the package's sucess was likely its use of known doctor names from the hospital from which the clinical notes were obtained in a look-up table, an example of use-case specificity that would not port well to other hospitals. The authors concluded that while accuracy was high, the software was probably insufficient for use to publicly disseminate medical data, a limitation reflected by other rule-based approaches.

The Laboratory for Computational Physiology has since then converted *deid* into Python and added new rules in a package called *pydeid*.[1] Similarly to *deid*, *pydeid* uses pattern matching to identify numerical PHI instances and look-up tables with context checks to identify non-numerical PHI instances. This move to Python, with an accompanying Jupyter notebook to show usage, has made the de-identification tool more accessible to a broader audience of researchers looking to de-identify their data. However, the same limitations of *deid* apply to *pydeid*, so the package is best used in conjunction with other de-identification methods.

2.3.2 Machine learning algorithms

In contrast to purely rule-based approaches, machine learning methods use various algorithms to train themselves to recognize patterns without the need to define these

[1] https://github.com/MIT-LCP/pydeid

patterns. In practice, machine learning methods train neural networks (also known as artificial neural networks, or ANNs), which are comprised of node layers: an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, has its own associated weight and threshold, and when it receives data from the previous layer, it can be activated and send the data to the next layer. In natural language processing, the data is sequential, so many de-identification efforts have focused on recurrent neural networks, or RNNs, which contain memory cells that can store information over time. More recently, transformer architectures have been explored for de-identification.

Previous research has found success with long short term memory (LSTM) [35], bi-directional long short term memory (BiLSTM), and conditional random field (CRF) models [32, 10]. After the development of BERT (bidirectional encoder representations from transformers), a deep learning model that uses attention through bidirectional transformers, researchers have applied BERT and subsequent BERT-variant models to the task of deidentification, achieving state-of-the-art performance [21, 38]. Most recently, there has been the emergence of large language models (LLMs), which are deep learning algorithms with many parameters (on the order of billions or more weights) trained on large quantities of unlabeled text using supervised or semi-supervised learning. With the ability to perform well on NER with little to no additional fine-tuning, LLMs are a promising method of de-identification.

Long Short-Term Memory networks (LSTMs)

Long short-term memory (LSTM) models are a type of RNN that can learn long-range dependencies in sequences of data [17]. LSTMs use a memory cell and gating mechanisms to regulate the flow of information in a cell. While the memory cell stores the previous state of the LSTM at that node, three gates–the input gate, forget gate, and output gate–control the flow of information in and out of the cell. This allows for LSTMs to "forget" information that is no longer relevant. The ability to understand long-range context dependencies is useful, but conventional LSTMs are only able to make use of previous context. To overcome this, bi-directional LSTMs (BI-LSTMs) were introduced to process sequential data in both directions [16]. Implementations of BI-LSTMs in de-identification involve one LSTM trained on the regular sequence and a second LSTM trained on a sequence with the words in reverse order.

Madan et al. [35] used a BI-LSTM with character-level embeddings concatenated with POS (part-of-speech) tag embeddings for de-identification. The character-level embeddings allowed the LSTMs to incorporate character-level information while encoding each word into a vector (embedding), and the POS were added to the final input embedding to be fed into the model. The BI-LSTM achieved a strong micro-averaged F1 score of 0.9592.

While BI-LSTMs can account for context in both left-to-right and right-to-left fashion, they are subject to locality bias, which is the tendency to weight short-distance context over long-distance context [24].

Hybrid methods

To mitigate some of the limitations of purely rule-based and machine learning methods, hybrid methods have combined the two. Liu et al. (2017) [32] tested four de-identification subsystems, both individually and ensembled together: BI-LSTMs, BI-LSTMs with features (an additional hidden layer), conditional random fields (or CRFs, another type of machine learning), and rules-based (regular expressions). When the ensemble classifier of three machine learning-based subsystems and the rules-based subsystem were merged, the final system achieved the highest "strict" micro-averaged F1-scores of 0.9143 on the 2016 N-GRID corpus, ranking first in the 2016 CEGS N-GRID NLP challenge [32].

2.3.3 State of the art

In 2017, the transformer neural network architecture was introduced [61], and it used an attention mechanism that allowed for the input sequence to be processed in parallel rather than sequentially, as in the case of RNNs. This allowed transformers to process words concurrently and better learn the context of words (i.e. learn from both directions simultaneously). Transformers are composed of encoder-decoder stacks; the encoder takes in words and generates embeddings that encapsulate the meaning of the word and the context, while decoders take the embeddings and generate the output sequence. Transformers are powerful tools for NLP, achieving state-of-the-art results on a variety of NLP tasks. Applications of different types of transformers to de-identification have similarly shown strong performance.

BERT (and RoBERTa)

The objective of the original transformer architecture was to generate output sequences from input sequences, as in the case of machine translation [61]. Devlin et al. (2019) [11] took apart the transformer architecture and focused on stacking just encoders, resulting in the development of BERT (Bi-directional Encoder Representations from Transformers). BERT was designed to learn bidirectional representations from unlabeled text from pre-training, which is the process of training a model on a dataset that is similar to the target task but not exactly the same, with the goal to initialize the weights of the model based on knowledge tained from the pre-training dataset. Following pre-training, the model is a viable option for transfer learning, a technique where a model trained on one task is used as a starting point for training a mdoel on a different task. By using the weights from the pre-trained model as a starting point, the model can then be fine-tuned (trained) on a smaller labeled dataset relevant to the intended task to achieve high performance.

BERT was pre-trained via self-supervised learning on the BooksCorpus (800M words) [70] and English Wikipedia (2,500M words). The resulting pre-trained BERT model can generate deep representations of input sequences and can then be fine-tuned with just one additional output layer for application to a wide range of tasks, including named entity recognition. Upon its release, BERT achieved new state-of-the-art results on eleven NLP tasks [11]. Johnson et al. (2020) [21] used BERT for clinical text de-identification, tokenizing sentences as inputs to different versions of pre-trained BERT, with a final linear layer with the outputs of BERT as inputs and the log-likelihood of different PHI classes as outputs. Both BERT-base and the larger

BERT-large outperformed the previous state-of-the-art de-identification models based on RNNs.

Following BERT, several BERT-variant models have arisen, including AlBERT (A Lite BERT) [27] and RoBERTa (Robustly Optimized BERT Pre-Training Approach) [30]. While the RoBERTa model shares the same architecture of BERT, the two differ in training corpus size–RoBERTa's is 160 GB compared to BERT's 16-GB corpus– and pre-training loss functions, and training hyperparameters. RoBERTa achieved new state-of-the-art results on top of records set by BERT, and de-identification was no exception. AlBERT was introduced in an effort to address issues associated with larger/deeper models: increased hardware requirements, memory utilization, and training times. In a comparison of several transformer architectures (i.e. BERT, RoBERTa, and AlBERT) for clinical text de-identification using the i2b2 2014 corpus, RoBERTa-large was found to be the best-performing model ($>$ 0.99 accuracy, 0.967 precision/recall) [38]. In the study, the authors concluded that transformer model-based architectures could, after suitable hyperparameter optimization, be a satisfactory solution to clinical text-deidentification.

Large language models (LLMs)

In a similar way to the BERT architecture, which is based on stacks of encoders, decoders have been stacked to form models like GPT (Generative Pre-trained Transformer) [13]. GPT-2, GPT-3, and other large language models (LLMs) have been shown to perform well on a variety of NLP tasks. GPT-4 [45] in particular, with its enhanced text data processing capabilities compared to its predecessor GPT-3, has sparked renewed interest in zero-shot and few-shot learning, where models are trained on limited or no data for a new task and subsequently tested on that task. This avenue of research holds promise for the field of de-identification, as it could enable the development of de-identification models that generalize well to new datasets without extensive fine-tuning.

One notable application of GPT-4 in de-identification is the "DeID-GPT" model, which achieved a remarkable accuracy of over 0.99 in a zero-shot scenario when provided with a specified prompt for de-identifying the 2014 i2b2 data [33]. However, the evaluation of DeID-GPT solely based on accuracy raises questions about its overall performance, as other crucial metrics such as precision, recall, and F1 score are not reported. Consequently, it becomes challenging to make meaningful comparisons between DeID-GPT and other existing de-identification models.

While the proliferation of LLMs has significantly advanced the field of NLP, it has also raised ethical concerns regarding the potential misuse and privacy implications associated with these models. Particularly worrisome is the continuous training of LLMs on user input data, as exemplified by models like ChatGPT and GPT-4. Adversarial attacks, for instance, have been shown to extract sensitive information from the training data utilized by LLMs [34], underscoring the need for robust privacy safeguards and responsible deployment of these powerful language models. As an even more direct risk, data breaches targeting the companies that own and operate LLMs could expose personal information, further jeopardizing the privacy and security of individuals whose data has been used to train these models. Therefore, stringent measures are needed to safeguard the data repositories and infrastructure supporting these LLMs, ensuring the protection of individuals' privacy and preventing any unauthorized access or misuse.

2.4 What tools are available for doing de-identification today?

Despite numerous recent advances in de-identification algorithms, the actual task of de-identifying a clinical dataset remains difficult. Studies reporting high performance metrics for their de-identification models often do not make their code available, making it a challenge for a researcher to piece together what they did. Out of the de-identification methods that are available, there are three main categories: general-purpose natural language processing libraries, commercial de-identification tools, and open-source de-identification tools. Among these approaches, there is often a significant tradeoff between cost and performance. Many free methods fall noticeably short of the current state of the art, and those that–potentially–perform better have a cost of usage proportional to the amount of data processed.

In Table 2.1., we present some of the presently available de-identification tools. We observe that there is no standardized method of evaluating performance of de-identification tools. While all of the studies surveyed in Table 2.1 report F1 score, the metric can be calculated in different ways; for example, some studies focus on binary PHI classification [10], while others show metrics on multi-class classification [32, 21]. While many studies use the i2b2 2014 dataset for evaluation and use the set of PHI tags defined by the challenge, others use PHI tags that are often more general, such as those based on a looser interpretation of HIPAA guidelines [42]. There are also varying levels of strictness when evaluating performance. Two more common methods are token-level evaluation, which evaluates performance on each token individually, and entity-level evaluation, which evaluates performance on each entity (which can be composed of multiple tokens). While many de-identification tools report token-level metrics, general NLP tools such as spaCy are often evaluated at the entity level. All this variation makes it difficult to compare performance across studies, and it is important to keep this in mind when evaluating the performance of any single de-identification tool. While Table 2.1 records reported F1 scores for many de-identification tools, we note that since the calculation of the F1 score can vary, comparisons of tools using different evaluation approaches should be made with caution.

2.4.1 General-purpose natural language processing libraries

There are several popular NLP libraries available in Python, the language we use for the project. One popular example, SpaCy,[2] is an industrial-strength, open-source library for performing NLP tasks. The library is specifically designed to build complex industrial systems, featuring integration with TensorFlow, PyTorch, sci-kit-learn, Gensim, and others in Python's AI ecosystem [28]. The library's NER capabili-

[2] https://spacy.io/

Table 2.1: Summary of existing de-identification software. Note that evaluation metrics are not directly comparable across studies due to differences in datasets and evaluation methods. *: original model used available in Huggingface Transformers library. **: F1 score was not provided, so precision/recall provided instead. ***: Instead of an F1 score, the value reported was the F2 score, which is a weighted average of recall and precision that values recall twice as much as precision.

