RETHINKING CLINICAL DE-IDENTIFICATION

by

Mohamed Mohamed Saad Atia Abdalla

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy

Graduate Department of Computer Science
University of Toronto

Abstract

The restricted availability of free-text clinical notes and embedding-models trained on clinical notes is a bottleneck in deploying machine learning in clinical settings. To ameliorate concerns regarding the confidentiality of patient data, computer scientists have undertaken the task of developing methods which automatically remove sensitive personal information from notes. While these methods appear to perform exceedingly well, often with reported precision and recall well above 95%, automated approaches to clinical notes are still not trusted by clinicians because there remains non-zero risk.

In this work, we present various fundamental limitations associated with current approaches to de-identification that cannot be solved by incremental improvements to model performance. These limitations stem from the fact that current approaches are all trained in a supervised manner. To address these limitations, this thesis proposes the first unsupervised approach to the de-identification on free-text clinical notes. The proposed algorithm replaces all tokens with other tokens pseudo-randomly sampled from trained embeddings and is most useful for tasks where humans are not required to read the de-identified notes (e.g., training word embeddings for public release, piloting the feasibility of end-to-end machine learning models). Our approach successfully side-steps the issues facing supervised approaches (e.g., having to decide what constitutes sensitive personal information).

The second part of the thesis argues for an expansion to the scope of clinical de-identification. Whereas existing de-identification approaches focus solely on protecting the patient's identity, we argue that given the relationship between healthcare provider and patient, de-identification should also focus on protecting the identity of healthcare providers. First, we demonstrate that authorship attribution in clinical notes is a very easy task when compared to many traditional author attribution datasets. This demonstrates a need for specialized and improved author obfuscation techniques. However, the data to develop such

techniques is difficult to obtain due to privacy concerns; it is impractical to manually label paired sentences and difficult to crowd-source the task given data-sharing limitations. To enable the development of automated means of evaluating semantic relatedness, we developed a novel sentence-pair dataset ordered by semantic relatedness. This dataset can serve as a catalyst for future author obfuscation evaluation, and we draw insights from this dataset to better understand existing work.

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

Acknowledgments

The work in this thesis was financially supported by a Vanier Canada Graduate Scholarships, an Ontario Graduate Scholarship, and funding from the Vector Institute for Artificial Intelligence. The Vector Institute provided, in part, the computing resources for work in this thesis.

I am indebted to my supervisors Professor Graeme Hirst and Professor Frank Rudzicz for their supervision and guidance through my academic journey. Their feedback has been, and remains to be, invaluable in improving myself as a researcher. I would also like to express my gratitude to my committee members Dr. Liisa Jaakkimainen and Dr. Yang Xu as well as the internal examiner Dr. Muhammad Mamdani and external examiner Dr. Khaled El Emam. Their feedback has been instrumental to improving the thesis. I am thankful for my collaborators: Krishnapriya Vishnubhotla, Haoran Zhang, Dr. Moustafa Abdalla, and Dr. Saif Mohammed for the technical help, detailed discussions, and general support helped make this possible. To the many colleagues I've not published with, or whose papers couldn't fit in the theme of this thesis I am truly grateful for the opportunity to have worked with and learned from you.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Clinical applications of machine learning (ML) and natural language processing (NLP) have been growing alongside the increasing availability of clinical data and computational power. As the percentage of health-care facilities with digitally stored data nears 100% (Chang and Gupta, 2015), the potential impact on patients continues to grow. However, despite the widely reported advancements in available algorithms, as demonstrated by yearly articles proclaiming above-human performance on a wide variety of tasks (McKinney et al., 2020; Esteva et al., 2017), computational approaches to healthcare have largely remained niche and experimental, with few deployments (Joyner et al., 2016; Taylor and Fenner, 2018).

Why is this the case? Current thought hypothesizes that the lack of deployment is due to a wide variety of factors including the lack of context, especially during evaluation, resulting in hesitancy to adopt and test predictive algorithms in the clinical setting (Kelly et al., 2019; Cabitza et al., 2017; Taylor and Fenner, 2018). Researchers have also highlighted how the lack of interoperability (Joyner et al., 2016; Taylor and Fenner, 2018), poor data quality (Joyner et al., 2016; Cabitza et al., 2017), and privacy concerns (Joyner et al., 2016) affect the adoption of ML and NLP in the clinical setting.

The aim of this thesis is to develop and better evaluate approaches to protecting patient confidentiality. In this work, I purposefully differentiate between confidentiality and privacy. Traditionally, privacy was used to refer to the societal good (often considered a human right), whereas confidentiality referred to protecting sensitive information from adversarial actors. While more recent work uses the term "privacy" to refer to both these concepts (Evans, 2019), it is useful to differentiate between the concepts both as an attempt to pre-

vent further ethics-washing[1] by corporations (Evans, 2019; Abdalla and Abdalla, 2021), and because my work focuses concretely on confidentiality.

The over-arching hypothesis of this thesis is that the de-identification of free-text clinical notes can successfully be done using unsupervised methods, greatly reducing the amount of human effort required and that the resulting de-identified notes will be useful for a variety of end-to-end machine learning tasks. To explore this hypothesis, the thesis is split into three parts. In the first part, to motivate the need for unsupervised approaches to de-identification, I highlight the limitations of existing approaches, and present novel demonstrations of risk in **Chapter 2**. Having motivated the need for an unsupervised de-identification method, I present my work proposing a novel unsupervised de-identification technique (RaNNA) in **Chapter 3**. In **Chapter 4**, I explore the risks associated with RaNNA, and compare these risks relative to the risks associated with existing supervised de-identification techniques. In the second part of the thesis, I propose the expansion of scope for the goals of clinical de-identification techniques. To motivate this expansion, **Chapter 5** uncovers and demonstrates the risk associated with clinical author attribution. To facilitate the creation of automated natural language generation processes to perform author obfuscation (i.e., change style while maintaining the meaning) of clinical notes, **Chapter 6** presents a novel dataset of English sentence pairs annotated for semantic relatedness. The final part of the thesis, Chapter 7, summarizes the contributions of this work and plots a path for future research.

1.2 Background

The advent of electronically stored health data has led to an increase of researchers interested in applying ML techniques on this data to improve patient outcomes (Maclagan et al., 2021; Abdalla et al., 2017; Liaqat et al., 2019), increasing system efficiency (Abdalla et al., 2020c) or reducing cost (Zhang et al., 2019) among many other motivations. To facilitate this research while preserving the confidentiality of their patients, data holders apply different de-identification methods to reduce any risk to patient confidentiality associated with note sharing (Abdalla et al., 2020b; Neamatullah et al., 2008; Dernoncourt et al., 2017). This de-identification, often a legal obligation (Health Insurance Portability and Accountability Act, 2012b), can take various forms depending on the type of data and the stringency required by the law or by data holders. This thesis focuses solely on the de-identification of unstructured free-text clinical notes.

[1] Ethics-washing is the action of giving lip service to ethics (in this case: privacy) by companies to make it seem as though they are acting responsibly.

1.2.1 Terminology: Anonymization versus De-identification

Colloquially, and in many publications, the terms "anonymization" and "de-identification" are used interchangeably. Often, within the literature, the term "de-identification" is used as a "general term for any process of removing the association between a set of identifying data and the data subject" ISO-25237 (2008). In this work, I make use of this definition, until Chapter 6, where I argue for the expansion of de-identification to encompass the dis-association of identifying data from any individual associated with the data, thus including both subject and author.

The term "anonymization" is *supposed* to refer to the specific subset of de-identification processes that remove the association between a set of identifying data and the data subjects in such a way that it is impossible to re-associate the original data-subject with their previously associated data. This is in contrast to "pseudonymization" which refers to processes where identifiers (e.g., ID number) are replaced with pseudo-identifiers (e.g., fake ID number), but the original data holder keeps a table of the identifier and pseudo-identifier relationships. In these instances, the original data holder is able to easily re-identify the data.

In theory, it should be impossible to re-identify data that has been anonymized. However, there are many datasets that claim to have undergone anonymization yet have had some re-identification (Scaiano et al., 2016). The question of whether the term "anonymization" should be applied to processes based on the intent of their developers or only upon (mathematical) proof of anonymity as discussed by Garfinkel et al. (2015). In this work, I follow the approach of Garfinkel et al. (2015) and use the broader term "de-identification" for all processes that try to remove or obfuscate PII regardless of whether they can be re-identified or not. I also follow the approach of Garfinkel et al. (2015) for all other matters of terminology (e.g., using PII rather than PHI throughout the thesis).

1.2.2 Terminology: Clinical notes

Clinical data can be stored either as structured data or unstructured data. Structured data will require different de-identification techniques from unstructured data. In this work, I focus solely on the de-identification of unstructured free-text clinical notes (henceforth referred to both as clinical notes and as clinical texts).

Clinical notes are written by authors (i.e., healthcare providers) when subjects (i.e., patients) interact with the healthcare system either immediately during or some time after the interaction. There are many different types of clinical notes (e.g., progress notes which are taken during routine clinical encounters, discharge notes which summarize the course of a

patient following a health system interaction). Clinical notes, depending on their type, are produced by many different healthcare providers (e.g., nurses, physicians, and specialists). There are multiple ways of creating clinical notes (e.g., handwriting, typing, transcription from voice). There are no widely accepted standardized clinical note formats (in structure or writing style); this varies both inter- and intra-institution. Despite the wide variety, most clinical notes also have many commonalities. For example, due to time pressures on healthcare providers, clinical notes tend to be short, succinct, and filled with many abbreviations. Handwritten and typed notes also have many misspellings, which complicates automated approaches to their analysis and de-identification.

Clinical notes have many uses. For healthcare, clinical notes are used to keep track of patient progression to inform care. Researchers have demonstrated that clinical notes can be used to perform a wide array of tasks such as de-identification (Abdalla et al., 2020b), and clinical predictions (Kalyan and Sangeetha, 2020) among others.

1.2.3 Terminology: Identifiers

In the clinical de-identification literature, the terms "personal health information" and "protected health information" (PHI) are used to refer to health information which is created, stored, or transmitted by healthcare providers (e.g., name, medical history, contact information). "Personally identifiable information" (PII) is used to refer to "identifiers specific to individuals" (Garfinkel et al., 2015). The overlap between PII and PHI is not clear-cut and varies by the jurisdiction of the data. For example, in Canada and the US, names are considered both PHI and PII. At the same time, diseases or care received (e.g., a record of arriving at the emergency department with a sprained ankle and receiving an X-ray) are considered PHI but *generally* not considered as PII in Canada and the US. However, disease and care received can, in the case of rare diseases, also be considered PII.

Another common way to discuss information which may be used to re-identify patients (herein referred to as 'sensitive information') is to use the terms "direct" and "indirect" identifiers (Scaiano et al., 2016; Garfinkel et al., 2015). "Direct identifiers" are pieces of information which alone can be used to confidently re-identify an individual patient (e.g., full name). On the other hand, "indirect identifiers" (also referred to as "quasi-identifiers") are pieces of information that when considered alone cannot directly re-identify any patient, but where a combination of multiple indirect identifiers can pose a risk to re-identification (e.g., age, address, occupation).

There is often legislation stating that data must be de-identified when shared with researchers outside the those providing direct healthcare to a patient. Under United States

law (Section 164.514(a) of the HIPAA Privacy Rule), there are two methods of determining whether a clinical record has been satisfactorily de-identified (Health Insurance Portability and Accountability Act, 2012b):

- **Expert Determination:** an expert assesses the risk that the anticipated recipient of the data cannot meaningfully identify any individual in the data, and
- **Safe Harbor:** the removal of 18 types of identifiers, Table 1.1.

Personally Identifying Information	
Names	All geographic subdivisions smaller than a state
All elements of dates (except year)	Telephone numbers
Vehicle identifiers and serial numbers	Fax numbers
Device identifiers and serial numbers	Email addresses
Universal Resource Locators (URLs)	Social security numbers
Internet Protocol (IP) addresses	Medical record numbers
Biometric identifiers	Health plan beneficiary numbers
Full-face photographs & comparable images	Account numbers
Any other unique identifying numbers	Certificate/license numbers

Table 1.1: The 18 fields defined as personally identifying information by HIPAA which must be legally protected.

In Ontario, the Personal Health Information Protection Act (PHIPA), 2004, S.O.@ 2004, c@ 3, Sched.@A (Part 1, Section 2)[2] does not specifically define how a data holder should make the determination that their data has been successfully de-identified. In practice, this is equivalent to having only the 'Expert Determination' provision of HIPAA. Furthermore, according to PHIPA, PHI which has undergone de-identification is no longer considered PHI (Information & Privacy Commissioner of Ontario et al., 2011).

This variability in what is legally considered PII by jurisdiction, or how to determine whether a dataset has successfully been de-identified, is a tall hurdle for researchers attempting to develop automated approaches for use by data holders in multiple jurisdictions.

1.2.4 Methodology: De-identification of free-text clinical notes

To demonstrate the different approaches to de-identification, below I present the same following hypothetical sentence fragment de-identified in different ways:

[...] Principal Bob diagnosed with high fever; Alice (wife) in to visit. [...]

[2] https://www.ontario.ca/laws/statute/04p03#BK3

The first approach to de-identification is to delete all tokens except what is expected to be relevant information. This is highly dependent on the task at hand. In the example below, I highlight in gray tokens that would be removed under this de-identification approach.

> *[...] Principal Bob diagnosed with high fever; Alice (wife) in to visit. [...]*

The second approach to de-identification is to *secure* all tokens that belong to a pre-defined set of PHI (or PII) and keep all other tokens as they are. There are two ways of securing a token which has been deemed sensitive: 1) PII deletion, and 2) PII replacement. Deleting tokens involves either simple deletion (i.e., replacing the token with a white space), or with a placeholder token indicating removal. Replacing tokens involves replacing the sensitive token with another token of the same type. As stated in the previous subsection, the definition of which tokens are sensitive (i.e., are in the defined subset of PHI or PII that must be dealt with or removed) varies by jurisdiction. Below, I present the same sentence fragment secured using PII deletion and PII replacement. For clarity, I highlight in pink tokens that have been changed.

PII deletion (used by MIMIC-III (Johnson et al., 2016)):

> *[...] Principal **Name** diagnosed with high fever; **Name** (wife) in to visit. [...]*

PII replacement (used by ICES):

> *[...] Principal John diagnosed with high fever; Betty (wife) in to visit. [...]*

Search-based De-identification

All of the approaches to de-identification discussed above follow a search-based approach. That is, tokens belonging to a specifically defined group (clinically relevant words in approach one, or chosen PII in approach two), are sought.

Human-based Search: Humans, when parsing a sentence, attempt to determine whether each encountered word fits the desired category. While simple to describe, and intuitive to execute, the task of de-identification is non-trivial even for humans. Individually, trained experts (in this case: medical house officers) had a large variance in their performance at the task of PII labelling (as defined by HIPAA) with sensitivity (i.e., recall) ranging from 0.63 to 0.94 (Douglass et al., 2004). I re-create the Table 1.2 from Douglass et al. (2004) to demonstrate the full range of human performance. While using multiple annotators results

in great improvements, it is still not enough to guarantee the security of patients (missing 2% of PII on 400,000 notes means missing PII in 8000 notes).

# Human De-identifiers		Min	Max	Mean
1 Person	**Recall**	0.63	0.94	0.81
	Precision	0.95	1.00	0.98
2 People	**Recall**	0.89	0.98	0.94
	Precision	0.95	0.99	0.97
3 People	**Recall**	0.98	0.99	0.98
	Precision	0.95	0.99	0.97

Table 1.2: Table re-created from Douglass et al. (2004)'s study demonstrating human performance at detecting all PII.

Automated Approaches to Search: Automated approaches to perform de-identification can be grouped into three large categories: i) Dictionary-based Approaches, ii) Model-based Approaches, and iii) Hybrid Approaches.

Machine-based Search: Dictionary-based Approaches Dictionary-based approaches are, as the name implies, approaches to capturing PII that use complied dictionaries of PII or rules to capture PII.

With the goal of "produc[ing] an open source, HIPAA compliant, de-identification tool", Beckwith et al. (2006) developed a dictionary-based approach which employs three different passes to capture all PII. The first pass searches for known PII such as patient names, medical numbers, etc. The second pass looked for predictable patterns using regular expressions to capture other PII. The last pass removed all locations which existed in a pre-compiled dictionary. This approach removes 98.3%[3] of all PII according to the tests performed by the authors. The specific instances where the algorithm failed was determined to be misspellings of items that were in the dictionaries. Other such approaches, like that of Miller et al. (2001), purport to have comparable performances using the same approach of matching PII to a dictionary and known patterns.

Despite these strong performances, it is clear that a dictionary-based approach is quite limited because it is not robust to misspellings, new locations, or different types of notes (e.g., family physician progress note vs cardiology consultation note). This lack of robustness means these approaches cannot be confidently applied to novel datasets, and are not

[3]Personal conversations with the Data Management team at ICES (formerly known as the Institute for Clinical Evaluative Sciences) discovered that they were not able to reproduce the strong results of such an approach.

well suited to the noisy data that arises from the clinical setting which contains many errors, and abbreviations.

Machine-based Search: Model-based Approaches Model-based approaches make use of supervised algorithms to automatically capture PII. These tools can use either the text directly or extracted information (e.g., part-of-speech tags) to aid in the search process. Here, I cover two models, one which uses a support vector machine (SVM) classifier on text in conjunction with extracted information to capture PII (Sibanda, 2006), and a more recent approach that makes use of recurrent neural networks to capture PII (Dernoncourt et al., 2017).

Sibanda's statistical approach uses an SVM with a linear kernel at the word-level to classify whether a given word is PII or non-PII. The SVM was trained using human labelled data, and was represented using a variety of features including, but not limited to: the target word, a context window of ±2 nearby words, part-of-speech tags, capitalization of the target among other selected features. This approach claims to out-perform dictionary-based methods, accurately recognizing 94.27%[4] of PII presented.

Dernoncourt et al.'s statistical approach made use of neural network-based approach to de-identifying clinical notes. Inspired by previous approaches, like that of Aberdeen et al. (2010), Dernoncourt et al. pass in notes tokenized using Stanford's CoreNLP tokenizer to a character-level embedding layer, which is then passed through a label prediction layer, to a sequence optimization layer. With a combination of trained models, they were able to achieve a recall of 97.8% on 2014 i2b2 dataset and a recall of 99.4% on MIMIC.

Statistical approaches are more robust than dictionary-based approaches to novel examples and misspellings, although they are not perfect.

Machine-based Search: Hybrid Approaches Hybrid approaches which make use of both dictionaries and statistical techniques are also quite common in capturing PII. Neamatullah et al. (2008)'s approach makes use of a combination of lexical lookup tables, regular expressions, and simple heuristics to locate PII. This is considered a hybrid approach as the lookup tables and regular expressions are pre-defined, whereas the heuristics are also learned from the training corpus. On a test corpus, they achieved a recall of 94%.

Sweeney (1996) developed a system which made use of multiple approaches in tandem to capture PII. More specifically, she attempted to match words to known PII templates using both known dictionaries as well as heuristic probability tables which are adjusted

[4]Clearly this is lower than the 98% touted by dictionary-based approaches and is the topic of further discussion in Section 2.

given a training set. This approach achieves a recall of 99–100%.

Dehghan et al. (2015) made use of dictionary and regex taggers, as well as a conditional random field (CRF) classifier, to tag PII. As inputs to the CRF they took lexical features as well as other features such as capitalization, presence of numeric characters, positional features, and semantic features to aid in classification. Using multiple passes through the data to classify a single note, they were able to achieve a recall of 92%.

Combining dictionaries with statistical techniques to form hybrid approaches is the most robust way to ensure that PII will be captured. This is because we can use the different classes to compensate for weaknesses of the other. However, we have seen that there is a large variance in the results reported, often with contradictory performance statistics. This is something I expand on in Chapter 2.

1.2.5 Methodology: Learning Methodologies

In this section, I will briefly define a few of the many possible learning methodologies used in machine learning: 1) supervised learning, 2) unsupervised learning, and 3) semi-supervised learning. This is not a comprehensive list of all learning strategies (e.g., excluding reinforcement learning), but covers what is relevant to this thesis.

Supervised Learning is a machine learning paradigm where the goal is to learn to predict an output from each input. To enable this prediction, during training, the model is provided an input for each output. Generally, outputs are regarded as the label of the input data. Unsupervised learning is generally viewed as the opposite of supervised learning. In unsupervised learning there are no output labels associated with the inputs. Rather, the algorithm is meant to learn patterns from unlabelled data (e.g., perform clustering). Semi-supervised learning lies between these two learning methodologies. It is an approach that combines a small amount of labelled data with a large set of unlabelled data.

Part I

Clinical De-identification

Chapter 2

Limitations of Supervised De-identification

2.1 Introduction

In the previous section, I motivated the need for the de-identification of clinical notes and explored the multiple proposed approaches to fulfill this need manually and in an automated manner. In this section, I critically examine existing approaches to clinical de-identification and highlight their assumptions and limitations.

All automated approaches to de-identification presented in Section 1.2.4 can be classified as supervised approaches. This means that these approaches are trained using a training set and evaluated using a test set. The algorithms are trained to classify all the tokens in a text either as sensitive (i.e., posing a risk of re-identification to the patient and needing to be dealt with) or as not. This per-token classification can be thought of as functionally being a search algorithm whereby the de-identification algorithm is searching for sensitive tokens. As such, it would be fair to classify all existing approaches, whether automated or not, as being search-based approaches. Framing these approaches as searching for sensitive tokens can help us uncover some underlying assumptions. More specifically, using search asks us to accept the assumption that is:

- **Assumption 1:** possible to define a comprehensive list of PII.
- **Assumption 2:** possible (or feasible) to design and train a perfect search algorithm to detect all PII.

In the sections below, I explore the limitations that stem from accepting each assumption.

2.2 Limitations due to Assumption 1: Defining PHI

The first limitation of supervised approaches is that the development of models requires agreement on what should be classified as sensitive. In the literature, there is active debate on what should be considered PII with multiple competing proposals. On one side of this debate, there are many developed approaches that stick to HIPAA's 18 chosen identifiers, Table 1.1. On the other hand, there are researchers who argue for the inclusion of more identifiers to be secured during de-identification.

For example, Dernoncourt et al. (2017) define PHI using HIPAA's definition. Other works have decided to narrow down what is considered PII from HIPAA's 18 using various subsets. Taira et al. (2002) focuses only on classifying names, Zuccon et al. (2014) considers a subset of 7 of the 18 and Aberdeen et al. (2010) classifies 15 of the 18 classes. There have also been authors advocating for capture more than simply what is defined in HIPAA. Ferrández et al. (2013) included larger geographic locations (e.g., states as PII), and Sweeney (1996) explicitly captured nicknames. Both Feder et al. (2020) and Abdalla et al. (2020b) raise concerns about occupation not being included in HIPAA and attempt to secure them as well. Scaiano et al. (2016) provides a literature review of the number of PII considered by de-identification algorithms and we can see that the majority of works do not use a consistent subset.

2.3 Limitations due to Assumption 2: Token Classification

Supervised approaches work under the assumption that the training data is correctly labelled. This means that we are confident that for each clinical note in the training and test set, we have successfully found and labelled all tokens belonging the selected PII groups. However, we know that this is not truly the case. In Section 1.2.4, we discussed the recall of multiple humans at detecting PII in clinical notes where we observed that having 3 human annotators try to detect PII would result in an average PII of 98%. Because of this, we cannot be fully confident in the performance evaluation of our algorithms (*if the goal is perfect de-identification*).

However, even if our training and test sets were perfectly labelled, it would not be possible to prove that our trained algorithms or classifiers would perform perfectly on unseen data: there is a large body of work demonstrating that state-of-the-art (SotA) algorithms perform worse on un-seen data (e.g., data from domains not similar to their training data) (Zhou et al., 2021).

Furthermore, drawing from logic, it is not possible to prove a general and negative claim through sampled observations. That is, induction cannot be used to arrive, *with complete certainty*, at the conclusion that our trained algorithm is perfect and will continue to be perfect simply because it has done so once before. As learned from Karl Popper's 'Theory of Falsification' (Popper, 1963), additional observations to confirm a theory is not enough to prove the theory correct if there is always the possibility that a future observation could refute it. Concretely, induction cannot yield certainty. It is for this reason that supervised search-based approaches can never be shown to have achieved perfection.

2.4 Demonstration of risk: evaluating privacy of embeddings

In this subsection, I demonstrate how failing to realize and deal with the limitation stemming from the two assumptions above can have negative consequences for data holders. More specifically, I present a novel methodology developed to calculate the risk or possible leakage from publicly releasing word embeddings that have been trained on clinical notes secured using PII removal. If we assume a supervised approach with good performance (i.e., 99% recall on sensitive information), then a cursory glance could indicate that releasing such embeddings has no associated risk because of the unordered nature of these models; all that is released is a list of tokens, arbitrarily ordered, with dense numeric vectors associated with each token. However, through experiments with three of the most popular traditional embedding techniques, I show that the released embeddings can be leveraged to learn information about the patients in the dataset.

This work (Abdalla et al., 2020a) is the first to develop a methodology to study the privacy implications of releasing word embeddings trained on clinical data. The developed methodology, which demonstrates how anonymizing clinical notes using PII removal can leave sensitive patient information vulnerable, has been adopted by other researchers to assess the risk of their model (Lehman et al., 2021).

Applying this methodology to multiple datasets, I show that, depending on the type of word embedding model used and the hyperparameters selected: (1) it is possible to associate name tokens together to form true name pairs, (2) there is a significant difference between the distances of diagnoses that have been associated with a patient and those of diagnoses not associated, and this is true both at the population level and at the patient level, and (3) it is possible for a malicious actor to determine diagnoses assigned to multiple

patients, using only precomputed embeddings.[1] I argue that, given our results, data holders and providers should explore whether other paradigms, such as PII replacement or RaNNA, are more successful in securing sensitive information when compared with PII removal.

2.4.1 Background and Motivation

Word embeddings are often used as the first step of representing text for neural approaches to various tasks. Word embeddings trained on health care data are strongly correlated with human-annotated word relatedness metrics for medical terms (Wang et al., 2018), although their performance on clinical classification tasks is strongly dependent on their size and type of data from which they are created (Lai et al., 2016). Previous studies have shown that, for a variety of tasks, embeddings created from clinically related data (e.g., clinical notes and biomedical texts, such as a collection of all PubMed Central articles and PubMed abstracts), often performed better than, and never performed worse than, unspecialized corpora (Wang et al., 2018). To enable more NLP research for the clinical setting, there has been a concerted effort to make clinical word embeddings publicly available, because they are often too expensive for smaller institutions to train due to costs associated with gathering and de-identifying the large amount of data involved in creating good embeddings.

Until recently, there had been no publicly released embeddings trained on clinical data (Alsentzer et al., 2019; Huang et al., 2019; Peng et al., 2019; Si et al., 2019). However, some newly released embeddings (Alsentzer et al., 2019; Huang et al., 2019; Peng et al., 2019; Si et al., 2019) are trained using contextual word embedding models on MIMIC-III (Johnson et al., 2016), which itself uses PII removal to abide by HIPAA regulations. This work demonstrates how, if no additional security measures are taken, then traditional (i.e., non-contextual) word embedding models may be compromised.

2.4.2 Data and Methods

Data

For the following experiments, I used consultation notes from Electronic Medical Records in Primary Care (EMRPC) housed at ICES. In Appendix 2.A, I demonstrate how these findings are reproducible with an experiment performed with a selected subset of Wikipedia pages. The latter is made publicly available alongside the code. For all texts, I removed all

[1] In this section, I refer to diagnostic codes and diagnoses interchangeably, although this is not, of course, a general equivalence. Here, I take the diagnostic code simply as an indication of the condition that the patient is suspected of having, which is sensitive information that must be protected.

punctuation and numeric characters, and lowercased all text but performed no lemmatization, tokenization, or any other preprocessing.

Access to the consultation notes is provided by ICES (formerly known as the Institute for Clinical Evaluative Sciences) under data sharing agreements with physicians for the purposes of evaluation and research. Consultation notes are written by specialist physicians and other health care consultants to a patient's family physician. They describe the tests performed, results observed, and other details that the specialist physician or health care consultant considers relevant. It is important to note that these codes are often not cover all the diagnoses of patients (as only 1 code can be recorded per visit). For this work, I compile patients' consultation notes and all their prescribed diagnostic codes that are indicative of suspected diagnoses and ordered tests, and are therefore sensitive health information that must not be connected to patient identities. The billing codes table includes text fields describing each code in 1 to 3 words (e.g., "colon screening"). These data sets are linked using unique encoded identifiers and analyzed using ICES.

Although this work is conducted at ICES, ICES does not grant its research affiliates access to true patient names but replaces them in the manner described earlier (PII replacement), using a semi-manual, dictionary-based masking process to consistently replace each true name with a randomly chosen fake name. I used heuristics to detect names in the notes. More concretely, I looked for semi-structured notes that have Name: str1, ... , strN (representing a series of alphabetical tokens separated by commas followed by a semicolon) to indicate the presence of a replaced name. The heuristic is not 100% accurate, which is why, in Appendix 2.B, I can provide only an estimate of how many true names exist by manually analyzing a randomly sampled set.

I performed these experiments on clinical consultation notes for which we can locate the associated fake patient name. For these experiments, I treat the fake names as if they were the true names and removed 99% of them, thus emulating current PII removal algorithms (Dernoncourt et al., 2017). This protected data set is then used as the first step of these experiments, as shown in Figure 2.1. Detailed information regarding the data is provided in Appendix 2.B.

Experimental Methods

The intuition behind re-identifying patient information solely from word embeddings stems from the distributional hypothesis (Sahlgren, 2008) — words appearing in similar contexts tend to have similar meanings and therefore have closer vector representations than other words. Knowing this, I hypothesized that there would be differences between both:

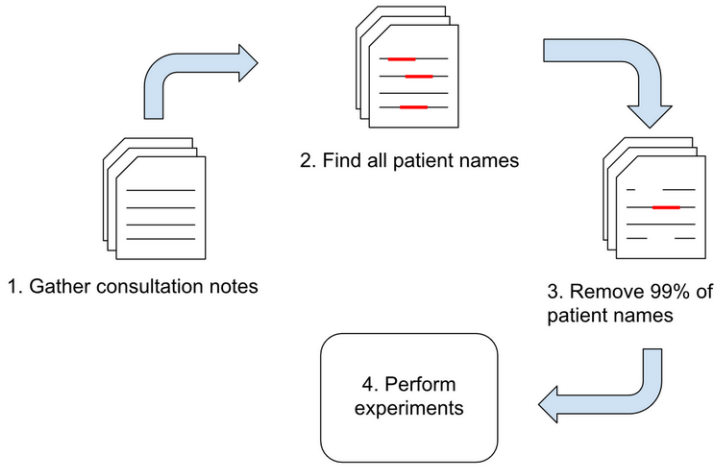

Figure 2.1: Process flow for gathering and preparing the clinical notes for embedding generation and experimentation.

1. The average distance between the tokens that make up a person's name, compared with tokens from different names.

2. The average distance in vector space between a person's name and their diagnoses (referred to as the *in-group*), compared with the average distance between their name and those diagnoses with which they are not associated (referred to as the *out-group*).

If there is a large enough distance between a person's *in-group* and *out-group*, then this distance could be used to extract sensitive information thought to have been hidden by the unordered nature of embeddings. In the following sections, I validate this hypothesis empirically.

Experiment 1: Name Reconstruction Experiment In the first experiment, I test whether it is possible to reconstruct true name pairs simply from a list of individual name tokens. Figure 2.2 presents the steps of this experiment, picking up from the last step of Figure 2.1.

A list of individual name tokens, corresponding to the fifth step in Figure 2.2, is easily generated by manual exploration of the words. However, as I left 1% of the names, to emulate the imperfect de-identification algorithms, I knew all the tokens (i.e., the 1% of name tokens purposefully left in place).

This experiment was performed on the consultation notes data set, where over 99% of names were removed to emulate a PII removal approach and only 1054 unique name tokens (from 650 full names) remained in the text.

I performed the experiment with 3 commonly used traditional word embedding algorithms (CBOW, Skipgram, and GloVe) for clinical prediction and modeling tasks (Khattak

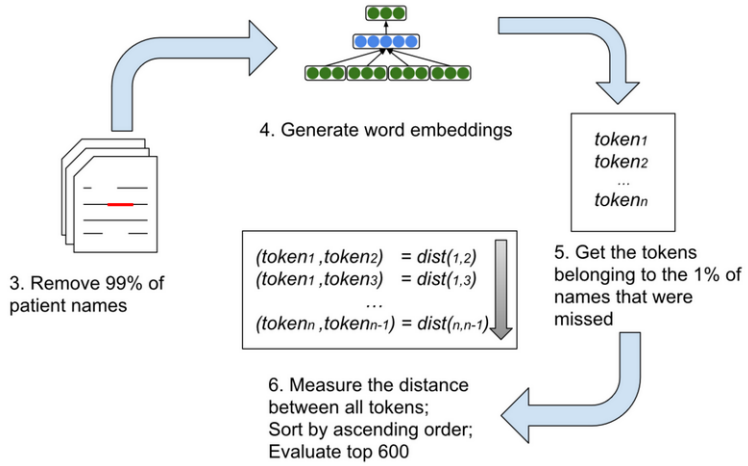

Figure 2.2: Process flow for generating word embeddings and performing the name reconstruction experiment.

et al., 2019). For each, I tested a variety of hyperparameters. Where a specific hyperparameter is not explicitly mentioned, I used the default hyperparameter of the training model as specified by the Python package Gensim[2], which can be found in Appendix 2.B.

However, for the sixth step, an attacker would not know how many full names were in the data set. Assuming that each name is composed of 2 tokens and none of the names share any name tokens; we would expect the number of complete names to be half the number of name tokens (i.e., 1052/2 complete names). Relaxing both assumptions increases the expected names. Given name tokens A and B, we considered a name to exist if either $\langle$A,B$\rangle$ or $\langle$B,A$\rangle$ exist as names (i.e., ignoring ordering). On this data set, I created many word embedding models (n = 88) with a wide set of hyperparameters (i.e., model specifications) that included variations in the distance metric (cosine or cityblock) and context window size.

Experiment 2: Name-Diagnostic Code Association Experiment The second experiment explores the second part of the hypothesis: is there a difference between the average distance in vector space between a person's name and their diagnoses (their in-group) compared with the average distance between their name and those diagnoses with which they are not associated (their out-group)?

For this experiment, I used the same data and tested the properties of the same word embedding algorithms for various hyperparameters, as in the last experiment. I first define a patient's name vector as the average of the vectors of its components (i.e., first, last, and possibly middle names). Here, numtoken is the number of space-separated tokens in a

[2] https://radimrehurek.com/gensim_3.8.3/models/word2vec.html

string and is the vector representation of the i-th token of the name:

$$\text{name\_vector} = \frac{1}{\text{numtoken(name)}} \sum_{i=1}^{\text{numtoken(name)}} n_i \tag{2.1}$$

Second, I defined the in-group d_{in} as the set of diagnoses for name and the out-group d_{out}, as all other diagnoses, with d_i representing any individual diagnosis. The average distance for each of these groups from their respective names are referred to as in_group and out_group, respectively:

$$\text{in\_group} = \frac{1}{|d_{in}|} \sum_{d_i \in d_{in}} abs(cityblock(\text{name\_vector}, d_i)) \tag{2.2}$$

$$\text{out\_group} = \frac{1}{|d_{out}|} \sum_{d_i \in d_{out}} abs(cityblock(\text{name\_vector}, d_i)) \tag{2.3}$$

Below, I present the results using the cityblock distance (i.e., the Manhattan distance) instead of the cosine distance because it performs better at this task (by uncovering more information), and past work has shown that the vector magnitude (i.e., the sum of all dimensions) is affected by the number of times that the word occurs in the corpus (Schakel and Wilson, 2015). However, the experiments were performed using the cosine distance metric as well, and complete results are presented in Appendix 2.B.

Initially, I explored the raw data (i.e., without any de-identification algorithm) by plotting the difference between the in- and out-groups for names that occur below different frequency thresholds. A name is below the threshold if the average counts of its components are below that threshold. For example, if "James" occurs 201 times in the corpus and "Qwerty" appears twice, then "James Qwerty" is below an arbitrary threshold of 200 ($101.5 < 200$).

Figure 2.3 shows that the more frequently a name occurs, the smaller the difference between the in-groups and out-groups. Nonetheless, the difference is still pronounced when all names are considered, with the lowest value being just under 5. Surprisingly, against my intuition, the in-group is larger than the out-group. This result is consistently observed throughout our testing described in the following sections.

Given initial observation that, on raw data, there is a difference between in- and out-groups on the population level on raw data, next I examined if the observed differences are statistically significant at both the population and patient levels for various embedding algorithms and hyperparameters on the de-identified data set (i.e., 99% of names have been removed to emulate an optimum real-life data sharing scenario). A diagram of the experimental process is shown in Figure 2.4.