Name	**Approach**	**Reference**	**Year released**	**Eval data**	**Eval method**	**F1, %**	**Availability**
Liu et al.	CRF + Bi-LSTM + hand-crafted features	Liu et al. (2017)	2017	i2b2	multi-class i2b2 token	96.98	unavailable
Dernoncourt et al.	CRF + Bi-LSTM	Dernoncourt et al. (2017)	2017	i2b2	binary HIPAA token	97.87	unavailable
Johnson et al.	BERT_large, uncased	Johnson et al. (2020)	2020	i2b2	multi-class i2b2 token	98.4	unavailable
Meaney et al. (AlBERT)	Albert-XXLarge	Meaney et al. (2022)	2022	i2b2	unknown	96.44	unavailable*
Meaney et al. (RoBERTa)	Roberta-Large	Meaney et al. (2022)	2022	i2b2	unknown	96.75	unavailable*
PhysioNet deid	lookup tables, regular expressions, heuristics	Neamatullah et al. (2008)	2008	MIMIC II	multi-class HIPAA token	74.9/96.7**	open
PHIlter	pattern matching, blacklists, whitelists	Norgeot et al. (2020)	2020	i2b2	[unknown classification] i2b2 token	94.77***	open
spaCy	Roberta-Large	-	2015	CoNLL03	multi-class entity	91.6	open
Stanza	Pretrained NER models	Qi et al. (2020)	2020	CoNLL03	multi-class entity	92.1	open
flair	Pretrained NER models	Akbik et al. (2019)	2019	CoNLL03	multi-class entity	92.7	open
Amazon Comprehend Medical	DetectPHI API	-	2018	-	-	-	commercial
Microsoft Azure Cognitive Service for Language	PII detection feature	-	-	-	-	-	commercial
Google Cloud Data Loss Prevention	De-identification API	-	-	-	-	-	commercial
MIST	CRF	Aberdeen et al. (2010)	2010	i2b2	multi-class i2b2 token	96.5	open
NeuroNER	CRF + LSTM	Dernoncourt et al. (2017)	2017	i2b2	unknown	97.7	open

ties are based on CNNs (convolutional neural networks) and transformers, which come pre-trained on general text and can be fine-tuned for specific purposes like de-identification. SpaCy utilizes `Document` objects to store text and annotations, which can be accessed through the `ents` property after being processed through an NER pipeline.

Stanza [49] is a set of language processing tools that offers high accuracy and efficiency for multiple human languages. It includes a Python interface to the CoreNLP Java package. The toolkit includes dedicated tools and models tailored for biomedical and clinical applications, making it particularly suitable for addressing specific tasks in the medical domain [69]. The clinical models have been trained on the MIMIC dataset. Within Stanza, the `NERProcessor` performs named entity recognition (NER) and can be invoked using the name `ner`. Upon completion of the NER pipeline, the resulting `Document` comprises a list of `Sentence`s, with each `Sentence` containing a list of `Token`s. The flexibility of Stanza's architecture allows the use of multiple NER models concurrently by specifying a list in the package dictionary.

Flair [3], another NLP framework, streamlines the training and dissemination of cutting-edge models for sequence labeling, text classification, and language modeling tasks. The central idea behind this framework is to provide a unified interface for different types of word and document embeddings through its text embedding library, enabling researchers to combine and utilize diverse embeddings. Additionally, the framework incorporates standard routines for model training, hyperparameter selection, and a data fetching module, which can efficiently download and convert publicly available NLP datasets into suitable data structures for rapid experimentation. Flair also includes a repository of pre-trained models, allowing users to readily employ state-of-the-art NLP models. To run NER using flair, the steps are to make a `Sentence`, load a pre-trained model, and use the model to predict tags for the sentence. The standard model uses flair embeddings, is pre-trained over the English CoNLL-03 task [51], and can recognize 4 different entity types, while the Ontonotes-pretrained models [62] can classify 18 different types of entities.

Recent studies have explored the utility of natural language processing toolkits for de-identifying unstructured medical text documents. One study comparing spaCy with OpenNLP, another open-source toolkit, found that a spaCy model achieved higher performance at deidentification [46]. A spaCy model was also shown to outperform Bi-LSTM and CRF models when all were trained and evaluated on the 2014 i2b2 data [19]. While neither study specified the spaCy model architecture used, it is likely that they used spaCy's default NER system, which consists of a deep convolutional neural network (CNN). Both studies found the spaCy model to achieve an F1 score of 0.91, which indicate that spaCy is a viable tool for deidentification, though further models need to be tested to achieve sufficient performance.

2.4.2 Commercial de-identification tools

There are also software products that address text de-identification as part of the scope of data privacy. These commercial tools use a mix of rules and machine learning to detect sensitive data, though companies do not go into more detail about the rules or model architectures. For example, Amazon Comprehend Medical[3] is a HIPAA-eligible natural language processing service that has been pre-trained to extract health data from medical text. When using the `DetectPHI` operation, Amazon Comprehend Medical creates a file in the output location with information on all the entities detected, including position, confidence score, text, and category. However, it is not free to use and could quickly become costly when deidentifying text with tokens in the order of millions, as requests are measured and charged in units of 100 characters (1 unit = 100 characters). Initially, users can utilize the free tier, which covers 85k units of text (8.5 million characters, or 1000 5-page 1700-character per page documents) for the first month.[4] Following the initial month, for requests of under 1 million units, the cost of the DetectPHI API is $0.0014 per unit. Given an example clinical note from the i2b2 2014 de-identification corpus containing 1,760 characters, de-identification would cost $0.02464 per note, or $246.40 for 10,000 notes. While this cost could be

[3] https://docs.aws.amazon.com/comprehend-medical/latest/dev/comprehendmedical-welcome.html

[4] https://aws.amazon.com/comprehend/medical/pricing/

covered by some labs, it represents a financial barrier that could potentially deter many from de-identifying and sharing their data.

PII (personally identifiable information) detection is one of the features offered by Microsoft's Azure Cognitive Service for Language.[5] This feature can identify, categorize, and redact sensitive information in unstructured text. By utilizing the PII Detection "skill" (feature), users can detect PII in input text and choose to mask it, if desired. The feature has two outputs: `piiEntities`, an array of complex types including the extracted text, type, and score (indicating the likelihood of being a real entity); and `maskedText`, the masked text processed according to the specified `maskingMode` (if applicable). When using the PII detection feature, users can process 5,000 text records per month free, with an additional $700 monthly per 1 million text records. Compared to Amazon Comprehend Medical, assuming the user uses the free tier during the first month, the cost of de-identifying 10,000 notes would still be $700, as pricing appear to be a flat rate for different tiers of usage.

Google Cloud Data Loss Prevention can detect PII and use a de-identification transformation to mask, delete, or otherwise obscure the data.[6] The API allows users to specify configurations for de-identification techniques such as masking by replacing characters with symbols, replacing entities with tokens or surrogates, and encrypting and replacing sensitive data. The output file includes the de-identified text and the type of PII detected. Cloud DLP content method pricing is billed based on bytes inspected and bytes transformed (separately), with up to 1 GB monthly free for each followed by $3/GB and $2/GB for further inspection and transformation, respectively. To calculate the expected cost of de-identifying 1,000 notes, we consider that the median note size of the i2b2 2014 dataset is around 7 kB. Assuming that the notes all average out to a similar size, the cost of de-identifying 10,000 notes would be free for the first GB. If considering additional requests beyond the first GB, the calculation for cost would be 0.000007 GB x $5/GB * 10,000 notes = $0.35. This is significantly cheaper than the other two commercial tools, though it is unclear how

[5] https://learn.microsoft.com/en-us/azure/search/cognitive-search-skill-pii-detection

[6] https://cloud.google.com/dlp/docs/deidentify-sensitive-data

the performance of the de-identification model compares to the other two.

2.4.3 Freely available de-identification tools

There are also a few open-source de-identification tools available for users to de-identify their clinical notes for free. For example, Philter (Protected Health Information filter), an open-source, command line-based clinical text de-identification software, relies purely on pattern matching in the form of whitelists, blacklists, and regular expressions to flag and remove PHI [43]. De-identification is done with a single command line function call specifying various parameters like file paths and output format. By default, Philter outputs PHI-reduced notes (.txt format) in the specified output directory.

When considering machine learning-based approaches, researchers can use MIST (MITRE Identification Scrubber Toolkit) [1], a suite of tools for identifying and redacting PHI powered by a conditional random field-based sequence tagger. While not open source, the package is free to download and use. Users can use the provided web interface to annotate text, and the documentation provides instructions on how to train and evaluate a de-identification model from the command line. PII can be masked either with obscuring fillers, such as "[NAME]", or with synthesized but realistic English fillers. While useful at the time of release, due to its age, MIST now has reduced utility due to outdated dependencies. At the time of writing this thesis, the latest version of the package is 2.0.4, which was last updated in 2014. The package requires Python 2 to run, as Python 3 is known not to work with the software. This could pose a challenge for users who do not have Python 2 installed on their machines, as they would have to install an older version of Python to use the software. The system requirements also hint at potential issues, as the last recorded MacOS system that has been (partially) tested is Snow Leopard (10.8.x). At the time of writing, the latest MacOS system is MacOS Ventura (13.3.x). Issues of compatibility with newer systems and software versions are common in open-source software, and they motivate the need for a new, regularly maintained de-identification tool.

NeuroNER is an open-source NER tool based on LSTMs that focuses on making (previously) state-of-the-art NER available to anyone, with an emphasis on usability [9]. The tool allows users to create or modify annotations for new or existing data by interfacing with the web-based annotation program BRAT [54]. The NER engine consists of a model pre-trained on CoNLL 2003 and i2b2 2014, making NeuroNER well-suited PHI detection. Using default hyperparameters, the system performs marginally better than the state of the art at the time on i2b2 2014 data: a 97.7% F1 score compared to the previous highest of 97.9%, also by Dernoncourt et al. (2017). NeuroNER is `pip` installable, uses the spaCy English langauge module, can be run using a Python interpreter or command line, and offers pretrained models for use. The tool is also well-documented, with a tutorial and a user guide available on the GitHub repository. However, NeuroNER tool is not actively maintained, with the last commit being in 2019. This could pose a problem for users who encounter issues with the tool, as there is no guarantee that the developers will respond to issues or fix bugs.

These freely available de-identification tools, while undoubtedly helpful, vary in performance and are often rather difficult to use due to incompatible or outdated versioning. This motivates the need for a new, easy-to-use de-identification tool that is actively maintained.

Chapter 3

Datasets for de-identification

3.1 Aim

While HIPAA is intended as legal guidance, it falls short of a technical definition of language tokens. As a result, there may be ambiguity as to what exactly constitutes a specified identifier. This ambiguity has led to variation in labeling in the few datasets available for training and evaluating models for de-identification. This may have also indirectly contributed to a paucity of publicly available datasets for developing de-identification models. This is concerning, as access to data is key to advancing machine learning in healthcare. The lack of data is both a limitation for developing models and a motivation for our work. In this chapter, we review currently available datasets for de-identification and introduce a new dataset, which we use as the basis of our bias audit in Chapter 4 and hope will be a useful resource for the de-identification community. While we would like to work toward a technical specification for de-identification, it is beyond the scope of this thesis.

3.2 Introduction

One recent finding that highlights the need for more de-identified data is presented by Wornow et al. (2023) [65]. The authors point out that clinical language models (CLaMs), which are a subtype of large language models (LLMs), are almost all trained on a single database: MIMIC-III [22], which contains approximately 2 million notes written between 2001 and 2012 in the ICU of the Beth Israel Deaconess Medical Center. In particular, 17 out of the 23 CLaMs surveyed have been trained on MIMIC-III/IV, with the remainder of the CLaMs having been trained on private electronic health record data. This is problematic because MIMIC is a small dataset and not representative of most populations, as Beth Israel Deaconess Medical Center is a well-resourced private hospital in a major metropolitan area. Furthermore, MIMIC data would lack any new diseases, treatments, or practices discovered after 2012. These limitations may cause CLaMs to encode weights or even biases unique to characteristics of MIMIC data. This motivates the need for more data from health centers of varying sizes, locations, and patient demographics.

Here we present five datasets commonly used to develop de-identification methods. Three of the most commonly used datasets were introduced by three competitions: the 2006 Informatics for Integrating Biology and the Bedside (i2b2) competition [60], the 2014 i2b2/UTHealth shared task [57], and the 2016 Centers of Excellence in Genomic Science (CEGS) and Neuropsychiatric Genome-Scale and RDOC Individualized Domain (N-GRID) shared task [56]. In addition to competition datasets, another common source of data was MIMIC (‘Medical Information Mart for Intensive Care’), an extensive database comprising data from critical care units at a large tertiary care hospital [22]. MIMIC has undergone several updates and is now at MIMIC-IV. MIMIC-II was the source of nursing notes for the PhysioNet corpus [12, 42, 15]. The Dernoncourt-Lee corpus drew discharge summaries from MIMIC-III [10]. Table 3.1 provides summary statistics on these datasets, which are all publicly available. There are also numerous datasets used to train and evaluate models that are unavailable, however. For example, one study found high performance for the models it evaluated using a dataset of 50 IME (independent medical examination) reports, which were manually de-identified but not available for others to use [46]. Thus, datasets for the de-identification tasks are either overused (in the case of the challenges and MIMIC) or unavailable for use, highlighting the need for a new dataset, which is addressed as part of the project.