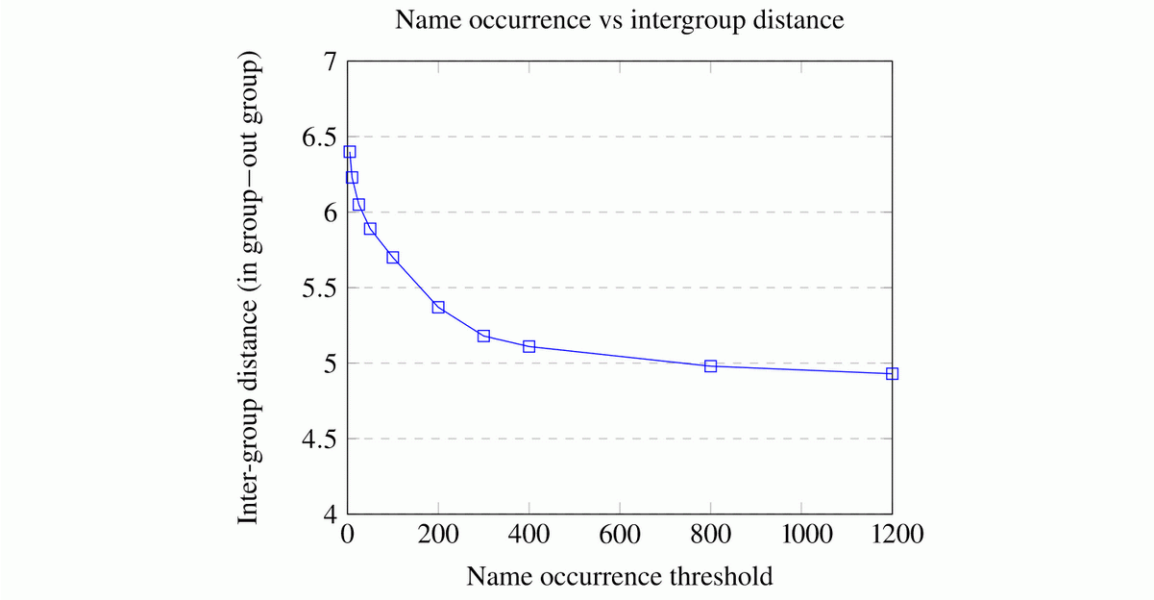

Figure 2.3: Relationship between frequency of name occurrence and the average difference between the in-group and out-group for patients. This graph is generated from an experiment run on a GloVe model with a dimension of 100, window of 10, learning rate of 0.05, minimum occurrence of 1, and alpha of .75.

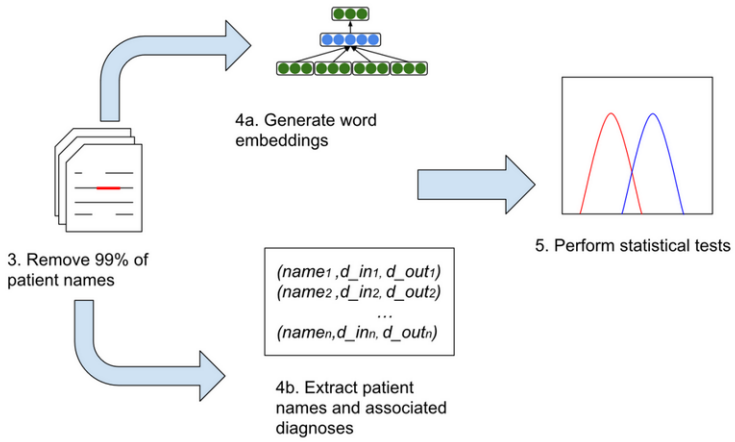

Figure 2.4: Process flow for generating word embeddings and performing statistical testing. For population-level statistical testing, we performed a Wilcoxon signed-rank test, and for patient-level statistical testing, we calculated empirical P values using 1000 randomly generated permutations.

Experiment 2a: Population-Level Statistical Testing The aim of Experiment 2a is to determine whether the difference between the in- and out-groups on the population level is statistically significant.

Here, as with all the clinical text experiments, the embedding model is trained using all consultation notes after 99% of the names have been removed. Using the same setup as in the previous section to obtain distances between in- and out-groups, I used the Wilcoxon signed-rank test to compare the pairings of in- and out-groups for each name on the population level. The Wilcoxon signed-rank test is non-parametric and, unlike the paired Student two-tailed t test, makes no assumptions regarding normality.

This experiment was performed for various embedding algorithms, distance metrics, and hyperparameter ranges.

Experiment 2b: Patient-Level Statistical Testing The next experiment explored whether there is a statistically significant difference between the in- and out-groups for each patient, which would indicate that an individual patient is at risk of having their diagnostic code uncovered.

In this experiment, I compared the average difference between a patient's in-group and the out-group. Although each comparison results in a P value for each patient, for brevity and privacy, I do not report the per patient analysis of the ICES data, but instead report the number of patients for which the difference is significant after correcting for multiple comparisons. To determine statistical significance at the patient level, we calculated empirical P values by randomly sampling in- and out-groups generated using 1000 permutations of the same size from the same data set.

I experimented with various embedding algorithms, distance metrics, and hyperparameter ranges.

Experiment 3: Scenario Simulation

In this experiment, I performed a hypothetical attack to examine whether the results of the previous 2 experiments demonstrate an actionable level of risk. Assuming the role of an attacker who has access only to released embeddings built from doctor-patient consultation notes that have been secured by using PII removal, I test whether the attacker would be able to associate name tokens that were missed by PII removal to arrive at a list of complete patient names and whether they are able to associate these names with some target diagnoses.

For this hypothetical scenario, I used the same data and tested the properties of the same word embedding algorithms for various hyperparameters as in the last experiment.

The hypothetical attack scenario is designed as follows:

- Identify a list of target diagnoses that the attacker wishes to attribute to patients. As an example, I considered the following set of diagnoses: constipation, diarrhea, vaginitis, sexual dysfunction, urinary infection, herpes genitalis, dementia, anorexia, alcoholism, threatened abortion, and AIDS.
- For each name, the attacker will calculate the 5 diagnoses that are farthest from the name.
- Using these 5 diagnoses as the basis for prediction, we calculated Top-1 (A@1) and Top-5 (A@5) accuracy.

To ensure that our results are not an artifact of the selected diagnoses, we repeated the above experiment 1000 times for each tested hyperparameter, randomly selecting 30 target diagnoses. To be as stringent as possible, we chose from diagnoses that appeared at least 10 times in the data (which likely results in a pessimistic bias, as demonstrated in Appendix 2.B.

2.4.3 Results

Experiment 1: Name Reconstruction Experiment The results of this experiment demonstrate that it is possible to reconstruct true name pairs simply from a list of individual name tokens and their respective embeddings.

In this section, I present the results for various context window sizes, an expected name list of size 600, and a cosine distance metric. We can see that up to 68.5% (411/600) of the paired tokens come from true names, as shown in Table 2.1 and Figure 2.5. As there are over 170,000 name-pair combinations, these embeddings clearly carry patient information that can be identified, thus affirming the initial hypothesis. The complete results for other hyperparameters, the number of names expected, and the cityblock distance metric are presented in Appendix 2.B.

Experiment 2: Name-Diagnostic Code Association Experiment

Experiment 2a: Population-Level Statistical Testing The results of this experiment indicate that, at the population level, the average difference between the in- and out-groups per patient is statistically significant. Table 2.2 and Figure 2.6 present the results for various embedding algorithms, varying context window sizes, and a cityblock distance metric. The

Context window size	Skipgram names, n (%)	CBOW[a] names, n (%)	GLoVe[b] names, n (%)
1	51 (8.5)	17 (2.8)	8 (1.3)[c]
3	369 (61.5)	265 (44.2)	158 (26.3)
5	393 (65.6)	323 (53.8)	278 (46.3)
7	410 (68.3)	331 (55.2)	317 (52.8)
9	411 (68.5)	340 (56.7)	323 (53.8)

Table 2.1: The number and percentage of paired tokens that are part of true names as a function of context window size, using the cosine distance metric of the first 600 paired tokens sorted in ascending order. [a]CBOW: Continuous Bag of Words. [b]GLoVe: Global Vectors. [c]Result not significant after correcting for multiple comparisons using the Holm-Bonferroni correction.

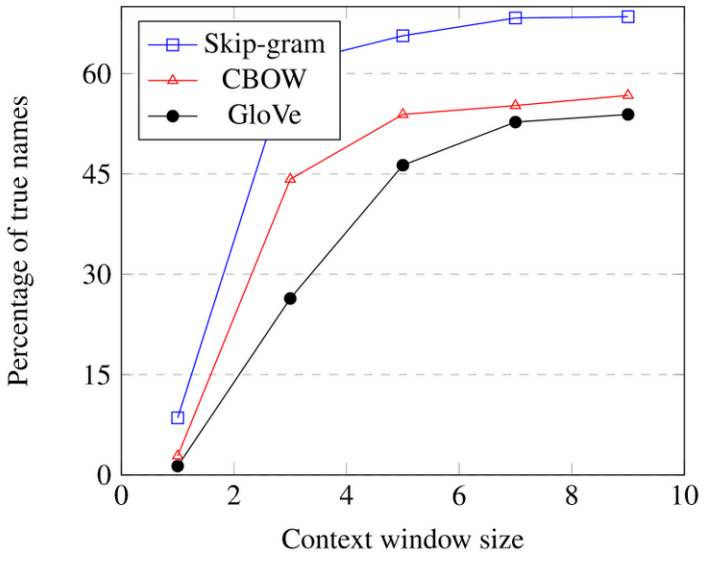

Figure 2.5: Visual representation of the percentage of paired names belonging to true names from the first 600 paired tokens when sorted in ascending order.

complete results for other hyperparameters, other distance measures, and absolute distances are shown in Appendix 2.B.

Context window size[a]	Skipgram difference	CBOW[b] difference	GLoVe[c] difference
1	3.91	7.59	4.85
3	2.88	28.53	5.69
5	2.33	39.55	5.45
7	1.84	47.1	5.12
9	1.51	51.61	5.54

Table 2.2: Difference between the in-group and out-group as a function of context window size for various word embedding algorithms using the cityblock distance metric. The differences are relative distances between word embedding vectors in an n-dimensional space. [a]All differences were statistically significant after correcting for multiple comparisons. [b]CBOW: Continuous Bag of Words. [c]GLoVe: Global Vectors.

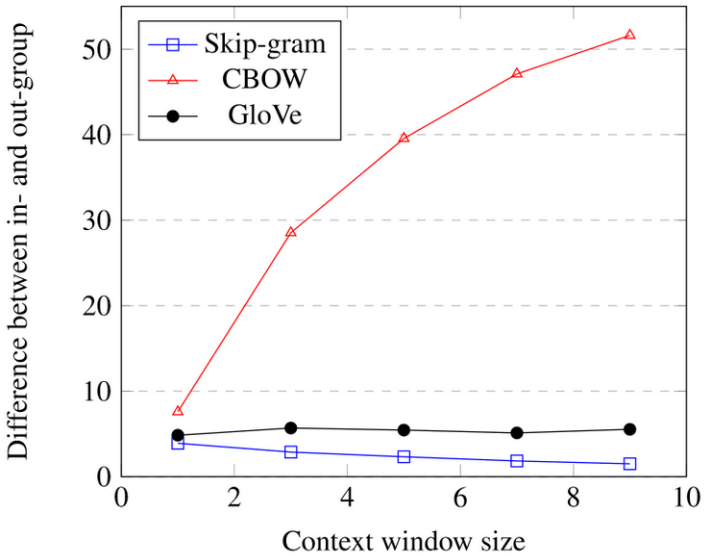

Figure 2.6: Visualization of the difference between the in-group and the out-group as a function of context window size for various word embedding algorithms using the cityblock distance metric.

Given our selected hyperparameters, we can see that for all sizes tested and for all embedding techniques, the difference between the in- and out-groups on the population level was statistically significant with $p < 0.001$ calculated using the Wilcoxon test, after correcting for multiple comparisons using the Holm-Bonferroni correction (Holm, 1979). The Holm-Bonferroni correction is a sequentially rejective procedure for correcting multiple comparisons that keeps the family-wise type I error bounded. Figure 2.6 shows that the

difference between the in-group and out-group decreases for embeddings created with the Skipgram algorithm as the context window increases. Conversely, the difference grows for CBOW, while it remains relatively stable for all GloVe models.

Experiment 2b: Patient-Level Statistical Testing Building on our previous observations, the results of this experiment indicate that, at the patient level, for a percentage of examined patients (up to 449/638, 70.4%), the average difference between in- and out-groups per patient is statistically significant.

Table 2.3 and Figure 2.7 show the results for various embedding algorithms, varying context window sizes, and a cityblock distance metric. The complete results for other hyperparameters, other distance measures, and absolute distances are available in Appendix 2.B.

Size	Skipgram patients, (%)	CBOW[a] patients, (%)	GLoVe[b] patients, (%)
1	49 (7.7)	77 (12.1)	400 (62.7)
3	41 (6.4)	149 (23.4)	401 (62.8)
5	33 (5.2)	152 (23.8)	403 (63.2)
7	16 (2.5)	153 (24.0)	380 (59.6)
9	12 (1.9)	153 (24.0)	449 (70.4)

Table 2.3: The percentage of patients whose diagnoses are identifiable due to a statistically significant difference between the in-group and out-group as a function of context window size for various word embedding algorithms using the cityblock distance metric. [a]CBOW: Continuous Bag of Words. [b]GLoVe: Global Vectors.

Table 2.3 presents the patient-level analysis for different context window sizes. As shown in Figure 2.7, using the CBOW algorithm, an increasing window size initially correlates positively with the number of vulnerable patients, defined as having a significant difference between the in-group and out-group. The opposite trend can be observed for the Skipgram model. Context window size does not appear to have an effect on word embeddings created using GloVe, as the number of patients remains relatively stable.

Experiment 3: Scenario Simulation

Having demonstrated that the difference between in- and out-groups is statistically significant, in this section, we can see that our hypothetical attack results in a low, yet possibly actionable level of risk. That is, an attacker who has access only to released embeddings built from doctor-patient consultation notes that have been secured by using PII removal may be able to arrive at a list of complete patient names, and associate these names with target diagnoses (depending on the model type and hyper-parameters of the trained model).

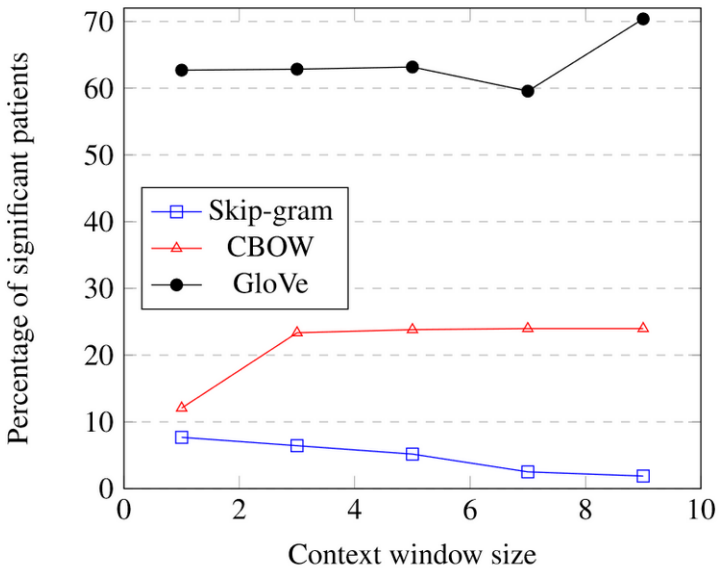

Figure 2.7: Visualization of the percentage of patients who have a significant difference between their in- and out-groups as a function of context window size for multiple word embedding algorithms using the cityblock distance metric.

We observe that for our chosen target diagnoses (i.e., constipation, diarrhea, vaginitis, sexual dysfunction, urinary infection, herpes genitalis, dementia, anorexia, alcoholism, threatened abortion, and AIDS) our approach out performs the majority baseline for both top-1 (A@1) and top-5 (A@5) accuracy of 0.00 and 0.70, respectively, (top-n rate is the fraction of examples for which the correct label is among the n labels considered most probable by the model). The complete results for all hyperparameters as well as both distance metrics are present in Appendix 2.B.

We observe similar results when the above experiment was repeated 1000 times for each tested hyperparameter, randomly selecting 30 target diagnoses. Table 2.4 shows how often our attacker's approach surpasses the baseline of choosing the majority diagnoses for both top-1 and top-5 accuracies. The results show that we can consistently beat the proposed baselines, although the highest top-1 and top-5 accuracies are modest at 0.08 and 0.15, respectively. The complete results for all hyperparameters as well as both distance metrics are present in Appendix 2.B.

2.4.4 Summary and Conclusion

In this section, I have shown the following:

- There is a statistically significant difference between the distance of patients' in- and out-groups at the population level.

Context window size[a]	Skipgram A@1, A@5	CBOW[b] A@1, A@5	GLoVe[c] A@1, A@5
1	55.8, 56.7	61.8, 61.8	55.4, 56.9
3	55.6, 53.1	51.2, 52.6	60.5, 59.5
5	57.4, 55.6	53.6, 54.5	59.4, 57.2
7	57.4, 53.5	54.6, 53.9	55.9, 54.0
9	57.2, 53.2	53.7, 51.2	60.6, 56.7

Table 2.4: The percentage of times using a word embedding–based attack beats the majority baseline for A@1 and A@5 for various context window sizes over 1000 random diagnosis selections. [a]We observed that the majority baseline is surpassed consistently and up to 60% of the time. [b]CBOW: Continuous Bag of Words. [c]GLoVe: Global Vectors.

- For many patients, the difference between their personal in-group and out-group is also statistically significant.
- A malicious actor working only with word embeddings may identify full names occurring in the training corpus of the embeddings as well as sensitive attributes associated with these names.

This exploration of the induced privacy (or lack of privacy) of embeddings created from medical notes was done to empirically highlighted the security risks of sharing embeddings trained on clinical data. Although their nature does serve to obfuscate information, the experiments above show it is still possible to connect PII to names from word embeddings secured using PII removal. There is much variation in the risks observed in this work, which are dependent on imperfect de-identification algorithms and very skilled attackers. Therefore, the actual risk to patient information, while nonzero, remains small and dependent on many variables such as the attack strategy, de-identification method, and embedding algorithm. It is also unclear whether or not the risks translate directly to contextual word embeddings; Lehman et al. (2021) were not able to replicate the results of this work using contextual word embeddings. However, this replication was not fully true to our methodology as they: 1) did not test many hyperparameters, 2) utilized cosine distance as their main distance metric (instead of cityblock), and 3) made no changes to account for the contextual nature of the embeddings (e.g., introducing novel evaluation methodology).

2.5 Discussion

In this chapter, I have highlighted the limitations inherent to supervised search-based approaches (regardless of whether they use deletion or replacement). These limitations can

affect the risk to patient confidentiality when sharing data. Section 2.4 demonstrated the risk associated with publicly sharing word embeddings that have been de-identified using a search-and-delete approach because of limitations inherent to said approach. While the risk resulting from using search-and-delete approaches to create embeddings can be alleviated by using search-and-replace approaches, the risks associated with sharing full clinical notes, stemming from the limitations highlighted in Sections 2.2 and 2.3, cannot be addressed by supervised approaches convincingly.

As such, in the next chapter, I present a novel class of unsupervised de-identification approaches that: 1) do not have the same limitations inherent to supervised approaches, and 2) reduce the cost associated with de-identification of free-text clinical notes to enable data sharing for smaller institutions.

Appendix

2.A Wikipedia

2.A.1 Scenario Simulation on Wikipedia Data

While data-sharing agreements prohibit ICES from making our clinical dataset publicly available, access may be granted to those who meet pre-specified criteria for confidential access, available at www.ices.on.ca/DAS.

In this section, however, we aim to replicate the results of the ICES experiments with a novel dataset that can be publicly released, and to answer questions that were not possible with the ICES dataset. To model patient data in a dataset that can be publicly released without privacy concerns, we built a synthetic dataset composed of approximately 20,000 biographies of politicians by scraping the "Politician" category in Wikipedia. The characteristics of the table are described in Table 2.A.1. Each biography can be regarded as a model "clinical note" about its subject, upon which we can experiment. We "de-identified" our dataset of 20,000 politicians' biographies by removing 99% of the names, leaving behind only 200 names as if they had been missed by a state-of-the-art PHI removal algorithm. We considered only space-separated tokens in the title as "names", choosing to ignore other possible names. The embedding model used for all the following experiments is a CBOW word embedding model with a context window of 5 trained on this "secured" version of the 20,000 biographies.

Number of pages	19,686
Mean words per page	251

Table 2.A.1: Characteristics of Wikipedia dataset.

First, we show that it is possible to reconstruct entire name pairs simply from a list of individual name tokens. To do this, we measure the cosine distance between each pairing of individual name tokens and sort the resulting list in ascending order. We observe that the first 10 paired tokens are correctly paired and that, of the top 50 paired tokens, 36 (72%)

are correctly paired. This result indicates that it is possible to reconstruct full names (or parts of full names) simply from the embedding model itself. This lowers the cost of an attack by making it easier to identify people in a released model.

Second, we attempt to determine the nationality of each politician using only the word embedding model. Here, nationality is analogous to a diagnostic code in the previous experiment. As observed before, the in-group has a higher distance than the out-group with average distances of 64.4 and 51.0, respectively. Performing the same hypothetical attack as Experiment 3 results in top-1 and top-5 accuracies of 9.8% and 21.3% respectively, with a large increase in top-1 accuracy and a maintained performance in the top-5 accuracy.

Finally, we manually explore the nearest neighbors of name vectors as well as words most likely to be predicted as the focal word given the name as context. This analysis is done qualitatively for the first 5 named pairs that were correctly paired. The full list of predicted words for each name is presented in uploaded as part of the supplemental materials (as a Jupyter notebook). We observe that such information, although quite noisy, contains terms relevant for some names (including the birthplace, and "*guilty, tribunal, murder*"). This information could be used by a malicious actor to gain more information about a patient.

2.B ICES

2.B.1 Word Embedding Model Parameters

For CBOW and Skipgram the initial parameters are: Window Size = 5, Learning Rate = 0.025, min_count = 1, epochs = 25.

For GloVe the parameters are: Window Size = 10, Learning Rate = 0.05, Alpha = 0.75, min_count = 1, epochs = 35.

Number of patients	89,990
Notes with a name	402,793
Words per note	Min = 1; avg = 212; max = 10,473
Word count	84,278,374
Unique word count	366,977
Unique code words	9,094
Code words per patient	Min = 1, avg = 21, max = 163

Table 2.B.1: Characteristics of ICES dataset.

Number of patients with Diagnostic Code	Number of Diagnostic Codes
constipation	4146
diarrhea	9064
vaginitis	6770
sexual dysfunction	1043
urinary infection	1174
herpes genitalis	172
dementia	1851
anorexia	466
alcoholism	1581
threatened abortion	546
AIDS	24

Table 2.B.2: Frequency of diagnostic codes used in the hypothetical scenario presented in the paper.

2.B.2 Studying effect of Frequency

In this section, we study the effect of diagnostic code frequency on statistical significance. In particular, we attempt to see whether the statistically significant distances observed in the experiments are due to the long-tail in the dataset (i.e., the singly occurring diagnoses that make up the majority of the dataset). To this end, we recreate the experiment at both the population and the patient level while excluding diagnoses that occur fewer than 5 times and fewer than 10 times (Table 4).

We observe that the more restrictive we are with the diagnostic codes, the less statistical significance we observe at the patient level. The drop is especially apparent for the CBOW and Skipgram algorithms, while the GloVe models still reveal much about patients. However, on the population level, the in-group is still significantly higher than the out-group consistently for all three algorithms, Appendix Table 2.B.4 (b) (d) and (f). To demonstrate the "actionable risk" posed even by the most highly-restricted dataset, we randomly sample from the restricted set of diagnostic codes in the last scenario performed in the work where we play a hypothetical attacker. We also explored whether the mean frequency of diagnostic codes from the in-group when compared to the mean frequency of diagnostic codes from the out-group was correlated to the difference between the in- and out-groups. We found that there is little to no correlation between the frequency of diagnostic codes in the in-group vs out-group and difference between in- and out-groups, with most comparisons lacking statistical significance, Appendix Table 2.B.5.

Number of patients with Diagnostic Code	Number of Diagnostic Codes
1	9707
2	1904
3	703
4	454
5	291
6	239
7	182
8	179
9	129
10–50	1503
50–300	870
300–Max (31472)	598

Table 2.B.3: Number of diagnostic codes that appear for varying number of patients.

2.B.3 Considering Effect-Size

Appendix Table 2.B.4 demonstrated the statistical significance at the patient and population level. In this section we present the effect-size of the difference between the in- and out-groups both at the population and the patient level. The effect size, Appendix Equation 2.4, serves to communicate the magnitude of difference between two groups as opposed to the binary test of significance. Furthermore, unlike the statistical tests performed in the paper, effect size is independent of sample size.

$$\text{Effect Size} = \frac{[\text{mean of group1}] - [\text{mean of group2}]}{\text{Standard Deviation}} \tag{2.4}$$

Appendix Table 2.B.6 presents the effect size when comparing the in- and out-groups for both the population level and the patient level. In this comparison, we only use diagnostic codes that occur over 10 times (as this was the most-restrictive setting observed in Appendix Table 2.B.4). We observe that at the population level we have large effect sizes for CBOW and GloVe, and medium to small effect sizes for Skipgram. At the patient level, we observe a small average effect size.

2.B.4 Name Reconstruction Parameters

For the name reconstruction experiments in the paper and full results in the appendix, we explore what percentage of the first 600 names sorted by ascending order are part of existing patient names. In this section, we explore the effect of choosing different limits. We can see that expanding the list of names which we look at does not greatly change the

Context Window Size – Cityblock Measure – Diagnostic Codes (ALL)

Size	Skipgram %patients	CBOW %patients	GLoVe %patients
1	7.7	12.1	62.7
3	6.4	23.4	62.8
5	5.2	23.8	63.1
7	2.5	24.0	59.6
9	1.9	24.0	70.4

(a)

Size	Skipgram distance	CBOW distance	GLoVe distance
1	3.91	7.59	4.85
3	2.88	28.53	5.69
5	2.33	39.55	5.45
7	1.84	47.10	5.12
9	1.51	51.61	5.54

(b)

Context Window Size – Cityblock Measure – Diagnostic Codes (N ≥ 5)

Size	Skipgram %patients	CBOW %patients	GLoVe %patients
1	0.9	3.8	51.4
3	1.2	3.9	49.5
5	1.1	3.9	48.7
7	0.6	4.2	46.2
9	0.2	4.2	59.2

(c)

Size	Skipgram distance	CBOW distance	GLoVe distance
1	1.86	3.73	3.04
3	1.34	14.58	3.40
5	0.94	20.25	3.29
7	0.71	23.71	2.87
9	0.50	26.18	3.22

(d)

Context Window Size – Cityblock Measure – Diagnostic Codes (N ≥ 10)

Size	Skipgram %patients	CBOW %patients	GLoVe %patients
1	1.6	2.5	45.4
3	1.4	1.6	42.5
5	0.8	2.0	43.4
7	0.5	1.4	39.3
9	0.3	1.6	54.2

(e)

Size	Skipgram distance	CBOW distance	GLoVe distance
1	1.51	3.03	2.47
3	1.15	11.58	2.81
5	0.79	16.10	2.73
7	0.61	18.65	2.31
9	0.43	20.79	2.67

(f)

Table 2.B.4: The percentage of patients whose diagnoses is identifiable due to a statistically significant difference distance between in-group and out-group as a function of various hyperparameter setting, using the cityblock measure. Sub-tables a) and b) consider all diagnostic codes. Sub-tables c) and d) consider diagnostic codes that occur at least 5 times across all patients. Sub-tables e) and f) consider diagnostic codes that occur at least 10 times across all patients. To determine statistical significance at the patient level, we calculated empirical p-values by randomly sampling the in- and out-groups generated using 1000 permutations of the same size from the same dataset. At the population level, we use the Wilcoxon signed-rank test to compare the pairings of in- and out-groups for each name. All presented distances are significant after correcting for multiple comparisons using Holm-Bonferroni correction.

Context Window Size – Cityblock Measure – Diagnostic Codes ($N \geq 5$)

Size	Skipgram correlation	CBOW correlation	GLoVe correlation
1	–0.023 [a]	0.005 [a]	0.140
3	–0.014 [a]	0.074 [a]	0.158
5	–0.008 [a]	0.075 [a]	0.134
7	–0.023 [a]	0.072 [a]	0.125 [a]
9	–0.041 [a]	0.062 [a]	0.158

(a)

Context Window Size – Cityblock Measure – Diagnostic Codes ($N \geq 10$)

Size	Skipgram correlation	CBOW correlation	GLoVe correlation
1	–0.026 [a]	0.001 [a]	0.136
3	–0.016 [a]	0.074 [a]	0.155
5	–0.010 [a]	0.073 [a]	0.131
7	–0.021 [a]	0.072 [a]	0.122 [a]
9	–0.041 [a]	0.060 [a]	0.154

(b)

Table 2.B.5: Spearman's rank correlation between in-group frequency and in- and out-group differences as a function of varying context window sizes for various word embedding algorithms using the cityblock distance for diagnostic codes that appear more at least (a) 5 times, (b) 10 times across all patients. A superscript 'a' denotes lack of significance after correcting for multiple comparisons using the Holm-Bonferroni method. We see that there is little to no correlation between the two variables.

Context Window Size – Cityblock Measure – Population Level

Size	Skipgram Effect Size	CBOW Effect Size	GLoVe Effect Size
1	0.463	0.684	0.852
3	0.330	0.911	0.997
5	0.221	0.911	0.970
7	0.161	0.905	0.836
9	0.108	0.912	0.973

(a)

Context Window Size – Cityblock Measure – Patient Level

Size	Skipgram Effect Size	CBOW Effect Size	GLoVe Effect Size
1	0.237	0.278	0.293
3	0.211	0.352	0.372
5	0.146	0.342	0.365
7	0.117	0.335	0.310
9	0.078	0.333	0.367

(b)

Table 2.B.6: (a) Effect size comparing the in- vs out-group distances as a function of context window size for multiple word embedding algorithms using the cityblock distance measure at the population level. (b) Mean effect size comparing the in- vs out-group distances for each patient as a function of the context window size for multiple word embedding algorithms using the cityblock distance measure.

percentage of names that belong to true name pairs, although as expected it does decrease slightly, Appendix Table 2.B.7.

Context Window Size	First 150 names (%)	First 200 names (%)	First 250 names (%)	First 300 names (%)	First 350 names (%)	First 400 names (%)	First 450 names (%)
1	6.7	6.5	6	8.7	8.3	8.3	9.3
3	85.3	85.0	80.0	75.7	72.0	70.0	66.7
5	94.7	93.0	88.0	81.0	77.7	74.8	70.7
7	96.7	95.0	90.0	86.3	82.0	78.5	74.2
9	95.3	93.5	90.8	86.3	83.1	80.0	75.8
Context Window Size	**First 500 names (%)**	**First 550 names (%)**	**First 600 names (%)**	**First 650 names (%)**	**First 700 names (%)**	**First 750 names (%)**	**First 800 names (%)**
1	8.8	8.5	8.5	8.5	8.5	8.5	8.5
3	63.8	61.5	61.5	61.5	61.5	61.5	61.5
5	68.2	65.7	65.7	65.7	65.7	65.7	65.7
7	71.4	68.3	68.3	68.3	68.3	68.3	68.3
9	71.4	68.5	68.5	68.5	68.5	68.5	68.5

Table 2.B.7: Comparing the effect of choosing a different number of tokens to look at paired tokens, sorted by ascending order, the percentage that are part of existing patient names as a function of context window size, using the cosine distance metric. To determine statistical significance at the patient level, we calculated empirical p-values by randomly shuffling all (n¦2) (n choose 2) combinations of name tokens 1000 times. All results are significant after correcting for multiple comparisons using Holm-Bonferroni correction.

Note: While iterating through the chosen list names, we disregard names that have been seen before (assuming that they have already been correctly assigned to their first guess). Therefore, when there is no difference between two different settings, it is because newly added pairs (e.g., the 50 new names pairs ranked from 500 to 550) have had one of the pair seen already in the first 500 and are therefore disregarded when gathering statistics.

2.B.5 Complete Results

In this section, we present the complete results for all the experiments performed in the paper. We explore varying hyperparameters as well as different measures. Our aim in providing the complete set of results is to demonstrate the robustness and generalizability of our observations across the hyperparameter space as opposed to cherry-picking specific statistics.

Name Reconstruction Experiment

Context Window Size – Cosine Measure

Size	Skipgram %-names	CBOW %-names	GLoVe %-names
1	8.5	2.8	1.3 [a]
3	61.5	44.2	26.4
5	65.6	53.9	46.3
7	68.3	55.2	52.8
9	68.5	56.7	53.9

Context Window Size – Cityblock Measure

Size	Skipgram %-names	CBOW %-names	GLoVe %-names
1	8.2	2.8	0.8 [a]
3	50.5	33.6	23.2
5	57.3	39.5	39.1
7	58.4	40.6	43.6
9	59.6	46.5	45.0

Embedding Size – Cosine Measure

Size	Skipgram %-names	CBOW %-names	GLoVe %-names
20	56.2	35.7	20.5
50	64.3	47.2	40.2
100	65.6	53.9	46.3
200	66.8	51.6	47.6
300	66.8	53.1	47.6

Embedding Size – Cityblock Measure

Size	Skipgram %-names	CBOW %-names	GLoVe %-names
20	50.7	25.8	18.2
50	55.4	36.6	33.4
100	57.3	39.5	39.1
200	56.7	40.6	41.0
300	56.2	42.3	41.4

Learning Rate – Cosine Measure

Size	Skipgram %-names	CBOW %-names	GLoVe %-names
0.0125	65.1	51.0	26.8
0.025	65.6	53.9	39.7
0.05	65.5	50.5	46.3
0.1	66.6	51.8	52.0

Learning Rate – Cityblock Measure

Size	Skipgram %-names	CBOW %-names	GLoVe %-names
0.0125	57.7	40.6	25.4
0.025	57.3	39.5	33.6
0.05	56.2	37.4	39.1
0.1	56.2	39.5	43.6

Negative Sampling Rate – Cosine Measure

Size	Skipgram %-names	CBOW %-names
1	66.4	52.0
5	65.6	53.9
10	65.8	51.4
30	66.0	49.9
64	65.3	49.9

Negative Sampling Rate – Cityblock Measure

Size	Skipgram %-names	CBOW %-names
1	55.8	40.8
5	57.3	39.5
10	55.6	38.5
30	54.5	39.1
64	56.4	38.1

Table 2.B.8: Of the first 600 paired tokens, sorted by ascending order, the percentage that are part of existing patient names as a function of various hyperparameter setting, using different measures. To determine statistical significance at the patient level, we calculated empirical p-values by randomly shuffling all n choose 2 combinations of name tokens 1000 times. All results are significant after correcting for multiple comparisons using Holm-Bonferroni correction except for those followed by a superscript ‘a’.

Name-Diagnostic Code Association Experiment

Context Window Size – Cosine Measure

Size	Skipgram distance	CBOW distance	GLoVe distance
1	0.037	0.027	–0.019
3	0.020	0.053	–0.003 [a]
5	0.009	0.046	–0.001 [a]
7	0.000 [a]	0.025	–0.004 [a]
9	–0.006 [a]	0.012	–0.001 [a]

Context Window Size – Cityblock Measure

Size	Skipgram distance	CBOW distance	GLoVe distance
1	3.91	7.59	4.85
3	2.88	28.53	5.69
5	2.33	39.55	5.45
7	1.84	47.10	5.12
9	1.51	51.61	5.54

Embedding Size – Cosine Measure

Size	Skipgram distance	CBOW distance	GLoVe distance
20	0.022	0.074	–0.005 [a]
50	0.013	0.057	0.004 [a]
100	0.009	0.046	–0.001 [a]
200	0.009	0.034	0.003 [a]
300	0.010	0.031	0.002 [a]

Embedding Size – Cityblock Measure

Size	Skipgram distance	CBOW distance	GLoVe distance
20	0.82	15.78	1.63
50	1.47	26.70	3.83
100	2.33	39.55	5.45
200	3.68	57.84	7.98
300	4.90	71.60	10.25

Learning Rate – Cosine Measure

Size	Skipgram distance	CBOW distance	GLoVe distance
0.0125	0.007	0.039	–0.006 [a]
0.025	0.009	0.046	–0.006 [a]
0.05	0.009	0.039	–0.001 [a]
0.1	0.009	0.043	0.004 [a]

Learning Rate – Cityblock Measure

Size	Skipgram distance	CBOW distance	GLoVe distance
0.0125	2.23	39.74	5.80
0.025	2.33	39.55	5.63
0.05	2.21	39.61	5.45
0.1	2.35	40.94	5.12

Negative Sampling Rate – Cosine Measure

Size	Skipgram distance	CBOW distance
1	0.008	0.041
5	0.009	0.046
10	0.010	0.043
30	0.009	0.043
64	0.006 [a]	0.038

Negative Sampling Rate – Cityblock Measure

Size	Skipgram distance	CBOW distance
1	2.31	40.24
5	2.33	39.55
10	2.37	40.11
30	2.28	39.62
64	2.23	39.62

Table 2.B.9: Difference between the in-group and outgroup as a function of various hyperparameter settings, using different measures. We use the Wilcoxon signed-rank test to compare the pairings of in- and out-groups for each name on the population level. All results are significant after correcting for multiple comparisons using Holm-Bonferroni correction except for those followed by a superscript 'a'.

Context Window Size – Cosine Measure

Size	Skipgram %patients	CBOW %patients	GLoVe %patients
1	6.4	4.2	0
3	3.9	5.5	0
5	3.3	5.8	0
7	3.1	2.8	0
9	2.2	2.8	0

Context Window Size – Cityblock Measure

Size	Skipgram %patients	CBOW %patients	GLoVe %patients
1	7.7	12.1	62.7
3	6.4	23.4	62.8
5	5.2	23.8	63.2
7	2.5	24.0	59.6
9	1.9	24.0	70.4

Embedding Size – Cosine Measure

Size	Skipgram %patients	CBOW %patients	GLoVe %patients
20	4.4	5.5	0
50	3.0	7.0	0
100	3.2	5.8	0
200	2.4	6.3	0
300	2.8	5.6	0

Embedding Size – Cityblock Measure

Size	Skipgram %patients	CBOW %patients	GLoVe %patients
20	6.3	22.9	68.5
50	3.8	22.7	75.2
100	5.2	23.8	63.2
200	4.1	24.9	58.8
300	3.6	24.1	50.8

Learning Rate – Cosine Measure

Size	Skipgram %patients	CBOW %patients	GLoVe %patients
0.0125	3.0	4.6	0
0.025	3.3	5.8	0
0.05	2.2	3.9	0
0.1	3.0	6.1	0

Learning Rate – Cityblock Measure

Size	Skipgram %patients	CBOW %patients	GLoVe %patients
0.0125	3.9	23.7	62.2
0.025	5.2	23.8	65.8
0.05	4.6	23.7	63.2
0.1	3.9	25.6	65.5

Negative Sampling Rate – Cosine Measure

Size	Skipgram %patients	CBOW %patients
1	3.0	5.6
5	3.3	5.8
10	2.8	5.6
30	3.1	5.0
64	2.7	4.9

Negative Sampling Rate – Cityblock Measure

Size	Skipgram %patients	CBOW %patients
1	4.9	24.8
5	5.2	23.8
10	4.2	23.8
30	4.6	24.6
64	3.9	24.3

Table 2.B.10: The percentage of patients whose diagnoses is identifiable due to a statistically significant difference between in-group and out-group as a function of various hyperparameter settings, using different measures. To determine statistical significant at the patient level, we calculated empirical p-values by randomly sampling the in- and out-groups generated using 1000 permutations of the same size from the same dataset.

Scenario Experiment

Context Window Size – Cosine Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5	GLoVe A@1, A@5
1	55.8, 56.7	61.8, 61.8	55.4, 56.9
3	55.6, 53.1	51.2, 52.6	60.5, 59.5
5	57.4, 55.6	53.6, 54.5	59.4, 57.2
7	57.4, 53.5	54.6, 53.9	55.9, 54.0
9	57.2, 53.2	53.7, 51.2	60.6, 56.7

Context Window Size – Cityblock Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5	GLoVe A@1, A@5
1	35.0, 54.1	40.6, 56.8	38.3, 53.7
3	43.7, 55.8	31.7, 54.9	40.8, 55.7
5	44.1, 55.6	31.9, 58.8	39.5, 54.6
7	40.9, 53.3	30.3, 57.7	38.3, 53.8
9	43.9, 55.9	32.9, 57.4	45.2, 57.4

Embedding Size – Cosine Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5	GLoVe A@1, A@5
20	58.0, 51.9	48.3, 51.4	60.5, 56.9
50	57.1, 52.4	48.7, 55.3	59.1, 55.4
100	57.4, 55.6	53.6, 54.5	59.4, 57.2
200	56.8, 56.6	56.3, 58.2	60.3, 58.1
300	55.0, 55.4	59.2, 57.7	57.9, 57.6

Embedding Size – Cityblock Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5	GLoVe A@1, A@5
20	48.8, 55.2	36.3, 56.5	41.3, 56.1
50	43.0, 54.2	33.8, 56.6	39.1, 55.4
100	44.1, 55.6	31.9, 58.8	39.5, 54.6
200	40.3, 56.3	32.0, 57.9	38.7, 56.2
300	39.8, 56.4	35.0, 59.8	40.6, 55.4

Learning Rate – Cosine Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5	GLoVe A@1, A@5
0.0125	59.1, 54.1	54.3, 50.9	55.6, 56.1
0.025	57.4, 55.6	53.6, 54.5	58.3, 55.5
0.05	54.9, 56.0	51.9, 55.6	59.4, 57.2
0.1	54.6, 53.9	56.8, 57.5	59.7, 55.5

Learning Rate – Cityblock Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5	GLoVe A@1, A@5
0.0125	41.6, 58.5	30.2, 57.6	38.6, 51.1
0.025	44.1, 55.6	31.9, 58.8	38.9, 51.6
0.05	45.2, 54.7	30.4, 55.9	39.5, 54.6
0.1	45.4, 54.9	32.4, 58.1	40.6, 57.8

Negative Sampling Rate – Cosine Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5
1	58.2, 52.4	55.4, 52.6
5	57.4, 55.6	53.6, 54.5
10	58.0, 53.2	53.5, 52.4
30	58.4, 53.8	53.2, 53.1
64	59.2, 54.0	50.1, 55.3

Negative Sampling Rate – Cityblock Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5
1	36.1, 51.3	32.0, 55.1
5	44.1, 55.6	31.9, 58.8
10	39.0, 52.9	31.8, 57.4
30	43.4, 56.5	33.3, 56.6
64	40.4, 54.9	31.8, 59.6

Table 2.B.11: Percentage of times (of 1000 random diagnosis selections) where using a word embedding–based attack beats the majority baseline for A@1 and A@5 for various hyperparameters and distance metrics.

Context Window Size – Cosine Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5	GLoVe A@1, A@5
1	0.02, 0.15	0.04, 0.13	0.02, 0.11
3	0.02, 0.12	0.01, 0.14	0.02, 0.13
5	0.02, 0.12	0.02, 0.11	0.03, 0.13
7	0.01, 0.10	0.02, 0.10	0.02, 0.12
9	0.01, 0.11	0.02, 0.11	0.02, 0.12

Context Window Size – Cityblock Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5	GLoVe A@1, A@5
1	0.06, 0.12	0.03, 0.11	0.08, 0.15
3	0.03, 0.11	0.00, 0.15	0.08, 0.15
5	0.05, 0.10	0.02, 0.15	0.08, 0.14
7	0.01, 0.13	0.02, 0.15	0.08, 0.15
9	0.02, 0.11	0.03, 0.15	0.08, 0.15

Embedding Size – Cosine Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5	GLoVe A@1, A@5
20	0.03, 0.11	0.01, 0.13	0.02, 0.14
50	0.02, 0.11	0.01, 0.12	0.02, 0.14
100	0.02, 0.12	0.02, 0.11	0.03, 0.13
200	0.02, 0.12	0.02, 0.11	0.03, 0.14
300	0.02, 0.12	0.03, 0.12	0.03, 0.13

Embedding Size – Cityblock Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5	GLoVe A@1, A@5
20	0.01, 0.10	0.01, 0.15	0.08, 0.13
50	0.05, 0.12	0.08, 0.15	0.01, 0.15
100	0.05, 0.10	0.02, 0.15	0.08, 0.14
200	0.05, 0.12	0.03, 0.15	0.08, 0.15
300	0.06, 0.12	0.02, 0.17	0.07, 0.15

Learning Rate – Cosine Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5	GLoVe A@1, A@5
0.0125	0.02, 0.12	0.01, 0.12	0.03, 0.13
0.025	0.02, 0.12	0.02, 0.11	0.02, 0.12
0.05	0.01, 0.12	0.02, 0.12	0.03, 0.13
0.1	0.01, 0.11	0.01, 0.12	0.03, 0.14

Learning Rate – Cityblock Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5	GLoVe A@1, A@5
0.0125	0.04, 0.11	0.02, 0.15	0.08, 0.14
0.025	0.05, 0.10	0.02, 0.15	0.01, 0.15
0.05	0.05, 0.11	0.03, 0.15	0.08, 0.14
0.1	0.05, 0.12	0.03, 0.15	0.08, 0.12

Negative Sampling Rate – Cosine Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5
1	0.02, 0.11	0.01, 0.12
5	0.02, 0.12	0.02, 0.11
10	0.01, 0.11	0.01, 0.11
30	0.02, 0.12	0.02, 0.12
64	0.01, 0.11	0.00, 0.11

Negative Sampling Rate – Cityblock Measure

Size	Skipgram A@1, A@5	CBOW A@1, A@5
1	0.05, 0.12	0.02, 0.16
5	0.05, 0.10	0.02, 0.15
10	0.03, 0.11	0.03, 0.15
30	0.05, 0.12	0.03, 0.15
64	0.02, 0.11	0.03, 0.15

Table 2.B.12: A@1 and A@5 for the set of diagnosis codes (constipation, diarrhea, vaginitis, sexual dysfunction, urinary infection, herpes genitalis, dementia, anorexia, alcoholism, threatened abortion, and AIDS) for varying hyperparameters and distance measures. The majority baseline is A@1 and A@5 of 0.00 and 0.07.

Chapter 3

Introducing Unsupervised De-identification (RaNNA)

3.1 Introduction

Having motivated the need for de-identification (Chapter 1) and demonstrated the limitations inherent to supervised approaches (Chapter 2), in this chapter I introduce an unsupervised approach to clinical de-identification.

The proposed method does not require any training; it employs a new "random replacement" paradigm (replacing each token in clinical notes with neighboring word vectors from the embedding space) to achieve 100% recall on the removal of sensitive information. The approach, named Random Nearest Neighbour Anonymization (RaNNA) achieves performance better than any current supervised "search-and-secure" paradigms (as measured by recall). In addition to achieving perfect recall on all PII (including sensitive information often not considered by current work), it is also the first technique which does not require data-holders to specify what types of PHI needs to be secured. After introducing the method, I demonstrate the utility of this paradigm on multiple corpora in a diverse set of classification tasks. Following this, Chapter 4 provides an in-depth risk assessment of this method by extending an existing probabilistic approach to risk assessment.

3.2 Background

Section 1.2.4 discusses the current approaches to de-identification. In summary, all existing approaches are supervised and make use of a variety of models to detect and secure sensitive tokens. Securing sensitive tokens can be done in one of two ways: deletion or

replacement. Deletion is often used because it is both simple and there is a minimal information loss which is a critical concern to most people working with clinical notes (Thomas et al., 2002). However, since no perfect search algorithm exists, sensitive data missed by deletion can be found by combing through the secured data for names and other sensitive information that was not removed.

Replacement is a more secure approach to de-identifying clinical records, as it is no longer clear which names have been missed by the de-identification algorithm and which have been randomly replaced. Unfortunately, this approach is more difficult to effectively implement and remains susceptible to attack by malicious actors (depending on the specifics of the replacement) (Carrell et al., 2013): malicious actors can look at instances of notes where there exist multiple differing names and leverage both context in the notes and external world knowledge to deduce real names (Carrell et al., 2013).

3.3 Methods: Random Nearest Neighbour Anonymization

My proposed approach, **Ra**ndom **N**earest **N**eighbour **A**nonymization (RaNNA), works by replacing each token t_i in a clinical note with another token t_r randomly selected from the nearest neighbours to the original token t_i in the embedding space. This replacement is done for all tokens in all notes and thus does not rely on training a model to detect sensitive tokens. This approach works by relying on the semantic properties of word embeddings: tokens which are related (i.e., appearing in similar contexts) will have closer positions in the numeric vector space (i.e., likely to be nearest neighbours and thus more likely to be replaced with each other). This means that the secured text should have similar properties (at minimum: similar numeric embeddings) for downstream tasks as the original text.

Replacing each token by randomly selecting from its nearest neighbours in an embedding space can be implemented in various ways:

- **Dataset-level vocabulary replacement:** In this implementation, all tokens in the vocabulary of *all notes* have a singular replacement value. For example, if the replacement set for X is $\{A, B, C, D, E\}$, and the randomly selected token for X is C, then all instances of X in the entire dataset are replaced by C.

- **Patient-level vocabulary replacement:** In this implementation, all tokens in the vocabulary of *a single patient's notes* have a singular replacement value. For example, if the replacement set for X is $\{A, B, C, D, E\}$, and the randomly selected token for X in the notes of a particular patient is C, then all instances of X in that patient's

records are replaced by C. However, it is possible that all instances of X for another patient are replaced by another token in the replacement set (e.g., A).

- **Note-level vocabulary replacement:** In this implementation, all tokens in the vocabulary of *a single note* have a singular replacement value. For example, if the replacement set for X is $\{A, B, C, D, E\}$, and the randomly selected token for X in a particular note is C, then all instances of X in that specific note are replaced by C. However, it is possible that all instances of X for another note (regardless of the patient) are replaced by another token in the replacement set (e.g., A).

- **Note-agnostic vocabulary replacement:** In this implementation, tokens do not have a singular replacement value. For example, if the replacement set for X is $\{A, B, C, D, E\}$, then *within the same note* different instances of X may be replaced different selections of $\{A, B, C, D, E\}$.

Each of the above implementation guarantees that the original sensitive tokens are no longer in the notes (since all tokens have been replaced). However, different implementations have different associated risks to patient notes if publicly released. The differences in terms of risk reduction for each of these implementations is explored in great detail the next chapter.

The examples above sample from a set of 5 nearest neighbours (i.e., the examples have used an obfuscation parameter of 5; represented by the equation N = 5). It is important that the degree of obfuscation is not too small (as it would then be too easy to reconstruct the original note), nor too large (as the new tokens would be completely unrelated). Below I explore the effects of using different values for the obfuscation parameter.

Table 3.3.1 shows a sample (artificial) clinical note along with de-identified versions of the note that result from traditional de-identification algorithms and from RaNNA with 3 different degrees of obfuscation (i.e., values of N). In this example, I highlight different groups of tokens in different colors to allow for easy tracking of how tokens are changed by the replacement mechanism of our proposal.

- Pink highlights name tokens. we can see that names are consistently replaced with other name tokens.

- Light green is used to highlight tokens associated with the patient's age. Here, we see that the replacements are related to age, but do lose precision. For example, with 'N = 5', '*fifty year old*' turns to '*sixty years olf*' (10 year difference in age and slight misspelling of the word old). However, looking at N = 7, we observe '*fifty year*

Note Type	Text
Original note	arnold smith is a fifty year old male, with a history positive for alcholic cirrhosis , hcv, and variceal bleeds, presenting to the ed with syncope and an inner lip laceration after fall on face
PII removal	*NAME* *NAME* is a *AGE* year old male, with a history positive for alcoholic cirrhosis, hcv, and variceal bleeds, presenting to the ED with syncope and an inner lip laceration after fall on face .
PII replacement	John Bobby is a sixty year old male, with a history positive for alcoholic cirrhosis, hcv, and variceal bleeds, presenting to the ED with syncope and an inner lip laceration after fall on face .
$N = 3$	muller doug was another seventy ycar monthold man, wth an hislory equivocal ibr alcohcllc cirhosis, hbv, arid varlceal bdoands, chief restrainting this er wth palpitations however a outer lid lacerations afer falling onthe cheeks
$N = 5$	seth joe remains another sixty years olf female, wit another hx positivity forthe abstainer steatohepatitis, ebv, however varicies bleed, chief restrainting its ahc vith presyncopc but acardiogenic supralateral lid abrasion thereafter concussion onthe forehead
$N = 7$	howard doug looks the thirteen decade monthoid man, wilh wiowill histoiy postive ofr exdrinker cirrhotic, hepc, similarly hemorroidal epistaxis, longstanding insalin ihe ahc wtih presyncopc similarly a posterior gingiva lacn before summer brewere scalp

Table 3.3.1: An artificial clinical note, and the result of applying RaNNA with 3 different degrees of obfuscation. RaNNA does not assume proper spelling or grammar from the input. The obfuscated notes have less readability but maintain important information for ML applications while covering PII.

old' turns to '*thirteen decade monthoid*'. '*monthoid*' is a misspelling of '*month old*' which makes the age difference here 40 years (130 months is 10 years).

- Blue highlights the gender which is replaced with either same words for the same gender or other genders.

- Orange, brown, and light grey are used to replace relevant medical terms. In orange, '*Alcoholic cirrhosis*' (scarring of the liver due to alcohol abuse) is replaced by '*alcohcclc cirhosis*' (a misspelling of the same symptom), '*abstainer steatohepatitis*' (abstainer is close to alcoholic, and steatohepatitis is a type of fatty liver disease), and '*exdrinker cirrhotic*' (again relevant to alcohol, and the adjective form of the noun). In brown we observe the medical term '*hcv*' (hepatitis C) being replaced with '*hbv*' (hepatitis B, a common coinfection), '*ebv*' (a virus in the hepatitis family), or '*hepc*' (alternative shorthand for hepatitis C). In grey, we have description of the fall which is replaced in a similar manner to that of '*Alcoholic cirrhosis*'.

The misspellings come from the corpus itself, as clinical texts are invariably filled with grammatical and spelling errors; correcting misspellings is still an unsolved research problem. It is important to stress that these replacements are not truly interchangeable (e.g., *hcv* and *hbv*) as they represent differing patient pathologies. Nonetheless, our empirical experiments below show that both our upstream evaluations and our downstream classification tasks are not affected by these substitutions. More experimentation is required to understand what the effect of applying RaNNA on a very clean dataset (i.e., no abbreviations or spelling mistakes) would be. It may be the case that overly standardized texts (e.g., where is no variation in the representation of any topic — no misspellings or abbreviations) could affect RaNNA negatively by forcing each token replacement to be a completely different topic. However, this needs to be explored further. In the discussion of the next chapter, we talk about the effects of the presence of misspellings on the privacy risk to RaNNA.

We also observe that all the names have been replaced with other names and not with misspellings of the original name. We hypothesize that this is because, considering all other tokens that occur in similar contexts, misspellings are less likely to occur than name tokens of other patients with the same ailment. This is the opposite to the situation for most other kinds of tokens (e.g., grammatical and medical terms) where misspelling replacement is much more likely, because the context of misspelled tokens is likely to be extremely similar.

3.4 Experiments

In this section, I quantify the effect that the proposed technique has on the performance of end-to-end machine learning models in various tasks (e.g., creating word embeddings and undertaking two clinical classification tasks).

First, I test the quality of word embeddings created by training on a set of data (before and after de-identification using RaNNA). To do this, I apply various intrinsic tests on both sets of word embeddings to measure the change in embedding quality on de-identified data. The results of this experiments has practical ramifications in two ways: 1) it would present a method to deal with the risks associated with releasing word embedding trained on clinical data 2.4, and 2) it is reasonable to assume that some researchers will want to create and use embeddings from data that have been provided to them after securing and using our method. For this analysis, I compare the performance of different degrees of obfuscation (i.e., how many nearest neighbours are replacement tokens sampled from) alongside the performance of out-of-domain datasets to assess the relative decrease.

Second, I evaluate the performance of models trained on fully anonymized notes for various downstream tasks (diagnostic code classification, International Classification of Diseases-9 (ICD-9) classification, and sentiment analysis). For each of these classification tasks, I perform the task using the original data (e.g., progress notes or movie reviews) and then once again using the same dataset de-identified using RaNNA. When applying RaNNA for each dataset, I train a novel embedding model on the same data used for classification and create the replacement set from this embedding model.

3.4.1 Intrinsic evaluation

To test the quality of word embeddings generated from the anonymized clinical data, I follow the testing strategy of Wang et al. (2018). They compared word embeddings generated from a variety of sources against human-annotated values of word relatedness for a list of clinically relevant terms.

To generate word embeddings for comparisons, I used consultation notes provided to ICES (previously known as the Institute for Clinical Evaluative Sciences) under data-sharing agreements with physicians for the purposes of evaluation and research. Consultation notes are written by physicians (specialists) and healthcare providers after interacting with a patient. These notes describe history collected, results observed, tests performed, and other details that a physician thinks are important for the treatment of the patient. I used all patient consultation notes (9,051,707 notes), composed of 949,782,513 tokens (2,612,592 unique tokens), Table 3.4.1.

	Counts
Number of patients	542,651
Number of notes	9,051,707
Number of tokens	949,782,513
Number of unique tokens	2,612,592

Table 3.4.1: Description of the consultation notes dataset.

For data preprocessing, all tokens were converted to lowercase, and had special characters and numbers removed. Tokens were split on whitespaces and punctuation tokens. Using these notes, I trained word embeddings using the continuous bag-of-words (CBOW) algorithm with an embedding size of 100, a context window of 5, and a negative sampling rate of 5. These values were picked only once as standard values because they have been shown to work in the clinical setting (Wang et al., 2018). RaNNA was then applied to secure the entire set of notes, sampling randomly from the 3, 5, or 7 nearest neighbours. From the newly anonymized set of consultation notes, I created new embeddings with the exact same set of parameters. These newly created embeddings were then compared, using intrinsic evaluation measures, against embeddings created on the original consultation notes. These intrinsic measures quantify the quality of embeddings created from anonymized consultation notes, defining quality as correlation to human judgments. I also included a comparison to the quality of embeddings trained on biomedical literature and news corpora to see whether the drop in quality from anonymization renders the specialized data useless in comparison to cheaper and lower-risk alternatives. The biomedical embeddings were trained on a snapshot of the Open Access Subset[1] of the PubMed Central in March 2016. PubMed Central is an online digital collection of freely available full-text biomedical literature containing more than 1.25 million biomedical articles, with 2 million distinct tokens in the vocabulary. The news corpus used was the Google News dataset.[2] This corpus is trained on approximately 100 billion tokens (composed of 3 million unique words or phrases).

For this evaluation, I used 4 word-pair lists composed of pairs of biomedical words and human annotated values of semantic relatedness between the word-pairs. The semantic relatedness is based on human judgments from medical coders and physicians that are provided in the datasets. Specifically, I analyzed the performance of word embeddings on the following datasets: 1) Pedersen et al. (2007) (30 medical term pairs), 2) Hliaoutakis

[1]Website: http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/ Accessed December 13, 2019.

[2]Website: https://code.google.com/archive/p/word2vec/ Accessed December 13, 2019.

(2005) (34 medical term pairs), 3) MayoSRS (Pakhomov et al., 2011) (101 clinical term pairs), and 4) UMNSRS (Pakhomov et al., 2010) (566 medical term pairs). Following Wang et al. (2018), if a term is composed of multiple words, the term is represented using the average of all the individual word vectors. FastText (Bojanowski et al., 2017) was used to generate word embeddings for out-of-vocabulary words. For each of the paired terms, I measured the cosine distance and presented the Pearson correlation in Figure 3.4.1.

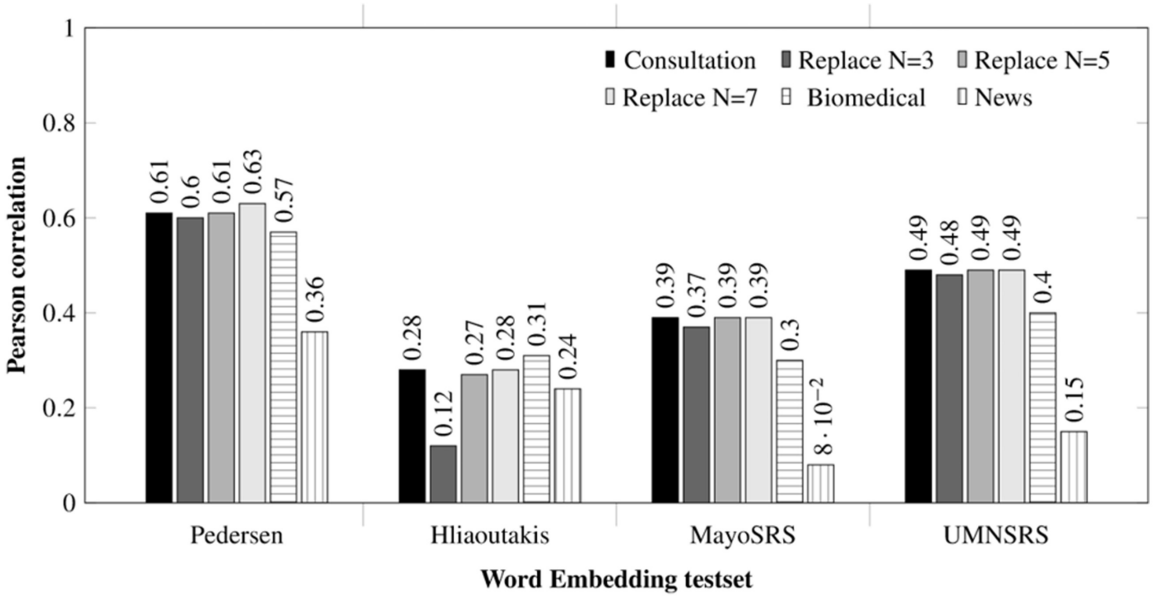

Figure 3.4.1: Pearson correlations of the intrinsic word embedding test. The baseline is in solid black, outputs from RaNNA are in shades of grey, and nonclinical sources are in horizontal and vertical grey lines. As shown, increasing the degree of obfuscation does not greatly impact the quality of the word embeddings.

The result of this experiment demonstrates that our anonymization technique does not greatly impact the quality of the embeddings (except for N = 3 on the Hliaoutakis word-pair list). Believing that this poor performance was simply caused by chance during the shuffling of the data, I re-ran the models 5 times using the same settings and observed that this bad run was, in fact, caused by chance. The average Pearson correlation is over 10 points higher and within 2 points of the un-anonymized model performance, shown in Table 3.4.2.

As observed, the quality of the anonymized word embeddings, as measured by these tests, is still higher than that of embeddings trained on out-of-domain corpora, informing us that: i) the noise added to the corpora by replacing each token with a random neighbor generally maintains the overall co-occurrence statistics (hence no significant change in the positive or negative direction), and ii) the embeddings created from anonymized data remain more informative (insofar as they correlate better with human annotations) than

	Consultation	N = 3	N = 5	N = 7
Pedersen	0.61	0.54 (0.51, 0.56)	0.64 (0.62, 0.65)	0.62 (0.61, 0.63)
Hliaoutakis	0.28	0.26 (0.19, 0.32)	0.26 (0.24, 0.27)	0.24 (0.19, 0.29)
MayoSRS	0.39	0.38 (0.37, 0.39)	0.39 (0.39, 0.40)	0.39 (0.38, 0.40)
UMNSRS	0.49	0.49 (0.48, 0.49)	0.49 (0.49, 0.49)	0.48 (0.48, 0.49)

Table 3.4.2: Pearson correlations (with 90% confidence interval bracketed beneath) of the intrinsic word embedding test done 5 times for each setting of $N = 3, 5, 7$ to measure the effect of randomly shuffling. As can be seen, conclusions drawn regarding comparable performance can still be observed. This also demonstrates that the bad result shown in the body was a result of bad luck/randomization.

embeddings trained on out-of-domain corpora, demonstrating that the anonymized data remains useful.

3.4.2 Extrinsic evaluation

Now, I will present multiple experiments which test the effect of our anonymization technique on classification tasks. I will test three different classification tasks:

1. Task 1: Diagnostic code classification using ICES data.
2. Task 2: ICD-9 using MIMIC III (Johnson et al., 2016).
3. Task 3: Sentiment Analysis classification using IMDB movie reviews (Maas et al., 2011).

For each of the tasks, I experiment with using embeddings created from 2 of the most popular word embedding algorithms (CBOW and Skipgram) to demonstrate that our results do not hinge on any single algorithm. These models are used both in classification (i.e., text representation), as well as de-identification (the embedding model used in RaNNA). I also test a variety of ML models to demonstrate that our technique preserves enough signal to remain useful for many different classifiers. In addition to the relevant clinical classification tasks, I also chose to do a sentiment analysis classification task because tokens of opposing sentiments tend to appear in similar contexts (e.g., "*This movie was **good***" and "*This movie was **bad***") and are therefore possible candidates for replacement and this task is likely to have the largest performance drop (Abdalla et al., 2019). If these models are to achieve respectable performance on sentiment analysis task, then it is likely that our algorithm does not substantially negate the signal or information encoded in text.

Task 1: ICES Diagnostic Code Classification

The first task tackled is diagnostic code prediction from progress notes. The progress notes and diagnostic codes used for this experiment are progress notes provided to ICES under data-sharing agreements with physicians for the purposes of evaluation and research. Progress notes are short notes written by the healthcare providers during or shortly after a patient encounter, and diagnostic codes are inputted by the healthcare providers for billing purposes. Table 3.4.3 presents a summary of the dataset. Due to the work and time pressures on healthcare providers, progress notes are often filled with shorthand, acronyms, and many errors both of spelling and of grammar.

	Counts
Number of patients	526,868
Number of notes	2,639,164
Number of tokens	703,698,773
Number of unique tokens	1,114,870

Table 3.4.3: Description of the progress notes dataset.

From the entire set of progress notes and diagnostic codes, I narrowed down our selection to notes that have been assigned to the top 10 most-common diagnostic codes. These codes represent a large variety in the type of care provided, ranging from hypertension to asphyxia. This results in 2,639,164 progress notes composed of 703,698,773 tokens (1,114,870 unique tokens). The resulting dataset does not suffer from any large class imbalance issues. Table 3.4.4 presents all classes as well as their relative distribution. The largest class (Anxiety Neurosis) constitutes 17% of the dataset, and the smallest class (Coccydynia) constitutes 5% of the dataset.

Diagnostic Code	**Counts**
Anxiety neurosis	446,764
Hypertension	386,414
Diabetes mellitus	286,930
Common cold	281,442
Arthralgia	278,446
Annual health examination	229,647
Well baby care	210,917
Abdominal pain	208,901
Asphyxia	166,181
Coccydynia	143,522
Total	**2,639,164**

Table 3.4.4: Description of the progress notes dataset.

For data preprocessing, all words were lower-cased, special characters and numbers were removed and words were split on space and punctuation tokens. All notes were truncated to 150 words in length. For this experiment, I performed a 10-way classification task. For the traditional classifiers, I tested a logistic regression classifier[3] and an SVM classifier[4] with base settings from the scikit-learn package (version 0.20.3). For these classifiers I represented each note using TF-IDF vectors[5], considering an n-gram range from 1 to 3 with a minimum occurrence of at least 3 times. The experiment followed a stratified k-fold ($k = 3$) validation scheme.

Table 3.4.5, presents the results of the various models attempted, and Figure 3.4.2 plots the change in performance as a result of differing degrees of obfuscation caused by our algorithm.

Obfuscation	**N = 0**	**N = 3**	**N = 5**	**N = 7**	**N = 9**	**N = 3-14**
ICES_SG0_CNN	72.3	72	71	71	71	71.6
ICES_SG1_CNN	75.7	75	74	74.33	73.7	74.7
ICES_SG0_LOG	78	77.3	77	76.3	76.3	77.3
ICES_SG1_LOG	78	77.3	77	76.3	76.3	77.3
ICES_SG0_SVM	78	77.3	77	76.3	76.3	77
ICES_SG1_SVM	78	77	77	76.3	76.3	77

Table 3.4.5: Performance (F_1 score) of different models and varying degrees of obfuscation for diagnostic code classification. Each model name is broken into three parts: 1) The task performed (ICES for diagnostic code classification), 2) the word embedding representation used to randomly replace the tokens (either SG0 or SG1 for CBOW or Skipgram), and 3) the type of model used to classify the texts.

We observe the general trend that increased obfuscation results in decreased classification performance. However, the observed decreases are small in magnitude and bottom out at approximately 2 percentage points. The very slight decrease in performance lends credence to the claim that data secured using RaNNA remains useful for clinical researchers for initial pilots exploring the feasibility of automated classification.

Task 2: MIMIC ICD-9 Code Classification

The second task tackled is another clinical classification task. However, to further demonstrate the signal-preserving properties of our privacy technique, I use a new dataset and

[3] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[4] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

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

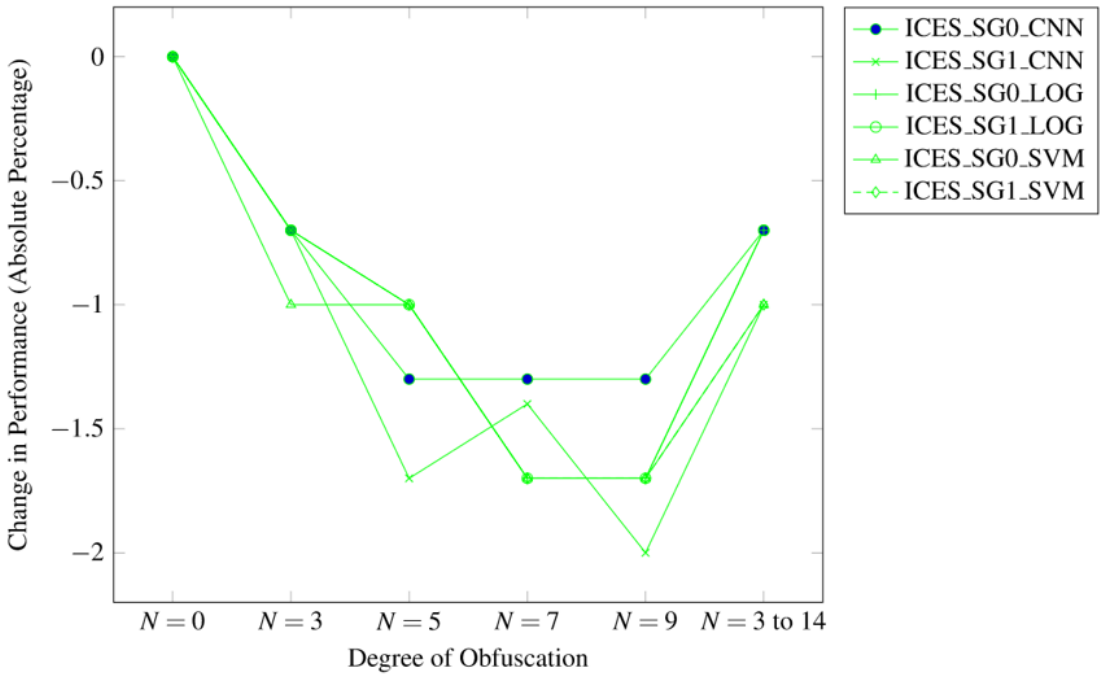

Figure 3.4.2: Absolute percentage change of performance (F_1 score) as a function of different obfuscation settings for diagnostic code classification with varying degrees of obfuscation. Each model name is broken into three parts: 1) The task performed (ICES for diagnostic code classification), 2) the word embedding representation used to randomly replace the tokens (either SG0 or SG1 for CBOW or Skipgram), and 3) the type of model used to classify the texts.

apply different methods. This classification task makes use of the Multiparameter Intelligence Monitoring in Intensive Care (MIMIC-III) dataset which consists of electronic health records for 38,597 adult patients and 7,870 neonates admitted to the intensive care unit of the Beth Israel Deconess Medical Center between 2001 and 2012 (Johnson et al., 2016). The dataset contains approximately 2 million clinical notes of varying types (discharge summaries, nursing notes, radiology reports, etc.).

This classification task made use of all 52,000 discharge notes. All the ICD-9 codes were re-labelled into smaller classes of codes by taking advantage of the ICD hierarchy as done by Liendo et al. (2019), presented in Figure 3.4.3.

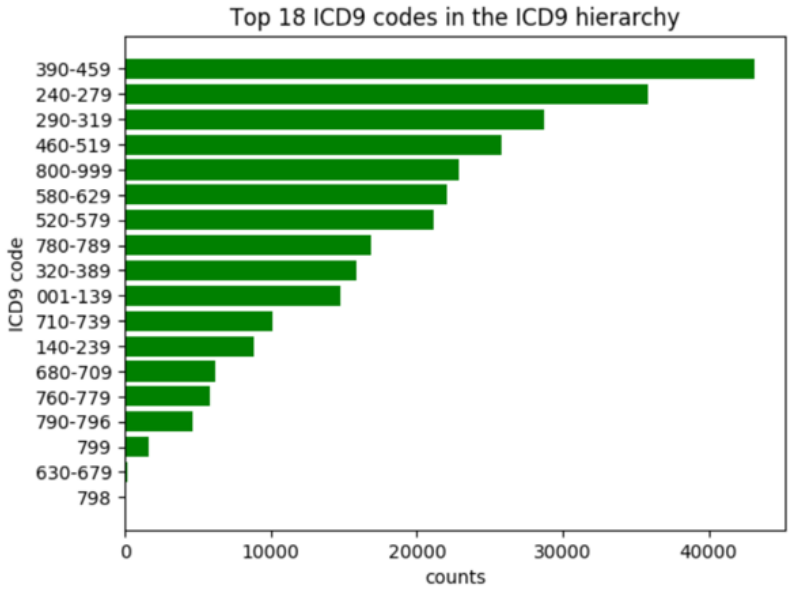

Figure 3.4.3: Frequency of the top 18 ICD-9 codes in the ICD-9 hierarchy.