Table 3.1: Summary of available datasets for the de-identification task. The included datasets are five of the most commonly used datasets to develop and evaluate models for de-identification. The datasets are ordered by year of release. Partners HealthCare is now known as Mass General Brigham. MIMIC: Medical Information Mart for Intensive Care. BIDMC: Beth Israel Deaconess Medical Center. *: Track 1.A. was split with 60% as test data, Track 1.B. was split with 60% as training data.

Dataset	**Year**	**Note type**	**Note source**	**# patients**	**# notes**	**# tokens**	**# PHI**	**Train/test**
i2b2-2006	2006	discharge notes	Partners HealthCare	889	889	487k	19.5k	75%/25%
physionet	2008	nursing notes	MIMIC II (BIDMC)	163	2434	345k	1.9k	59%/41%
i2b2-2014	2014	diabetic longitudinal records	Partners HealthCare	296	1304	738k	28.8k	61%/39%
CEGS-N-Grid-2016	2016	psychiatric intake records	Partners HealthCare	1000	1000	1,862k	34.4k	60%/40%*
Dernoncourt-Lee	2016	discharge summaries	MIMIC III (BIDMC)	1635	1635	2,945k	60.8k	80%/20%

3.3 i2b2 2014 de-identification challenge data

We use the 2014 i2b2/UTHealth corpus for training and evaluating our de-identification models [58]. The corpus contains 1,304 longitudinal medical records describing 296 patients, with 2-5 records selected per patient, and a total of 805,118 whitespace-sparated tokens; an average of 617.4 tokens per file. The records come from Partners Healthcare and present a snapshot of diabetic patients' health at different points in time. All records are annotated according to a risk-averse interpretation and extension of HIPAA as detailed in the corpus paper [58]. Each record is stored as an XML file with a <TAGS> node containing annotations for the document text. The downloadable corpus data comes in three sets: training set 1 (521 notes), training set 2 (269 notes), and test set (514 notes). We use training set 1 as our training data, training set 2 as our validation data, and the test set as our test data. The data is available for download from the i2b2 2014 de-identification challenge website.[1]

[1] https://portal.dbmi.hms.harvard.edu/projects/n2c2-nlp/

3.4 Challenges in annotation

Based on the 18 identifiers listed under the Safe Harbor provision, de-identification dataset annotators have produced differing annotation schema, which forms a barrier to cross-dataset evaluation. For example, HIPAA only considers ages over 89 as PHI, and the 2006 i2b2 de-identification corpus treats all ages as PHI [60]. Continuing along a risk-averse interpretation of HIPAA guidelines, creators of the 2014 i2b2 de-identification dataset expanded their definition of PHI to other information indirectly related to patients that could potentially identify them. This information includes doctors' and nurses' names, all parts of dates (including years), and all locations (including states and countries) [58]. In 2016, the CEGS N-GRID de-identification dataset went on to include "generic" organizations such as "deli" or "gas station" in the LOCATION: ORGANIZATION tag [56]. These annotation differences present a challenge to researchers aiming to evaluate their de-identification method on multiple datasets, requiring the grouping together of different PHI categories into larger bins and subsequent shifts in preprocessing data and calculating metrics [?].

The varied landscape of PHI classification is especially a challenge for free text de-identification, as models must adapt to the idiosyncracies of clinical notes, including medical terminology, abbreviations, and writing errors. Manual de-identification is able to address this somewhat, as medical professionals in the same domain or health center as the clinical notes can draw from their expertise to identify PHI, but the process is time-consuming and human error-prone [2]. Recent de-identification approaches incorporate natural language processing (NLP), the branch of artificial intelligence focused on giving computers the human-like ability to understand text and spoken words, into their workflows, with promising results.

3.5 Creation of a new de-identification dataset

In Chapter 4, we detail an audit for potential demographic biases in de-identification tools. Many of the evaluated tools have been trained on the same datasets, particularly the i2b2 2014 dataset, motivating the need for a new dataset not yet "seen" by the de-identification models. In order to facilitate our bias audit and contribute a new dataset for future de-identification research, we create a new dataset of 100 hospital admission notes. Because of the objective of the bias audit to test the ability of de-identification methods to remove names, we focus on constructing realistic name surrogates that are in line with popular demographics. We describe the process of creating the dataset in this section, and more details on the usage of the dataset are discussed in Chapter 4.

To create the dataset, we first construct 16 name groups with distinct combinations of gender, race, popularity, and decade demographics. The purpose of these groups is for comparative evaluation of model performance on different groups. Then, we present how we prepare and populate 100 hospital admission notes for evaluation inputs.

3.5.1 Construction of name sets

To prepare name lists with diverse gender, race, popularity, and decade backgrounds, we first aggregate the number of people of each gender having the same first name and born in each of the following decades—the 1940s, 1970s, and 2000s—based on the data from the US Social Security Administration.[2] We then assign each first name to a racial or ethnic group by referencing the demographic aspects in [59]. We process the surnames in a similar fashion based on 2000 US Census data,[3] with the assumption that the population of each surname does not change much over time.

In this way, we create 16 mutually exclusive name lists in Table 3.2, where each name list contained 20 names. In particular, the 20 most popular names of each decade are chosen such that they do not appear in the 50 most popular names of the other two decades, which ensures that the names are sufficiently representative of people born in each decade. The 20 least popular names are chosen by breaking ties randomly. The names of medium popularity are randomly sampled from the

[2] https://www.ssa.gov/oact/babynames/limits.html

[3] https://www.census.gov/topics/population/genealogy/data/2000_surnames.html

Table 3.2: Description of 16 name sets of diverse demographic backgrounds and examples of first and last names for each set. Name Sets 1-6 are names with top, medium, and bottom popularity associated with the White racial group in the 2000s. Name Sets 7-12 are names with medium popularity associated with the Black, Asian, and Hispanic racial groups in the 2000s. Name Sets 13-16 are names with top popularity in the 1970s and 1940s associated with the White racial group. Reproduced from Xiao et al. (2023) [66].

Name Set	**Gender**	**Race**	**Popularity**	**Decade**	**First Name Examples**	**Last Name Examples**
1	Male	White	Top	2000s	Jacob, Ethan, Tyler, ...	Smith, Davis, Brown, ...
2	Female	White	Top	2000s	Emily, Emma, Olivia, ...	Smith, Davis, Brown, ...
3	Male	White	Medium	2000s	Wade, Ted, Brien, ...	Waldon, Clapp, Bogle, ...
4	Female	White	Medium	2000s	Mabel, Liz, Terressa, ...	Waldon, Clapp, Bogle, ...
5	Male	White	Bottom	2000s	Nicki, Leslee, Marti, ...	Lofft, Lyna, Tamaro, ...
6	Female	White	Bottom	2000s	Glenn, Lyle, Heath, ...	Lofft, Lyna, Tamaro, ...
7	Male	Black	Medium	2000s	Cedric, Marlon, Ollie, ...	Booker, Grier, Spikes, ...
8	Female	Black	Medium	2000s	Aisha, Ebony, Jamila, ...	Booker, Grier, Spikes, ...
9	Male	Asian	Medium	2000s	Zhi, Nguyen, Rajeev, ...	Ngo, Mao, Ahmed, ...
10	Female	Asian	Medium	2000s	Neha, Priya, Xin, ...	Ngo, Mao, Ahmed, ...
11	Male	Hispanic	Medium	2000s	Leonel, Camilo, Cruz, ...	Ceja, Amaro, Recinos, ...
12	Female	Hispanic	Medium	2000s	Celina, Rebeca, Luisa, ...	Ceja, Amaro, Recinos, ...
13	Male	White	Top	1970s	Patrick, Brian, Eric, ...	Smith, Davis, Brown, ...
14	Female	White	Top	1970s	Amy, Lisa, Laura, ...	Smith, Davis, Brown, ...
15	Male	White	Top	1940s	Jerry, George, Frank, ...	Smith, Davis, Brown, ...
16	Female	White	Top	1940s	Linda, Carol, Nancy, ...	Smith, Davis, Brown, ...

names ranked 400 to 8000 by popularity, since the most popular names identified with Black, Hispanic, or Asian groups fell into this range. Then, we can compare the de-identification performance of models along name dimensions.

3.5.2 Preparation of clinical templates

We develop a new clinical text de-identification dataset with 100 selected hospital admission notes from Beth Israel Lahey Health between 2017 and 2019 that have not previously been made publicly available. Each note is used as a template for PHI substitution. We follow the HIPAA Safe Harbor provisions by marking the occurrence of names in the templates and replacing other PHI classes with realistic, synthetic values. We note that our templates are more complex than those used in existing benchmark datasets [37, 39, 40], with an average of 12,893 characters and 3.5 unique names per template and each unique name appearing an average of 2.1 times per template [66]. This design is more reflective of real-world de-identification applications and more likely to expose flaws in less effective methods.

Chapter 4

Auditing bias in de-identification software

4.1 Aim

In this section, we audit the performance of existing de-identification software on a new dataset. The dataset has been annotated to enable the evaluation of the performance of each de-identification tool on different demographic groups, using names as a proxy for these groups. We also explore a bias mitigation technique of fine-tuning de-identification models on more diverse data and show that doing so can improve the models' performance on underrepresented groups. A note on contribution: I did initial work on the project with Tom Pollard and found interesting results, which then led to a collaboration with Yuxin Xiao and Marzyeh Ghassemi, and together we carried out an extensive further analysis and wrote a paper [66].

4.2 Introduction

Considering the growing utility of natural language processing approaches to de-identification, it is crucial to consider potential biases embedded in the models. All five of the common, publicly available datasets used for de-identification are sourced from American hospitals whose patient populations may not adequately reflect the diversity of patient data for de-identification. One characteristic component of patient data that can vary widely is names. Names appear in a spectrum of formats reflecting diverse naming practices based on religion, language, or geography. Names originating from cultures outside America may not be represented as highly as typical American names in the training datasets for language models. The skewed representation of names in data could lead to the models unintentionally embedding biases that reflect the datasets, including a decreased ability to recognize and remove uncommon names from patient data before it is shared. While names are not a proxy for race/ethnicity, if most of the people with a given name self-identify with a particular group, and models are not as efficient at removing this name, it then follows that members of this group may be at higher risk of data disclosure [37].

Multiple NER studies have used names as a proxy to detect biases against different demographic groups. Mehrabi et al. (2020) [39] examined the difference in NER models' ability to recognize male and female names as PERSON entity types highlighted gender-based discrepancies. The model chosen for evaluation was Stanford CoreNLP [36], an NLP toolkit used widely in research, government, and commercial circles. When evaluating the model on a dataset of U.S. Census-obtained baby names, researchers found that relatively more female names were not recognized as PERSON entities as compared to male names. In particular, when female names were also common location names (e.g. "Charlotte), the NER model almost always wrongfully tagged the name as a location, despite clear context that the entity should be a person.

Another paper similarly explored NER models for demographic bias in the categories of gender and ethnicity. Mishra et al. (2020) [40] evaluated a BiLSTM CRF, spaCy, and Stanford CoreNLP with a dataset composed of names across 8 demographic groups, which were a combination of race (or ethnicity) and gender. By assessing if NER models varied in their accuracy of identifying first names from various demographics as PERSON entities, the researchers found that models were better at identifying White-associated names with higher confidence as compared to other demographics.

To examine biases in name detection with a privacy lens, Mansfield et al. (2022) [37] evaluated three off-the-shelf PII (personally identifiable information) masking systems on name detection and redaction, using names and templates from customer service messaging conversations. The authors found significant disparities in name recognition based on demographics, particularly for names associated with Black and Asian/Pacific Islander groups. These disparities were more pronounced in the commercial models tested than an open-source RoBERTa-based system. Our work in de-identification-related bias extends this work and the aforementioned studies in the following key aspects: considering first and last names together, adding two dimensions of evaluation (age and name popularity), changing the domain/length of the templates (hospital admission notes, which are much longer), and increasing the number and type of models evaluated (in total: 3 open-source NLP libraries, 3 open-source clinical text de-identification-specific models, and 3 commercial models).

4.3 Bias evaluation of existing de-identification tools

As the first audit in the existing literature to evaluate clinical text models for potential bias, we identified four dimensions to reflect the diversity and trends in names found in the United States: gender, race, popularity of a name, and decade of name popularity. In particular, we define the demographic dimensions as follows:

- The **gender** of a name refers to the sex assigned at birth to someone with that name, as the phonological property of a name can suggest the associated gender [33]. We examine two gender groups in this study: male and female.
- The **race** of a name refers to the expected racial or ethnic identity of someone with that name, reflecting the variation in name distributions that exists between different self-reported racial or ethnic groups [60]. We consider four racial or ethnic groups: White, Black, Asian, and Hispanic. Other groups are skipped due to prohibitively small community sizes.
- The **popularity** of a name refers to the size of the population of a gender within

a decade having that name. We compare three groups here: top, medium, and bottom popularity.