For preprocessing, I followed the same approach taken by Liendo et al. (2019). That is, all characters were lower-cased, special characters were removed, contractions were separated, numbers were canonized, and words were split on space tokens. Although the maximum discharge note had a length of 10,000 words, I chose a representation length of 5,000 as that covered over 99.5% of the notes. The CNN model has 4 concurrent filter sizes of 2,3,4, and 5 with 100 filters each, which are concatenated and passed to a final dense layer. The LSTM model contains a single LSTM layer with 50 hidden units. For the embedding layer we used a CBOW model that was pretrained on the anonymized text undergoing classification. The full code for each of the models is publicly available[6].

As discharge notes could be associated with more than ICD-9 code, this task was treated as a multi-label classification problem. I employed the stratified k-fold (k = 5) validation

[6] https://github.com/zliendo/AI_MedicalNotes

scheme. Table 3.4.6 presents the results of the various models attempted, and Figure 3.4.4 shows the change in performance as a result of differing degrees of obfuscation caused by our algorithm.

Obfuscation	**N = 0**	**N = 3**	**N = 5**	**N = 7**	**N = 9**	**N = 3-14**
MIM_SG0_CNN	77.16	77.18	76.22	75.64	75.02	74.62
MIM_SG0_CNNwATTN	64.58	63.12	63.4	62.3	61.66	61.78
MIM_SG0_LSTM	59.02	59.06	59.06	59.08	59.08	59.1
MIM_SG0_LSTMwATTN	70.92	72.68	72.08	72	70.16	71.4

Table 3.4.6: Performance (F_1 score) of different models and varying degrees of obfuscation for the ICD-9 code classification task on MIMIC III. Each model name is broken into three parts: 1) The task performed (MIM for ICD-9 code classification on MIMIC III), 2) the word embedding representation used to randomly replace the tokens (SG0 for Skipgram), and 3) the type of model used to classify the texts.

Again, we observe the general trend that increased obfuscation results in decreased classification performance. However, the observed decreases are small in magnitude, sometimes positive, and bottom out at approximately 3 percentage points. There are certain instances where applying RaNNA resulted in improved classification performance. Such an increase can be a result of multiple factors. First, as these gains in performance are not substantial, they could be a result of noise and would be lost with repeated experiments — akin to our results in the intrinsic evaluation. Another possible explanation is that the increases in performance after obfuscation indicate that the original model was over-fitting the original training set, an observation noticed by Liendo et al. (2019) (the original authors of the approach mirrored). Regardless of the underlying reason, the maintained slight decrease in performance despite different models used and a different underlying dataset further strengthens the claim that data secured using RaNNA remains useful for clinical researchers for initial pilots exploring the feasibility of automated classification.

Task 3: Sentiment Analysis Classification Task

To demonstrate that the previous results are both reproducible and generalizable, I performed a third extrinsic evaluation in a different domain, sentiment analysis. I purposefully chose sentiment analysis as tokens of opposite sentiment tend to appear in similar contexts (i.e., "*This movie was good*" and "*This movie is bad*"). This would result in our algorithm switching tokens of opposite sentiment in the movie reviews thereby tricking the classifier (Abdalla et al., 2019). By showing that RaNNA does not greatly negate the signal in sentiment analysis classification I highlight its ability to preserve information, and its utility

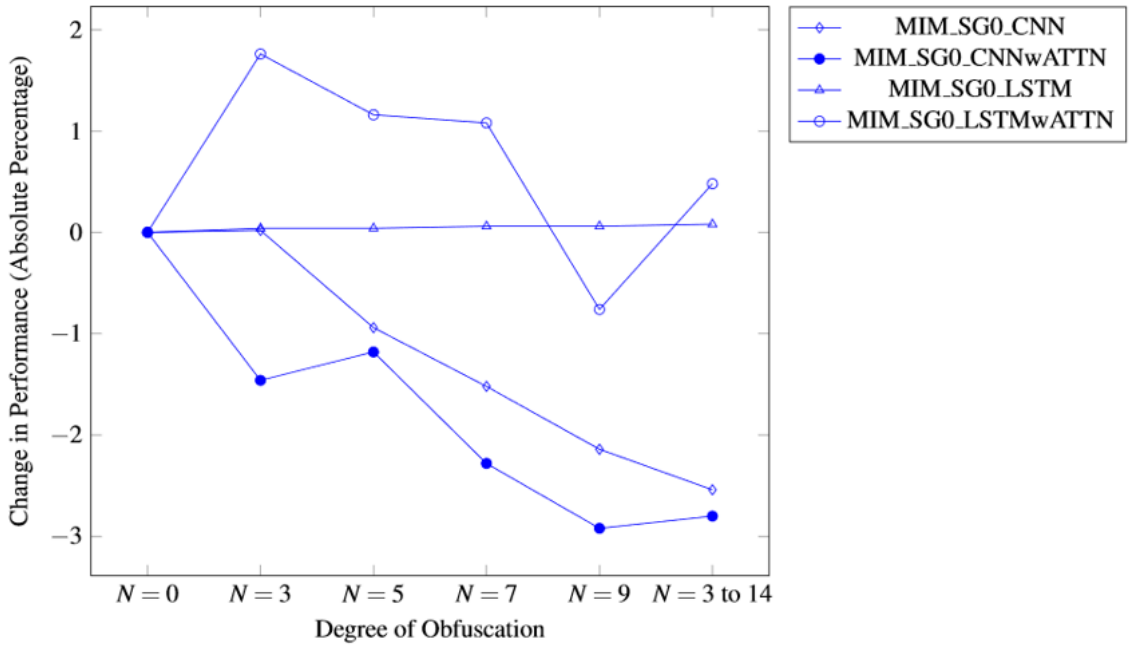

Figure 3.4.4: Absolute percentage change of performance (F_1 score) as a function of different obfuscation settings for ICD-9 code classification task on MIMIC III with varying degrees of obfuscation. Each model name is broken into three parts: 1) The task performed (MIM for ICD-9 code classification on MIMIC III), 2) the word embedding representation used to randomly replace the tokens (SG0 for Skipgram), and 3) the type of model used to classify the texts.

across multiple tasks. These experiments were performed using the Large Movie Review dataset (Maas et al., 2011).

The Large Movie Review dataset is composed of 50,000 movie reviews with an equal number of positive and negative reviews. The dataset is constructed such that no more than 30 reviews are allowed for any single movie. Movies are rated on a 10-point scale. Reviews with 4 or fewer stars are labelled as negative, and reviews greater than 7 are labelled as positive. Neutral reviews are not included in this dataset. For this task both neural networks as well as more traditional classifiers. For the traditional classifiers, we tested a logistic regression classifier[7] and a SVM classifier[8] with base settings from the scikit-learn package (version=0.20.3). For these classifiers we represented the data using TF-IDF vectors[9], considering an n-gram range from 1 to 3 with a minimum occurrence of at least 3 times. For the neural based approach, a CNN was used with 2 concurrent filter sizes of 3 and 8 with 10 filters each, which are concatenated and passed to a dense layer of size 50[10]. For all classifiers, I employ the stratified k-fold (k = 5) validation scheme.

Table 3.4.7 presents the results of the various models attempted, and Figure 3.4.5, shows the change in performance as a result of differing degrees of obfuscation caused by RaNNA.

Obfuscation	**N = 0**	**N = 3**	**N = 5**	**N = 7**	**N = 9**	**N = 3-14**
Sent_SG0_CNN	90	86.4	85.2	84	83.4	82.8
Sent_SG1_CNN	90	87	85.6	84.6	84	84.2
Sent_SG0_LOG	89.8	86.2	84.8	83.6	83	83
Sent_SG1_LOG	89.8	86.6	85.4	84.4	83.8	83.8
Sent_SG0_SVM	91.2	87	85.8	84.4	83.8	83.4
Sent_SG1_SVM	91.2	87.6	86.2	85.2	84.4	84.2

Table 3.4.7: Performance (F_1 score) of different models and varying degrees of obfuscation for the sentiment classification task. Each model name is broken into three parts: 1) The task performed (Sent for Sentiment Classification), 2) the word embedding representation used to randomly replace the tokens (either SG0 or SG1 for CBOW or Skipgram), and 3) the type of model used to classify the texts.

As with the previous tasks, we observe that increased security correlates with a decrease in performance. The decreases observed for this task are approximately double those ob-

[7] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[8] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

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

[10] Code for our approach can be found at the following link: https://github.com/alexander-rakhlin/CNN-for-Sentence-Classification-in-Keras which is based on the work of Kim (2014)

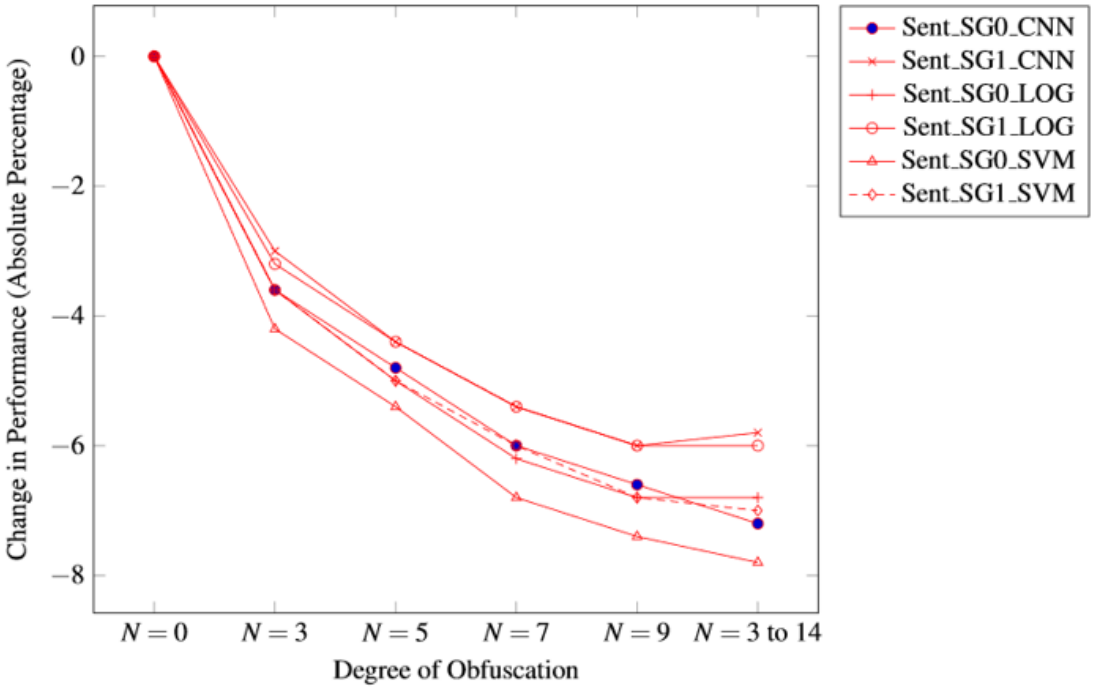

Figure 3.4.5: Absolute percentage change of performance (F_1 score) as a function of different obfuscation settings for sentiment classification task with varying degrees of obfuscation. Each model name is broken into three parts: 1) The task performed (Sent for Sentiment Classification), 2) the word embedding representation used to randomly replace the tokens (either SG0 or SG1 for CBOW or Skipgram), and 3) the type of model used to classify the texts.

served for the clinically relevant tasks. This is expected as tokens of opposite sentiment are expected to be switched as they often occur in the same contexts. However, with a maximum performance drop of less than 8 percentage points, the secured data remains useful.

Summary Results

Table 3.4.8 presents the complete set of experiments conducted. Figure 3.4.6 presents the average decrease in performance for different degrees of obfuscation for all tasks performed.

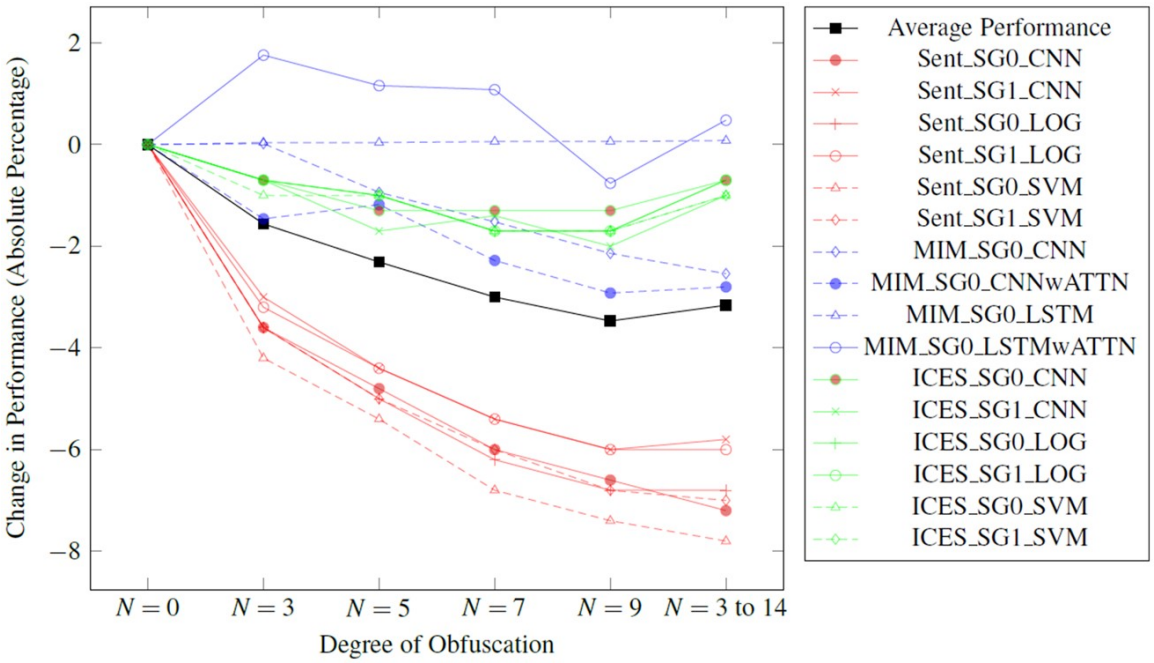

Figure 3.4.6: Absolute percentage change of performance (F_1 score) as a function of different obfuscation settings for various tasks, settings, and models. Each model name is broken into 3 parts: 1) The task performed, of which there are 3 (Sent for Sentiment Classification, MIM for MIMIC III ICD-9 code classification, or ICES for ICES diagnostic code classification); 2) the word embedding representation used to learn randomly replace the tokens (either SG0 or SG1 for CBOW or Skipgram); and 3) the type of model used to classify the texts. More details regarding each of these settings and models can be found in the Supplementary Material.

From Figure 3.4.6, we observe that increased obfuscation generally results in decreased classification performance (measured using F_1 score). However, the observed decreases are small in magnitude, no more than approximately 4 percentage points for clinical tasks, thereby demonstrating the utility of data protected using our method.

Obfuscation (N)	Models for ICES task	Models for MIMIC task	Models for IMDB task
N = 0	Logistic regression[a]	CNN[a]	Logistic regression[a]
N = 3	SVM[a]	CNN with attention[a]	SVM[a]
N = 5	CNN[a]	LSTM[a]	CNN[a]
N = 7	Logistic regression[b]	LSTM with attention[a]	Logistic regression[b]
N = 9	SVM[b]		SVM[b]
N = 3-14	CNN[b]		CNN[b]

Table 3.4.8: Summary of all experiments. The list of models is organized column-wise by task. In brackets, we present the word embedding algorithm used to randomly replace each token (CBOW or Skipgram). We also present the size of the nearest neighboring set of obfuscating tokens from which we randomly sample. For obfuscation settings, N = 0 is the evaluation on the original unprotected dataset, and for $N = 3 - 14$, we varied the size of the nearest neighbor set for each word between 3 and 14 instead of holding it constant for each token. [a] CBOW [b] Skipgram

3.5 Discussion

Our experiments demonstrate that the obfuscated data created by RaNNA remains useful for many ML tasks. By replacing tokens with other tokens that occur frequently in the same context, RaNNA does not change the underlying positioning of word tokens in the embedding space greatly.

By maintaining the relative position of tokens in an embedding space, the performance of neural-based ML and NLP classification methods is not greatly impacted and may be used for pilot research projects. Of course, there are tasks for which our technique may not be the optimal approach for anonymizing data — for example, clinical named-entity recognition and tasks requiring human interaction or interpretability. There may also be other data sources where RaNNA is not as effective. For example, progress notes are often much more loosely structured and a grammatical than consultation notes, which could create much nosier embeddings resulting in poorer replacement sets. Fortunately, we did not observe this with our set of progress notes, but this remains possible for other noisier or smaller data sources. More research is required to evaluate the impact of our method on other tasks and data.

Our classification experiments have been performed on tasks where each class can be considered large and quite distinct from other classes. Quantifying how RaNNA performs in the face of great data imbalances, the presence of rare classes (e.g., the long-tail problem (Lindvall and Molin, 2020)), or classes that are very similar to each other is left for future work.

Theoretically, this approach could also use contextual word embeddings (Devlin et al., 2019) — embeddings that change depending on the context. Future work would have to show that the trends observed above hold and study whether model size and its contextual nature have any negative effects regarding random neighbor generation or privacy.

The perfect recall of RaNNA comes at the cost of agrammatical, and sometimes even unreadable, transformations. Using a dictionary-based search method to preclude certain words from being replaced would increase readability; however, choosing to keep a pre-determined list of informative words, for example, stop-words or medical names (some of which might also be human names, such as Parkinson's) would increase readability as well as risk. This reintroduction of risk should only be done for specific use-cases and under controlled access measures.

Quantitative assessment of the risk of publicly sharing clinical notes secured using RaNNA is a challenge, since mainstream measures of security (often simply classification measures) are insufficient, and therefore so are existing shared tasks (Stubbs et al., 2017, 2015). Using precision, recall, or Carrell et al. (2013)'s approach would not be appropriate because RaNNA, by design, has perfect recall and very low precision. Chapter 4 presents a quantitative risk-assessment of the various implementations of RaNNA.

I am not advocating that this method should be used on the input of a model deployed in real clinical settings — the loss of precision for notes and lack of explainability for any models trained on texts secured using RaNNA present issues. Rather, I propose that this method can be used in pilot classification tasks very quickly and at low cost. For example, to explore the possibility of automatically classifying text, data holders can share data that have been anonymized using our method at reduced risk. If any of the research groups involved were able to achieve acceptable performance, then that collaboration or development could be brought in-house to work on unobfuscated data. RaNNA allows data holders to outsource ML research and data analytics to outside research groups without the overhead of creating and maintaining a manually secured data repository.

3.6 Conclusion

In this work, I introduced a novel anonymization technique for clinical notes that can be applied to any body of text. The method:

- is generalizable across different types of text data, as demonstrated by our application to consultation notes, progress notes, and movie reviews,

- guarantees that all PHI will be randomly replaced with perfect recall, a claim that cannot be made of algorithms that currently exist in the literature, and
- does not result in a significant decrease in performance for classification tasks using either neural networks or more traditional machine learning.

RaNNA provides complete coverage on all sensitive information at the cost of introducing some noise that reduces human readability. However, we have seen through our intrinsic tests (i.e., correlation scores with human annotated word-pair lists) and extrinsic tests (i.e., 10-way diagnostic code classification and binary sentiment classification) that the amount of noise introduced does not negate the benefits of having a specialized corpus to create embeddings for certain ML and NLP classification tasks.

Chapter 4

Risk Analysis of RaNNA

4.1 Introduction

In the previous chapter, I introduced Random Nearest Neighbour Anonymization (RaNNA), a method that randomizes every token in a clinical note while retaining the properties that make the note suitable for use as data for machine learning. More specifically, by replacing each token t_i by token t_j, randomly selected from the nearest neighbours of t_i in an embedding model trained on the data[1], RaNNA guarantees that the new token t_j is semantically related to some degree and that all personally identifying information (PII) has technically been removed.

In this chapter, I assess the risk associated with publicly releasing data that has been secured using RaNNA. To this end, I first explore how past work in clinical de-identification and cryptology has assessed the risk to their proposed techniques. Then, I quantify the risk associated with releasing embeddings and notes secured using RaNNA using various measures and in different hypothetical attack scenarios.

4.2 Background

4.2.1 Risk assessment in clinical de-identification

As highlighted in the previous chapter, the most common method to protect the identities of patients in clinical notes is to remove or replace a pre-defined set of PHI. These methods are evaluated using classification measures (e.g., precision and recall) (Uzuner et al., 2007; Taira et al., 2002; Dernoncourt et al., 2017; Yang et al., 2019; Neamatullah et al., 2008;

[1]The set of nearest neighbours of a token is referred to by the following terms interchangeably: replacement set and semantically-proximate token set.

Douglass et al., 2004). Scaiano et al. (2016) identified three limitations of using classification measures to evaluate de-identification approaches. First, classification measures do not distinguish between misclassifications all happening within a single note versus the same number of errors spread across multiple notes; the risk of re-identification is greater in the first instance. Second, when using recall, all entities are treated as having equivalent risk, yet a missed name presents greater risk than a missed address. Lastly, these classification measures do not take into account the distribution of PHI across datasets. To address these limitations, Scaiano et al. (2016) proposed assessing the risk associated with publicly releasing notes by estimating the probability of re-identification given a dataset's properties and a de-identification method's performance. Their approach incorporates the probability of attacks by bad faith actors taking place, the probability of the de-identification method missing PHI, and the probability of an attacker identifying such missed attempts. Furthermore, they differentiate between direct identifiers (information that can confidently be attributed directly to a single patient, e.g., names) versus indirect identifiers (where multiple indirect identifiers are required to infer the identity of a patient, e.g., sex, ethnicity, age). In later sections, I will extend this approach of risk assessment to quantify the risk associated with released notes secured using RaNNA.

Carrell et al. (2013) evaluate the risk associated with hiding PHI missed by de-identification methods 'in plain sight' by testing whether experts can detect when PHI has been missed after obfuscation (Hirschman and Aberdeen, 2010). In these experiments, the de-identification method used belongs to the search-and-replace approach, where detected PHI is replaced with PHI of a similar type. They observe that, on their small pilot study, using this method of obfuscation resulted in a 10-fold reduction in the risk of accidentally disclosing sensitive information.

In addition to the hypothetical risk assessments presented above, there is a multitude of work demonstrating re-identification of sensitive information on real health data. However, a systematic review performed by El Emam et al. (2011), found that most of these studies were performed on small datasets and that most of these datasets did not meet high privacy standards; only two out of the fourteen datasets used best practices.

4.2.2 Cryptanalysis

At first glance, RaNNA appears quite similar to a substitution cipher: a cryptographic algorithm that substitutes each letter or symbol (in the case of RaNNA, tokens) by a different plain-text letter or symbol (Shimeall and Spring, 2013). If one was to ensure that, for each original token, RaNNA would select a singular unique replacement token, then RaNNA

would functionally be a substitution cipher. However, explicitly defining the components and purpose of a cipher makes the distinction between ciphers and RaNNA clear.

First, the goal of a using a cipher is to encrypt a text. Informally, the output of a cipher is secure if "regardless of any information an attacker already has, a ciphertext [leaks] no additional information about the underlying plaintext" (Katz and Lindell, 2020). This is not the goal of RaNNA or other de-identification methods applied in the clinical setting. Clinical de-identification does not attempt to prevent the leakage of any additional information to those with access (as is the case with ciphers); rather its focus is on preventing the association of any particular note with any real-world individual. A note encrypted using traditional ciphers would be of little use in medical research if not decrypted[2].

In addition to difference of purpose, there are also structural differences between the algorithmic components of RaNNA and of private-key encryption methods (e.g., substitution ciphers). Traditionally, private-key encryption methods have the following components: 1) a procedure Gen for generating keys k, 2) a procedure Enc for encrypting message m, and 3) a procedure Dec for decryption. An encryption scheme must satisfy the following equation, Equation 4.1.

$$\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m \tag{4.1}$$

RaNNA differs from private-key encryption in multiple ways. First, RaNNA has no procedure for decryption. While it is possible to recreate the original text m, after applying RaNNA on m, since RaNNA is not deterministic there can be no confidence regarding possible attempts at decryption (without already possessing m). Second, RaNNA does not use a key-based encryption. For these reasons, directly building on the mathematics used to analyze the security of ciphers will not be possible when analyzing RaNNA as they assume the existence of the components of traditional ciphers.

While the mathematics of cryptanalysis may not directly apply to RaNNA, generating a risk assessment by reasoning about threat models remains useful. Threat models specify what level of "power" (often in the form of access to information) an adversary has. Borrowing from Katz and Lindell (2020), I list all threat models in order from least to greatest power below. The listed threat models are customized to fit our clinical use-case.

- **Ciphertext-only attack:** This is the most basic threat model where the attacker only has access to the de-identified clinical notes.

[2]There is ongoing research into homomorphic encryption: an encryption scheme that enables meaningful computation on data that is still encrypted (Naehrig et al., 2011; Zhou and Wornell, 2014). However, research on this topic is still preliminary and as of writing there have not been any meaningful demonstrated applications of homomorphic encryption in an applied NLP setting.

- **Known-plaintext[3] attack:** This threat model assumes the attacker to have access to both the de-identified clinical notes and one or more plaintext notes. We assume, in this threat model, that the plaintext notes are paired with their de-identified counterparts.

- **Chosen-plaintext attack:** In this threat model, the attacker is able to obtain the de-identified note for any chosen plaintexts.

- **Chosen-ciphertext attack:** In this threat model, the attacker is able to obtain the plaintext note for any chosen ciphertext.

Not all these threat models apply to the situation RaNNA was developed for. For example, attackers will not be able to request researchers encrypt a text for their attack (as is the case for the last threat model 'Chosen-ciphertext attack'). As clinical notes secured using RaNNA are to be released at once (i.e., it is not a system meant for querying), the second-last threat model 'Chosen-plaintext attack' is also not applicable.

In the two sections below, we will perform a risk assessment of RaNNA. First, in section 4.3, we will assess the risk surrounding the release of secured word embeddings models. Second, in section 4.4, we will assess the risk of releasing notes secured using RaNNA.

4.3 Risk assessment: Releasing word embeddings

In this section, I calculate the risk surrounding publicly releasing word embeddings trained on data secured using RaNNA. First, I list the different ways RaNNA's randomization can be implemented. Recall from Chapter 3 that there were four ways that RaNNA's replacements can be implemented:

- Dataset-level vocabulary replacement.

- Patient-level vocabulary replacement.

- Note-level vocabulary replacement.

- Note-agnostic vocabulary replacement.

For each of the above implementations, there are two threat models that should be considered:

[3]The term 'plaintext' is used to refer to the original and unsecured text.

- **Ciphertext-only attack:** In this threat model, we assume the attacker only has access to the de-identified notes.

- **Known-plaintext attack:** In this threat model, we assume the attacker has all of the plaintext notes of some (but not all) patients in the dataset. The level of risk is dependent on the number of notes and the number of unique patients' notes acquired by the attacker. In our analysis, we tackle the higher risk threat model assuming that the attacker has all the original notes for multiple patients in the dataset.

Below, I quantify the risk associated with releasing embeddings securing using each implementation of RaNNA for both threat models. When attacking word embeddings secured using RaNNA there are two steps in the attack: 1) Reconstructing the replacement set of each token, and 2) Given all replacement sets, correctly (re-)identifying a patient. We know that if embeddings trained on unsecured text can likely be attacked, Chapter 2. Thus, if reconstruction is trivial, then the same risk to unsecured datasets will apply.

The difficulty of reconstructing the replacement set depends on various factors. One factor is the obfuscation parameter used (i.e., the size of the replacement set); the smaller the size of the replacement set the easier reconstruction will be. Another factor is the implementation method. To get an understanding of the risk of re-creating the replacement set of the original text from released notes or released embeddings we present various measures that help us understand the associated risk.

4.3.1 Local Clustering Co-efficient

The first such measure is the local clustering co-efficient (LCC) of token sets. The LCC is a measure from graph theory to arrive at a fine-grained understanding of how inter-connected replacement sets are. A graph G is defined as having a set of vertices V and edges E. For this analysis, the vertices V is the vocabulary T. Using the embedding model an edge is placed from token t_i to t_j if t_j is in the replacement set of t_i. An edge in the opposite direction is placed if t_i is in the replacement set of t_j.

After having constructed the graph the local clustering coefficient C_i is measured for each token t_i; a measure of how close the neighbours s_i of token t_i are to being a clique (i.e., a complete graph). The higher the coefficient, the easier it is to learn the full replacement set. The local clustering coefficient is defined in Equation 4.2. This sort of measure, illustrated in Figure 4.3.1, would indicate how easy a ciphertext-only attack will be on released embeddings.

Let $G = (V, E)$ be a graph built from vertex set V and edge set E. Let $e_{i,j}$ be an edge from vertex v_i and vertex v_j. Then the neighbourhood N_i for vertex v_i is defined as its immediately connected neighbours: $N_i = \{v_i\} \cup \{v_j : e_{i,j} \in E\}$. The vertex v_i is defined as being part of the neighbourhood set per the traditional LCC set-up where the central node is included as part of set of nodes considered. This is also required for the math of the co-efficient to work out cleanly (i.e., have a maximum value of 1).

$$C_i = \frac{|e_{i,j} : v_i, v_j \in N_i, e_{i,j} \in E|}{|N_i|(|N_i| - 1)} \tag{4.2}$$

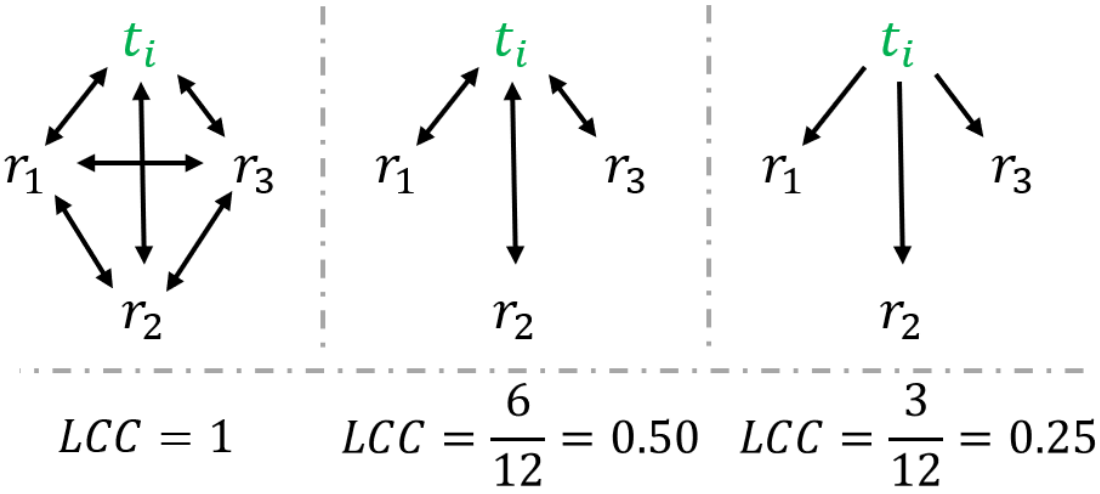

Figure 4.3.1: Illustration of the LCC of token t_i in different scenarios. The more interconnected the nearest neighbours of t_i are, the higher the LCC.

Table 4.3.1 presents an analysis of the local clustering coefficient for various replacement set sizes. Unlike other measures presented below, since the LCC is defined using the original embedding before the application of RaNNA, the implementation used will not affect the LCC. Thus, we only present 1 set of results; later results will present different calculations for different implementations. We can see that the larger the replacement set size the less cliquey the replacement sets are (both by mean and median LCC). We observe an LCC of approximately 0.3 for the larger replacement set sizes. Therefore, I can say, that subjectively, it is not a trivial task for an adversarial actor to confidently reconstruct the nearest neighbours of a word.[4]. This is especially true if they use a naïve approach of simply placing all the nearest neighbours as the proposed replacement set as this will (as demonstrated by Table 4.3.1 likely not be correct.

[4]My subjective analysis is inspired by the widespread interpretation of correlation values. Here, values of 0.8 and greater are considered to be indicative of high correlation, 0.5–0.8 is considered medium and below 0.5 is considered low

Replacement Set Size	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	0.25	0.47	1.0	0.25	0.33	0.42	0.58	0.75
5	0.17	0.42	1.0	0.17	0.27	0.40	0.53	0.67
7	0.12	0.39	0.98	0.16	0.25	0.36	0.50	0.64
9	0.10	0.37	0.97	0.13	0.23	0.34	0.48	0.61
3-14	0.08	0.27	0.95	0.10	0.17	0.24	0.34	0.47

Table 4.3.1: Local clustering coefficient for various replacement set sizes.

4.3.2 Reciprocity of replacement sets

Another measure that I propose to capture the ease of reconstructing the replacement sets from released notes (or embeddings trained on secured notes) is what I term the *reciprocity of replacement sets*. I use this term to describe, for a token t_i, the proportion of tokens t_j from the secured embeddings that have t_i in their replacement set and are in the replacement set of token t_i in the original embedding. For example, given an original token A with a replacement set of $\{B, C, D\}$, it would be easiest to rebuild if the replacement sets of B, C, and D, when trained on the secured notes/in released embeddings, all had A in them *AND* no other token in the secure embeddings had A in their replacement set (the reciprocity would be 1 as all 3 tokens with A in their replacement set are also in the replacement set of A). However, if the tokens that had A in their replacement set (when trained on the secured embeddings) were B, C, D, E, F, and G but A's replacement set was still only $\{B, C, D\}$, then the reciprocity would be 0.5 (only 3 of the tokens that had A in their replacement set were also in the replacement set of A). A higher value indicates that it would be trivial for an attacker to rebuild the original replacement sets given secured embeddings or if they train a model only with access to the ciphertext.

Table 4.3.2 presents the results of this analysis for all four implementations of RaNNA for various replacement set sizes. We can see that the reciprocity tends to be a stable metric across both implementation variation as well as replacement set size (i.e., obfuscation parameter). Within the variation that exists, we observe that increasing the obfuscation parameter increases the security of released embeddings by reducing the value observed. This is observed across all implementations. Comparing across implementations, we can see that implementation 1 has substantially lower reciprocity values than the other three implementations. However, all values are quite low, thus presenting little risk of a straightforward approach to uncovering the original replacement sets from released embeddings.

Obfuscation Parameter	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	0	0.10	1.0	0	0	0	0	0.43
5	0	0.09	1.0	0	0	0	0.0	0.33
7	0	0.08	1.0	0	0	0	0.08	0.31
9	0	0.08	1.0	0	0	0	0.1	0.29
3-14	0	0.05	1.0	0	0	0	0.05	0.2

(a) Reciprocity of secured embeddings for various replacement set sizes and implementation #1 of RaNNA.

Obfuscation Parameter	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	0	0.16	1.0	0	0	0	0.29	0.60
5	0	0.16	1.0	0	0	0	0.33	0.50
7	0	0.17	1.0	0	0	0	0.33	0.50
9	0	0.17	1.0	0	0	0	0.32	0.50
3-14	0	0.11	1.0	0	0	0	0.20	0.33

(b) Reciprocity of secured embeddings for various replacement set sizes and implementation #2 of RaNNA.

Obfuscation Parameter	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	0	0.16	1.0	0	0	0	0.33	0.60
5	0	0.17	1.0	0	0	0	0.33	0.50
7	0	0.17	1.0	0	0	0	0.33	0.50
9	0	0.17	1.0	0	0	0	0.33	0.50
3-14	0	0.11	1.0	0	0	0	0.20	0.33

(c) Reciprocity of secured embeddings for various replacement set sizes and implementation #3 of RaNNA.

Obfuscation Parameter	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	0	0.17	1.0	0	0	0	0.33	0.6
5	0	0.17	1.0	0	0	0	0.33	0.5
7	0	0.17	1.0	0	0	0	0.33	0.5
9	0	0.18	1.0	0	0	0	0.33	0.5
3-14	0	0.12	1.0	0	0	0	0.21	0.35

(d) Reciprocity of secured embeddings for various replacement set sizes and implementation #4 of RaNNA.

Table 4.3.2: Analysis of the replacement mechanism (Reciprocity).

4.3.3 Percent overlap of replacement sets

The final measure I propose to capture the ease of reconstructing the replacement sets from released notes (or embeddings trained on secured notes) is the percent overlap of replacement sets, illustrated in Figure 4.3.2. For this measure, we measure what percentage of its nearest neighbours when trained on the secured notes are among its nearest neighbours when trained on the original notes used to perform RaNNA. For example, if the token A has the replacement set $\{B, C, D, E\}$ when trained on the original data and the replacement set $\{B, E, F, G\}$ when trained on the secured data, then the percent overlap would be 0.5 (since both B and E are shared between both. A higher percent overlap indicates that it is simple to recreate the original replacement sets from releasing secured word embeddings.

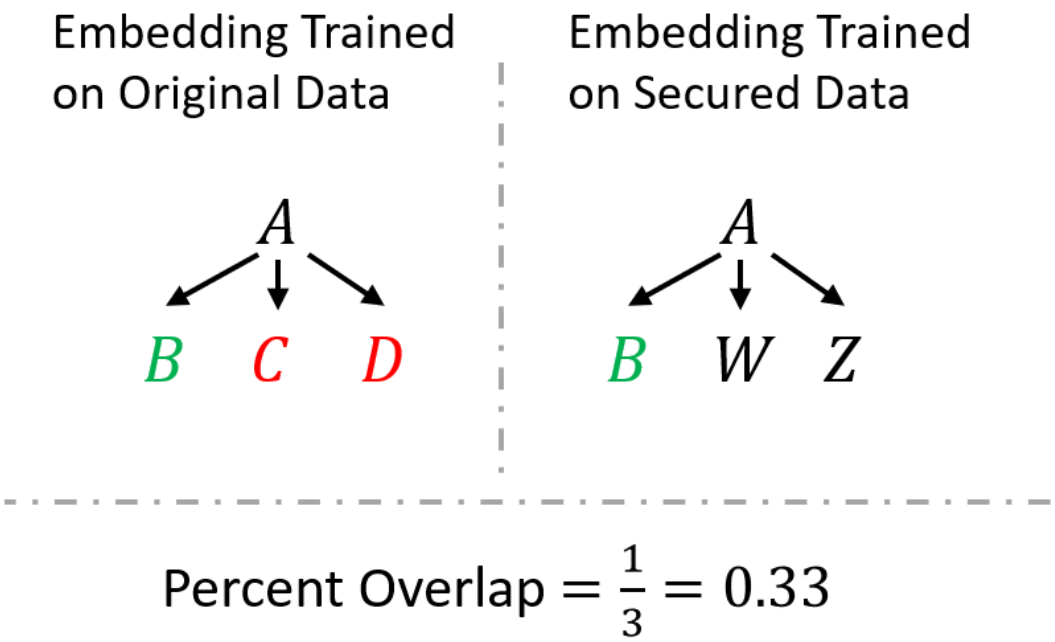

Figure 4.3.2: Illustration of the percent overlap of token A. The greater the overlap between the nearest neighbours of a model trained on the original text and text secured using RaNNA, the higher the percent overlap.

Table 4.3.3 presents the results of this analysis for all four implementations of RaNNA for various replacement set sizes. We can see that the reciprocity tends to be a stable metric across the last three implementations of RaNNA. Increasing the obfuscation parameter increases the security of released embeddings by reducing the percent overlap observed. This is observed across all implementations. Comparing across implementations, we can see that implementation 1 is has substantially lower percent overlap. However, like before, all values are quite low thus presenting little risk of a straightforward approach to uncovering the original replacement sets from released embeddings.

Obfuscation Parameter	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	0	0.14	1.0	0.0	0.0	0.0	0.33	0.67
5	0	0.12	1.0	0.0	0.0	0.0	0.20	0.40
7	0	0.11	1.0	0.0	0.0	0.0	0.14	0.43
9	0	0.10	1.0	0.0	0.0	0.0	0.22	0.33
3-14	0	0.06	0.75	0.0	0.0	0.0	0.08	0.25

(a) Percent overlap of the replacement set for tokens trained on original and secured embeddings for various replacement set sizes using implementation #1 of RaNNA.

Obfuscation Parameter	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	0	0.22	1.0	0	0	0	0.33	0.67
5	0	0.22	1.0	0	0	0	0.40	0.60
7	0	0.22	1.00	0	0	0.14	0.43	0.57
9	0	0.23	1.00	0	0	0.11	0.44	0.67
3-14	0	0.16	0.92	0	0	0.08	0.25	0.42

(b) Percent overlap of the replacement set for tokens trained on original and secured embeddings for various replacement set sizes using implementation #2 of RaNNA.

Obfuscation Parameter	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	0	0.22	1.00	0	0	0	0.33	0.67
5	0	0.22	1.00	0	0	0	0.40	0.60
7	0	0.23	1.00	0	0	0.14	0.43	0.57
9	0	0.23	1.00	0	0	0.11	0.44	0.67
3-14	0	0.16	0.92	0	0	0.08	0.25	0.42

(c) Percent overlap of the replacement set for tokens trained on original and secured embeddings for various replacement set sizes using implementation #3 of RaNNA.

Obfuscation Parameter	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	0	0.22	1.00	0	0	0	0.33	0.67
5	0	0.22	1.00	0	0	0	0.40	0.60
7	0	0.23	1.00	0	0	0.14	0.43	0.57
9	0	0.23	1.00	0	0	0.11	0.44	0.67
3-14	0	0.17	0.92	0	0	0.08	0.33	0.42

(d) Percent overlap of the replacement set for tokens trained on original and secured embeddings for various replacement set sizes using implementation #4 of RaNNA.

Table 4.3.3: Analysis of the replacement mechanism (Percent Overlap).

4.3.4 Effect of frequency on measures

To gain further insight to the risk of reconstructing the replacement set associated with obfuscation parameter, I present an analysis of the three previously introduced measures now stratified by the frequency of the tokens. More specifically, for the fourth implementation of RaNNA and an obfuscation parameter of 3–14 we present the LCC (Table 4.3.4), reciprocity score (Table 4.3.5), and percent overlap (Table 4.3.6). Each row in all tables is a bucket representing 20% of the tokens assigned by their frequency percentile.

Percentile Bucket	Min	Mean	Max	Percentile				
				5	25	50	75	95
[0-20]	0.08	0.22	0.95	0.09	0.13	0.17	0.27	0.40
(20-40]	0.08	0.25	0.95	0.10	0.15	0.21	0.31	0.44
(40-60]	0.08	0.28	0.92	0.12	0.18	0.25	0.35	0.47
(60-80]	0.08	0.31	0.90	0.14	0.21	0.28	0.38	0.51
(80-100]	0.08	0.30	0.92	0.13	0.21	0.27	0.37	0.47

Table 4.3.4: Local clustering coefficient for implementation #4 and replacement set size 3–14 for various buckets of token frequency.

Percentile Bucket	Min	Mean	Max	Percentile				
				5	25	50	75	95
[0-20]	0	0.04	1.0	0	0	0	0	0.17
(20-40]	0	0.07	1.0	0	0	0	0.08	0.29
(40-60]	0	0.11	1.0	0	0	0	0.2	0.33
(60-80]	0	0.16	1.0	0	0	0.14	0.25	0.36
(80-100]	0	0.21	1.0	0	0.07	0.19	0.31	0.43

Table 4.3.5: Analysis of the replacement mechanism stratified by token frequency. Reciprocity of secured embeddings for implementation #4 and replacement set size 3–14 for various buckets of token frequency.

We can see that as token frequency increases so does the LCC. This means that more frequent tokens are at a larger relative risk of having their replacement sets re-constructed when compared to infrequent tokens. We observe a similar trend, with respect to risk for both the reciprocity measure and the percent overlap. This trend is most substantial for the evaluation of percent overlap where the percent overlap of the replacement set for the most frequent tokens is over 5 times that of the least frequent tokens (0.05 vs 0.28). As individual name tokens are unlikely to be the most frequent tokens in clinical notes, this analysis indicates that they would not be at the most risk compared to other tokens.

Percentile Bucket	Min	Mean	Max	Percentile				
				5	25	50	75	95
[0-20]	0	0.05	0.92	0	0	0	0	0.25
(20-40]	0	0.10	0.92	0	0	0	0.17	0.33
(40-60]	0	0.17	0.92	0	0	0.08	0.25	0.42
(60-80]	0	0.23	0.92	0	0.08	0.25	0.33	0.50
(80-100]	0	0.28	0.92	0	0.08	0.25	0.42	0.58

Table 4.3.6: Analysis of the replacement mechanism stratified by token frequency. Percent overlap of the replacement set for tokens trained on original and secured embeddings for implementation #4 and replacement set size 3–14 for various buckets of token frequency.

4.3.5 Effect of part of speech tags

Next, I recalculate the three previously introduced measures now stratified by the part of speech (POS) tag of the tokens. We present the LCC (Table 4.3.7), reciprocity score (Table 4.3.8), and percent overlap (Table 4.3.9) for the fourth implementation of RaNNA and an obfuscation parameter of 3–14.

Using the ScispaCy (Neumann et al., 2019) — a Python package with tailored spaCy models for processing biomedical, scientific and clinical text — I assigned a POS tag for each token[5]. The results for the four most frequent POS tags (nouns, pronouns, adjectives and verbs) are presented below. We can observe no substantial difference between any of the three measures when grouped by part of speech tags.

POS Tag	Min	Mean	Max	Percentile				
				5	25	50	75	95
NOUN	0.08	0.26	0.95	0.11	0.17	0.24	0.33	0.44
PROPN	0.08	0.29	0.95	0.10	0.17	0.26	0.37	0.49
ADJ	0.08	0.27	0.90	0.11	0.17	0.24	0.34	0.46
VERB	0.08	0.25	0.90	0.10	0.16	0.22	0.31	0.43

Table 4.3.7: Local clustering coefficient for implementation #4 and replacement set size 3–14 for various buckets of token frequency.

4.3.6 Re-identifying from an embedding given the replacement sets

In this subsection, I answer the question: "Given all replacement sets, what is the risk of correctly (re-)identifying a patient?" using the embedding.

[5]It is true that a single token can be correctly assigned multiple POS tags depending on the sense and the context. However, this analysis does not take this complexity into consideration as the underlying methodology used (that of traditional word embeddings) does not deal with this complexity.

POS Tag	Min	Mean	Max	Percentile				
				5	25	50	75	95
NOUN	0	0.12	1.0	0	0	0	0.22	0.35
PROPN	0	0.11	1.0	0	0	0	0.20	0.35
ADJ	0	0.12	1.0	0	0	0	0.22	0.36
VERB	0	0.11	1.0	0	0	0	0.20	0.33

Table 4.3.8: Analysis of the replacement mechanism stratified by part of speech tags. Reciprocity of secured embeddings for implementation #4 and replacement set size 3–14 for various buckets of part of speech tags.

POS Tag	Min	Mean	Max	Percentile				
				5	25	50	75	95
NOUN	0	0.17	0.92	0	0	0.08	0.33	0.42
PROPN	0	0.17	0.92	0	0	0.08	0.33	0.50
ADJ	0	0.17	0.92	0	0	0.08	0.33	0.42
VERB	0	0.16	0.92	0	0	0.08	0.25	0.42

Table 4.3.9: Percent overlap for implementation #4 and replacement set size 3–14 for various part of speech tags.

For this analysis, I will start by making some simplifying assumptions and considerations:

- I assume that the attacker has access to the replacement set for all tokens in the original notes. This is an upper-bound/worst-case scenario assumption so the estimated risks will likely be an upper-bound on truly observed risk on a dataset found in-the-wild.
- I consider successfully re-identifying a name to be a successful breach. In reality, as most names are shared by multiple people, simply identifying the name (alone) might not be enough for re-identification.
- I assume that the data will be publicly released.
- Only RaNNA is applied on this dataset. No other security mechanism is used.

Given these assumptions, the maximal risk associated with correctly identifying a token (e.g., a name) is as follows:

$$Pr(t_i \mid t_j) = \frac{1}{|\{t_k : t_k \in s_j\}|} \tag{4.3}$$

That is, the probability that a replacement token t_j has replaced an original token t_i is one over the number of all other tokens that have t_j in their replacement set. To simplify

the math of later estimations, we consider the risk of the dataset as the maximal risk of any token. This risk is calculated by finding the replacement token t_j that has the least number of original tokens that could have been replaced by it, Equation 4.4.

$$max\_risk = \frac{1}{min(|\{t_k : t_k \in s_j\}|)} \tag{4.4}$$

Table 4.3.10 performs this calculation for each implementation of RaNNA for varying replacement set sizes.

For consistency with other sections, I also present a stratification of the results for Table 4.3.10 by token frequency (Table 4.3.11) and part of speech tag (Table 4.3.12).

As expected, increasing the size of the replacement set increases the mean and median measurements across implementations. We also see that there is a correlation with token frequency and the associated score (the more frequent the token, the more original tokens it can stand in for). At the same time, we can observe that regardless of the implementation choice, replacement set size, token frequency, or POS tag the minimum value is always 1.

Having a consistent minimum value of 1 is undesirable because, practically, it means that there is at least one token in the secured text that stands in only for one other token. If, in the worst-case this token belonged to a name token, and the attacker had access to the replacement sets (as we assumed in the start of this subsection) then it would be trivial for them to violate the of multiple patients who share that name.

To better understand when this worst-case happens we performed further analysis. We observe that, in our data, 13% of all replacement words have this worst case. However, at the same time only 10% of the original tokens had replacements that were not shared by any other original tokens. This means for a few tokens they had multiple replacements that were unique to them. How often did these original tokens appear? On average these tokens appeared 4497 times in the entire corpora, with a median occurrence of 14 times. This means that the vast majority of these words with unique replacements are rare words, helping explain the trends observed in Table 4.3.11.

4.3.7 Methodological fix

To remedy this issue, we propose an approach with two implementations. Currently, the replacement set of a token t_i is chosen to be the closest N neighbours in the embedding space. I now change the composition of the replacement set to be the closest N tokens in the embedding space that are also in the replacement set of at least k other tokens.

As an example, consider looking at the replacement set of t_i, $N = 3$ and $k = 10$. Let r_i represent the list of tokens in the embedding space ordered by their closeness to t_i such

Obfuscation Parameter	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	1	2.8	134	1	1	2	3	4
5	1	2.15	93	1	1	2	3	4
7	1	2.14	74	1	1	2	3	4
9	1	2.12	62	1	1	2	3	4
3-14	1	2.06	32	1	1	2	3	4

(a) Counting the number of tokens that could have been replaced by a single token from the secured notes. This table presents these counts for varying replacement set sizes for implementation #1.

Obfuscation Parameter	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	1	4.31	311	1	1	3	6	9
5	1	6.01	320	1	2	4	8	14
7	1	7.43	285	1	2	5	10	17
9	1	8.71	265	1	2	6	12	20
3-14	1	10.24	153	1	3	7	14	24

(b) Counting the number of tokens that could have been replaced by a single token from the secured notes. This table presents these counts for varying replacement set sizes for implementation #2.

Obfuscation Parameter	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	1	4.32	335	1	1	3	6	9
5	1	6.04	312	1	2	4	8	14
7	1	7.49	316	1	2	5	10	17
9	1	8.76	293	1	2	6	12	20
3-14	1	10.36	165	1	3	7	14	24

(c) Counting the number of tokens that could have been replaced by a single token from the secured notes. This table presents these counts for varying replacement set sizes for implementation #3.

Obfuscation Parameter	Min	Mean	Max	Percentile				
				5	25	50	75	95
3	1	4.35	331	1	1	3	6	9
5	1	6.14	349	1	2	4	8	14
7	1	7.63	334	1	2	5	10	18
9	1	8.99	310	1	2	6	12	21
3-14	1	10.67	185	1	3	7	15	25

(d) Counting the number of tokens that could have been replaced by a single token from the secured notes. This table presents these counts for varying replacement set sizes for implementation #4.

Table 4.3.10

Percentile Bucket	Min	Mean	Max	Percentile				
				5	25	50	75	95
[0-20]	1	2.84	185	1	1	2	3	5
(20-40]	1	8.94	111	1	3	6	11	20
(40-60]	1	14.97	100	2	6	11	20	32
(60-80]	1	15.49	90	2	7	13	21	31
(80-100]	1	11.12	79	1	4	9	15	23

Table 4.3.11: Counting the number of tokens that could have been replaced by a single token from the secured notes. This table presents these counts for the replacement set size of 3–14 for implementation #4 stratified by frequency of token.

POS tag	Min	Mean	Max	Percentile				
				5	25	50	75	95
NOUN	1	10.85	102	1	3	7	15	25
PROPN	1	10.67	185	1	2	6	15	26
ADJ	1	10.57	93	1	3	7	15	24
VERB	1	11.05	80	1	3	8	15	25

Table 4.3.12: Counting the number of tokens that could have been replaced by a single token from the secured notes. This table presents these counts for the replacement set size of 3–14 for implementation #4 bucketed by part of speech tags.

that r_1, r_2, r_3 are the three closest tokens to t_i. Per the original implementation of RaNNA, $\{r_1, r_2, r_3\}$ would be final replacement set. However, in the new implementation (i.e., the methodological fix) we seek to ensure that each replacement token r_i is in the replacement set of at least 10 other tokens. We use a greedy algorithm to achieve this token–replacement pairing:

1. Create temporary replacement sets as defined in the original implementation. In our example the replacement set for t_i will be $\{r_1, r_2, r_3\}$.

2. For each replacement token r, calculate the number of original tokens t that have r in their replacement set. If all replacement tokens occur in the replacement sets of at least k original tokens, then break.

3. For each original token t, replace all replacement tokens r that do not occur in the replacement set of at least k original tokens with the next available replacement tokens (without caring how many tokens t have the new replacement tokens in their replacement sets). For example, if for t_i, r_1 and r_2 were not in the replacement set of at least 10 other tokens then they cannot be in the final replacement set, and at this stage r_1 and r_2 would be replaced by r_4 and r_5. We do not care at this stage if either

r_4 or r_5 are in the replacement sets of 10 tokens. This will be re-calculated after this change is propagated for all tokens.

4. Go back to Step 2.

When following the proposed changes to RaNNA, we can run it until completion or we can define an arbitrary threshold after which no more replacement is done (to ensure that the selected replacement tokens are still somewhat close to the original token). For example, continuing the above example, if we define the threshold as 40 and imagine of none of the replacement tokens from r_4 to r_{40} satisfied our replacement condition then the final replacement set would be $\{r_3\}$. By stating that we can only choose the closest three replacement tokens within the forty closest tokens that fulfill our condition, it is possible that we will not have enough replacement tokens for each token. Using an arbitrary limit (of 40) is the first implementation of the methodological fix. The max_risk recalculated using this implementation is presented in Table 4.3.13.

Obfuscation Parameter	k **(smallest set size)**	**Min**	**Mean**	**Max**	**Percentile**				
					5	**25**	**50**	**75**	**95**
3	3 (3)	1	7.31	363	3	4	6	9	13
5	5 (5)	2	12.03	372	5	7	10	15	21
7	7 (3)	1	16.18	387	7	10	14	20	28
9	9 (4)	3	19.93	374	9	12	17	24	34
3-14	14 (2)	7	28.65	232	14	19	25	34	46

Table 4.3.13: Counting the number of tokens that could have been replaced by a single token from the secured notes. This table presents these counts for multiple replacement set sizes for the first methodological fix.

The second implementation does not institute any limit on where the replacement tokens have to be sampled from. With this implementation the replacement set of t_i can be quite far from the original token (e.g., $\{r_3, r_{45}, r_{110}\}$). The max_risk recalculated using this implementation is presented in Table 4.3.14, and we observe that the minimum number is better in this implementation, though generally there is not much difference between the two implementations on this dataset.

4.4 Risk assessment: Releasing clinical notes

In the remainder of this chapter, we will explore the risk of re-identification stemming from publicly releasing clinical notes that have been secured using RaNNA. The following experiment will assume that only secured context notes are released and not the original

Obfuscation Parameter	k (smallest set size)	Min	Mean	Max	Percentile				
					5	25	50	75	95
3	3 (3)	1	7.30	356	3	4	6	9	13
5	5 (5)	2	12.04	380	5	7	10	15	21
7	7 (7)	3	16.17	371	7	10	14	20	28
9	9 (9)	3	19.92	349	9	12	17	24	34
3-14	14 (12)	7	28.71	233	14	19	25	34	46

Table 4.3.14: Counting the number of tokens that could have been replaced by a single token from the secured notes. This table presents these counts for multiple replacement set sizes for the second methodological fix.

embedding. First, I will quantify the answers to some questions regarding the risks of releasing the notes in different situations. Second, I will re-derive and then extend the risk-scores following the framework of Scaiano et al. (2016) to arrive at a probabilistic understanding of the risk associated with releasing notes secured using RaNNA (both with and without an additional traditional search-and-replace approach as suggested by Abdalla et al. (2020b)).

4.4.1 Attacking a Replacement Token

Unlike simply releasing the traditional word embeddings, the release of a set of clinical notes provides contextual information to bad actors. Consider a hypothetical dataset that contains note numbers, patient ID for each note, and the text of each note:

NOTE:1, PATIENT ID: 1, TEXT: "*Title: John ...*
NOTE:2, PATIENT ID: 1, TEXT: "*Names: Adam ...*
NOTE:3, PATIENT ID: 1, TEXT: "*Named: Smith ...*

In this situation, if the text is secured using RaNNA alone, the attacker can likely guess that *John*, *Adam*, and *Smith* are all replacement tokens for an unknown original token t_i. With this information, and the entire set of notes: can the attacker identify the original token t_i?

If the attacker had access to the original embeddings, discovering t_i would be trivial; they just need to look for any token that had all of the observed replacement tokens (i.e., *John*, *Adam*, and *Smith*) in their replacement set. Without the original embedding, the attacker must train an embedding model on the released (and secured) data. We will assume that the attacker knows the hyperparameters used to train the original model to provide an upper-bound (i.e., worst-case) prediction of the associated risk. For discussion on and

quantification of the differences between the actual embeddings and embeddings trained on secured notes please refer to the previous section and the various measures that were calculated.

After training an embedding model on the secured released data (termed: "secured embeddings") using the correct hyperparameters, we ask the following question:

> **What is the likelihood of picking the correct token t_i by choosing the token that occurs in the plurality of nearest neighbours of the known replacement words?**

Assume that the data is highly structured (e.g., the quote in subsection 4.4.1). Such a release is not uncommon (e.g., MIMIC-III (Johnson et al., 2016)), and is required if researchers are to perform tasks for patients.[6] From the example found above, given that the text of the notes all belong to the same patient ID, we know that *John*, *Adam*, and *Smith* are all likely replacement tokens for an unknown original token t_i. For an attacker it would make sense to look at the nearest neighbours of all of these words in an attempt to identify t_i. In the previous section, we calculated the overlap between embeddings trained on the original text and embeddings trained on the same text secured using RaNNA. If we assume that reconstructing the datasets approximates the original replacement datasets (and it does to some degree), it would make sense for the attacker to guess t_i to be the token that occurs in the plurality of the nearest neighbours of the replacement tokens for the unknown original token t_i.

Now, I follow this plan of attack: I calculate the probability that choosing the most common token from a set of grouped replacement sets results in selecting the correct original token t_i. On our dataset, averaging over all tokens, I found that such an attack plan will result in the attacker guessing the correct word only 4.82% of the time.

4.4.2 Attacking Released Clinical Notes

Now, we would like to compare the risk associated with publicly sharing clinical notes secured using RaNNA against releasing notes secured using only search-based approaches or using the combination of both search-based approaches and RaNNA. To perform this comparison, we will introduce Scaiano et al.'s (2016) framework for the probabilistic risk assessment of de-identification methods. We will then extend this framework to incorporate RaNNA to enable data holders to directly compare the risks of these approaches.

[6] However, if information identifying which note comes from which patient is publicly released it likely makes more sense to use Implementation 3 rather than Implementation 4 of RaNNA.

First, we will introduce estimates for the various required parameters to assess the re-identification risk:

- h is the "hiding in plain sight" factor. This is loosely translated to, given a search-and-replace approach, what is the probability that an attacker can recognize when a token has been missed. Per the literature, we set $h = 0.1$ (Carrell et al., 2013). Other work has predicted a more optimistic value of $h = 0$ through experiments (Meystre et al., 2014), but we will take the more conservative value to increase the possible risk.

- w_i is the probability that a direct identifier i appears in a note and is measured as the number of notes i appears in as a proportion of all notes. This is dependent on the identifier and the dataset. If we simplify our situation, claiming only names to be direct identifiers and our defined hypothetical dataset above, we will assume w_i to be $\frac{15}{1500} = 0.01$.

- r_i is the all-or-nothing recall of the search-based algorithm for the specific identifier i. This value is dependent on the identifier, dataset, and the classifier used. As stated earlier, we will over-estimate the performance of search-based approaches (relative to RaNNA) by assuming that the reported micro-average recall is equal to their all-or-nothing recall. A value of 0.98 is prevalent within the literature (Dernoncourt et al., 2017; Scaiano et al., 2016), though this is likely an optimistic estimate. Past work has shown a drop in recall of 20 – 30 absolute points for names when training on one dataset and testing on another (Yue and Zhou, 2020). Therefore, we will test values of r_i at 0.98 as well as 0.90 and 0.80 to estimate the performance of generalization.

- r_q is the macro-average recall for all indirect identifiers, and like r_i, it is dependent on the identifier, dataset, and the classifier used. In the literature, the recall for indirect identifiers tends to be lower than the recall for direct identifiers (Dernoncourt et al., 2017; Scaiano et al., 2016). For this work, we will test values of 0.95, 0.90, and 0.80.

- m is the average number of times that a specific indirect identifier appears in a single document. Again, this is solely dependent on the dataset and can vary greatly. For our hypothetical dataset, we will set m to 2.

- n_q is the average number of unique indirect identifiers in a document (or clinical note). This value is solely dependent on the dataset and can vary greatly depending on the type of note being de-identified. For example, for radiology notes this value

will be around 0, whereas for consultation notes it will be much higher. For our hypothetical dataset we will set n_q to 3.

- c_i represents the probability that the attacker will be able to reconstruct the replacement set for an original token t_i that they are hoping to attack. This value is dependent on the dataset. Highly structured and repetitive datasets will result in very easy reconstruction, whereas very unstructured and free-flowing texts will make the reconstruction very difficult. For this work, we are assuming that the attacker will be able to successfully construct the datasets for 70% of all words. However, through manual experimentation on small texts, it appears that this number will likely be lower.

- s_i represents the probability of uncovering the replaced token t_i, given a correctly re-constructed replacement set for t_i. If we assume that the attacker knows the hyper-parameters of the embedding used by RaNNA, then this risk is quantitatively calculated to be 0.05. This value is highly dependent both on the dataset, the hyper-parameters of the algorithm, and chance. We are also simulating the worst-case scenario (in favour of the attacker) by assuming they know the hyper-parameters and the replacement set size, which increases this probability.

Risk Assessment: Search-based Approaches If we apply a search-based algorithm to a note there are two possibilities for any PII in the note. Either the algorithm performs correctly and *catches* all PII, or it has a false negative and *leaks* at least one PII. Thus, the risk of re-identification can be represented as follows:[7]

$$P(\text{reid}) = P(\text{reid}, \text{catch}) + P(\text{reid}, \text{leak}) \tag{4.5}$$

Catch

If the algorithm correctly classifies the PII with some probability, which I represent as $P(\text{catch})$, then it is not possible for attackers to re-identify the specific token that has been caught. However, if another de-identification method was used (e.g., RaNNA) it may be theoretically possible to use information around in the note to still re-identify the PII. We express this probability using $P(\text{reid})$. The probability of re-identification given an

[7]Note: $P_i(\text{leak})$ represents the probability that a specific identifier i is leaked. On the other hand, $P(\text{leak})$ represents the probability that any PII is leaked and will later be represented as the product of the probability of leakage for all PII in a note. This notation — P_i for probability relating to a specific identifier vs P for probability on the dataset level — will be used throughout this subsection.

algorithm catching PII can be represented as follows.

$$P(\text{reid}, \text{catch}) = P(\text{reid} \mid \text{catch}) \cdot P(\text{catch}) \tag{4.6}$$

For traditional search-based approaches, $P(\text{reid} \mid \text{catch})$ is assumed to be 0 (Scaiano et al., 2016). That is, if an instance of PII is successfully caught by the algorithm, we assume that attackers will not be able to attack that specific PII. Thus, it is also assumed that there is no risk to releasing a dataset where all PII has been caught.

Leak

However, if the algorithm missed an instance of PII, probability represented using $P(\text{leak})$, then it can be used to re-identify a patient. The probability of re-identification and a leak happening is dependent on the probability of a leak occurring and the probability re-identification given a leak.

$$P(\text{reid}, \text{leak}) = P(\text{reid} \mid \text{leak}) \cdot P(\text{leak}) \tag{4.7}$$

The probability of a leak occurring for a specific token i, by definition, is the opposite of recall r (i.e., the false negative rate):

$$P_i(\text{leak}) = 1 - r \tag{4.8}$$

We will assume the worst-case scenario: complete re-identification ability such that if there is any leak, it will be trivial for an attacker to re-identify the patient:

$$P(\text{reid} \mid \text{leak}) = 1 \tag{4.9}$$

In truth, this is an over-simplification. For example, knowing that a certain note belongs to a John does not mean that the attacker knows exactly which John the note belongs to. The more PII is leaked, the higher the risk of re-identification. However, to simplify the math, and to place an upper-bound on the risk (i.e., worst-case scenario), we follow this assumption from past work (Scaiano et al., 2016).

Incorporating the combined risks discussed above, we can frame the total risk of a search-based de-identification algorithm as follows:

$$P(\text{reid}, \text{catch}) + P(\text{reid}, \text{leak}) = P(\text{reid} \mid \text{leak}) \cdot P(\text{leak}) \tag{4.10}$$

Since analysis of traditional search-based approaches assume $P(\text{reid} \mid \text{catch})$ to be 0, the risk of re-identification for search-based approaches is wholly dependent on the risk of

leakage. We will explore the risk of leaking direct identifier (i.e., PII that alone can re-identify an individual) and indirect identifiers (i.e., PII that alone can't re-identify an individual). separately.

Direct identifiers

The probability that a direct identifier is leaked AND is in the corpus:

$$P(\text{leak}, \text{appears}) = P(\text{leak} \mid \text{appears}) \cdot P(\text{appears}) \tag{4.11}$$

We define w_i as the probability that a direct identifier i is in a note, d_i as the number of notes that i appears in and n as the total number of notes.

$$w_i = \frac{d_i}{n} \tag{4.12}$$

The probability of a leak occurring is related to the recall of the model. Scaiano et al. (2016) discussed the concept of "all-or-nothing recall" in opposition to the widely used "micro-average recall". Imagine that there is a dataset composed of 10 notes, each belonging to a different patient. In each note, the patient's name occurs 10 times. If the search-based algorithm performed perfectly on nine of the ten notes and had a recall of 0 on the last note, the micro-average recall would be 0.90. At the same time, if the algorithm missed a single name for each of the ten notes but caught all other nine the micro-average recall would still be 0.90 despite a much greater risk to many more patients. This is because the micro-average recall does not take into account the distribution of PII. To address this, the "all-or-nothing" recall is calculated at the note level. Assuming the same dataset, if the search-based algorithm performed perfectly on nine of the ten notes and had a recall of 0 on the last note, the micro-average recall would be 0.90. However, if the algorithm missed a single name for each of the ten notes but caught all other nine the all-or-nothing recall would be 0.

The difference between micro-average recall and all-or-nothing recall for direct identifiers such as names can be as little 0–3% (El Emam and Arbuckle, 2013) or as high as 43% (Scaiano et al., 2016). For the remainder of this work, to provide an upper-bound on the risk (i.e., examine the worst-case scenario), we will assume that the all-or-nothing recall matches the micro-average recall (and refer to it using only the term recall). Doing so will over-estimate the capability of search-based approaches in our comparison with RaNNA.

With the above assumption, we define r_i as the all-or-nothing recall of the search-based model when evaluated for identifier i. We can then rewrite the combined probability that a

single direct identifier i appears and is leaked:

$$P_i(\text{leak}, \text{appears}) = w_i(1 - r_i) \tag{4.13}$$

The probability that an instance of PII i is not leaked is $1 - P_i(\text{leak}, \text{appears})$. Considering the entire set of all PII, the probability that at least one direct identifier will be leaked can then be represented using:

$$1 - \prod_i (1 - w_i(1 - r_i)) \tag{4.14}$$

Indirect identifiers

A single indirect identifier (e.g., occupation or age) is not enough to uniquely re-identify an individual. Let m be the average number of times that a specific indirect identifier is repeated in a document (i.e., the average number of instances per indirect identifier value in a document). It is important to note that the value m is static for all indirect identifiers. That is, if the average number of times that an indirect identifier appears is twice, then for the risk calculation we will assume all direct identifiers appear twice (which is required to allow us to use the binomial distribution). Let r_q be the micro-average recall computed across all indirect identifiers. Then the probability that at least one instance of the indirect identifier is missed would be:

$$1 - {r_q}^m \tag{4.15}$$

Let n_q be the average number of unique indirect identifiers in a text. Assume that just two identifiers are required to re-identify an individual. The probability that two indirect identifiers are leaked given n_q indirect identifiers in a note can be calculated using a binomial distribution $B(a, b)$ with a trials and b probability of success:

$$P(X \geq 2) \text{ for } X \sim B(n_q, 1 - {r_q}^m) \tag{4.16}$$

Re-synthesis

Equations 4.14 and 4.16 do not account for the difficulty of re-identification for a search-and-replace approach. When search-and-replace is used, simply leaking the token is not enough for re-identification. In this case, the attacker must also be able to detect the fact that the token has been leaked for re-identification to happen, represented using $P(\text{recognize})$; unlike before we no longer assume that there is perfect re-identification after a leak. Incorporating this into our analysis:

$$P(\text{recognize}, \text{leak}, \text{appear}) = P(\text{recognize} \mid \text{leak}, \text{appear}) \cdot P(\text{leak} \mid \text{appear}) \cdot P(\text{appear}) \tag{4.17}$$

We will represent $P(\text{recognize})$ using h (i.e., the probability that a token is recognized by an attacker to have been missed by the algorithm). For example, if there are two names *John* and *Avneet* in a note that discusses in great detail health issues revolving a recent celebration of Diwali, then an attacker is likely to guess that *Avneet* is the true name.[8] Incorporating this probability, we can then update the probability that at least one direct identifier appears and is leaked 4.14 and the probability that at least two indirect identifiers appear and are leaked 4.16:

$$1 - \prod_i (1 - h \cdot (w_i(1 - r_i))) \tag{4.18}$$

$$P(X \geq 2) \text{ for } X \sim B(n_q, h(1 - (r_q)^m)) \tag{4.19}$$

Scaiano et al. (2016) claim that the above adjustments are too optimistic. The adjustment of h should only be applied in cases where the recall r_i is above a certain threshold. They choose (somewhat arbitrarily) the threshold of 0.9 for direct identifiers and 0.7 for indirect identifiers. While the specific threshold is arbitrary, applying them increases the risk of re-identification. That is, if we had applied the adjustment for all examples the risk of re-identification would be lower. By requiring recall to meet a certain threshold before applying this adjustment the authors are calculating an upper-bound of the risk. The underlying motivation for creating a threshold is that in cases where recall is too low, it will be too trivial for an attacker to predict the correct name (i.e., h will no longer apply). Following this reasoning, if we restrict the performance in this way, then the probability that a single direct identifier appears, is leaked, and is detected by an attacker can be represented as follows:

$$1 - \prod_{i \,|\, r_i \geq 0.9} (1 - h \cdot (w_i(1 - r_i))) \prod_{i \,|\, r_i < 0.9} (1 - (w_i(1 - r_i))) \tag{4.20}$$

$$\begin{aligned} P(X \geq 2 \text{ if } r_i \geq 0.7, \text{ or } Y \geq 2 \text{ if } r_i < 0.7) \text{ for} \\ X \sim B(n_q, h(1 - (r_q)^m)), \\ Y \sim B(n_q, (1 - (r_q)^m)) \end{aligned} \tag{4.21}$$

Confidence Intervals

Thus far, we have been treating everything as a point estimate. However, measurements such as w_i and r_i are measured from samples of the data and are difficult to get for each identifier i. Scaiano et al. (2016) proposed sampling these values for each identifier i from

[8] This is only a guess based on probabilities observed in the real world. There is no reason that John could not indeed be the true name — it just less likely.

a normal distribution with mean r_i and variance $r_i(1-r_i)/d_i$ for recall values and a normal distribution with mean w_i and variance $w_i(1-w_i)/n$ for $P(\text{appears})$. There is no strong motivation for these specific distributions, and they may not represent the distributions observed for any specific dataset. However, as a framework, these distributions can easily be re-defined to better fit whatever data is being secured. For the purposes of this work, we will follow Scaiano et al. (2016). Updating the probability of re-identification given a leak, appearance, and a human recognition the leak for indirect and direct identifiers:

$$P(\text{reid}, \text{leak}, \text{appears}) = 1 - \prod_{i \,|\, r_i \geq 0.9} (1 - h \cdot (W_i(1-R_i))) \prod_{i \,|\, r_i < 0.9} (1 - (W_i(1-R_i)))$$

for

$$W_i \sim N(w_i, \sqrt{w_i(1-w_i)/n}), R_i \sim N(r_i, \sqrt{r_i(1-r_i)/d_i}) \tag{4.22}$$

We will also sample these variables from distributions for the risk assessment of indirect identifiers. The micro-average recall for all indirect identifiers is sampled from a normal distribution with mean r_q and variance $r_q(1-r_q)$. The number of unique indirect identifiers present in a note will be sampled from a Poisson distribution N_q with $\lambda = n_q$, and the number of occurrences for each is sampled from a Poisson distribution M with $\lambda = m$.

$$P(X \geq 2 \text{ if } r_q \geq 0.7, \text{ or } Y \geq 2 \text{ if } r_q < 0.7)$$

for

$$X \sim B(N_q, h(1-(R_q)^M)), Y \sim B(N_q, (1-(R_q)^M)) \tag{4.23}$$

where

$$R_q \sim N(r_q, \sqrt{r_q(1-r_q)/n}), N_q \sim \text{Pois}(n_q), M \sim \text{Pois}(m)$$

Risk Assessment: RaNNA To facilitate a direct comparison between RaNNA and search-based approaches, we will need to formulate the risk assessment of RaNNA in a similar fashion. This means evaluating both the risk for direct identifiers and indirect identifiers separately. The following section closely mirrors the structure and approach of Scaiano et al. (2016), but represents a novel expansion of and contribution to that framework. We start by following a similar framing of catching or leaking identifiers.