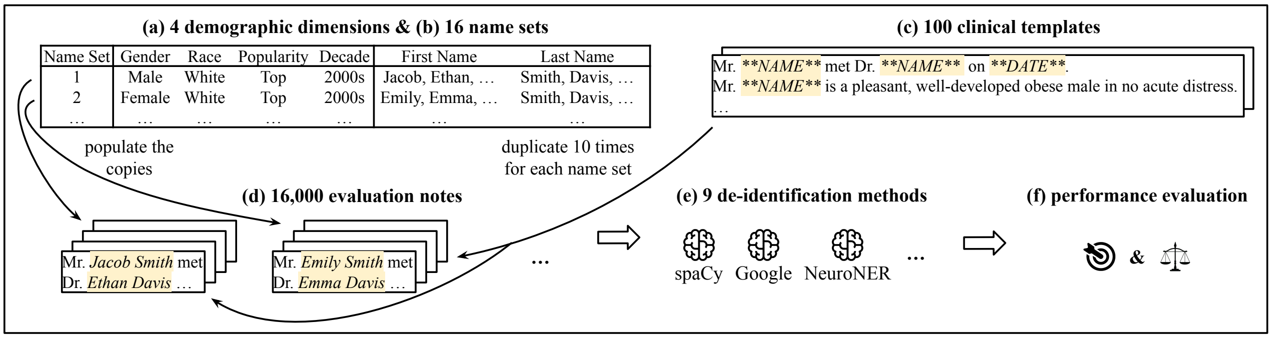
- The **decade** of popularity refers to the decade in which a name is popular in the U.S. in terms of babies being given the name, as naming trends change over time. We assess three decade groups: 2000s, 1970s, and 1940s.

Limitations of Standardized Demographic Categories. We acknowledge the limitation of using standardized self-reported racial categorizations and binary gender groups when composing the name sets. More fine-grained racial and gender categorizations can be explored in future work, as there can be variety in the linguistic norms and naming traditions even within each racial group considered. Transgender and non-binary gender groups are also important to consider in future work, as these groups may use gender-neutral names or have variations in name usage between records.

We conduct the bias audit according to the workflow in Figure 4-1 We first prepare 16 name sets (Table 3.2) by combining the four demographic dimensions. To construct the dataset, we duplicate each of the 100 selected clinical templates ten times and populate the copies with randomly selected names for each of the name sets. We then use the 16,000 evaluation notes to assess nine de-identification methods. The nine methods are chosen from three categories: general-purpose natural language processing libraries, commercial services for PHI detection, and tools designed specifically for the purpose of de-identification. We evaluate the methods on their ability to detect names in the notes, with a focus on performance along the four demographic dimensions.

4.3.1 De-identification methods for bias evaluation

Here we evaluate models of three different categories: general-purpose, open-source libraries for natural language processing, commercial services for PHI detection, and models designed specifically for the purpose of de-identification. For each category, we identify three popular options for assessment.


(a) 4 demographic dimensions & (b) 16 name sets
(c) 100 clinical templates
Mr. **NAME** met Dr. **NAME** on **DATE**.
Mr. **NAME** is a pleasant, well-developed obese male in no acute distress.
…
populate the copies
duplicate 10 times for each name set
(d) 16,000 evaluation notes
Mr. Jacob Smith met Dr. Ethan Davis …
Mr. Emily Smith met Dr. Emma Davis …
…
(e) 9 de-identification methods
spaCy
Google
NeuroNER
…
(f) performance evaluation
&

Name Set	Gender	Race	Popularity	Decade	First Name	Last Name
1	Male	White	Top	2000s	Jacob, Ethan, …	Smith, Davis, …
2	Female	White	Top	2000s	Emily, Emma, …	Smith, Davis, …
…	…	…	…	…	…	…

Figure 4-1: Workflow of the bias audit study. We identify (a) four demographic dimensions and prepare (b) 16 name sets with diverse settings. For each name set, we duplicate each of the (c) 100 clinical templates ten times and populate the copies with randomly generated names. We then uses these (d) 16,000 evaluation notes to assess (e) nine de-identification methods. Reproduced from Xiao et al. (2023) [66]

.

General-purpose, open-source libraries for natural language processing:

- spaCy [18] is a widely-adopted open-source library for industrial information extraction tasks. We use Roberta-base [30], which is pre-trained on a massive general-purpose corpus, as the backbone of its named entity recognition pipeline.

- Stanza [49] is a Python natural language analysis package built with neural network components for named entity recognition. We use its 18-class NER model variant based on contextual string representations [4] and pre-trained on the OntoNotes corpus [62].

- flair [3] is a powerful library based on state-of-the-art natural language processing models for named entity recognition. We use its large four-class NER model variant built on XLM-R embeddings [8] and document-level features [52] and pre-trained on the CoNLL03 corpus [51].

Commercial services for protected health information detection:

- Amazon Comprehend Medical[1] extracts useful information in unstructured clinical text. We leverage its DetectPHI API to extract names in hospital admission notes.

[1] https://docs.aws.amazon.com/comprehend-medical/latest/dev/comprehendmedical-welcome.html

- Microsoft's Azure Cognitive Service for Language[2] uses natural language understanding to extract key phrases from unstructured text, such as personally identifiable information.

- Google Cloud Data Loss Prevention[3] inspects sensitive data in text and removes any personally identifiable information.

Open-source models designed specifically for the purpose of de-identification:

- Philter [43] is a command line-based clinical text de-identification software that uses pattern matching to detect and scrub PHI.

- NeuroNER [9] performs named entity recognition by leveraging a long short-term memory (LSTM) architecture. We use the model pre-trained on the 2014 i2b2 de-identification corpus with GloVe word embeddings [47].

- MIST [1] is a suite of tools for identifying and redacting personally identifiable information in free-text medical records. We pre-train the model supplied by the Carafe engine, a conditional random field-based [26] sequence tagger, on the 2006 i2b2 de-identification corpus.

4.4 Evaluation of Bias

To quantify the bias of each method along each dimension, we follow Mansfield et al. (2022) [37] by evaluating the recall equality difference: the average absolute difference between the recall of each demographic group and that of all the groups along the corresponding demographic dimension. More specifically, for dimension D and its entailed set of demographic groups $\mathcal{G}^D = \{G_1^D, G_2^D, \ldots\}$, recall equality difference $= \frac{1}{|\mathcal{G}^D|} \sum_{G_i^D \in \mathcal{G}^D} |Recall(G_i^D) - Recall(D)|$ [66]. We use the recall equality difference as the fairness metric since it demonstrates the difference in recall each demographic group would experience while expecting the reported average performance.

[2] https://learn.microsoft.com/en-us/azure/cognitive-services/language-service/

[3] https://cloud.google.com/dlp

We carry out the Wilcoxon signed-rank test [64] for the dimension of gender and the Friedman test [14] for the dimensions of race, popularity, and decade to assess the null hypothesis that a de-identification method treats all the groups equally well along a demographic dimension. After applying the Bonferroni correction, the adjusted significance levels for gender, race, popularity, and decade are 5%, 0.833%, 1.667%, and 1.667%, respectively [66].

4.5 Results

We present the results of the evaluation, considering overall performance and performance across the four analyzed dimensions: gender, race, popularity, and decade. We focus on the following performance metrics: precision, recall, and F1 score, with a particular emphasis on recall. We conclude with promising results on the possibility of mitigating bias with fine-tuning on diverse datasets.

4.5.1 Overall performance

We first describe the overall accuracy of various models in Table 4.1. We observe that flair, Amazon, and NeuroNER have the highest recall for their respective model category: NLP library, commercial, and free de-identification tool. The Microsoft model has the highest recall overall of 0.960, while NeuroNER has the highest F1 score overall of 0.945. We also note that the spaCy model has the lowest recall overall of 0.629, and the Philter model has the lowest F1 score overall of 0.353.

4.5.2 Name group recall

Next, we look at the average recall for each name set across all models used for de-identification. Figure 4-2 shows each group's average recall sorted from highest to lowest. The six groups on the left (sets 16, 13, 14, 2, 15, and 1) with the highest recall are all composed of White-associated names. The three groups on the right with the lowest recall (sets 10, 7, and 9) are all composed of non-White-associated

Table 4.1: Bias audit results: Overall performance (higher is better) and bias along demographic dimensions (lower is better) of the examined de-identification methods. We measure the bias with recall equality difference and bold the best two scores in each column. In particular, flair achieves the highest recall and F1 and the lowest bias for race and popularity. Moreover, the asterisk next to a bias score indicates a statistically significant difference in performance at an adjusted significance level (5% for gender, 0.833% for race, 1.667% for popularity and decade). A majority of the examined methods exhibit statistically significant performance gaps along most demographic dimensions. Reproduced from Xiao et al. (2023) [66].

	Overall Performance			**Bias along Dimensions**			
Method	**Precision**	**Recall**	**F1**	**Gender**	**Race**	**Popularity**	**Decade**
spaCy	0.917	0.629	0.746	0.002*	0.013*	0.028*	0.007*
Stanza	0.678	0.881	0.766	0.002*	0.016*	0.011*	0.005*
flair	0.920	**0.974**	**0.946**	0.003*	**0.006***	**0.008***	0.002*
Amazon	**0.923**	0.925	0.924	0.005*	0.022*	0.032*	**0.001**
Microsoft	0.664	**0.960**	0.785	0.003*	0.023*	0.010*	0.006*
Google	0.609	0.869	0.716	0.009*	0.025*	0.014*	0.010*
NeuroNER	**0.946**	0.944	**0.945**	**0.001**	0.045*	0.026*	0.002
Philter	0.227	0.794	0.353	**0.000**	**0.000**	**0.003***	**0.000**
MIST	0.474	0.751	0.581	0.013*	0.022*	0.017*	0.003*

names, namely Asian-associated and Black-associated names. Recall was poorest for Asian-associated names (sets 9 and 10) and next poorest for male, Black-associated names (set 7). Hispanic-associated names fared better overall, with recall for sets 12 and 11 comparable to the recall for White-associated names of equal popularity. When considering gender, the recall between name sets of male and female names appears roughly similar, with female name sets overall having slightly higher recall than male name sets. Looking at name eras, we find that name sets with names from older eras (1970s and 1940s) generally have higher average recall than the names from the 2000s era.

4.5.3 Recall along demographic dimensions

We then consider the difference in recall by the models on each name set and plot Figure 4-3. Along the dimension of gender (Figure 4-3a), most models appear to have little difference in performance between male and female names. Similarly, recall be-

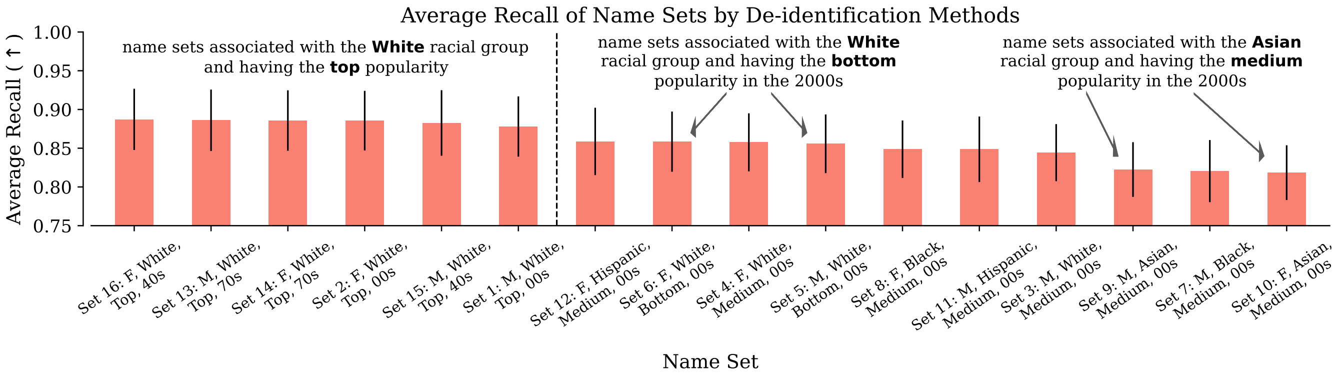


Figure 4-2: Average recall and standard error of each name set by the examined de-identification methods, ordered by decreasing recall. The average recall on name sets with top popularity exceeds the other sets by a clear margin. Moreover, the methods are, on average, more capable of recognizing less popular names associated with the White racial group compared to more popular names associated with the Asian racial group. Reproduced from Xiao et al. (2023) [66].

tween different decades of name popularity also does not appear to very within models (Figure 4-3d). Interestingly, when considering the race demographic (Figure 4-3b), we note that most of the models have a lower recall on names associated with minority racial groups, including Black and Asian groups, even though the names in each non-White racial group are from the same popularity level in the 2000s. One partial exception is Hispanic names, which often have comparable–and often improved–recall relative to the White names. Along the dimension of popularity (Figure 4-3c), we observe that most models achieve a higher recall on more popular names, with a more notable drop in recall between name sets of top and medium popularity as compared to the drop between name sets of medium and bottom popularity.