Unlike search-based approaches, RaNNA guarantees that each token will be replaced by another token. As such $P(\text{leak}) = 0$ and $P(\text{catch}) = 1$. This means that the risk of

re-identification can be wholly encapsulated by $P(\text{reid} \mid \text{catch})$.

Direct Identifiers

The probability that a direct identifier appears in a note, is caught by the algorithm, and is re-identified:

$$P(\text{reid}, \text{catch}, \text{appears}) = P(\text{reid} \mid \text{catch}, \text{appears}) \cdot P(\text{catch} \mid \text{appears}) \cdot P(\text{appears}) \quad (4.24)$$

The probability of re-identification is based largely on two factors. First, the ability of the attacker to reconstruct the replacement set of an unknown token, which we represent by $P(\text{construct})$, and their ability to select the right word from the nearest neighbours of the reconstructed replacement set, represented by $P(\text{select})$. We define s_i as the $P(\text{select})$ for identifier i and know from our past experiment, Section 4.4.1, that given the replacement set $P(\text{select}) = s_i = 0.05$. We will let c_i represent $P(\text{construct})$. From this, we can rewrite the combined probability that a direct identifier appears, is caught, and re-identified as:

$$P(\text{reid}) = P(\text{select} \mid \text{construct}) \cdot P(\text{construct}) \quad (4.25)$$

For a specific token i, this can be written as:

$$P_i(\text{reid}, \text{catch}, \text{appears}) = w_i \cdot c_i \cdot s_i \quad (4.26)$$

The probability that a direct identifier i is not leaked is $1 - P_i(\text{reid}, \text{catch}, \text{appears})$. Considering the entire dataset, the probability that at least one direct identifier will be leaked can then be represented as:

$$1 - \prod_i (1 - (w_i \cdot c_i \cdot s_i)) \quad (4.27)$$

To enable the calculation of confidence intervals, we again follow the approach of Scaiano et al. (2016) and redefine each of these variables as being sampled from normal distributions. For each identifier i, we will use a normal distribution with mean w_i and variance $w_i(1-w_i)/n$ for $P_i(\text{appears})$. For $P_i(\text{construct})$, we will sample from a normal distribution with mean c_i and variance $c_i(1-c_i)/n$. For $P_i(\text{select})$, we will sample from a normal distribution with mean s_i and variance $s_i(1-s_i)/d_i$. As mentioned above, these distributions are hypothetical and can be replaced with empirically determined distributions for the dataset being secured. Updating the above equations:

$$P(\text{reid, leak, appears}) = 1 - \prod_i (1 - (W_i \cdot C_i \cdot S_i))$$

for

$$\begin{aligned} W_i &\sim N(w_i, \sqrt{w_i(1-w_i)/n}), \\ C_i &\sim N(c_i, \sqrt{c_i(1-c_i)/n}), \\ S_i &\sim N(s_i, \sqrt{s_i(1-s_i)/d_i}) \end{aligned} \tag{4.28}$$

Indirect Identifiers

If m is the average number of times that a specific indirect identifier appears in a document, and $c_i \cdot s_i$ is the probability of reconstruction and correct selection of an identifier, then the probability that at least one instance of the indirect identifier is successfully re-identified would be:

$$1 - (1 - (c_i \cdot s_i))^m \tag{4.29}$$

Again, if we assume that just two indirect identifiers are required to re-identify an individual then the probability of a leak of two indirect identifiers can be calculated using the binomial distribution:

$$P(X \geq 2) \text{ for } X \sim B(n_q, 1 - (1 - (c_i \cdot s_i))^m) \tag{4.30}$$

To enable the calculation of confidence intervals, we will sample each of the variables from distributions defined previously for each of n_q, c_i, and s_i.

Risk Assessment: Search-based Approaches and RaNNA Now we will assess the risk associated with releasing notes that have both a search-based approach and RaNNA applied to them. When applying both of these methods there two possible options: (i) the identifier is caught by the search algorithm, in which case RaNNA does not provide any additional security, or (ii) the identifier is leaked, in which case RaNNA helps to obfuscate the token that was leaked.

Direct Identifiers

The probability that a direct identifier i appears, is leaked, and is detected as a leak is represented as:

$$h \cdot w_i \cdot (1 - r_i) \tag{4.31}$$

The probability that an identifier appears in the text and has its replacement set re-constructed, and the attacker selects the correct word from all the re-constructed replacement sets is:

$$w_i \cdot c_i \cdot s_i \tag{4.32}$$

Combining the two situations: the probability that an identifier i is present, has been leaked by the search algorithm, is identified as being leaked (Equation 4.18), has had its replacement set built, and the correct word selected from the replacement sets (Equation 4.26) is then:

$$h \cdot w_i \cdot (1 - r_i) \cdot c_i \cdot s_i \tag{4.33}$$

Incorporating all of these variables into the probability that at least one direct identifier will be re-identifiable can be represented using:

$$\begin{aligned} P(\text{reid}, \text{leak}, \text{appears}) \; &= 1 - \prod_i (1 - h \cdot W_i \cdot C_i \cdot S_i \cdot (1 - R_i)) \\ &\text{for} \\ &W_i \sim N(w_i, \sqrt{w_i(1 - w_i)/n}), \\ &C_i \sim N(c_i, \sqrt{c_i(1 - c_i)/n}), \\ &S_i \sim N(s_i, \sqrt{s_i(1 - s_i)/d_i}), \\ &R_i \sim N(r_i, \sqrt{r_i(1 - r_i)/d_i}) \end{aligned} \tag{4.34}$$

The distinction in calculated probability for $r_i \geq 0.9$ is dropped because the additional noise of RaNNA makes it no longer trivial to determine when an algorithm has leaked an identifier.

Indirect Identifiers

Similarly, for indirect identifiers, we can incorporate the effects of RaNNA on identifiers that are missed by the de-identification algorithm:

$$P(X \geq 2)$$

for

$$X \sim B(N_q, h \cdot C_i \cdot S_i \cdot (1 - (R_q)^M))$$

where

$$\begin{aligned} R_q &\sim N(r_q, \sqrt{r_q(1-r_q)/n}), \\ C_i &\sim N(c_i, \sqrt{c_i(1-c_i)/n}), \\ S_i &\sim N(s_i, \sqrt{s_i(1-s_i)/d_i}), \\ N_q &\sim \text{Pois}(n_q), \\ M &\sim \text{Pois}(m) \end{aligned} \tag{4.35}$$

Again, the distinction in calculated probability for $r_q \geq 0.7$ is dropped because the additional noise of RaNNA makes it no longer trivial to determine when an algorithm has leaked an identifier.

Calculating Risk Having formulated equations to estimate the risks associated with releasing clinical notes secured using search-based approaches, we can now use them for a quantitative analysis. For this analysis, we will be using a hypothetical dataset of 1500 notes, 100 patients, and 15 notes per patient.

Using the value estimates introduced at the beginning of Section 4.4.2 above, we can now directly compare the risks of sharing a dataset of clinical notes. Table 4.4.1 presents the risk of the re-identification of at least one direct identifier for the release of a set of clinical notes secured using various de-identification methods: (i) search-and-remove methods calculated using Equation 4.22 with $h = 0$, (ii) search-and-replace methods calculated using Equation 4.22 with $h = 0.1$, (iii) RaNNA calculated using Equation 4.28, and (iv) applying RaNNA after a search-and-replace approach calculated using Equation 4.34.

We observe that using search-and-replace method in tandem with RaNNA provides the most security for direct identifiers. The probability of any direct identifiers being leaked is orders of magnitude less than the risk of using search-and-replace or RaNNA alone. Using this combined approach reduces the risk even if the search-based classifier performs poorly (e.g., recall dropping to 0.80) — something most sites should be able to easily achieve. Comparing RaNNA with search-based methods alone, we can see that if the search-based

	Mean	Percentile 2.5	Percentile 97.5
Search and Remove			
r=0.98	2.62E-02	2.07E-02	3.19E-02
r=0.90	9.87E-02	8.50E-02	1.13E-01
r=0.80	1.82E-01	1.64E-01	2.01E-01
Search and Replace			
r=0.98	4.01E-03	2.13E-03	7.25E-03
r=0.90	7.99E-02	6.44E-02	9.62E-02
r=0.80	1.76E-01	1.56E-01	1.97E-01
RaNNA			
	3.25E-02	1.68E-02	4.98E-02
Search and Replace + RaNNA			
r=0.98	8.84E-05	6.80E-05	1.09E-04
r=0.90	3.48E-04	2.94E-04	4.02E-04
r=0.80	6.79E-04	5.99E-04	7.73E-04

Table 4.4.1: The risk of the re-identification of at least one direct identifier for the release of the hypothetical dataset described secured using various de-identification methods.

method has an all-or-nothing recall of 0.98, there is less risk than using RaNNA. However, if the all-or-nothing recall drops below that (e.g., to 0.90) then RaNNA is the more secure option to use.

Table 4.4.2 presents the risk of the re-identification of at least two indirect identifiers for the release of a set of clinical notes secured using various de-identification methods: (i) search-and-remove methods calculated using Equation 4.23 with $h = 0$, (ii) search-and-replace methods calculated using Equation 4.23 with $h = 0.1$, (iii) RaNNA calculated using Equation 4.30, and (iv) applying RaNNA after a search-and-replace approach calculated using Equation 4.35.

We observe trends similar to those observed for the risks to direct identifiers. As in past literature, the risk of re-identification from two indirect identifiers is lower than the risk of re-identification from a single direct identifier (Scaiano et al., 2016).

	Mean	Percentile 2.5	Percentile 97.5
Search and Remove			
r=0.98	4.20E-02	0.00E+00	2.65E-01
r=0.90	1.24E-01	0.00E+00	6.52E-01
r=0.80	2.70E-01	0.00E+00	9.20E-01
Search and Replace			
r=0.98	5.55E-04	0.00E+00	4.00E-03
r=0.90	1.91E-03	0.00E+00	1.40E-02
r=0.80	5.94E-03	0.00E+00	3.40E-02
RaNNA			
	2.38E-02	0.00E+00	1.70E-01
Search and Replace + RaNNA			
r=0.98	6.67E-07	0.00E+00	0.00E+00
r=0.90	2.00E-06	0.00E+00	0.00E+00
r=0.80	8.67E-06	0.00E+00	0.00E+00

Table 4.4.2: The risk of the re-identification of at least two indirect identifiers for the release of a set of clinical notes secured using various de-identification methods.

4.5 Discussion and Conclusion

In this chapter, we conducted a quantitative risk analysis of releasing data secured using RaNNA. First, we explored the infeasibility of using cryptanalysis to understand the risk of using such a method because of RaNNA's discordance with the assumptions of cryptology techniques. Second, we quantified the difficulty of reconstructing the replacement set of released traditional word embeddings. We presented many measures, some novel and some adapted from existing fields, to better quantify the probability of reconstruction. Last, we extended the probabilistic analysis of Scaiano et al. (2016) to evaluate the risk of releasing a dataset of free-text secured using search-based approaches, RaNNA, and the combination of both. As was suggested by Abdalla et al. (2020b), using the combination of search-and-replace and RaNNA results in the lowest risk for released notes.

It is important to note that for the risk analysis performed in this chapter, we purposely made assumptions that would result in an increased privacy risk in order to present a high estimate — we assumed that a dataset (that is to be released) would be de-identified with embeddings trained on the data. In practice, it would likely be safer to get the replacement sets from a dataset that is similar (in subject) but not the same as the dataset to be re-identified: the underlying relationships between PII in clinical notes from different hospitals are likely to be different enough to increase security, while the relationship between medical concepts should be much more static. Quantifying the impact of this decision is

left for future work.

There are additional limitations associated with this risk analysis. Most vitally, there has been no proof of "full security". That is, although we believe that we have empirically shown the risk to released data to be minimal if the approaches are implemented correctly, it is possible that there are undetected edge-cases that we have not caught. Furthermore, although we tried to attack data secured using RaNNA in the worst-case scenario, it may be possible that future work will develop techniques hitherto unknown to the author that increases the risk to the publicly released data (i.e., we have not provided an upper-bound). Rather our analysis is a relative risk analysis of the average case using estimated values. I believe that such an analysis remains useful for directly comparing between two techniques.

We have observed, through manual exploration, that names are often replaced by different names but diseases are often replaced by misspellings of the same disease. While from the traditional de-identification perspective, this is usually not an issue (diseases are not what we are trying to hide), if we consider an attacker re-identifying the idea behind the word (e.g., a word and all its misspellings) rather than just the strict word itself as successful re-identification, then the probabilities would need to be adjusted using the probability that a misspelling is within the nearest neighbours of any individual token. Alternatively, the sampling mechanism from the nearest neighbours could attempt to avoid misspellings by looking for character overlap and avoiding it. This is an area left for future exploration.

It is important to stress that the contribution of this work is the demonstration of how texts secured using RaNNA can be assessed for risk of release. The probabilities or values presented in this paper (e.g., 5% risk of re-identification given perfect reconstruction) should not be used as global truths applicable to all situations. The values used to represent the parameters are hypothetical (although based on "reasonable" values found in the literature). As such, all values measured in this chapter are subject to change both due to the random nature of RaNNA (where a bad run can result in increased risk), but also due to the underlying data. The risk will likely change depending on the note-type being analyzed (e.g., progress vs consultation notes) and we suggest that these values be calibrated for the specific dataset used. Furthermore, as stated earlier, the estimates are not upper-bounds on the estimated risk.

There is also future work that can help further decrease the risk of re-identification. In this work, we only considered the risk of re-identification given the full release of a dataset secured using RaNNA (i.e., the entire dataset used to create the embeddings to enact RaNNA is secured and released). The underlying risks are likely to decrease if we use a different dataset to secure the text. For example, it may be possible to perform the replacement of tokens using an embedding model trained on a different (possibly larger dataset).

As the relationships between sensitive tokens would change, this would likely significantly reduce the risks associated with release, though proving this is a difficult endeavor.

As mentioned in Section 3.5, it may be possible to use contextual embeddings to implement RaNNA. However, the risk assessment would likely have to take an entirely different approach to properly account for the contextual nature of the models as well as the larger number of parameters in the contextual embedding models. Such an extension is not trivial and would constitute a research project on its own.

Part II

Expanding the scope of clinical de-identification

Chapter 5

Authorship Attribution Increases the Risk to Patient Privacy

5.1 Introduction

State-of-the-art (SotA) approaches to protecting patient privacy in clinical notes often define a set of personally identifying features (e.g., names, addresses) to delete or replace (Neamatullah et al., 2008; Szarvas et al., 2007; Ferrández et al., 2013; Uzuner et al., 2008; Dernoncourt et al., 2017; Liu et al., 2017; Yadav et al., 2016; Ahmed et al., 2020). Recent work has challenged such approaches by disputing: 1) the possibility of defining a comprehensive list of identifying features, and 2) the feasibility of designing an algorithm that perfectly captures all identifiers (Abdalla et al., 2020b). To remedy this, we proposed a statistical approach that randomizes every word in a clinical note (Abdalla et al., 2020b). RaNNA (Random Nearest Neighbour Anonymization), ensures that the anonymized text retains properties that make it suitable for use as data for training classifiers. Regardless of the specific method, existing literature for clinical note de-identification exclusively focuses on removing indicators of patient identity.

However, both traditional search-based approaches and newer approaches crucially overlook the interaction between the identity of the patient (i.e., the subject of the clinical note) and the healthcare provider (i.e., the author of the note).[1] Correctly identifying the healthcare provider (HCP) who has written a clinical note dramatically narrows the list of possible patients who could be the subject of that note. Each patient becomes more identifiable as the number of patients with specific attributes *AND* the same HCP is lower. Alternatively, the HCP is an additional attribute that can be used to aid in the de-identification

[1] Most techniques remove PII also belonging to the HCP, however they make no attempt to alter or remove other features indicative or authorship (e.g., writing style).

of the patients.

Publicly available clinical datasets are often released *with* anonymized author labels provided as part of the dataset (e.g., MIMIC-III (Johnson et al., 2016)). As such, an attacker can try to leverage outside information to link an anonymized ID with a real-world identity. If the attacker has a model, trained on the publicly available dataset, to predict which anonymized ID has written a note, they can apply that model to predict the ID of a single clinical note for which they have the true author. To get note(s) from the same hospital (such that the author of the note would correspond to one of the authors in the dataset), the attacker can either have received care at the same institution and requested their personal health records, or attempt to buy records from patients who have, this process has illustrated using a flow chart in Figure 5.1.1. Alternatively, if two hospitals release datasets of the same type, an HCP who worked in both hospitals can be linked and later identified with access to employee records or upon the compromise of one of the datasets. Without such labels, it may still be possible to cluster authors or use publicly available lists of physicians and their specialities (e.g., from the College of Physicians and Surgeons in Ontario) to coordinate some sort of attack; however, we leave such unsupervised attacks to future work.

The structure of this chapter is as follows:

- First, I successfully train a classifier to identify the author of de-identified clinical notes, thus demonstrating that patient de-identification (termed 'subject de-identification') does not protect against author attribution.
- Second, I show that this result holds even when controlling for note and author details (e.g., role, and type of note). That is, the classifiers that we used for HCP attribution do not overly rely on differences in note type or on HCP roles.
- Third, I deploy various SotA de-identification methods to demonstrate that, regardless of the de-identification method used (Dernoncourt et al., 2017; Ahmed et al., 2020; Abdalla et al., 2020b), HCPs can be identified with a high degree of accuracy in the MIMIC-III dataset. This highlights the fact that existing methods are unable to mitigate this risk to patient privacy.
- Last, to gain a better understanding of how the algorithms distinguish between authors, I examine the most informative features for author attribution and present some differences of writing style.

In light of this argument, this work is the first to:

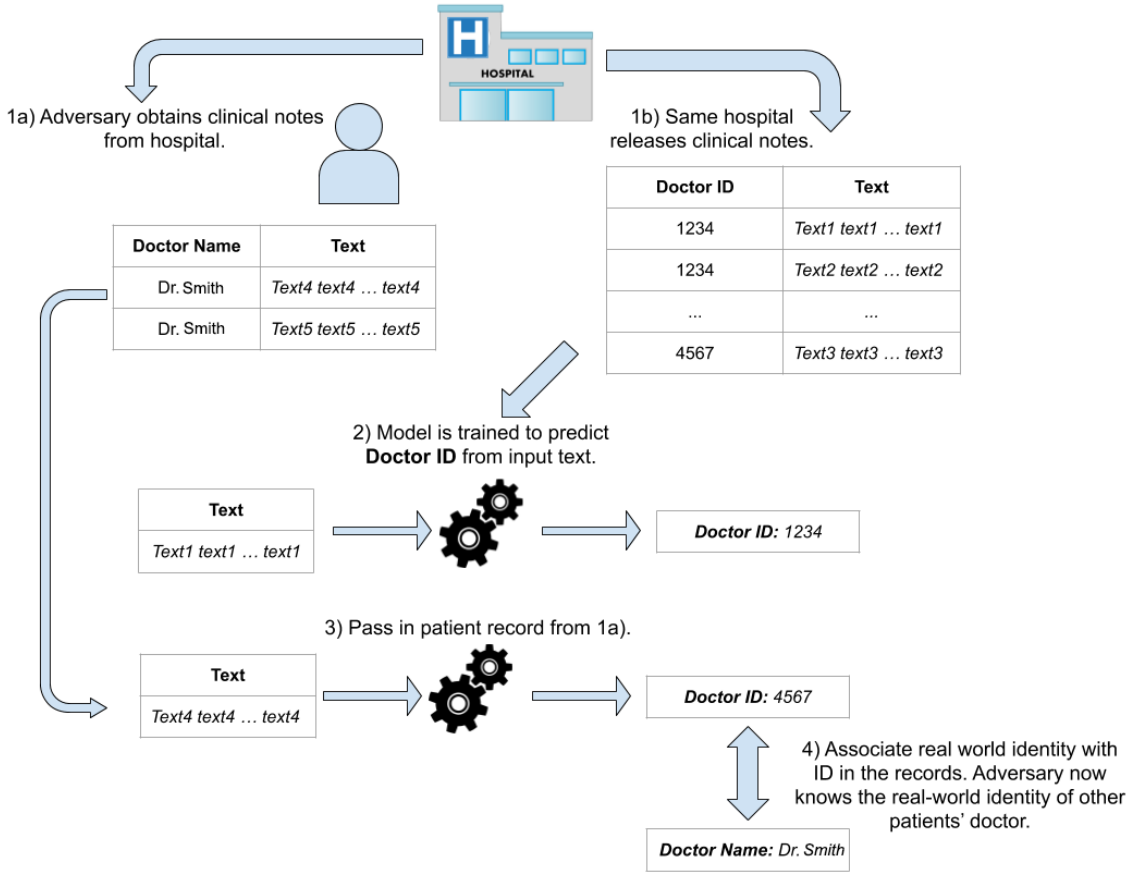

Figure 5.1.1: Illustration of scenario to associate anonymized HCP ID with real-world identity, thereby gaining additional information on all other patients of the doctor. This figure demonstrates how an adversarial actor is able to leverage real-world knowledge (as little as one note with true doctor label – possibly their own) to re-identify an anonymized HCP ID (even if their specific note was not in the publicly released dataset). Knowing the real-world identity of HCP can then be used to narrow down the list of possible patients in certain circumstances.

- Empirically demonstrate that author attribution of clinical notes is possible, as suggested by Thaine and Penn (2020).
- Test the effect of clinical de-identification methods on author attribution.

5.2 Background

To date, most de-identification in the clinical setting has focused on protecting the privacy of patients by removing a predefined set of sensitive attributes (e.g., names, addresses, occupations) (Ferrández et al., 2013; Dernoncourt et al., 2017; Liu et al., 2017; Yadav et al.,

2016; Ahmed et al., 2020; Meystre et al., 2010). Such approaches use a variety of means to achieve this goal, including dictionaries of sensitive attributes Thomas et al. (2002), and training end-to-end machine learning models (Neamatullah et al., 2008; Dernoncourt et al., 2017; Liu et al., 2017; Yadav et al., 2016; Ahmed et al., 2020). These approaches are able to remove between 90 and 99% of the sensitive attributes, though their performance on unseen datasets is often much worse Steinkamp et al. (2020). How to define a sensitive attribute is a matter of debate Kayaalp (2017); Abdalla et al. (2020b).

Most of these approaches do remove surface-level identifiers of HCPs (i.e., their names), but do not deal with writing style. The main contribution of this work is to demonstrate the writing style alone, *with no surface-level identifiers*, is enough to perform author attribution. There has been limit work demonstrating the ability to perform author attribution on health-related informatics — Bobicev et al. (2013) demonstrate author attribution on Health Forums — but no work demonstrating this on clinical notes.

Algorithms for author obfuscation often use algorithms to create new text in a different style from the input text (Bo et al., 2019; Shetty et al., 2018; Shen et al., 2017) or offer live suggestions to authors about how to obfuscate their writing style (McDonald et al., 2012, 2013), but have not been discussed in the context of clinical de-identification — something our work seeks to change.

5.3 Data

In this section, I present details of the dataset used to perform the experiments in this chapter. I make use of the MIMIC-III dataset (Johnson et al., 2016). Below we will provide descriptive statistics of the dataset and describe what subsets we made use of. I will also discuss idiosyncrasies of the MIMIC-III dataset and the possible effects on the results.

In these experiments, author identity was defined using the field for caregiver ID (i.e., "CGID") in the relevant MIMIC tables ("CAREGIVERS" and "NOTEEVENTS"). This field is intended to be a unique identifier of authorship; however, "due to imprecision in the storage of unique identifiers across the database" it is possible that a small percentage of notes may not be correctly classified. The MIMIC-III documentation states that this is an unlikely occurrence and as such we do not believe it has an effect on the conclusions of the experiments.

Author role was identified using the "LABEL" field in the "CAREGIVERS" table. As this field is a free-text field, there are many typographical errors (e.g., MD, M.D., MDs would all represent the same label). To account for this, when choosing authors with the same role, I manually went through the authors to ensure that these differences in input did

not result in our rejection of authors.

5.3.1 Top 50 Author Dataset

In this subsection I provide descriptive statistics of the dataset used in the first experiment: 50-way author identification experiment between the 50 most prolific authors in the MIMIC-III dataset.

Descriptive Statistics:

Total number of notes: 257,029

Number of notes per author:

Mean: 5141

Median: 3996

25th, 75th percentile range: 3161–5347

Number of tokens per note:

Mean: 105

Median: 68

25th, 75th percentile range: 42–129

5.3.2 Specialty Author Dataset

Here, I describe the dataset for the second and third experiments: for each of the groupings, defined using HCP role and note category, we perform 10-way author classification between the most prolific authors. Table 5.3.1 presents the MIMIC Category & Caregiver ID labels that were used to select authors and notes.

Speciality (Note-Type)	**MIMIC Category**	**Caregiver ID Label (Author-Role)**
Nursing	"Nursing" or "Nursing/Other"	"MD"
Physician	"Physician"	"MD"
General	"General"	"MD"
Nutrition	"Nutrition"	"RD/DI"
Rehab	"Rehab Services"	"Rehab"
Respiratory	"Respiratory"	"RT/RRT"

Table 5.3.1: Defining the specifics behind the note-type groupings based on chosen MIMIC Category and Caregiver ID label.

Tables 5.3.2 and 5.3.3 present the descriptive statistics for the number of tokens per

note and notes per type given the defined selection criteria. As we can see there is large variation in the number of notes and tokens in total and per author.

Speciality (Note-Type)	# of notes	Mean #	Median #	25th, 75th Percentile of number of notes
General	3137	285.2	154	85–190
Nursing	76707	7670.7	7178	5703–9706
Nutrition	8957	814.3	1006	263–1224
Physician	21193	1926.6	1982	1518–2232
Rehab	4105	410.5	316	246–470
Respiratory	8362	836.2	842	754–889

Table 5.3.2: Descriptive statistics for the number of notes per note-type.

Speciality (Note-Type)	Mean number of tokens per note	Median number of tokens per note	25th, 75th Percentile of number of tokens per note
General	166	128	92–194
Nursing	98	81	60–108
Nutrition	319	331	246–406
Physician	833	792	644–971
Rehab	421	473	221–602
Respiratory	162	166	137–192

Table 5.3.3: Descriptive statistics for the number of tokens in all notes.

5.4 Experiments

In this section, I present a series of experiments that explore author identification of clinical notes. All experiments are performed using Scikit-learn (Pedregosa et al., 2011).

5.4.1 Simple Author Identification

The first experiment demonstrates that it is possible to accurately identify the HCP who authored a specific note. I train a classifier that takes a clinical note as input and predicts the author. I counted the number of notes written by each author in the dataset and selected the top 50 authors by number of notes (257,029 notes in total). Each note was then represented by a TF-IDF (term frequency $\times$ inverse document frequency) vector (Oleynik et al., 2019; Havrlant and Kreinovich, 2017) with uni-, bi-, and trigrams as features. I performed 50-way author classification using a multinomial logistic regression model and 5-fold stratified cross-validation. Evaluating the macro precision and macro recall across each of the 5

folds, this approach is able to achieve a precision of 98.26% (95% CI: 98.14–98.38%) and a recall of 97.18% (95% CI: 97.01–97.35%). This greatly exceeds the performance (i.e., informativeness) of naïvely choosing the most frequent author (henceforth: majority baseline) for which precision is 0.20% and recall is 2.00%.

5.4.2 Controlling for Note Type and Author Role

The experiment above demonstrated that a relatively simple model can be used to identify an individual author using only the distribution of words in the text. An author's particular choice of words, or vocabulary, may arise from : 1) writing style, 2) the role of the author[2] (e.g., physician or nurse), 3) the type of note[3] (e.g., discharge or nursing), or 4) other influences. If these other factors are controlled, the task becomes more challenging. To control for these differentiating factors, I split the data into sets with a single author-role and single note-type (Supplemental Table 1) and attempt to differentiate between individual authors under these conditions.

Specialty (Note-Type)	**Precision % (95% CI)**	**Recall % (95% CI)**	**Majority Baseline (Precision %)**
General	93.73 (87.47–99.99)	73.50 (70.32–76.69)	05.03 (05.02–05.04)
Nursing	99.25 (99.15–99.34)	98.85 (98.64–99.05)	01.54 (01.54–01.54)
Nutrition	69.42 (59.65–79.19)	60.37 (57.93–62.80)	01.75 (01.74–01.75)
Physician	70.86 (67.47–74.24)	69.73 (66.47–72.98)	01.10 (01.10–01.10)
Rehab	98.38 (98.04–98.72)	94.58 (93.05–96.12)	02.70 (02.70–02.71)
Respiratory	96.05 (95.34–96.76)	95.83 (95.06–96.60)	01.16 (01.16–01.17)

Table 5.4.1: Performance of logistic regression on various note-types after controlling for note-type and author-role. Each row represents a note-type and presents the average performance metric for all of the authors of that type. For each metric, the mean is presented above the 95% confidence interval which is calculated using the standard deviation across the folds. The majority baseline presents only the precision; the recall is always 10% (as there are ten classes).

[2]Author roles were extracted from the "LABEL" attribute in the "CAREGIVERS" table in MIMIC-III. The full list of author roles used can be found in Table 5.3.1.

[3]Note-types are specified using the "CATEGORY" attribute in the "NOTEEVENTS" table in MIMIC-III. The full list of note-types considered can be found in Table 5.3.1.

More specifically, for each note-type and author-role grouping, I select the top 10 authors of the same clinical role (with the most notes) and attempt to identify them — a traditional set-up in author identification research (Khonji and Iraqi, 2020; Patchala and Bhatnagar, 2018; Reddy et al., 2018). I follow the same experimental set-up as the previous experiment and report the same metrics.

Table 5.4.1 shows that controlling for note-type and author-role does not result in a substantial decrease in performance for most groups. This indicates that what allows the classifier to differentiate between HCPs is not simply the note-type or author-role but something more specific to each HCP. That is, even among those who, for example, work solely on neo-natal intensive-care patients, what they write is consistently unique enough to enable us to differentiate them.

Feature Analysis

In this section, I dive deeper into the classification of a single note-type and author-role grouping (nursing note-type written by the "MD" or medical doctor author-role) in an attempt to understand why the classifier performs so well. I re-ran the previous experiment on this subset of notes and looked at the most informative lexical features of the classifier in a single fold. The top 30 most informative lexical features (i.e., tokens) per author are presented in Table 5.6.1.

Next, I present the percentage of notes with certain words that are observed to be highly indicative of authorship. This exploration is meant to demonstrate how innocent (i.e., grammatical and correct and/or justified) differences between writing styles (e.g., presence or absence of acronyms) can easily rule out particular individuals as the authors of certain notes. I present three examples of this analysis: 1) the usage of the token "*we*" in a note (Table 5.4.2), 2) using "*week*" vs. "*wk*" (Table 5.4.3), and 3) using "*gram*" vs. "*gm*" (Table 5.4.4). For each of these tokens I present the percentage of notes, per author that make use of the tokens.

Author ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
"we"	1.56	5.41	0.74	1.05	1.15	0.9	22.07	1.31	0.12	1.26

Table 5.4.2: Percentage of nursing notes (per author) that have the word "*we*".

From the above analysis, we can see that there are several factors that aid the classifier in determining authorship. Among these factors are user templates and writing styles.

User Templates: Feature analysis revealed that note headings were among the most important features. Specifically, within our curated subset of the MIMIC-III corpus, certain

Author ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
"wk"	0.09	61.64	0.04	0.13	0.59	0.03	0.08	0.6	76.04	0
"week"	13.86	12.11	65.77	18.55	17.08	64.71	68.59	79.94	46.4	13.45

Table 5.4.3: Percentage of nursing notes (per author) that have the word "*wk*" and "*week*".

Author ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
"gm"	4.19	38.06	0.12	0.26	0.06	0.46	0.02	0.05	3.94	0.05
"gram"	2.86	0.54	0.51	0.26	4.54	0.94	1.09	83.29	66.52	54

Table 5.4.4: Percentage of nursing notes (per author) that have the word "*gm*" and "*gram*" per author.

authors always began their notes with specific titles (e.g., "*Neonatology Attending Note*", "*Neo Attend*", or just "*Neonatology*"). When I re-ran the classification after removing the first 5 tokens of each note (i.e., all headers), both precision and recall decreased by 1–2 absolute percentage points.

Writing Styles: It was observed that stylistic features were among the top features. Here, "stylistic feature" denotes tokens that don't directly carry relevant medical information. For example, we observe one author using "*we*" and "*we will*" much more than other authors. Other authors use the abbreviation "*wk*" for "*week*" or use "*gm*" for "*gram*".

Determining Authorship Between 25% of Authors Per Note-type

To demonstrate that the results still hold with a larger number of authors I performed the experiment again and instead chose 25% of the authors for each note-type and author-role (choosing the authors with the most notes). The results demonstrate that the results presented in Table 5.4.1 also hold for a larger number of authors (when the number of authors for a specific grouping was greater than 40), Table 5.4.5.

5.4.3 Testing State-of-the-Art Patient De-identification

In this section, I demonstrate that other SotA de-identification methods (personal identifying information (PII) deletion and RaNNA) do not adequately conceal the identity of the HCP writing the note.

The first SotA method, PII deletion, either just deletes all tokens relating to sensitive attributes or replaces them with tokens that contain no identifying information. For MIMIC-III we change names in the dataset from anonymized identifiers (e.g., [** NAME (NI) 1052**]) to "PII_NAME_PII". We apply PII deletion to the data used in the previous experiment and rerun the experiment.

Speciality (Number of Authors)	Precision % (95% CI)	Recall % (95% CI)	Majority Model (Precision %)	Majority Model (Recall %)
General	49.57[a]	38.94[a]	0.98	2.33
(43)	(43.93–55.20)	(35.55–42.33)	(0.98–0.98)	(2.33–2.33)
Nursing	99.36	96.95	0.47	3.85
(26)	(99.30–99.41)	(96.71–97.19)	(0.47–0.47)	(3.85–3.85)
Nutrition	95.55	94.78	5.21	20.0
(5)	(94.46–96.64)	(93.05–96.51)	(5.20–5.22)	(20.00–20.00)
Physician	57.13[a]	50.37[a]	0.08	1.64
(61)	(51.49–62.77)	(44.64–56.1)	(0.08–0.08)	(1.64–1.64)
Rehab	98.38	94.58	02.70	10.0
(10)	(98.04–98.72)	(93.05–96.12)	(02.70–02.71)	(10.00–10.00)
Respiratory	96.53	96.41	1.41	11.11
(9)	(95.74–97.32)	(95.56–97.26)	(1.40–1.41)	(11.11–11.11)

Table 5.4.5: Performance of a Logistic Regression classifier on various note-types after adjusting for note-type AND author-role. For each metric, the mean is presented above the 95% confidence interval. The numbers in the first column represent the number of authors represented in this classification task (arrived at by dividing the number of authors by 4). [a]The classifier used performed poorly. However, this is because this work did not aim to maximize performance of classification but instead to demonstrate that author identification was an issue with simple classifiers. This is still the case as simply changing the loss from L2 to L1 increases, for "General" the precision to 77.23% (70.46–84.01%) and recall to 66.91% (63.97–69.84%) and for "Physician" the precision to 74.18% (71.92–76.43%) and recall to 73.29% (70.99–75.59%).

The second SotA method, RaNNA (Abdalla et al., 2020b), replaces each token in a note with a randomly selected nearest neighbour (from a set of the nearest 3 to 15 words) using an embedding model trained on the entire MIMIC-III corpus. After applying RaNNA to the data used in the previous experiment, I rerun the previous experiment with the same setup.

We can observe that PII deletion has a negligible effect on our ability to predict which HCP wrote the note. RaNNA does reduce the performance of the author classifier, but the drop does not approach the majority baseline, Table 5.4.6.

5.5 Discussion

Data Limitations: While it is expected that the conclusions in this paper will be generalizable to other clinical datasets, there may be specialties or institutions where identifying the HCP who wrote a specific note is more difficult (e.g., an institution where all HCPs are required to use pre-defined templates).