4.6 Mitigating bias

We explore a fine-tuning setup and find that it not only improves the overall recall of the models tested but also reduces the bias significantly along most demographic dimensions.

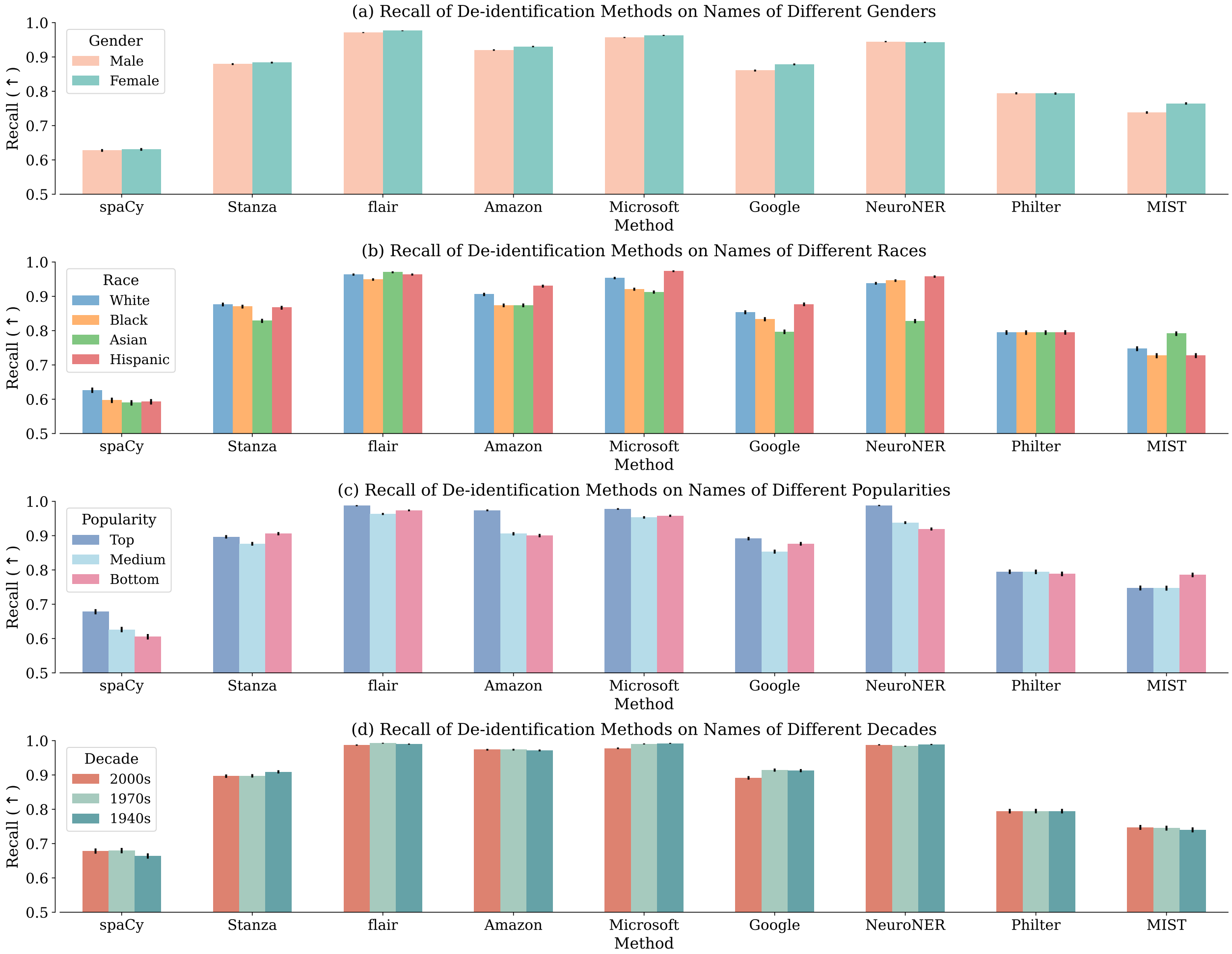


Figure 4-3: Recall and 95% bootstrapped confidence interval of the demographic groups along each dimension by each audited de-identification method. Disparities in performance between different groups are more observable along the dimensions of race and popularity than along the dimensions of gender and decade. Reproduced from Xiao et al. (2023) [66].

4.6.1 Fine-tuning de-identification methods

We prepare the fine-tuning de-identification datasets by considering two types of context and two types of names: general/clinical and popular/diverse. We treat the longitudinal clinical narratives in the 2014 i2b2 de-identification challenge as the clinical context and the Wikipedia articles in the DocRED dataset [67] as the general context. We define "diverse" here as a set of names composed of 16 subsets randomly sampled from each of the 16 name sets in Table 3.2, and "popular" as a set of names randomly sampled from the most popular names over the three chosen decades that do not appear in the 16 name sets. We generate diverse and popular name sets of 160 names each, with 10 names from each of the 16 name sets in the diverse set.

For each type of context, we randomly sample 1000 templates for training and 100 for validation. We then fill in each template with names from either the diverse or popular name sets. We thus end up with four fine-tuning setups: general-popular, general-diverse, clinical-popular, and clinical-diverse. We also generate 1600 test notes by filling in the 100 validation templates with the remaining 160 names not selected for the diverse names set, populating each template with each of the 16 name sets in Table 3.2. With this preparation, the test notes do not overlap with the fine-tuning context or names.

To compare the effectiveness of these setups, we fine-tune two de-identification methods with different performances from the previous analyses ("out-of-the-box"): spaCy and NeuroNER. SpaCy is a widely-adopted NLP library that has a low de-identification recall and a moderate demographic bias observed in Table ??. In contrast, NeuroNER has been pre-trained on the original 2014 i2b2 de-identification corpus, and it yields a fairly high recall and high bias along the dimensions of race and popularity. After fine-tuning with their respective default hyperparameters, these methods are evaluated on the test notes.

4.6.2 Clinical context and diverse names improve performance

Table 4.2 displays the overall performance and the demographic bias (measured using recall equality difference) of the two methods, spaCy and NeuroNER, after fine-tuning. Interestingly, despite distinct out-of-the-box performance for the two fine-tuned methods, the setup comprising clinical context and diverse names largely enhances the overall performance of both methods and diminishes their bias, especially along the dimensions of race and popularity.

In particular, although most of the fine-tuning setups improve spaCy's overall performance, fine-tuning with clinical context and diverse names results in the largest boost in spaCy's recall by over 0.3. On the other hand, most of the four fine-tuning setups do not improve NeuroNER's strong out-of-the-box performance, likely due to

Table 4.2: Overall performance (higher is better) and bias along demographic dimensions (lower is better) of two de-identification methods fine-tuned with different setups. We measure the bias with recall equality difference and bold the best score in each column for each method. For both methods, using clinical context and diverse names for fine-tuning improves the overall performance and reduces the demographic bias along most dimensions, especially race and popularity. Reproduced from Xiao et al. (2023) [66].

	Fine-tuning Setup		**Overall Performance**			**Bias along Dimensions**			
Method	**Context**	**Name**	**Precision**	**Recall**	**F1**	**Gender**	**Race**	**Popular**	**Decade**
spaCy	out-of-the-box		0.916	0.623	0.741	**0.003**	0.027	0.025	0.005
	clinical	diverse	0.984	**0.958**	**0.971**	0.009	**0.019**	**0.003**	0.004
	clinical	popular	0.999	0.729	0.843	0.009	0.101	0.131	**0.003**
	general	diverse	**1.000**	0.672	0.804	0.039	0.061	0.122	0.016
	general	popular	0.997	0.542	0.702	0.007	0.095	0.284	0.005
NeuroNER	out-of-the-box		0.955	0.953	0.954	0.005	0.044	0.030	0.001
	clinical	diverse	0.982	**0.986**	**0.984**	0.006	**0.011**	**0.011**	**0.000**
	clinical	popular	**0.994**	0.875	0.930	0.011	0.055	0.126	0.001
	general	diverse	0.975	0.896	0.934	0.025	0.066	0.067	0.015
	general	popular	0.921	0.772	0.840	**0.002**	0.061	0.337	0.003

its already having been pretrained with clinical text. The only exception is fine-tuning with clinical context and diverse names, which increases precision, recall, and F1 by 0.03 across the board.

Considering the change in bias along each dimension across fine-tuning, the most noteworthy changes are those of the dimensions of race and popularity, where the initial high bias is reduced by more than half after fine-tuning with clinical-diverse data. The fine-tuning setup did not affear to affect bias along the dimension of gender as much, with bias even increasing with many setups.

4.7 Conclusion

In this chapter, we have shown that de-identification methods can be biased along certain demographic dimensions, and the bias can be mitigated by fine-tuning with clinical context and diverse names. We suggest that fine-tuning de-identification methods with clinical context and diverse names should be done as an initial fix to improve fairness before the methods are applied to clinical tasks. The ability to equitably de-identify patient data would allow for the data's wider circulation in research, enabling the use of more diverse datasets to train machine learning models in healthcare and potentially mitigating the effects of medical bias on future predictions.

Chapter 5

Developing a de-identification package

5.1 Aim

In this chapter, we present the development of a de-identification package, HIPAAway. The package is designed to be simple and easy to use, providing the ability to use and combine different machine learning approaches to de-identification. Upon completion, the package will be installable from the Python Package Index (PyPI) and will support user customization for different de-identification needs. We describe the overall workflow, design decisions, and implementation details of the package. We also analyze and report the performance of multiple de-identification approaches provided through the package.

5.2 Software specifications

5.2.1 Objectives

After reviewing the current state of de-identification software and discussing with colleagues, we identified the following objectives:

- Create an open-source package that will de-identify a patient dataset according to a defined set of patient identifiers.

- Create a set of methods to train a model on local patient health data so that the package is adaptable to regional and system-specific variations.
- Add functionality to the package to allow users to generate a report on the levels of PHI in an input dataset (e.g., number and category of PHI entities).
- Create a testing framework to assess performance against benchmark datasets in terms of commonly reported metrics, such as precision, recall, and F1 score.

These objectives are used to guide the development of the package, and while the package has not yet been completed, we have made significant progress toward achieving these objectives.

5.2.2 Assumptions

The following assumptions have been made in the development of the package:

- The package will initially run on English-language data only, but it will be designed to be easily extensible to other languages.
- The package will be developed in Python 3.7+ and will be distributed via the Python Package Index (PyPI) and Conda package managers.
- Users will be familiar with running software in a Python interpreter on MacOS, Unix, or Windows machines.
- The software will be able to process both structured and unstructured data.
- Minimum hardware specification for applying the software to a dataset for de-identification will be an entry level CPU with 8 GB RAM. Training new models and achieving best possible performance metrics may require higher-spec machines with GPU.

- Runtime for training models and applying them to data will vary on hardware specification. The runtime should allow a relatively large corpus (e.g. 100,000 records of 1000 words) to be processed within 24 hours.

5.2.3 Risks

The following risks have been identified in the development of the package:

- The HIPAA guidelines only broadly define PHI, leaving room for interpretation of the details. We will draw on expert knowledge as appropriate.
- It is not possible to promise perfect performance (i.e. 100% precision and recall), so there should be an expectation that there will be false positives (i.e. non-PHI labelled as PHI) and false negatives (i.e. PHI not labelled as PHI).

5.2.4 Intended use

We intend for the package to provide the following de-identification workflow: loading and pre-processing data, tokenization, (optional) model training/fine-tuning, (optional) model validation, annotation, and scrubbing. At each step of model training, validation, and prediction, the user can evaluate model performance. The workflow is summarized in Figure 5-1.

5.2.5 User stories

To guide the development of the package, we create a set of user stories, which are summarized in Table 5.1. The user stories are written according to needs that arise for different users at different stages in the de-identification pipeline.

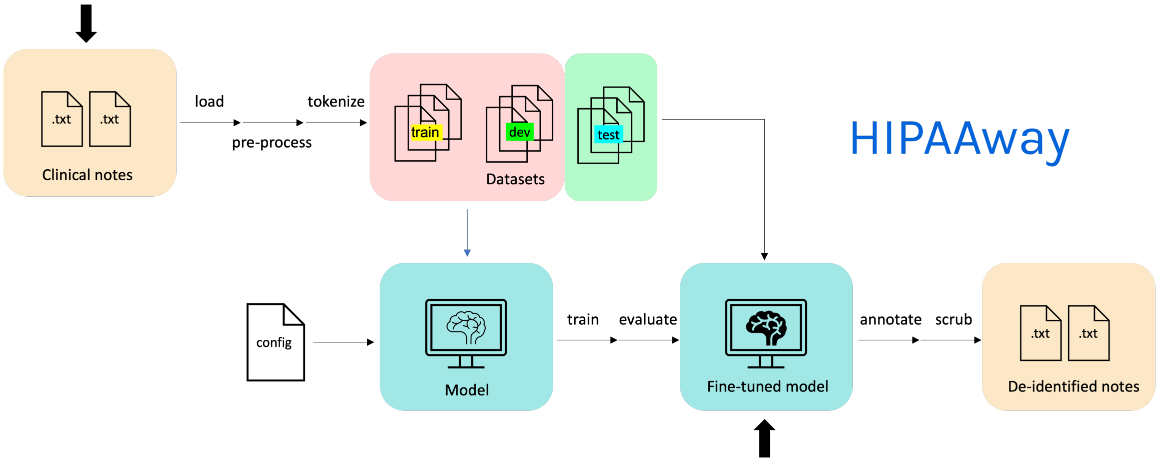


Figure 5-1: Overview of HIPAAway workflow. Users typically start with clinical notes in .txt, .csv, or .json files, which they can load, pre-process, and tokenize to produce Dataset objects, which can be split into training, development/validation, and test sets. The training and validation sets can be used to respectively train and validate a model, which is specified and loaded via a config file. The fine-tuned model can then be used to annotate and scrub PHI from the test set, which can be saved in the same format options as the input data. The large arrows indicate starting points in the workflow, as users can directly start from a fine-tuned model to annotate and scrub PHI from new data.

Table 5.1: User stories for HIPAAway. The user stories are generally organized by the steps in the de-identification workflow. The user stories are written in the format of "As a: [user], I need: [functionality], so that: [reason].

ID	**Story**	**Acceptance Test/s**
S1: Load data	**As a:** user of the de-identification package **I need:** a method to load my dataset so that it can be processed. **So that:** I can identify and remove PHI from the dataset.	T1: Load and parse txt, csv, and json files correctly.
S2: Annotate PHI	**As a:** user of the de-identification package **I need:** a method to annotate PHI within my dataset **So that:** I can review the annotations.	T2: Display the annotations in an understandable format.

ID	Story	Acceptance Test/s
S3: Scrub PHI	**As a:** user of the de-identification package **I need:** a method to scrub PHI from my data **So that:** I can generate a de-identified version of the data.	T3: Does the software remove PHI from my pre-generated test set as expected?
S4: Output data	**As a:** user of the de-identification package **I need:** to output my scrubbed data to file and optionally a file containing the annotations **So that:** I can share the de-identified version of the data.	T4: Does the software allow the de-identified data to be outputted to plain text?
S5: Output validation	**As a:** user of the de-identification package **I need:** a way to measure the degree of confidence that the output dataset is de-identified to an agreed standard **So that:** I can have confidence that I know the PHI levels in the output dataset.	T5: Does the software reach an acceptable minimum performance on my benchmark dataset?
S6: Package load	**As a:** person wanting to use the de-identification package **I need:** to be able to install the package into my development environment **So that:** I can use the tool.	T6: Users are able to install the software on local systems.
S7: Reporting	**As a:** non-technical dataset owner **I need:** to know the PHI levels pre- and post-running the tool as a human-readable report output **So that:** I can have assurance that the PHI in my data are at a certain level.	T7: Summary reports can be created using the software.

ID	Story	Acceptance Test/s
S8: Awareness and terms of use	**As a:** user and consumer of the tool or reports **I need:** to be aware and/or confirm that I understand that not all PHI data may be removed **So that:** I know and have acknowledged that the output dataset may still contain PHI.	T8: The software and output reports include clear guidance on terms of use.
S9: Documentation	**As a:** user of the software or reports **I need:** documentation **So that:** I clearly understand how it works.	T9: Documentation is publicly available.
S10: Publication	**As a:** user of the tool or reports **I need:** to be able to cite an authoritative source that describes the software **So that:** I can provide reassurance of the quality of my de-identification approach.	T10: Paper submitted to journal.

5.3 Implementation

In this section, we describe the development of HIPAAway. We first describe the NLP libraries we use and then detail the implementation of the de-identification pipeline.

5.3.1 Choosing a library

When developing software employing natural language processing, choosing the right NLP library is important, as the library provides developers with a wide range of algorithms for building and applying machine learning models. Which NLP library performs best depends on the task (e.g. tokenization or part-of-speech tagging) and the source (e.g. clinical text, which includes domain-specific technical language). Despite this, Omran and Treude (2017) [5] have discovered that only a small minority of software engineering conference papers that mention "natural language" mention the NLP library used, and an even smaller proportion justify their choice of the library. For the development of HIPAAway, we first build a framework using spaCy, which Omran and Treude (2017) [5] found to have the most promising performance on the NLP tasks they studied. After discovering some limitations with spaCy for our intended implementation, we then switch to the Huggingface Transformers library [63], which provides tools to easily download and train state-of-the-art pretrained models.

Initial approach: spaCy

The initial de-identification package work focused on building a pipeline in spaCy [18], which provides powerful language processing pipelines that take in text and return a processed Doc object. The Doc object contains a list of Token objects, which contain information about the tokenized text, such as the token's text, start and end indices, and label. Doc object text can undergo tokenization, part-of-speech tagging, named entity recognition, and other processes depending on the pipeline. spaCy also offers ways to incorporate pattern-matching approaches to the pipeline with components like Matcher and EntityRuler.

Challenges using spaCy

After building out a de-identification pipeline in spaCy, we evaluated the performance of the pipeline on the i2b2 2014 de-identification challenge data. We found that the pipeline performance on the data fell short of the state of the art. We identified several challenges with spaCy that contributed to this poor performance.

spaCy's available models for NER are limited. The NER model is based on either a CNN or transformer (RoBERTa) architecture. We found that that the CNN-based pipeline had poor performance, so we focused on the transformer-based pipeline. We used spaCy's wrapper for the Huggingface Transformers library [63] to fine-tune different transformers on the i2b2 2014 de-identification challenge data. However, we found that the fine-tuned models had poor performance on the i2b2 test data. We concluded that the poor performance of the fine-tuned models was due to the inability to use the NER heads from the pre-trained Huggingface transformers, losing valuable information from pre-training. While some modifications could be made to obtain predictions from the full Huggingface pre-trained model, these modifications would prevent the model from being trainable in spaCy. Based on these factors, we looked to building out the de-identification on top of the Huggingface transformers library directly.

The Huggingface Transformers library

The HuggingFace Transformers library is a popular Python library that provides APIs and tools to easily access and train state-of-the-art pretrained models.[1] The models support a broad range of machine learning tasks in different modalities, including natural language processing. Within NLP, models can be used for tasks such as text classification, named entity recognition, and question answering. The library also supports framework interoperability between PyTorch, Tensorflow, and JAX, three widely used deep learning frameworks. Similar to spaCy, the Transformers library also provides a pipeline that takes in text and returns the processed text. The pipeline supports a variety of tasks, including named entity recognition.

HIPAAway does not use the Huggingface pipeline directly, instead implementing each step in the pipeline separately for more flexibility. To transition from the spaCy workflow to Huggingface would be facilitated with a spaCy Doc-like object to represent each note. We use the Huggingface Dataset object, which–while it does not split up notes into separate objects–has access to the data and methods needed for implementing the de-identification workflow. The steps are detailed below.

[1] https://huggingface.co/transformers/

5.3.2 Pre-processing data

HIPAAway supports loading data in multiple file type formats, specifying an annotation format (if applicable) for each of .txt, .csv, and .json files. For data pre-processing, HIPAAway supports sentence and word tokenization, as well as the ability to split data into training, validation, and test sets. More details on data loading formats and pre-processing are provided in Section 4. The data is annotated using the BIO scheme, which annotates each token with B (beginning of a PHI entity), I (inside a PHI entity), or O (outside a PHI entity).

5.3.3 Tokenization

We define tokenization here as the model-dependent process of splitting text into tokens. HIPAAway uses the Huggingface Tokenizers library for tokenization[2]. As defined by the Tokenizers library, the tokenization involves the following steps: normalization, pre-tokenization, model, and post-processing. Normalization involves a set of operations to a raw string to make it "cleaner," including operations like removing whitespace and lowercasing text. Pre-tokenization splits the text into smaller parts that can be considered "words," of which the final tokens will be a part. The model step is when the `Tokenizer` applies a given model on the pre-tokens and is the part that requires either training on the corpus or a pretrained tokenizer. Currently, the Tokenizers library supports the following tokenization methods: WordLevel, byte-pair encoding (BPE), Unigram, and WordPiece. WordLevel simply maps words to IDs, while the other three options are subword tokenization algorithms, which further split words into smaller tokens. Finally, the last step of the tokenization pipeline is post-processing, which includes any additional transformations to the encoding, such as adding special tokens that indicate the beginning or end of sentences. HIPAAway uses the pre-trained tokenizers associated with each model in the Transformers library; for example, the BERT model has a corresponding BERT tokenizer that relies on the WordPiece algorithm.

[2] https://huggingface.co/docs/tokenizers/index

5.3.4 Training (fine-tuning)

HIPAAway allows users to specify any model in the Huggingface Transformers library they wish to use, as long as the model is suitable for the task of token classification (named entity recognition). Because the vast majority of the models available are not pre-trained in the clinical domain, fine-tuning–even if only based on a few examples with PHI tags–is crucial. That said, the package makes this step optional, allowing users to skip fine-tuning and use the desired model as-is, with the recommendation that they use a model that has been fine-tuned on a clinical de-identification task.

HIPAAway uses the Huggingface `Trainer` class[3] to train the models, which provides an API for training in PyTorch. In addition to training, the `Trainer` class also handles model evaluation and prediction. Users can specify training arguments such as number of training epochs, batch size, and learning rate.

Saving and loading models

Following fine-tuning, models can be saved locally or in a private Huggingface repository. To access these models, users can specify a local filepath or a Huggingface repository name in the config file as input into the de-identification pipeline.

5.3.5 Validation

Validation is an optional but recommended step in HIPAAway. The package uses the `Trainer` class's built-in evaluation method to evaluate the model on the validation data. Users can evaluate their models on validation data set aside from the training data or the training data itself (not recommended) to tune hyperparameters such as the training arguments mentioned above: number of training epochs, batch size, and learning rate.

[3] https://huggingface.co/docs/transformers/main_classes/trainer

5.3.6 Prediction

The prediction step of HIPAAway uses the `Trainer` class's built-in prediction method to predict instances of PHI on either test data or new, unannotated data. The package provides an `annotate` method that takes in a `Dataset` object and returns a list of predicted labels for each full-word token in the dataset.

5.3.7 Aggregation of predictions

HIPAAway allows for ensembling models by aggregating predictions from multiple models. The package provides a `combine_annotations` method that takes in a list of predictions from different models and returns a list of predicted labels for each full-word token in the dataset. The aggregation method is majority vote, and other methods may be added in the future.

5.3.8 Evaluation

The models are evaluated using micro-averaged precision, recall, and F1 score, so each prediction for a token is weighted equally. We focus on recall and F1 score, as recall emphasizes false negatives (missed PHI), and F1 score is the harmonic mean of precision and recall.