Specialty	PII Deletion		RaNNA	
	Precision % (95% CI)	Recall % (95% CI)	Precision % (95% CI)	Recall % (95% CI)
General	92.62 (85.26–99.99)	73.85 (70.88–76.83)	56.86 (54.49–59.22)	35.19 (34.10–36.27)
Nursing	99.25 (99.16–99.34)	98.86 (98.64–99.08)	97.47 (97.23–97.72)	96.20 (95.86–96.53)
Nutrition	76.08 (69.44–82.72)	66.85 (64.76–68.94)	50.75 (50.29–51.20)	62.60 (61.60–63.60)
Physician	74.46 (73.56–75.35)	75.59 (74.67–76.52)	60.68 (55.79–65.58)	54.87 (53.32–56.43)
Rehab	97.95 (97.52–98.38)	94.10 (92.50–95.70)	70.67 (65.63–75.70)	51.76 (50.73–52.79)
Respiratory	96.18 (95.43–96.92	95.96 (95.18–96.75)	76.90 (75.19–78.62)	72.75 (71.66–73.83)

Table 5.4.6: Performance of logistic regression on various note-types that have undergone different subject de-identification (PII deletion and RaNNA). For each metric, the mean is presented above the 95% confidence interval which is calculated using the standard deviation across the folds. The performance of the majority baseline is in Table 5.4.1.

Assessing Risk: Author attribution can be carried out by (relatively) unskilled attackers, if they are able to get the record of a patient who has received care at the institution from which the data is sourced (even if the notes are not in the specified dataset, so long as the authors also wrote notes in the public database). This is because government regulations require that healthcare records be provided to patients (Health Insurance Portability and Accountability Act, 2012a). However, to compromise patient privacy, the adversary would still need more information about the patient, and it is here that existing subject de-identification methods can help to mitigate risk. Demonstrating how subject and author obfuscation interact and quantifying the actionable risk that exists for patient notes given an attempted two-pronged attack is left for future work.

Protecting Author Identity: In certain jurisdictions, where what is considered PII is quite limited, HCPs are not entitled to privacy. This is the case in the United States according to the Supreme Court ruling in *Sorrell v. IMS Health* (564 U.S. 552) in 2011. However, there are other jurisdictions where the identity of the medical provider could be considered sensitive information (e.g., in Europe under GDPR[4] (Council of European Union, 2016)). Regardless, direct identifiers of authorship are often removed. However, we argue that this is not enough; altering writing style to obfuscate author identity should be done to reduce the risk of re-identification for patients.

[4]We are grateful to Khaled El Emam for informing us of this.

While this work has demonstrated that methods intended to protect patient identity fail to prevent author identification, and hence retain clues to patient identity, it may be possible to develop methods to obscure HCP identity. Automated techniques that detect and correct for idiosyncrasies of HCPs may aid in tackling this issue. Alternatively, a technique which is able to transform different personal templates into a general shared template would greatly reduce an adversary's ability to identify the author of clinical notes.

5.6 Conclusion

In this work, I have shown that state-of-the-art patient de-identification methods do not adequately conceal the identity of the HCP writing the note. Given the provider's identity, the relation between patient and doctor can then be exploited to vastly narrow down the subset of patients to which a specific note could refer. This highlights a critical gap in research that poses a serious risk to patient privacy. I hope that future work on the de-identification of clinical notes will consider both the effects of author and subject privacy.

Author	Most Important Features
1	neonatology attending note, attending note, cont, cc day, nl voiding, neonatology attending, tf, attending note day, note day, bs, gms, nl, rr30, note, nl voiding and, murmur hr, gms tf, day, rr40, no bs, crib, attending admission note, isolette, open crib, cl, feedings, cl and, open, tf 150, tol
2	neo, wnl, attend, neo attend, respr, neo attending, cv, wk, gm, uop, attend day, neo attend day, day, pt, clear bs, kg day, spo2, discussed, glu, as noted, discussed with, cc kg day, and discussed, tw, and discussed with, abd, abd wnl, staff, known, uop and
3	imp plan, premie, progress note dol, dstx, note dol, plan premie, imp plan premie, attending progress, neonatology attending progress, attending progress note, progress note, imp, inc, monitor, crit, calories, infant, due to, progress, pedi, above, premie infant, plan premie infant, due, monitor weight, afof, retx, normal s1s2, nontender, soft nontender
4	weight, neonatology attending day, attending day, stable temperature, temperature, gms, remains, neonatology attending, gaining weight, bp mean, blood glucose, gaining, stable temperature in, temperature in, closely, remains in ra, blood, glucose, will continue to, attending, wks remains, benign abdomen, clear breath sounds, clear breath, remains on, no bradycardia, control, tf at, breathing control, remains in
5	abdomen benign, neonatology will, abdomen, comfortable, neonatology, to be, benign, neonatology remains, at, neonatology doing, dc, neonatology doing well, feeds, range, be, comfortable appearing, am, apeparing, cal, of, patient, this am, appearing, comfortable apeparing, feeds at, to, abx, remains, tolerating feeds, at present
6	baby, assessment plan, feedings, assessment, progress, of, now day of, now day, attending progress, attending progress note, neonatology attending progress, kg of, progress note, of life, life, note now, progress note now, note now day, day of life, now, cvs, day of, attending admission note, neonatology attending admission, baby is, ca, attending admission, noted, well tolerated, normal urine
7	we, we will, tfi, attending dol, neonatology attending dol, on tfi, neonatology attending, week ga, the, ga, examination, neonatology attending addendum, kg day, dol, attending addendum, ga infant, week ga infant, addendum, cc kg day, with no, room, infant with, room air, ga infant with, attending, air with, room air with, normally, in room air, in room
8	grams, grams up, neonatology attending dol, attending dol, ml, visiting, weeks stable, ds, weeks stable in, on, ml kg, neonatology attending, voiding stooling, stable in ra, stable, and up to, and up, soft flat, af soft flat, stooling wt, bc, voiding stooling wt, weeks, up to date, to date, neonatology attending exam, attending exam, date, grams down, up to
9	plans, hemodynamically stable, hemodynamically, fen wt, cvr, plans continue, wk, dev, former, fen, neonatology dol, overall, imp former, dev in, dstik, wk infant, neonatology addendum, wks cvr, cvr remains, neonatology attending note, pmd, imp, wks, grams tf, exam, cvr remains in, addendum, infant, cx, fontanelles
10	will, attending note, day of, the, note day of, note, life, day of life, of life, note day, attending note day, imp, cal oz, day, cc kg day, mom, kg day, oz, weight, was, making, he, kg day of, imp infant, room, room air, will have, air rr, room air rr, imp stable

Table 5.6.1: Top 30 features per author when classifying between the top 10 most prolific authors in the "Nursing" category.

Chapter 6

Enabling Author Obfuscation – Evaluating Semantic Relatedness

6.1 Introduction

In the previous section, I demonstrated the need for author obfuscation methods that can be applied on clinical notes. Unlike subject de-identification, which can be satisfactorily achieved by the removal of select tokens, achieving satisfactory author obfuscation is likely to require significant re-writing of the text. That is, changing all aspects of the writing style cannot be achieved simply through deletion. However, we need to ensure that our significant rewriting of the text, especially if done in an automatic manner, does not change the underlying meaning of the text. To evaluate the change in meaning after author obfuscation we need to automatically measure semantic relatedness.

The semantic relatedness of two units of language —words, phrases, sentences, etc.— is the degree to which they are close in terms of their meaning (Mohammad, 2008). The linguistic units can be words, phrases, sentences, etc. Though our intuition of semantic relatedness is dependent on many factors such as the context of assessment, age, and socio-economic status (Harispe et al., 2015), it is argued that a consensus can usually be reached for many pairs (Harispe et al., 2015). Consider the two sentence pairs in Table 6.1.1. Most speakers of English will agree that the sentences in the first pair are closer in meaning to one another than those in the second. When judging the semantic relatedness between two sentences, humans generally look for commonalities in meaning: whether they are on the same topic, express the same view, originate from the same time period, one elaborates on (or follows from) the other, etc.

The semantic relatedness of two units of language has long been considered funda-

Pair 1: a. *There was a lemon tree next to the house.*
b. *The boy enjoyed reading under the lemon tree.*

Pair 2: a. *There was a lemon tree next to the house.*
b. *The boy was an excellent football player.*

Table 6.1.1: Most people will agree that the sentences in pair 1 are more related than the sentences in pair 2.

mental to understanding meaning (Halliday and Hasan, 1976; Miller and Charles, 1991); given how difficult it has been to define meaning, a natural approach to get at the meaning of a unit is to determine how close it is to other units. Semantic relatedness is also central to textual coherence and narrative structure. Usually, a large number of sentences in a document will be semantically related to each other, and this is a crucial component of meaningful communication (Halliday and Hasan, 1976; Morris and Hirst, 1991). Automatically determining semantic relatedness has many applications such as question answering, text generation, and summarization.

However, prior NLP work has focused on semantic similarity (a small subset of semantic relatedness), largely because of a dearth of datasets on relatedness. The few relatedness datasets that exist are only for word pairs (Rubenstein and Goodenough, 1965; Radinsky et al., 2011) or phrase pairs (Asaadi et al., 2019). Further, most existing datasets were annotated, one item at a time, using coarse rating labels such as integer values between 1 and 5 representing coarse degrees of closeness. It is well documented that such approaches suffer from inter- and intra-annotator inconsistency, scale region bias, and issues arising due to the fixed granularity (Presser and Schuman, 1996). Further, the notions of *related* and *unrelated* have fuzzy boundaries. Different people may have different intuitions of where such a boundary exists. Finally, for some tasks, it is more appropriate to train on a dataset of relatedness than similarity. (§6.2.1 discusses how relatedness and similarity are different.)

Unlike previous chapters, this work was not solely led by myself. Rather, the general conceptualization of this work was arrived at independently by Krishnapriya Vishnubhotla and me. We agreed to collaborate together under the supervision of Saif Mohammed. As such, for the remainder of the chapter, I will use the pronoun "*I*" to indicate contributions claimed by myself and "*we*" to indicate contributions claimed by others. In this work I present the first manually annotated dataset of sentence–sentence semantic relatedness annotating using a *comparative* annotation schema. In comparative annotations, two (or more) items are presented together and the annotator has to determine which is greater with respect to the metric of interest. Since annotators are making relative judgments, the

limitations discussed earlier for rating scales are greatly mitigated. Importantly, such annotations do not rely on arbitrary boundaries between arbitrary categories such as "strongly related" and "somewhat related". It includes fine-grained scores of relatedness from 0 (least related) to 1 (most related) for 5,500 English sentence pairs. The sentences are taken from diverse sources and thus also have diverse sentence structures, varying amounts of lexical overlap, and varying formality.

Using this data:

1. I explore to what extent do speakers of English intuitively agree on the relatedness of pairs of sentences? (Section 6.5)

2. We explore what makes a sentence pair more related than another sentence pair? (Section 6.6)

3. I explore how well existing approaches of unsupervised sentence representation capture semantic relatedness (by placing related sentence pairs closer to each other in vector space)? (Section 6.7)

4. We explore how well supervised approaches to sentence representation capture semantic relatedness. (Section 6.7)

The curated dataset is referred to as *STR-2021*, and the task of predicting relatedness between sentences as the *Semantic Textual Relatedness (STR)* task. The data, data statement, and annotation questionnaire are publicly available at: https://github.com/Priya22/semantic-textual-relatedness.

6.2 Background

The three subsections below discuss key ideas regarding the approach towards annotating data for semantic relatedness and similarity, existing datasets, and comparative annotation, respectively.

6.2.1 Annotating Relatedness and Similarity

Closeness of meaning can be of two kinds: semantic relatedness and semantic similarity. Two terms are considered semantically similar if there is a synonymy, hyponymy (hypernymy), or troponymy relation between them (examples include *doctor–physician* and *mammal–elephant*). Two terms are considered to be semantically related if there is any lexical semantic relation at all between them. Thus, all similar pairs are also related, but

not all related pairs are similar. For example, *surgeon–scalpel* and *tree–shade* are related, but not similar.

Analogous to term pairs, two sentences are considered semantically similar when they have a paraphrasal or entailment relation. Determining such an equivalence of meaning is useful in NLP tasks such as text summarization. Semantic relatedness, however, accounts for all of the commonalities that can exist between two sentences (Halliday and Hasan, 1976; Morris and Hirst, 1991). For example, the sentences in Table 6.1.1 Pair 1 are highly related, but they are not paraphrases or entailing. This expands the scope of the measure to include aspects such as the relatedness between their topics, their styles, the emotions expressed, and so on. Such a measure is highly relevant in applications such as question answering, information retrieval, and text generation. Even models of text summarization benefit from measures of relatedness as the sentences in a summary (especially those adjacent to one another) need to have some degree of continuity in meaning.

However, there are no widely agreed upon concrete definitions of relatedness. This presents a challenge for gathering annotations; one can either: (i) construct their own codified instructions on how to judge semantic relatedness under various scenarios (e.g., overlapping sentence structure, relatedness of topic, differences in facts stated, etc.), or (ii) abstain from explicitly and comprehensively defining relatedness (relying instead on annotators' intuitions). In this work, I chose to do the latter. This allows me to: (i) demonstrate the extent to which human intuition regarding relatedness of sentence pairs is reliable (without needing comprehensive definitions), and (ii) use the resulting relatedness dataset to empirically determine how speakers of a language naturally judge semantic relatedness.

6.2.2 Existing Relatedness and Similarity Datasets

There are several term-pair datasets capturing similarity and relatedness (Rubenstein and Goodenough, 1965; Finkelstein et al., 2001; Miller and Charles, 1991; Radinsky et al., 2011).

The datasets created for sentence pair similarity (e.g., STS (Agirre et al., 2012, 2013, 2014, 2015, 2016), MRPC (Dolan and Brockett, 2005), and LiSent (Li et al., 2006)) have multiple weaknesses. First, all past studies ask annotators to choose among coarse similarity labels — resulting in information loss. This also makes annotation difficult because distinctions between categories are often not clear; for example, the STS 2012–2016 questionnaires ask annotators to make the distinction between *2: not equivalent but share some details* and *1: not equivalent, but are on the same topic*, which is often not straightforward. Second, despite claiming to determine semantic similarity, the descriptions of categories 1

and 2 incorporate aspects of semantic relatedness — an amalgamation muddying the waters with respect to the phenomenon being annotated. Such an amalgamation is also a weakness of the SICK (Marelli et al., 2014) dataset which combines a labeling scheme from STS with those about entailment and contradiction. This results in an annotation schema that cannot be said to, by definition, belong cleanly to either similarity or relatedness, rather a specific hybrid defined by the authors.

For the annotations of STR-2021, we avoid fuzzy ill-defined levels of relatedness. Rather, we rely instead on the intuitions of fluent English speakers to judge relative rankings of sentence pairs by relatedness.

6.2.3 Comparative Annotations

The simplest form of comparative annotations is paired comparisons (Thurstone, 1927; David, 1963). Here, annotators are presented with pairs of examples and are asked to choose which item is greater with respect to the property of interest (semantic relatedness, sentiment, etc.). These choices can then be used to generate an ordinal ranking of items. While paired comparisons, as a methodology, does not suffer from the drawbacks mentioned previously, it requires a large number of annotations (N^2, where N = # items).

Best–Worst Scaling (BWS) is a comparative annotation schema that builds on pairwise comparisons and does not require as many labels (Louviere and Woodworth, 1991). Annotators are given n items at a time (for our work, $n = 4$ and an *item* is a pair of sentences). They are instructed to choose the best (i.e., most related) and worst (i.e., least related) item. Annotation for each 4-tuple provides us with five pairwise inequalities. For example, if a is marked as most related and d as least related, then we know that $a > b, a > c, a > d, b > d$, and $c > d$. From all the annotations (and corresponding inequalities) we can calculate real-valued scores, and thus an ordinal ranking of items, using a simple counting mechanism (Orme, 2009; Flynn and Marley, 2014): the fraction of times an item was chosen as the best (i.e., most related) minus the fraction of times the item was chosen as the worst (i.e., least related). Given N items, reliable scores are obtainable from about $2N$ 4-tuples (Kiritchenko and Mohammad, 2017).

6.3 Data Sources

Like previous work on semantic similarity, we decided to construct our dataset by sampling sentences from many sources to capture a wide variety of text in terms of sentence structure, formality, and grammaticality. In the following subsection, I will explicitly demarcate

which datasets were extracted by me.

We selected sentence pairs with varying amounts of lexical overlap because randomly sampling sentence pairings would result in mostly unrelated sentences. This also allowed us to systematically study the impact of lexical overlap on semantic relatedness. For the paraphrase datasets (Formality, ParaNMT, and Wikipedia), we obtained sentence pairs in two ways: by directly taking the paraphrase pairs (indicated by the suffix *_pp*), and by randomly pairing sentences from two different paraphrase pairs (suffixed by *_r*). The paraphrase pairs were selected at random from the source dataset, whereas the lexical overlap strategy was applied in the creation of the random pairs. From STS, we randomly sampled 50 sentence pairs having similarity scores in [0–1), 50 pairs having scores in [1–2), and so on. Table 6.3.1 lists the datasets used and summarizes key details of the sentence pairs in STR-2021.

Types of Pairs	**Key Attributes**	**# pairs**
1. Formality	paraphrases, style	
Formality_pp	paraphrases, differ in style	300
Formality_r	random pairs	700
2. Goodreads	reviews, informal	1000
3. ParaNMT	automatic paraphrases	
ParaNMT_pp	automatic paraphrases	450
ParaNMT_r	random pairs	300
4. SNLI	captions of images	750
5. STS	have similarity scores	250
6. Stance	tweet pairs with same hashtag, less grammatical	750
7. Wikipedia	formal	
Wiki_pp	paraphrases, formal	500
Wiki_r	random pairs, formal	500
ALL		5500

Table 6.3.1: Summary of sentence pair types in STR-2021.

Below, each subsubsection provides further information about the sources of data and how sentence pairs were sampled for their inclusion in STR-2021.

6.3.1 Formality Data

The first paraphrase corpus is the Formality dataset from Rao and Tetreault (2018); they refer to it as GYAFC. This corpus consists of human-written formal and informal paraphrases for sentences sourced from the Yahoo! Answers platform. The sampling procedure used for this dataset, described below, is also used for the sampling procedure of the ParaNMT

dataset.

Formality_pp: Sentences were assigned to one of 50 buckets based on their lexical overlap score as described previously. Each bucket was then uniformly sampled to extract 300 sentence pairs.

Formality_r: Sentences less than 5 or more than 25 tokens were removed. To create the sentence pairs, we looped in a random order through all possible pairing of sentences. Two sentences were paired if they share at least 25% of their tokens but less than 75% of their tokens AND the difference in length between both sentences did not exceed 25%. 700 such sentence pairs were extracted.

6.3.2 Goodreads Data

We created 1000 sentence pairs by sampling from the UCSD Goodreads Dataset (Wan and McAuley, 2018; Wan et al., 2019), which has book reviews from the Goodreads website. We limited the sampling to the 'Fantasy and Paranormal' genre, since it contained a relatively higher number of reviews per book, allowing for a higher possibility of sampling more related sentence pairs. Each review was first split into sentences using the default NLTK sentence tokenizer; we kept only those sentences with the number of tokens between 5 and 25. We then randomly examined pairs of sentences and quantified the lexical overlap between then with an IDF-weighted Dice overlap score, Equation 6.1. The pairs were then assigned to buckets based on this overlap score; the range of each bucket was obtained by first finding 50 equally-spaced percentiles of the entire score distribution. We then sampled exponentially increasing number of sentences from low to high weighted Dice overlap bins such that a total of 1000 sentence pairs were included.

6.3.3 ParaNMT Data

ParaNMT (Wieting and Gimpel, 2018) is a dataset of 51 million sentential paraphrases that were automatically generated using a neural machine translation system. We generated two sets of pairs from these sentences corresponding to paraphrases and random pairs:

ParaNMT_pp: We assigned paraphrases to buckets based on the Dice score between the two sentences. We divided the range of scores into 100 equally-sized percentiles. We then sampled pairs uniformly from each bucket, for a total of 450 sentence pairs.

ParaNMT_r: For the random, non-paraphrase sentence pairings, we used the Dice score to extract 300 pairs, analogous to the creation of the **Formality _r** pairs.

6.3.4 SNLI Data

I created 750 sentence pairs by sampling from the Stanford Natural Language Inference (SNLI) Dataset (Bowman et al., 2015). SNLI is composed of image description captions; for each caption, multiple premise sentences are generated, along with multiple possible hypothesis sentences that could possibly belong to each premise. To build the sentence pairs I sought to pair different premise sentences together. I chose not to pair premise and hypothesis sentences together as their sentence structure was significantly different (and simpler for the hypothesis sentences), as noted by the creators of the dataset. Even still, the majority of premise sentences are very short (with a mean token count of 14), often following very simple (and similar) grammatical structure.

To generate the sentence pairs, first I removed all sentences with less than 5 or more than 25 tokens. Then, for each token in all remaining sentences, I replaced each token with its most frequent synonym, using Roget's Thesaurus (Roget, 1911) to define synonymous relationships. Words which did not have synonyms were left unchanged. The intention behind replacing each word with its most frequent synonym was to ensure that synonymous phrasings would count as overlaps when we measure it. I then randomly selected 750 sentences to serve as the first sentence of our final pairings. To find the second sentence to each pairing, I looped through all premise sentences and returned the first sentence that satisfied two conditions: 1) The unigram overlap was greater than or equal to 25% and less than 75% of the first sentence, and 2) the difference in length between both sentences did not exceed 25%.

6.3.5 STS Data

We selected 250 sentence pairs from existing STS corpora. This selection was done to enable a small investigation into the interplay between relatedness and similarity, which could serve as motivation for further investigation in future work. For this dataset, we randomly sampled 50 sentence pairs from each of bucket of annotation (i.e., 50 sentence pairs having an STS similarity score falling in $[0, 1)$, 50 sentence pairs having scores in $[1, 2)$, and so on).

6.3.6 Stance Data

I created 750 sentence pairs by sampling from Mohammad et al. (2016)'s dataset of tweets labeled for stance. The original dataset is composed of individual tweets labelled for both stance ('*For*', '*Against*', '*Neither Inference Likely*') and sentiment ('*Positive*', '*Negative*', '*Neutral*'). The dataset was built from tweets focused on six targets: '*Atheism*', '*Climate Change*', '*Donald Trump*', '*Feminism*', '*Hillary Clinton*', '*Abortion*'.

When curating the sentence pairs, the possible targets were limited to '*Hillary Clinton*', '*Donald Trump*', and '*Abortion*'. Sentence pairs were chosen such that both sentences shared the same target. 500 sentence pairs shared their stance towards their target (i.e., 250 *for–for* pairs and 250 *against–against* pairs). 250 sentences pairs differed on their stance (i.e., *250 for–against* pairs). I did not use any lexical overlap heuristic to specify which tweets should be paired with each other because we were interested in studying whether overlap in topic was a strong enough signal to impact relatedness. That is, by choosing pairs with the same target, I was already pre-selecting for various degrees of relatedness.

6.3.7 Wikipedia Data

I sampled 1000 sentence pairs from a dataset that pairs sentences from English Wikipedia with sentences from Simple English Wikipedia. Created to enable the task of sentence simplification, the paired sentences are often very closely related. I used this dataset in two ways: 1) Extracting sentence pairs which serve as paraphrases or near paraphrases (referred to as Wiki_pp), and 2) pairing sentences to other random sentences in the dataset (referred to as Wiki_r).

Wiki_pp: First, I removed any pairings for which either sentence was less than 5 words or more than 25 words. Then I narrowed the list of pairings further by removing any pairings that did not share more than 25% but less than 75% of unique unigrams. From the remaining sentence pairs, I randomly selected 500 paired sentences.

Wiki_r: Here, I only made use of the full sentences from the original Wikipedia, discarding sentences from Simple Wikipedia. I removed all sentences that have less than 5 or more than 25 tokens. To create the sentence pairs, I looped in a random order through all possible pairing of sentences. I paired two sentences if they shared at least 25% of their tokens but less than 75% of their tokens AND the difference in length between both sentences did not exceed 25%. The process was stopped once 500 sentence pairs were generated.

6.4 Annotating For Semantic Relatedness

From the list of 5,500 sentence pairs, I generated 11,000 unique 4-tuples (each 4-tuple consists of 4 distinct sentence pairs) such that each sentence pair occurs in around eight 4-tuples.[1]

In framing this task, I did not use detailed or technical definitions; rather, I provided brief and easy-to-follow instructions, gave examples, and encouraged annotators to rely on their intuitions of the English language to judge relative closeness in meaning of sentence pairs (similar to Asadi et al.'s (2019) work on bigrams). Annotators were asked to judge the "*closeness in meaning of sentence pairs*". Inspired by early work in linguistics on cohesion in text (Halliday and Hasan, 1976), it was also specified that: "*Often sentence pairs that are more specific in what they share tend to be more related than sentence pairs that are only loosely about the same topic*" and "*If a sentence has more than one interpretation, consider that meaning which is closest to the meaning of the other sentence in the pair.*" This is in-line with application scenarios where often relatedness is to be determined between sentences from the same document.[2] The full questionnaire can be found online with the publicly released dataset.

6.4.1 Crowdsourcing Annotations

We used Amazon Mechanical Turk (MTurk) for obtaining annotations.[3] Each 4-tuple (also referred to as a question) in our MTurk task consists of four sentence pairs. Annotators are asked to choose the (a) most-related, and (b) least-related sentence pairs from among these four options. Each question is annotated by two MTurk workers.[4]

For quality control, the task was open only to fluent speakers of English, based in the US, and those MTurk workers with an approval rate higher than 98%. Furthermore, we inserted "Gold Standard" questions at regular intervals in the task. These questions were manually annotated by all the authors of the associated paper and had high agreement scores. If an annotator gets a gold question wrong, they are immediately notified and shown the correct answer. This has several benefits, including keeping the annotator alert

[1]The tuples were generated using the BWS scripts provided by Kiritchenko and Mohammad (2017): http://saifmohammad.com/WebPages/BestWorst.html.

[2]This also addresses the scenario of referent ambiguity: for example, it induces annotators to consider the mentions of *lemon tree* in the two sentences of Pair 1 in Table 6.1.1 to be referring to the same lemon tree. In general, often one cannot be certain of the referents of mentions just from text, and so relying on human judgments of likely coreferent mentions is a reasonable strategy for relatedness annotation.

[3]This project was approved by the University of Toronto's Institutional Research Ethics Board (Protocol #: 40736).

[4]Pilot studies showed that this results in reliable scores.

and clearing any misunderstandings about the task. Those who scored less than ~70% on the gold questions were stopped from answering further questions and were paid for their work. All their responses were discarded.

6.4.2 Annotation Aggregation

I aggregated information from various responses by using the counting procedure discussed in Section 6.2.3. Since relatedness is a unipolar scale, the resulting relatedness score was linearly transformed to fit within a 0–1 scale of increasing relatedness.

Figure 6.4.1 presents a histogram of relatedness scores for STR-2021. Observe that each of the subsets covers a wide range of relatedness scores; that the lexical overlap sampling strategy has resulted in a wide spread of relatedness scores; and that supposed paraphrases are spread across much of the right half of the relatedness scale.

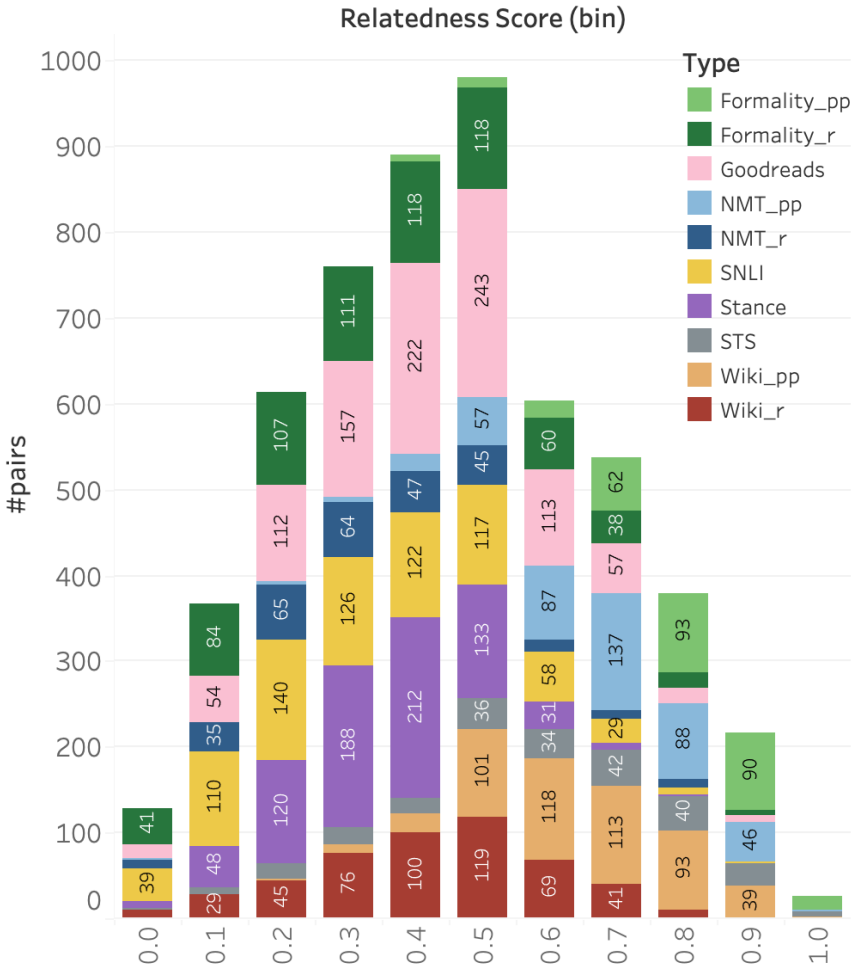

Figure 6.4.1: Histogram of STR-2021 relatedness scores.

6.5 Reliability of Annotations

For annotations producing real-valued scores, a commonly used measure of quality and reliability is *split-half reliability* (SHR) (Cronbach, 1951; Kuder and Richardson, 1937). SHR is a measure of the degree to which repeating the annotations would result in similar relative rankings of the items. To measure SHR, annotations for each 4-tuple are split into two bins. The annotations for each bin are used to produce two different independent relatedness scores. Next, the Spearman correlation between the two sets of scores is calculated—a measure of the closeness of the two rankings. If the annotations are reliable then there should be a high correlation. This process is repeated 1000 times and the correlation scores are averaged. Table 6.5.1 shows the result. SHR of 0.84 indicates high annotation reliability.

# Sentence Pairs	# Tuples	# Annotations Per Tuple	# Annotations	# Annotators	SHR
5,500	11,000	8	21,936	389	0.84

Table 6.5.1: Annotation statistics. SHR = split-half reliability (as measured by Spearman correlation).

6.5.1 STR vs STS

We also conducted experiments to assess fine-grained rankings of common sentence pairs as per our relatedness scores and as per STS's similarity scores. For each of the sets of 50 sentence pairs taken from STS (with scores in (0–1], (1–2], etc.), we calculated the Spearman correlations between the rankings by similarity and rankings by relatedness. We found that the correlations are only 0.25 (weak) and 0.19 (very weak) for the bins of (1,2] and (3,4], respectively, and only about 0.49 (moderate) for the bins of (2,3] and (4,5]. The (0,1] bin produces a correlation of 0.67 (moderate).

This analysis demonstrates that the fine-grained ranking of items in the STS dataset by similarity differs considerably (depending on the range) than the ranking found in the STR dataset. This is especially the case for sentence pairs which are in the intermediate range of similarity or relatedness. Without a third dataset, or extensive manual examination of these sentences pairs it is difficult to make claims of correctness regarding the observed rankings of sentence pairs in either dataset. However, as our methodology is designed to arrive at an ordinal ranking of elements through comparative annotations, unlike averaging Likert Scale annotations, we believe that of the two datasets, our labels are more trust-worthy.

6.6 What Makes Sentences More Semantically Related?

The availability of a dataset with human notions of semantic relatedness allows one to explore fundamental aspects of meaning: for example, what makes a pair of sentences more related than another? In this section, we examine some basic questions. We ask: "On average, to what extent is the semantic relatedness of a sentence pair impacted by presence of:"

- identical words (lexical overlap)? (Table 6.6.1, Q1)
- related words? (Table 6.6.1, Q2)
- related words of the same part of speech? (Table 6.6.1, Q3)
- related subjects, related objects? (Table 6.6.1, Q4)

6.6.1 Method

To explore the questions above, we first computed relevant measures for Q1 through Q4 (lexical overlap, term relatedness, etc.) for each sentence pair in our dataset. We then calculated the correlations of these scores with the gold relatedness scores.

Lexical Overlap. A simple measure of lexical overlap between two sentences X and Y is the Dice Coefficient (the number of unique unigrams occurring in both sentences, adjusted by their lengths):

$$\frac{2 \times |\, unigram(X) \cap unigram(Y) \,|}{|\, unigram(X) \,| + |\, unigram(Y) \,|} \tag{6.1}$$

Related Words: To calculate this measure, average the embeddings for all the tokens in a sentence and computed the cosine similarity between the averaged embeddings for the two sentences in a pair. This roughly captures the relatedness between the terms across the two sentences.[5] Token embeddings were taken from Google's publicly released Word2Vec embeddings trained on the Google News corpus (Mikolov et al., 2013a).

Related Words with same POS: The same procedure was followed as for Q2, except that only the tokens for one part of speech (POS) at a time were considered. We determined the part-of-speech of the tokens using spaCy (Honnibal et al., 2020).[6]

[5] Other ways to estimate relatedness between sets of words across two sentences may also be used.

[6] We used the simple (coarse-grained) UPOS part-of-speech tags: https://universaldependencies.org/docs/u/pos/

Related Subjects and Related Objects: For Q4, which examines the importance of different parts of a sentence, we employ the same process as Q2, except that for a given sentence: only tokens marked as subject are averaged; and only tokens marked as object are averaged. The packages spaCy (Honnibal et al., 2020) and Subject Verb Object Extractor (de Vocht, 2020) were used to determine all tokens that are the subject and object.

6.6.2 Results

Question	Spearman	# pairs
Q1. Lexical overlap	0.57	5500
Q2. Related words - All	0.61	5500
Q3a. Related words - per POS		
PROPN	0.50	1907
NOUN	0.45	4746
ADJ	0.36	2236
VERB	0.31	3946
PRON	0.30	1800
ADV	0.28	1147
AUX	0.25	2069
ADP	0.23	2476
DET	0.20	3265
Q3b. Related words - per POS group		
Noun Group	0.60	5478
Verb Group	0.32	4999
ADJ Group	0.29	4584
Q4. Related Subjects and Objects		
Subject	0.29	1611
Object	0.43	1618

Table 6.6.1: Correlation between features and the relatedness of sentence pairs. A rule of thumb for interpreting the numbers: 0–0.19: very weak; 0.2–.39: weak; 0.4–0.59: moderate; 0.6–0.79: strong; 0.8–1: very strong.

Table 6.6.1 shows the results. Row Q1 shows that simple word overlap obtains a correlation of 0.57 (traditionally considered to be at the high end of weak correlation). Figure 6.6.1 is a scatter plot where the x-axis is the word overlap score, the y-axis is the relatedness score, and each dot is a sentence pair where the color corresponds to the source of the sentence pair. Observe that the plurality of pairs fall along the diagonal; however, there are also a large number of pairs along the top-left side of this diagonal. This suggests that even though STR-2021 has pairs where the relatedness increases linearly with the amount of word overlap, there are also a number of pairs where a small amount of word overlap

results in substantial amount of relatedness. The sparse bottom-right side of the plot indicates that it is rare for there to be substantial word overlap, and yet very low relatedness. On average, occurrence of related words across a sentence pair leads to slightly higher relatedness scores than lexical overlap (row Q2).

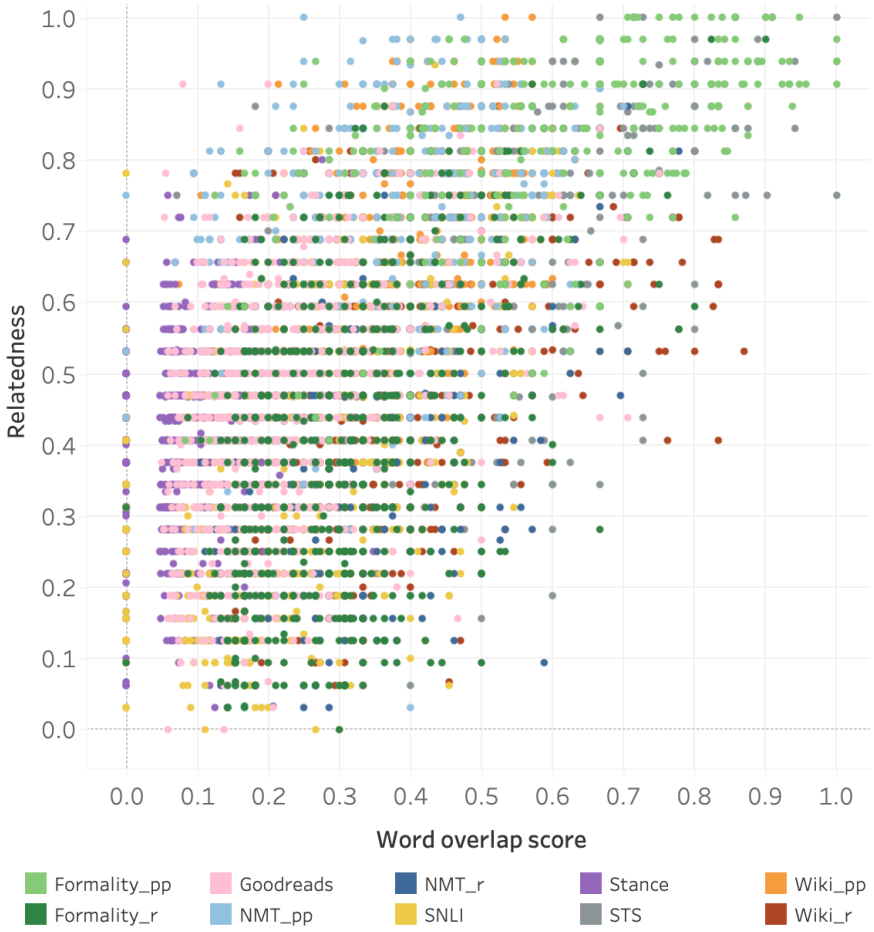

Figure 6.6.1: Scatter plot showing the relationship between lexical overlap and semantic relatedness of sentence pairs. Each dot in the plot is a sentence pair and the color of the dot represents the source from which the sentence pair is sampled.