In NER, there are two levels of evaluation: entity and token. Entity-level evaluation is the standard in NER, where a model is evaluated on whether it correctly predicts the entire entity. Token-level evaluation is more granular, as the model is evaluated on whether each token of an entity is correctly predicted. For example, given the ground truth label for "Jane Doe" is "B-NAME-PATIENT I-NAME-PATIENT," a model predicts "B-NAME-PATIENT O." The model would be considered correct for "Jane" but not "Doe" in token-level evaluation, and it would be entirely wrong under entity-level evaluation. HIPAAway uses entity-level evaluation, as it is important to correctly predict the entire PHI instance, not just some of its tokens.

5.3.9 Scrubbing

HIPAAway enables the scrubbing of PHI from data using the `scrub` method, which takes in a Huggingface Dataset object with annotated text to be scrubbed, the scrubbing method, and optional parameters for the scrubbing method. The package provides two scrubbing methods: `remove` and `replace`. The `remove` method removes all tokens with PHI tags from the text, in effect replacing them with empty strings. The `replace` method replaces all tokens with PHI tags with a specified replacement token. The replacement token can be the same for all PHI types (e.g. a string of underscores, ’___’), or it can be an indicator for each class (e.g. ’NAME-PATIENT’ for all tokens with the ’NAME-PATIENT’ tag).

5.4 Package evaluation

5.4.1 Performance on i2b2 2014 dataset

We evaluate the performance of HIPAAway on the i2b2 2014 dataset and compare it with the performance of other models in literature, namely the ones evaluated in Johnson et al. (2020). We evaluate for precision, recall, and F1 score with a focus on recall and F1 score, as recall emphasizes false negatives (missed PHI), and F1 score is the harmonic mean of precision and recall. The results are shown in Table 5.2. Note that the goal of this project is to develop software to allow de-identification models to be applied rather than developing the algorithms themselves.

One issue in evaluating performance is the lack of a standardized way to calculate metrics, which we highlight in Chapter 2 when reviewing currently available de-identification software. The metric calculation used by HIPAAway is `seqeval` [41], which underpins the Huggingface `Trainer` class’s evaluation method by default. `seqeval` is a Python framework for evaluating sequence labeling tasks like named entity recognition. `seqeval` calculates precision, recall, and F1 score at the entity level, in line with CoNLL-2000 [50] and other benchmark NLP tasks. However, de-identification papers in literature often calculate metrics at the token level due to

Table 5.2: Performance of HIPAAway on the i2b2 2014 dataset compared with other models in the literature. The best performance from past literature and HIPAAway are in bold. Note that the goal of this project is to develop software to allow de-identification models to be applied rather than developing the algorithms themselves. *: Due to computational constraints, we were unable to fine-tune and evaluate the BERT-large, uncased model on the i2b2 2014 dataset in this study.

Model	**Reference**	**Precision**	**Recall**	**F1-Score**	**Eval**
BERT-large, uncased*	Johnson et al.	**0.987**	**0.982**	**0.984**	token
BERT-large, cased	Johnson et al.	0.986	0.978	0.982	token
	HIPAAway	0.961	0.941	0.951	entity
BERT-base, uncased	Johnson et al.	0.986	0.979	0.983	token
	HIPAAway	0.963	0.959	0.961	entity
BERT-base, cased	Johnson et al.	0.984	0.974	0.979	token
	HIPAAway	0.957	0.944	0.950	entity
RoBERTa-base	HIPAAway	**0.970**	0.965	**0.967**	entity
RoBERTa-large	HIPAAway	0.963	**0.968**	0.965	entity
AlBERT-base	HIPAAway	0.961	0.949	0.955	entity
DistilBERT-base, uncased	HIPAAway	0.955	0.954	0.955	entity
DistilBERT-base, cased	HIPAAway	0.954	0.941	0.947	entity

an emphasis on catching any PHI tokens rather than getting entire PHI entities correct. For example, Johnson et al. (2020) [21] evaluate metrics at the token level, so their metrics are not directly comparable to ours. However, they serve as a helpful ballpark estimate of performance. While the transformer models tested using the HIPAAway framework have overall lower metrics than the models in HIPAAway, we view our current metrics as a lower bound on performance. We hypothesize that the lower metrics are due to the entity-level calculation of our metrics as opposed to the compared token-level metrics. We also note that the models in HIPAAway are trained using default hyperparameters, and we expect that performance can be improved with hyperparameter tuning. We leave the calculation of token-level metrics and hyperparameter tuning for future work.

5.4.2 Qualitative analysis of missed PHI

We continue with a qualitative analysis of PHI missed by language models in HIPAAway (false negatives), with a focus on the RoBERTa-base model, as it had the highest

F1 score in our analysis. Similar to previous work in Dernoncourt et al. (2017) [10], we notice patterns in the PHI missed. Dernoncourt et al. (2017) groups the sources of errors into four main categories described as follows:

- Abbreviations: PHI instances that are abbreviations are sometimes challenging to detect, especially when they are short and ambiguous.
- Ambiguities: Some PHI instances may not even be recognized as PHI by humans due to uncertainty, such as common words or numbers that could be dates (PHI) or test results (not PHI).
- Data sparsity: The training data may not contain many PHI instances similar to certain ones that are missed in the test set.
- Debatable annotations: Some tokens marked as PHI instances could also be considered as not PHI, such as names of medical conditions that happen to have common human names.

We hypothesize that the missed PHI can be categorized into the above categories and further extend the categories to include the following:

- Placement: PHI instances may be missed if they are split over multiple lines, especially in the case of sentence tokenization, when inter-sentence relationships between tokens are not taken into account.

Based on these definitions, we categorize the PHI missed by the RoBERTa-base model into the five categories: abbreviations, ambiguities, data sparsity, debatable annotations, and placement. We provide examples of each category in Table 5.3. Notably, an abbreviation for a doctor's name is missed ("O"), and we point out the ambiguity of the missed DATE entity, which only includes the last two digis of the year (i.e. 85 for 1985), which can be confused for a lab result or other non-PHI metric. We also highlight a missed AGE entity in the format "[age]y[months]m" (i.e. 56y0.5m), which is not a common age format and possibly appears little to no times in the training data. There are two "debatable" PHI annoations included: (1)

Table 5.3: Sample PHI missed by HIPAAway with context and hypothesized reason missed.

PHI type	**Missed PHI with context**	**Reason**
DOCTOR	Call me in a week to tell me how it is goin. Best wishes, **O**	Abbreviation
DATE	Has now gained 16lbs in less than a month 4/04/85 Cancelled two attempts for EGD/colonoscopy **85**	Ambiguity
AGE	11/18/2069 NEGATIVE Vital Signs BLOOD PRESSURE 128/68 WEIGHT 210 lbAGE **56y0.5m** Physical Exam Stable fatigued appearance NAD	Data sparsity
PROFESSION	Pt lives alone and has 2 daughters who live nearby on the same street. Former Computer **and** Network Operator. Smoked 1 PPD x 20 years, but stopped many years ago.	Debatable
ORGANIZATION	Improving tolerance of CPAP seems very key. Pt has started **atkins** dt with good success (6# in first week)	Debatable
PATIENT	It is certainly a pleasure for me to participate in **Shilpa** \n **Pickett**' care.	Placement

an example of the word "and" in a profession being considered PHI, which seems unnecessary, albeit likely not to impact downstream clinical model predictions on the de-identified text, and (2) an example of the word "atkins" in a diet being considered PHI, which is debatable as it is a common diet and not a person's name. The flagging and subsequent scrubbing of "atkins" in this context as PHI could potentially impact predictions made on this patient, as the diet is relevant to the patient's health. Finally, we include an example of a PATIENT entity missed likely due to its nature of having a line break in the middle, preventing the model does not recognize the entity as a whole.

This qualitative analysis of missed names highlights current hypothesized sources of error for de-identification models, only some of which can be easily mitigated. In the case of data sparsity, our proposed solution would be to increase the representation of data with similar formats to the missed PHI to improve their detection rate. This would work easily for numerical PHI, where structure rather than content is important for detection. In the case of PHI like names, content is more important than structure, and while at issue here is potentially also data sparsity, it is more difficult to anticipate all possible expected names in the training data. The other sources of error for missed PHI are more difficult to address, and in particular, the "debetable" annotations highlight the need for a standardized technical specification for PHI.

5.5 Discussion

In this work, we develop HIPAAway, a de-identification software tool that can be used to remove PHI from clinical text. We evaluate the performance of HIPAAway on the i2b2 2014 dataset and find that the best-performing models after fine-tuning are RoBERTa-large and RoBERTa-base, which achieve the highest recall (0.968) and F1 score (0.967), respectively, out of the models analyzed. These metrics are competitive for de-identification performance and are a promising start for the development and release of high-performing pre-trained models as part of HIPAAway.

That said, we have come to believe that consistently achieving 100% accuracy may remain an unattainable task. In addition to the possibility of new instances of PHI unlike anything seen before, the presence of PHI that are ambiguous and debatable means that there is no single "correct" answer for de-identification, complicating the process and evaluation of de-identification.

If de-identification models are unable to reach 100% accuracy, we have a problem. An accuracy of 99.9% would mean that for every 10,000 instances of PHI, perhaps 10 would be missed. These missed PHI instances have the possibility of being very rare names or other unique identifiers that would put their respective patients at a disproportionally high risk of de-identification. This is why we emphasize that even though HIPAAway is intended to remove PHI as defined by HIPAA, careful human review of all annotations is paramount.

Furthermore, HIPAA-defined de-identification works for data in the United States, but other countries (or continents) may have more stringent privacy rules, such as Europe's GDPR, the "toughest privacy and security law in the world."[4] One example of GDPR being stricter than HIPAA is in the case of anonymization vs. pseudonymization, both of which are possible methods of de-identification. In anonymization, data is fully scrubbed for any PHI. GDPR defines pseudonymization as the following: ”the processing of personal data in such a way that the data can no longer be attributed to a specific data subject *without the use of additional information.*” The "additional information" must be kept separately and is subject to measures to ensure that the personal data cannot be attributed to a person. While HIPAA allows de-identification in the forms of both anonymization and pseudonymization, GDPR considers pseudonymous data as still personal data that cannot be shared in the same way that anonymous data can.

One concern is that with the advances in natural language processing, the definition of pseudonymization may need to become broader in scope, and the task of de-identification in its current form may become nearly impossible. Taking a stringent view of privacy, it is possible to consider a free-text clinical note's very structure as the unique "noteprint" that serves as the key to be matched back to the original note. This means that the PHI-containing electronic medical record itself is the "additional information" that could turn de-identified text previously considered anonymous to pseudonymous instead and thus personal data covered by GDPR. This is why the package name of HIPAAway highlights the removal of HIPAA identifiers, which is not a complete solution but rather a first step of the de-identification task. We believe that the de-identification task will need to be redefined in the future to account for the possibility of the note itself being the key to re-identification via its unique noteprint.

[4] https://gdpr-info.eu/

Chapter 6

Future work and conclusion

In this thesis, we present contributions in the following areas: a review of the current state of the art in de-identification software, a new de-identification dataset, an audit of biases in existing de-identification tools, and a new de-identification software package. We conclude with a discussion of future work.

6.1 Future work

6.1.1 Package improvements

Looking ahead, two major avenues of enhancement for the package are improving the de-identification models' performance and increasing ease of use. Previous research has shown that hybrid approaches of pattern matching and machine learning can have better performance than each method on its own, though the hybrid approaches have only used LSTMs, not transformers [32]. We plan to incorporate pattern matching approaches into the package, which–when combined with the package's transformers models–also have the potential to improve performance. We also hope to apply the lessons learned from the bias audit and take a step toward mitigating biases in our own fine-tuned models by further fine-tuning them with a more diverse set of data. Finally, future work on the package could include designing a graphical user interface for the software, which would make it easier for non-technical users to use the package to de-identify their data.

6.1.2 Standards for de-identification

Considering the variability in PHI definitions, dataset attributes, and calculation of de-identification matrics, de-identification has much room for standardization. The lack of standardization not only makes comparison of de-identification models different, but it also leaves uncertainty concerning the degree to which data has been sufficiently de-identified under HIPAA. We propose for future work to develop a detailed technical specification for PHI that would make PHI annotations more uniform across datasets. We also suggest the curation of a single corpus of data which combines text data from all publicly available clinical datasets. The resulting corpus would contain notes from different clinical settings and medical disciplines and, when used for model training, may facilitate better generalizability of de-identification models. Lastly, we call for the identification of a standard set of de-identification metrics, with clear steps on how to calculate each one, as well as the potential development of a light software tool to calculate these metrics.

6.1.3 Beyond HIPAA de-identification

In light of some of the challenges facing de-identification software highlighted in Chapter 5, we come to a broader question: is de-identification enough? We believe the answer is no; de-identification needs to be augmented with methods that help prevent re-identification and disclosure of patient identifiers. One example method that would help is the "hiding in plain sight" approach: surrogate names are inserted in the place of real names so that if some real names are missed, they can "hide" among the artificially inserted identifiers. Another option is the generation of synthetic clinical notes, which shows more promise as large language models become increasingly more adept at generating realistic data in all types of domains.

6.2 Conclusion

This thesis has presented a review of the current state of the art in de-identification software, a new de-identification dataset, an audit of demographic biases in existing de-identification tools, and a new de-identification software package. We hope that the audit will be a wake-up call for the de-identification community, as well as a framework for future audits. We also hope that HIPAAway will be useful to researchers and clinicians who need to de-identify their patient data to enable sharing. Finally, we hope that the work presented in this thesis will help to advance the field of de-identification and contribute to the curation of more secure and useful de-identified data.

Resources

The thesis received proofreading help from my supervisors and ChatGPT.

Bibliography

[1] John Aberdeen, Samuel Bayer, Reyyan Yeniterzi, Ben Wellner, Cheryl Clark, David Hanauer, Bradley Malin, and Lynette Hirschman. The MITRE Identification Scrubber Toolkit: Design, training, and assessment. *International Journal of Medical Informatics*, 79(12):849–859, December 2010.

[2] Abdullah Ahmed, Adeel Abbasi, and Carsten Eickhoff. Benchmarking Modern Named Entity Recognition Techniques for Free-text Health Record Deidentification. *AMIA Summits on Translational Science Proceedings*, 2021:102–111, May 2021.

[3] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[4] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual String Embeddings for Sequence Labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[5] Fouad Nasser A Al Omran and Christoph Treude. Choosing an NLP Library for Analyzing Software Documentation: A Systematic Literature Review and a Series of Experiments. In *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*, pages 187–197, May 2017.

[6] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, April 2017.

[7] Mhairi Campbell, Srinivasa Vittal Katikireddi, Amanda Sowden, and Hilary Thomson. Lack of transparency in reporting narrative synthesis of quantitative data: a methodological assessment of systematic reviews. *Journal of Clinical Epidemiology*, 105:1–9, January 2019.

[8] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer,

and Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale, April 2020.

[9] Franck Dernoncourt, Ji Young Lee, and Peter Szolovits. NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 97–102, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[10] Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606, May 2017.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, May 2019.

[12] Margaret Douglass. Computer-Assisted De-Identification of Free-text Nursing Notes.

[13] Luciano Floridi and Massimo Chiriatti. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds and Machines*, 30(4):681–694, December 2020.

[14] Milton Friedman. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32(200):675–701, December 1937.

[15] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet. *Circulation*, 101(23):e215–e220, June 2000.

[16] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5):602–610, July 2005.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[18] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

[19] Rashmi Jain, Dinah Samuel Anand, and Vijayalakshmi Janakiraman. Scrubbing Sensitive PHI Data from Medical Records made Easy by SpaCy – A Scalable Model Implementation Comparisons, June 2019.

[20] Victor Janmey and Peter L. Elkin. Re-Identification Risk in HIPAA De-Identified Datasets: The MVA Attack. *AMIA Annual Symposium Proceedings*, 2018:1329–1337, December 2018.

[21] Alistair E. W. Johnson, Lucas Bulgarelli, and Tom J. Pollard. Deidentification of free-text medical records using pre-trained bidirectional transformers. *Proceedings of the ACM Conference on Health, Inference, and Learning*, 2020:214–221, April 2020.

[22] Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(1):160035, May 2016.

[23] Mehmet Kayaalp. Patient Privacy in the Era of Big Data. *Balkan Medical Journal*, 35(1):8–17, January 2018.

[24] Athar Khodabakhsh, Ismail Ari, Mustafa Bakır, and Serhat Murat Alagoz. Forecasting Multivariate Time-Series Data Using LSTM and Mini-Batches. In Mahdi Bohlouli, Bahram Sadeghi Bigham, Zahra Narimani, Mahdi Vasighi, and Ebrahim Ansari, editors, *Data Science: From Research to Application*, Lecture Notes on Data Engineering and Communications Technologies, pages 121–129, Cham, 2020. Springer International Publishing.

[25] Vjeko Kužina, Eugen Vušak, and Alan Jović. Methods for Automatic Sensitive Data Detection in Large Datasets: a Review. In *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, pages 187–192, September 2021.

[26] John Lafferty, Andrew McCallum, and Fernando C N Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.

[27] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, February 2020.

[28] Ivano Lauriola, Alberto Lavelli, and Fabio Aiolli. An introduction to Deep Learning in Natural Language Processing: Models, techniques, and tools. *Neurocomputing*, 470:443–456, January 2022.

[29] Leibo Liu, Oscar Perez-Concha, Anthony Nguyen, Vicki Bennett, and Louisa Jorm. De-identifying Australian Hospital Discharge Summaries: An End-to-End Framework using Ensemble of Deep Learning Models. *Journal of Biomedical Informatics*, 135:104215, November 2022. arXiv:2101.00146 [cs].

[30] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, July 2019.

[31] Zengjian Liu, Yangxin Chen, Buzhou Tang, Xiaolong Wang, Qingcai Chen, Haodi Li, Jingfeng Wang, Qiwen Deng, and Suisong Zhu. Automatic De-identification of Electronic Medical Records using Token-level and

Character-level Conditional Random Fields. *Journal of biomedical informatics*, 58(Suppl):S47–S52, December 2015.

[32] Zengjian Liu, Buzhou Tang, Xiaolong Wang, and Qingcai Chen. De-identification of clinical notes via recurrent neural network and conditional random field. *Journal of Biomedical Informatics*, 75:S34–S42, November 2017.

[33] Zhengliang Liu, Xiaowei Yu, Lu Zhang, Zihao Wu, Chao Cao, Haixing Dai, Lin Zhao, Wei Liu, Dinggang Shen, Quanzheng Li, Tianming Liu, Dajiang Zhu, and Xiang Li. DeID-GPT: Zero-shot Medical Text De-Identification by GPT-4, March 2023.

[34] Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing Leakage of Personally Identifiable Information in Language Models, April 2023.

[35] Apar Madan, Ann Mary George, Apurva Singh, and M.P.S. Bhatia. Redaction of Protected Health Information in EHRs using CRFs and Bi-directional LSTMs. In *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pages 513–517, August 2018.

[36] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, 2014. Association for Computational Linguistics.

[37] Courtney Mansfield, Amandalynne Paullada, and Kristen Howell. Behind the Mask: Demographic bias in name detection for PII masking, May 2022.

[38] Christopher Meaney, Wali Hakimpour, Sumeet Kalia, and Rahim Moineddin. A Comparative Evaluation Of Transformer Models For De-Identification Of Clinical Text Data, March 2022.

[39] Ninareh Mehrabi, Thamme Gowda, Fred Morstatter, Nanyun Peng, and Aram Galstyan. Man is to Person as Woman is to Location: Measuring Gender Bias in Named Entity Recognition. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media*, HT '20, pages 231–232, New York, NY, USA, July 2020. Association for Computing Machinery.

[40] Shubhanshu Mishra, Sijun He, and Luca Belli. Assessing Demographic Bias in Named Entity Recognition, August 2020.

[41] Hiroki Nakayama. seqeval: A python framework for sequence labeling evaluation, 2018. Software available from https://github.com/chakki-works/seqeval.

[42] Ishna Neamatullah, Margaret M. Douglass, Li-wei H. Lehman, Andrew Reisner, Mauricio Villarroel, William J. Long, Peter Szolovits, George B. Moody, Roger G. Mark, and Gari D. Clifford. Automated de-identification of free-text medical records. *BMC Medical Informatics and Decision Making*, 8(1):32, July 2008.

[43] Beau Norgeot, Kathleen Muenzen, Thomas A. Peterson, Xuancheng Fan, Benjamin S. Glicksberg, Gundolf Schenk, Eugenia Rutenberg, Boris Oskotsky, Marina Sirota, Jinoos Yazdany, Gabriela Schmajuk, Dana Ludwig, Theodore Goldstein, and Atul J. Butte. Protected Health Information filter (Philter): accurately and securely de-identifying free-text clinical notes. *npj Digital Medicine*, 3(1):1–8, April 2020.

[44] Christine M O'Keefe and Chris J Connolly. Privacy and the use of health data for research. *Medical Journal of Australia*, 193(9):537–541, 2010. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.5694/j.1326-5377.2010.tb04041.x.

[45] OpenAI. GPT-4 Technical Report, March 2023.

[46] Cole Pearson, Naeem Seliya, and Rushit Dave. Named Entity Recognition in Unstructured Medical Text Documents. In *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pages 1–6, December 2021.

[47] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[48] Fernanda Polubriaginof, Nicholas P. Tatonetti, and David K. Vawdrey. An Assessment of Family History Information Captured in an Electronic Health Record. *AMIA Annual Symposium Proceedings*, 2015:2035–2042, November 2015.

[49] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online, July 2020. Association for Computational Linguistics.

[50] Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the CoNLL-2000 Shared Task: Chunking, September 2000.

[51] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, June 2003.

[52] Stefan Schweter and Alan Akbik. FLERT: Document-Level Features for Named Entity Recognition, May 2021.

[53] Kenneth P. Seastedt, Patrick Schwab, Zach O'Brien, Edith Wakida, Karen Herrera, Portia Grace F. Marcelo, Louis Agha-Mir-Salim, Xavier Borrat Frigola, Emily Boardman Ndulue, Alvin Marcelo, and Leo Anthony Celi. Global healthcare fairness: We should be sharing more, not less, data. *PLOS Digital Health*, 1(10):e0000102, October 2022.

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

[55] Nicholas W. Sterling, Rachel E. Patzer, Mengyu Di, and Justin D. Schrager. Prediction of emergency department patient disposition based on natural language processing of triage notes. *International Journal of Medical Informatics*, 129:184–188, September 2019.

[56] Amber Stubbs, Michele Filannino, and Özlem Uzuner. De-identification of psychiatric intake records: Overview of 2016 CEGS N-GRID shared tasks Track 1. *Journal of Biomedical Informatics*, 75:S4–S18, November 2017.

[57] Amber Stubbs, Christopher Kotfila, and Özlem Uzuner. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/UTHealth shared task Track 1. *Journal of Biomedical Informatics*, 58:S11–S19, December 2015.

[58] Amber Stubbs and Ozlem Uzuner. Annotating longitudinal clinical narratives for de-identification: the 2014 i2b2/UTHealth Corpus. *Journal of biomedical informatics*, 58(Suppl):S20–S29, December 2015.

[59] Konstantinos Tzioumis. Demographic aspects of first names. *Scientific Data*, 5(1):180025, March 2018.

[60] Özlem Uzuner, Yuan Luo, and Peter Szolovits. Evaluating the State-of-the-Art in Automatic De-identification. *Journal of the American Medical Informatics Association*, 14(5):550–563, September 2007.

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[62] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. OntoNotes Release 5.0, October 2013.

[63] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-Art Natural Language Processing, October 2020.

[64] R. F. Woolson. Wilcoxon Signed-Rank Test. In *Wiley Encyclopedia of Clinical Trials*, pages 1–3. John Wiley & Sons, Ltd, 2008.

[65] Michael Wornow, Yizhe Xu, Rahul Thapa, Birju Patel, Ethan Steinberg, Scott Fleming, Michael A. Pfeffer, Jason Fries, and Nigam H. Shah. The Shaky Foundations of Clinical Foundation Models: A Survey of Large Language Models and Foundation Models for EMRs, March 2023.

[66] Yuxin Xiao and Shulammite Lim. In the Name of Fairness:Assessing the Bias in Clinical Record De-identification. 2023.

[67] Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. DocRED: A Large-Scale Document-Level Relation Extraction Dataset, August 2019.

[68] Vithya Yogarajan, Bernhard Pfahringer, and Michael Mayo. Automatic end-to-end De-identification: Is high accuracy the only metric? *Applied Artificial Intelligence*, 34(3):251–269, February 2020.

[69] Yuhao Zhang, Yuhui Zhang, Peng Qi, Christopher D Manning, and Curtis P Langlotz. Biomedical and clinical English model packages for the Stanza Python NLP library. *Journal of the American Medical Informatics Association*, 28(9):1892–1899, September 2021.

[70] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.