The Q3a rows in Table 6.6.1 show correlations for related tokens of a given part of speech.[7] (The rows are in order from highest to lowest correlation.) Observe that proper nouns (PROPN) and nouns have the highest numbers. It is somewhat surprising that related verbs do not contribute greatly to semantic relatedness; they have similar correlations as pronouns and adverbs, and markedly lower than adjectives and nouns. Not surprisingly, determiners (DET) are at the lower end of weak correlation.

[7]Only those POS tags that occur in both sentences of a pair in more than 10% of the pairs are considered (>550 pairs).

The Q3b rows show correlations of coarse POS categories: NOUN Group (NOUN, PRON, PROPN), VERB Group (VERB, AUX), and ADJ Group (ADJ, ADP, ADV). We see that presence of related nouns in a sentence pair impacts semantic relatedness much more than any other POS group.

Since related nouns were found to be especially important, we also wanted to determine what impacts overall relatedness more: the presence of related nouns in the subject position or in the object position. Q4 rows show that, on average, related objects lead to markedly higher sentence-pair relatedness than related subjects.

Effect of Relatedness on Correlation

	Spearman		
Question	0–1 pairs	< 0.5 pairs	≥ 0.5 pairs
Q1. Lexical overlap	0.57	0.14	0.52
Q2. Related words - All	0.61	0.14	0.50
Q3a. Related words - per POS			
PROPN	0.50	0.34	0.26
NOUN	0.45	0.18	0.37
ADJ	0.36	0.04	0.35
VERB	0.31	0.03	0.31
PRON	0.30	0.01	0.30
ADV	0.28	0.04	0.35
AUX	0.25	0.03	0.20
ADP	0.23	0.07	0.22
DET	0.20	0.03	0.19
Q3b. Related words - per POS group			
Noun Group	0.60	0.34	0.41
Verb Group	0.32	0.09	0.29
ADJ Group	0.29	0.04	0.32
Q4. Related Subjects and Objects			
Subject	0.29	0.00	0.32
Object	0.43	0.14	0.33

Table 6.6.2: Correlation between features and the relatedness of sentence pairs in STR-2021 when considering full relatedness range (0–1), only the pairs with relatedness < 0.5, and only the pairs with relatedness ≥ 0.5.
Note: The 0–1 pairs column was shown earlier in Table 6.6.1. It is repeated here for ease of comparison.

In order to examine whether lexical overlap and some POS are less or more relevant in low or high relatedness pairs, we repeated the experiment of Table 6.6.1, only for pairs

with relatedness scores < 0.5, and separately, only for pairs with scores ≥ 0.5. The results are presented in Table 6.6.2. Generally, for all measures, we observe that performance is worse on the < 0.5 relatedness pairs. We find that for the < 0.5 relatedness pairs, only the existence of related proper nouns across sentence pairs has moderate correlation with the semantic relatedness of sentences; the correlation is weak for nouns, and close to 0 for all other parts of speech. The notable importance of related proper nouns and nouns is likely because they indicate a common topic, person, or object being talked about in both sentences—making the two sentence pairs related. For the ≥ 0.5 relatedness pairs, the correlations are weak for most POS; highest for nouns; and the gap between nouns and adjectives, adverbs, and verbs is reduced. Lexical overlap in general has a much higher correlation for the ≥ 0.5 relatedness pairs than the < 0.5 pairs.

6.7 Evaluating Sentence Representation Models using STR-2021

Since STR-2021 captures a wide range of fine-grained relations that exist between sentence pairs, it is a valuable asset in evaluating automated sentence representation and embedding models. To evaluate both unsupervised sentence representation approaches and supervised embedding models we treat predicting semantic relatedness as a regression task. In this task, we will represent each sentence as a vector and then use the cosine similarity between the vectors as a prediction of their semantic relatedness. The Spearman correlation is used to measure the "goodness" of the relatedness predictions (and in turn the sentence representation).

The experiments below (unless otherwise specified) all involve 5-fold cross-validation (CV) on STR-2021. We report the average of the Spearman correlations across the folds. Note that even for models that do not require training (e.g., Dice score), to enable direct comparisons with supervised approaches, we evaluate their performance on each test fold independently and report the average of the correlations across folds.

6.7.1 Do Unsupervised Embeddings Capture Semantic Relatedness?

I first explore unsupervised approaches to sentence representation where the embedding of a sentence is derived from that of its constituent tokens. The token embedding can be of two types:

- **Static Word Embeddings:** I tested three popular models: Word2Vec (Mikolov et al.,

Model	Spearman
Baseline	
1. Lexical overlap (Dice)	0.57
Unsupervised, Static Embeddings	
2. Word2Vec (mean, Googlenews)	0.60
3. Word2Vec (max, Googlenews)	0.54
4. GloVe (mean, Common Crawl)	0.49
5. GloVe (max, Common Crawl)	0.56
6. GloVe (mean, 200_Twitter)	0.44
7. GloVe (max, 200_Twitter)	0.48
8. Fasttext (mean, Common crawl)	0.29
9. Fasttext (max, Common crawl)	0.24
Unsupervised, Contextual Embeddings	
10. BERT-base (mean)	0.58
11. BERT-base (max)	0.55
12. BERT-base (cls)	0.41
13. RoBERTa-base (mean)	0.48
14. RoBERTa-base (max)	0.47
15. RoBERTa-base (cls)	0.41
Supervised (Fine-tuning on portions of STR-2021)	
16. BERT-base (mean)	0.82
17. RoBERTa-base (mean)	0.83

Table 6.7.1: Average correlation between human annotated relatedness of sentence pairs and the cosine distance between their embeddings across the CV runs.

2013b), GloVe (Pennington et al., 2014), and Fasttext (Grave et al., 2018).

- **Contextual Word Embeddings:** We tested pretrained contextual embeddings from BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). We use the BERT-base-uncased and RoBERTa-base models from the HuggingFace library.[8]

We obtain sentence embeddings by both mean-pooling and max-pooling the token embeddings from the final layer. For the contextual embeddings, we also explore using the embedding of the classification token ([CLS]).

Table 6.7.1 presents the results. As baseline, we include how well simple lexical overlap (Dice score) predicts relatedness (row 1). Observe that mean-pooling with word2vec (row 2) obtains slightly higher correlation than the baseline, but the majority of the static embedding models fail to obtain better correlations (rows 3–9). The contextual embeddings from BERT and RoBERTa do not perform better than the word2vec embeddings (rows 10–15). Overall, the unsupervised methods leave much room for improvement.

[8] https://huggingface.co

6.7.2 Do Supervised Embeddings Capture Semantic Relatedness?

We now evaluate the performance of BERT-based models on STR-2021 when formulated as a *supervised* regression task. We use the S-BERT cross-encoder framework of Reimers and Gurevych (2019) and apply mean-pooling on top of the token embeddings of the final layer to obtain sentence embeddings. The model is trained using a cosine-similarity loss—the cosine between the embeddings of a sentence pair is compared to the gold semantic relatedness scores to obtain the Mean Squared Error (MSE) loss for each datapoint.

Table 6.7.1 rows 16 and 17 show the results: fine-tuning on STR-2021 leads to considerably better relatedness scores.

Impact of Domain on Fine-Tuning

The results above show that fine-tuning is critical for better sentence representation. However, it is well-documented that the domain of the data can have substantial impact on results, especially when quite different from the training data. With the inclusion of data from various domains in STR-2021 (Table 6.3.1), one can systematically explore performance on individual domains, as well as the extent to which performance may drop if no training data from the target domain is included for training.

Table 6.7.2 shows these results. The RoBERTA CV column shows a breakdown of results on sentence pairs from each source (domain). Essentially, these are results for the scenario where some portion of in-domain data is included in the training folds (along with data from other domains), and the system correlations are determined only on the test fold's target domain pairs. Observe that performance on most domains is comparable to each other, except for the stance domain where correlations are much lower.[9]

The LOO CV column shows correlations with a leave-one-out cross-validation setup: no in-domain training data is used and system correlations are determined only for the target domain pairs. Observe that this leads to drops in scores for all domains except STS. However, the drop is small; and scores are still much higher than the lexical overlap (Dice CV) baseline. This suggests that the diversity of data in the remaining subsets is useful in overcoming a lack of in-domain training data.

[9]The stance subset has a smaller range of relatedness scores than other subsets, and lower range is known to lead to lower correlations. Thus, its correlations are not directly comparable to those of the other subsets. (See: https://www.statisticshowto.com/restricted-range/)

	Dice	**SBERT(RoBERTa)**	
	CV	CV	LOO CV
STS	0.60	0.79	0.82
SNLI	0.53	0.80	0.77
Stance	0.20	0.49	0.39
Goodreads	0.44	0.73	0.70
Wiki	0.48	0.79	0.75
Formality	0.69	0.86	0.83
ParaNMT	0.44	0.80	0.79

Table 6.7.2: Breakdown of average test-fold correlations for each source: (a) using lexical overlap (Dice), (b) using SBERT and some in-domain data for fine-tuning (in addition to data from other domains), and (c) using SBERT and only out-of-domain data for fine-tuning (LOO CV). CV: cross-validation. LOO: leave-one-out.

6.8 Conclusion

In this work, we created STR-2021, the first dataset of English sentence pairs annotated with fine-grained relatedness scores. I used a comparative annotation method that produced a split-half reliability of 0.84 — showing that speakers of a language can reliably judge semantic relatedness. We used the dataset to explore several research questions pertaining to what makes two sentences more related.

Notably, we showed that word overlap and presence of related words are at the lower end of what would be considered as moderate correlation. Also, on average, occurrence of related proper nouns and nouns across a sentence pair increases their relatedness the most, compared to other parts of speech. Finally, we used STR-2021 to evaluate the ability of sentence representation methods to embed sentences in vector spaces such that those that are closer to each other in meaning are also closer in the vector space.

The dataset, STR-2021, is freely available to foster further research in semantic relatedness and sentence representation. There are two main lines of future work currently being explored: 1) creating STR datasets for more languages (to benefit NLP work in these languages and to explore broader trends in semantic relatedness) and 2) exploring the usefulness of STR-2021 in downstream applications such as quantifying emotion granularity.[10]

6.9 Discussion

There are various limitations and points of discussion related to this work.

[10]Emotion granularity, or relatedness of usages of different emotion words, has been shown to be predictive of health outcomes (Barrett, 2004; Kimhy et al., 2014).

First, the dataset is not representative of the entire English language. Although we sampled English sentences from a diverse array of sources from the internet, with a focus on social media, it is likely (almost certain) that several types of sentences (and several demographic groups) are not well-represented in STR-2021. The dataset likely includes more sentences by people from the United States and Europe and with a socio-economic and educational backgrounds that allow for social media access. This point can be addressed with gathering additional data from different sources; however, this is costly as annotation is very costly.

Related to this, we highlight that any sort of annotation will capture the beliefs of the humans performing the annotation. These biases may be systematically different for different socio-cultural groups. Our data was annotated by US annotators, but even within the US there are diverse socio-cultural groups. For example, one may have race or gender-related biases that may percolate subtly into one's notions of how related two units of text are. Our dataset curation was careful to avoid sentences from problematic sources, and we have not seen any inappropriate relatedness judgments, but it is possible that some subtle inappropriate biases still remain. Thus, as with any approach for sentence representation or semantic relatedness, we caution users to explicitly check for such biases in their system regardless of whether they use STR-2021.

We stress that we have not actually demonstrated such a bias when it comes to interpreting relatedness and demonstrating the manifestation of biases in the annotations of semantic relatedness remains a very interesting and open research problem.

On a higher level, the goal of creating this dataset was to identify common perceptions of semantic relatedness and demonstrate how comparative annotations can be used for this task. The resulting annotations (i.e., rankings of sentences) are not meant to be "correct" or "right" answers, but rather what the majority of the annotators believe based on their intuitions of the English language. Reasonable people may have well-founded disagreements about certain rankings without either being clearly wrong.

We would also like to highlight that the absolute values of the relatedness scores themselves have no meaning. The scores help order sentence pairs relative to each other. For example, a pair with a higher relatedness score should be considered more related than a pair with a lower score. No claim is made that the mid-point (relatedness score of 0.5) separates related words from unrelated words. One may determine categories such as *related* or *unrelated* by finding thresholds of relatedness scores optimal for their use/task.

Additionally, the relatedness scores do not indicate an inherent unchangeable attribute. The relatedness can change with time, but the dataset entries are largely fixed. They pertain to the time they are created.

6.9.1 Relation to Clinical Author Obfuscation

Recall that the purpose of creating STR-2021 and performing these analyses was to help inform those creating author attribution methods. In this respect, we have learned useful information.

The first and most critical observation is that simply averaging unsupervised embeddings (either traditional or contextual) will not result in very meaningfully correlation with human notions of relatedness. This is especially the case for sentence pairs that are on the lower end of the relatedness spectrum. This observation then has ramifications both for how one should go about evaluating their models, but also where such methods would not be appropriate.

We observed that supervised approaches to automatically measuring relatedness seem to perform much better. This is a good sign; however, it is unlikely that any models trained on STR-2021 will be appropriate to apply to clinical texts (due to the very large difference in vocabulary and background knowledge required). Here, future work could create their own annotation on clinical texts. To by-pass data privacy concerns it is possible for them to use publicly published case reports (Flamholz et al., 2022). However, if they are seeking to capture medical knowledge as part of relatedness (e.g., if a pair of medications are more similar than a differing pair), then annotation will be very costly as their annotator pool will have more education and thus likely require more compensation. The methodology used in this chapter can also be used to quantify the change in meaning resulting from the application of RaNNA.

Part III

Summary

Chapter 7

Conclusion

7.1 Summary

In this thesis, I set out to demonstrate the possibility and utility of unsupervised methods for the de-identification of clinical notes. First, to motivate the need for unsupervised approaches to clinical de-identification, I presented various limitations that are fundamental to search-based (i.e., supervised) approaches in Chapter 2. These limitations cannot be solved by incremental improvements to current search-based approaches. In the same chapter, I also demonstrated a novel methodology for assessing the risk of releasing traditional word embeddings that have been trained on clinical notes and secured using search-and-remove approaches.

Following this motivation, in Chapter 3, I introduced the first unsupervised clinical de-identification method: Random Nearest Neighbour Anonymization (RaNNA), a method which replaces all tokens in a note by randomly sampling from nearest neighbours in an embedding space. Following this, in Chapter 4, I developed a methodology to evaluate the risk associated with releasing traditional word embeddings trained on text secured using RaNNA. I also extended an existing risk assessment framework for the release of clinical notes to incorporate the risks associated with using RaNNA.

In the second part of this thesis, I advocated for the expansion of the scope of the goals of clinical de-identification techniques. In Chapter 5, I demonstrated that author attribution is possible on clinical notes and that existing clinical de-identification methods are not sufficient to protect author identity. In Chapter 6, I developed a semantic relatedness dataset using a comparative annotation schema. This dataset is meant to facilitate the creation of automated natural language generation processes to perform author obfuscation.

7.2 Discussion

Discussions surrounding the specifics of individual analyses or methods can be found at the end of each chapter in which the analyses are performed or the model is introduced. In this section, I will focus the discussion on higher-level analysis of the work. To do this, I will split the discussion into two parts. The first part, sub-section 7.2.1, will briefly discuss 'obvious' follow-ups to the work presented in this thesis. Despite being obvious, the ideas discussed will require significant effort and time to achieve. The second part, sub-section 7.2.2, will discuss clinical de-identification more broadly, whether this thesis has been successful, and what future research problems have been uncovered.

7.2.1 Low-Hanging Fruit

Most of the development and analysis in this thesis has used traditional word embeddings. With the advent of contextual word embedding models, a clear next step is to adjust much of the work to use these new technologies. However, this adjustment cannot be done haphazardly. For example, if the replacement of an implementation of RaNNA is to use contextual word embedding models, one needs to ask what is the context being used? What is the effect of adding context to word embeddings, both for performance and security?

Using contextual word embeddings will also require new thought about attack vectors. The methodology for evaluating the risk of releasing traditional word embeddings is likely not sufficient to attack contextual word embedding models. There are probably novel attack vectors for these sorts of models that would have to be developed independently of this work. Uncovering, documenting, and quantifying these attack vectors is a significant task left for future research.

7.2.2 Higher Aims

Initial Objective

The initial objective of the work in this thesis was to improve the methods used in the de-identification of clinical notes. To improve existing methods, I argued for the adoption of unsupervised approaches to de-identification. While this can increase the security of patient information and reduce the cost of de-identification for data holders, there is a significant reduction in the readability of the notes. This reduction in readability limits the uses of the data after de-identification, however there are many instances where this reduction is inconsequential (i.e., many of the tasks highlighted in Chapter 3). To better address issues where human readability is vital, it may be possible to alter the functionality

of RaNNA, e.g., reducing the obfuscation parameter to reduce noise, creating a dictionary of information words such as stop words that are not switched with other words, among other options. However, each of these changes does increase the risk of re-identification.

Subject vs Author De-identification

In this work, I was the first to demonstrate that author attribution is possible on clinical notes. The possibility of author attribution necessitates the development of author de-identification (i.e., author obfuscation). In the development of author de-identification there are two questions:

1. Should subject and author de-identification be treated as two different problems?

Currently, the work on subject de-identification is completely separate from the work on author de-identification. A first obvious step for the field is to evaluate new systems for both author and subject de-identification. However, theoretically, subject and author identification are two sides of the same coin; information on subjects can help narrow down authors and vice versa. As such, it is worthwhile to ask whether it makes sense for future researchers to frame these two tasks as a singular task. While the evaluations can remain separate, approaches that seek to remove both author and subject information at once will need to be fundamentally different than those that remove such information in sequence (i.e., as is currently possible).

2. Is it possible to have unsupervised author de-identification?

Regardless of the framing, it is interesting to consider if it is possible to frame author de-identification as an unsupervised task. While I cannot see a way of performing author de-identification without using trained models of some sort, it may be possible to develop a model that does not need to know which authors are present in the dataset being released. Conceptually, the development of a natural language generation model that forces all output to be in the style of a single author would satisfy both the requirements of author obfuscation and remain de-identified. However, it is not yet clear (in a technical sense) how such a model can be developed.

Alternative Solutions

In this work, we have implicitly assumed that the most effective way of reducing the risk to patients' confidentiality while enabling beneficial research (or access to free-text clinical notes) is to actively de-identify the data. However, there are other ways to share and access

data without having to perform de-identification. Here, we will discuss some of these alternate solutions, when they are useful, and what their limitations are.

1. Case Reports

To bypass the need for de-identifying data before creating word embeddings, Flamholz et al. (2022) propose using published case reports as the source of data. In healthcare, clinical case reports are detailed reports of medical encounters between a patient and the healthcare system. Such reports have long been an integral part of the medical literature, and play a vital role in educating healthcare practitioners and assisting researchers to detect novelties and generate hypotheses (Nissen and Wynn, 2014).

Unlike many publicly released clinical datasets, published case reports often require written and informed consent from patients (Nissen and Wynn, 2014). Despite undergoing manual de-identification, there is still a (small) risk for the re-identification of published case reports (Branson et al., 2020). However, given their public availability and informed consent, training word embeddings on such text presents no *additional* risk to those creating the embeddings.

Using case reports to train embeddings is a very clever solution to the issue of patient confidentiality; however, this approach has various limitations. First, the availability of case reports is not evenly distributed across all languages. For example, on PubMed there are 1,788,016 English case reports[1], 95,158 French case reports[2], and only 2 Arabic case reports[3]. As such, this approach will only work for the English language.

However, even for research done in English, the case reports are not evenly distributed across different medical specialities. For example, there are 53,041 English case reports about cardiology[4] but only 28,119 English case reports about urology[5]. Researchers developing embeddings specialized for a certain area of application may still need to gather their own data which will require de-identification.

2. Centralized Environments

Rather than providing de-identified data to researchers 'in-the-wild', there has been a

[1] https://pubmed.ncbi.nlm.nih.gov/?term=%28English%5BLanguage%5D%29&filter=pubt.casereports

[2] https://pubmed.ncbi.nlm.nih.gov/?term=%28French%5BLanguage%5D%29&filter=pubt.casereports

[3] https://pubmed.ncbi.nlm.nih.gov/?term=%28Arabic%5BLanguage%5D%29&filter=pubt.casereports

[4] https://pubmed.ncbi.nlm.nih.gov/?term=%28English%5BLanguage%5D%29+cardiology&filter=pubt.casereports

[5] https://pubmed.ncbi.nlm.nih.gov/?term=%28English%5BLanguage%5D%29+urology&filter=pubt.casereports

recent trend to host data locally in a compute cluster and provide researchers access to data only through that cluster. This approach has long been used by ICES in Toronto and has recently been adopted by the Nightingale Open Science Project[6]. Having researchers work in an environment controlled by the data holder provides the data holder with greater control. They can better enforce their policies as well as track for data egress, thus minimizing the possible damage caused by a third party.

However, these solutions often still use de-identification; it is generally regarded as best practice to de-identify data used by internal and trusted researchers. This increases the cost, as institutions now have to host the data as well as the computational resources to enable research. This increased cost is the greatest limitation of such an approach. It is difficult to scale such solutions to an international scale. For example, MIMIC-III (Johnson et al., 2016) has over 3000 citations. If we assume that each citing paper accessed the data, and each was single authored[7], it would be extremely costly to host the computation for 3000 researchers.

3. Homomorphic Encryption

Homomorphic encryption is a form of encryption that allows computation on ciphertexts (Naehrig et al., 2011; Zhou and Wornell, 2014). This means that it would be theoretically possible for a third party to train models on data without being able to see it, thereby negating the security risk associated with sharing sensitive information. However, I have not yet seen research on this topic (i.e., applied to clinical notes). Generally speaking, this field of research is still in its infancy and, as of this publication (early 2022), there have not been any meaningful demonstrated applications of homomorphic encryption in an applied NLP setting.

4. Differential Privacy

Unlike the literature studying homomorphic encryption, there is a mature body of work regarding differential privacy. Differential privacy is a popular novel privacy scheme that makes no assumptions about the attacker and works to ensure that the result of any query is not changed *too much* by the addition or removal of any single person's record in a database (Dankar and El Emam, 2013). Differential privacy has been used by the US Census Bureau (Abowd, 2018).

There are multiple competing definitions for the optimal formulation for differential privacy. Recent work has challenged the traditional formulation (i.e., requiring that the

[6] https://www.nightingalescience.org/about
[7] Almost certainly a dramatic under-count

presence or absence of any individual in the dataset to not affect results) as being too strong (Dwork and Rothblum, 2016; Soria-Comas et al., 2017; Wang, 2019). However, this work remains largely theoretical, with no direct application to readable texts.

There has been some work attempting to apply differential privacy on text. Yue et al. (2021), argue that existing literature that focuses on generating private numeric representations for text (e.g., through document-level features) (Weggenmann and Kerschbaum, 2018; Li et al., 2018) fails at being explainable or human-readable. To address this, they present an approach that is similar to our own RaNNA. For each document they run a:

> common text sanitization mechanism M over [a document] D on local devices. Specifically, M works by replacing every token x_i in D with a substitution $y_i \in V$, assuming that x_i itself is unnecessary for NLP tasks while its semantics should be preserved for high utility.

This mechanism is supposed to change the probability that sanitized text can be linked to the sensitive token. The authors run a variety of utility experiments, but very few privacy experiments. This is one of the biggest limitations of this work. It is unclear whether the assumption that all we care about is sensitive tokens is enough to truly make something private. There will likely be other textual features that also affect the privacy of the note that are not addressed in the privacy formulation of this algorithm (e.g., grammar, syntax, phrasing, writing styles). It's also unclear whether applying differential privacy to remove sensitive tokens would be enough to also hide author (i.e., doctor) identity. If we accept the focus of the work on solely removing subject information (i.e., the patients in clinical notes), it is still unclear if the traditional settings of differential privacy techniques (i.e., the hyper-parameters usually experimented with such as *epsilon*) correspond to traditionally accepted risk as measured by traditional classification metrics. This limitation or constraint could have been addressed by extending the evaluation framework of Scaiano et al. (2016).

On Risk: Cyber-attacks and Achieving Perfection

Throughout this thesis, I've implicitly assumed that aiming for 100% sensitivity was the goal of de-identification. This is an assumption commonly shared in the field. However, this assumption does not bear scrutiny. While it is vital to aim for perfection, when deciding to use de-identification algorithms it is also vital to take a risk-based trade-off approach. There are many benefits that come from granting researchers access to data, and while such access should be done safely with proper vetting and access controls, requiring perfection in de-identification will hamper research and thus eventually patient care.

I think it would be more fruitful to compare the risk of re-identification of secured clinical data against the risk of hackers attacking a healthcare database. Cyber-attacks on healthcare institutions have been steadily on the rise (Martin et al., 2017; Chigada and Madzinga, 2021). As such, rather than requiring perfection in de-identification before sharing, healthcare institutions should instead require that de-identified data be no more at risk, than the risk that data is exposed to simply sitting on the servers of healthcare institutions.

The risk-assessment of RaNNA performed in this paper was relative to other forms of de-identification. Future research should aim to ground this risk against the risk of cyber-attacks (which would have to be quantified as well). This area of work can help data holders better understand the risks associated with producing and storing vs de-identifying and sharing data and assist policy makers in producing meaningful legislation.

Other Applications of RaNNA

In this work, we demonstrated how unsupervised de-identification can be used to secure free-text clinical notes. However, the idea behind RaNNA (i.e., random replacement using nearest neighbours in an embedding space) could also be used to secure other types of data which can be represented using embeddings. More specifically, future work can explore applying RaNNA to secure audio. To do this consider the following steps: 1) a specific piece of audio can be split into small segments (e.g., 0.5 second splits), 2) the audio is passed through auto-encoders to create an embedding, 3) each segment is replaced by randomly replaced by other segments which have embeddings similar to the segment in question. This approach would allow entire pieces of audio from speaker X to be re-built using segments of many other speakers. This would hide the identity of the speaker. However, much experimentation and development is required to verify that this would actually work as expected.

Bibliography

Mohamed Abdalla and Moustafa Abdalla. The grey hoodie project: Big Tobacco, Big Tech, and the threat on academic integrity. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 287–297, 2021.

Mohamed Abdalla, Magnus Sahlgren, and Graeme Hirst. Enriching word embeddings with a regressor instead of labeled corpora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6188–6195, 2019.

Mohamed Abdalla, Moustafa Abdalla, Graeme Hirst, and Frank Rudzicz. Exploring the privacy-preserving properties of word embeddings: Algorithmic validation study. *Journal of Medical Internet Research*, 22(7):e18055, 2020a.

Mohamed Abdalla, Moustafa Abdalla, Frank Rudzicz, and Graeme Hirst. Using word embeddings to improve the privacy of clinical notes. *Journal of the American Medical Informatics Association*, 27(6):901–907, 2020b.

Mohamed Abdalla, Hong Lu, Bogdan Pinzaru, and Liisa Jaakkimainen. Using machine learning to measure specialist wait times from family physicians' electronic medical records linked to Ontario health administrative data. *International Journal of Population Data Science*, 5(5), 2020c.

Moustafa Abdalla, Danh Tran-Thanh, Juan Moreno, Vladimir Iakovlev, Ranju Nair, Nisha Kanwar, Mohamed Abdalla, Jennifer PY Lee, Jennifer Yin Yee Kwan, Thomas R Cawthorn, et al. Mapping genomic and transcriptomic alterations spatially in epithelial cells adjacent to human breast carcinoma. *Nature Communications*, 8(1):1–11, 2017.

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, 2010.

John M Abowd. The US Census Bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2867–2867, 2018.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. SemEval-2012 Task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, 2012.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, 2013.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. SemEval-2014 Task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, 2014.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. SemEval-2015 Task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, 2015.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez Agirre, Rada Mihalcea, German Rigau Claramunt, and Janyce Wiebe. SemEval-2016 Task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.* ACL (Association for Computational Linguistics), 2016.

Tanbir Ahmed, Md Momin Al Aziz, and Noman Mohammed. De-identification of electronic health record using neural network. *Scientific Reports*, 10(1):1–11, 2020.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, 2019.

Shima Asaadi, Saif Mohammad, and Svetlana Kiritchenko. Big BiRD: A large, fine-grained, bigram relatedness dataset for examining semantic composition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 505–516, 2019.

Lisa Feldman Barrett. Feelings or words? understanding the content in self-report ratings of experienced emotion. *Journal of Personality and Social Psychology*, 87(2):266, 2004.

Bruce A Beckwith, Rajeshwarri Mahaadevan, Ulysses J Balis, and Frank Kuo. Development and evaluation of an open source software tool for deidentification of pathology reports. *BMC Medical Informatics and Decision Making*, 6(1):12, 2006.

Haohan Bo, Steven HH Ding, Benjamin Fung, and Farkhund Iqbal. ER-AE: differentially-private text generation for authorship anonymization. *arXiv preprint arXiv:1907.08736*, 2019.

Victoria Bobicev, Marina Sokolova, Khaled El Emam, and Stan Matwin. Authorship attribution in health forums. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 74–82, 2013.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

Janice Branson, Nathan Good, Jung-Wei Chen, Will Monge, Christian Probst, and Khaled El Emam. Evaluating the re-identification risk of a clinical study report anonymized under EMA Policy 0070 and Health Canada regulations. *Trials*, 21(1):1–9, 2020.

Federico Cabitza, Raffaele Rasoini, and Gian Franco Gensini. Unintended consequences of machine learning in medicine. *JAMA*, 318(6):517–518, 2017.

David Carrell, Bradley Malin, John Aberdeen, Samuel Bayer, Cheryl Clark, Ben Wellner, and Lynette Hirschman. Hiding in plain sight: Use of realistic surrogates to reduce exposure of protected health information in clinical text. *Journal of the American Medical Informatics Association*, 20(2):342–348, 2013.

Feng Chang and Nishi Gupta. Progress in electronic medical record adoption in canada. *Canadian Family Physician*, 61(12):1076–1084, 2015.

Joel Chigada and Rujeko Madzinga. Cyberattacks and threats during COVID-19: A systematic literature review. *South African Journal of Information Management*, 23(1): 1–11, 2021.

Council of European Union. General data protection regulation (gdpr) article 4 definitions, 2016. URL https://gdpr.eu/article-4-definitions/.

Lee J Cronbach. Coefficient alpha and the internal structure of tests. *Psychometrika*, 16 (3):297–334, 1951.

Fida Kamal Dankar and Khaled El Emam. Practicing differential privacy in health care: A review. *Transactions on Data Privacy*, 6(1):35–67, 2013.

Herbert Aron David. The method of paired comparisons. In *Proceedings of the Fifth Conference on the Design of Experiments in Army Research Developments and Testing*, 1963.

Peter de Vocht. Python Package: Subject Verb Object Extractor. Github, 2020. URL https://github.com/peter3125/enhanced-subject-verb-object-extraction.

Azad Dehghan, Aleksandar Kovacevic, George Karystianis, John A Keane, and Goran Nenadic. Combining knowledge-and data-driven methods for de-identification of clinical narratives. *Journal of Biomedical Informatics*, 58:S53–S59, 2015.

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, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

Margaret Douglass, Gari D Clifford, Andrew Reisner, George B Moody, and Roger G Mark. Computer-assisted de-identification of free text in the MIMIC II database. In *Computers in Cardiology*, pages 341–344. IEEE, 2004.

Cynthia Dwork and Guy N Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016.

Khaled El Emam and Luk Arbuckle. *Anonymizing health data: case studies and methods to get you started*. " O'Reilly Media, Inc.", 2013.

Khaled El Emam, Elizabeth Jonker, Luk Arbuckle, and Bradley Malin. A systematic review of re-identification attacks on health data. *PloS ONE*, 6(12):e28071, 2011.

Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.

David Evans. Google federated privacy 2019: The dragon in the room, 2019. URL https://uvasrg.github.io/google-federated-privacy-2019-the-dragon-in-the-room/.

Amir Feder, Danny Vainstein, Roni Rosenfeld, Tzvika Hartman, Avinatan Hassidim, and Yossi Matias. Active deep learning to detect demographic traits in free-form clinical notes. *Journal of Biomedical Informatics*, 107:103436, 2020. Publisher: Elsevier.

Oscar Ferrández, Brett R South, Shuying Shen, F Jeffrey Friedlin, Matthew H Samore, and Stéphane M Meystre. BoB, a best-of-breed automated text de-identification system for VHA clinical documents. *Journal of the American Medical Informatics Association*, 20 (1):77–83, 2013.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*, pages 406–414, 2001.

Zachary N Flamholz, Andrew Crane-Droesch, Lyle H Ungar, and Gary E Weissman. Word embeddings trained on published case reports are lightweight, effective for clinical tasks, and free of protected health information. *Journal of Biomedical Informatics*, 125: 103971, 2022.

Terry N Flynn and Anthony AJ Marley. Best-worst scaling: Theory and methods. In *Handbook of Choice Modelling*. Edward Elgar Publishing, 2014.

Simson L Garfinkel et al. De-identification of personal information. *National Institute of Standards and Technology*, 2015.

Édouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomáš Mikolov. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman Group, 1976.

Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. *Semantic similarity from natural language and ontology analysis*. Morgan & Claypool Publishers, 2015.

Lukáš Havrlant and Vladik Kreinovich. A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation). *International Journal of General Systems*, 46(1):27–36, 2017.

Health Insurance Portability and Accountability Act. Guidance regarding methods for de-identification of protected health information in accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule. 2012a. URL https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html.

Health Insurance Portability and Accountability Act. Guidance regarding methods for de-identification of protected health information in accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule. 2012b. URL https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html.

Lynette Hirschman and John Aberdeen. Measuring risk and information preservation: Toward new metrics for de-identification of clinical texts. In *Proceedings of the NAACL HLT 2010 Second LOUHI Workshop on Text and Data Mining of Health Documents*, pages 72–75, 2010.

Angelos Hliaoutakis. Semantic similarity measures in MeSH ontology and their application to information retrieval on Medline. *Master's thesis*, 2005.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, pages 65–70, 1979.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. Python Package; spaCy: Industrial-strength Natural Language Processing in Python. Zenodo, 2020. URL https://doi.org/10.5281/zenodo.1212303.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. ClinicalBERT: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

Information & Privacy Commissioner of Ontario, CHEO Research Institute, and University of Ottawa. Dispelling the myths surrounding de-identification: Anonymization remains a strong tool for protecting privacy. *Information and Privacy Commissioner of Ontario*, 2011.

ISO-25237. Health informatics — Pseudonymization. Standard, International Organization for Standardization, Geneva, CH, December 2008.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, 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):1–9, 2016.

Michael J Joyner, Nigel Paneth, and John PA Ioannidis. What happens when underperforming big ideas in research become entrenched? *JAMA*, 316(13):1355–1356, 2016.

Katikapalli Subramanyam Kalyan and Sivanesan Sangeetha. SECNLP: A survey of embeddings in clinical natural language processing. *Journal of Biomedical Informatics*, 101:103323, 2020.

Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020.

Mehmet Kayaalp. Modes of de-identification. In *AMIA Annual Symposium Proceedings*, volume 2017, page 1044. American Medical Informatics Association, 2017.

Christopher J Kelly, Alan Karthikesalingam, Mustafa Suleyman, Greg Corrado, and Dominic King. Key challenges for delivering clinical impact with artificial intelligence. *BMC medicine*, 17(1):195, 2019.

Faiza Khan Khattak, Serena Jeblee, Chloé Pou-Prom, Mohamed Abdalla, Christopher Meaney, and Frank Rudzicz. A survey of word embeddings for clinical text. *Journal of Biomedical Informatics: X*, 4:100057, 2019.

Mahmoud Khonji and Youssef Iraqi. Evaluating author attribution on Emirati tweets. *IEEE Access*, 8:149531–149543, 2020.

Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

David Kimhy, Julia Vakhrusheva, Samira Khan, Rachel W Chang, Marie C Hansen, Jacob S Ballon, Dolores Malaspina, and James J Gross. Emotional granularity and social functioning in individuals with schizophrenia: An experience sampling study. *Journal of Psychiatric Research*, 53:141–148, 2014.

Svetlana Kiritchenko and Saif Mohammad. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470, 2017.

G Frederic Kuder and Marion W Richardson. The theory of the estimation of test reliability. *Psychometrika*, 2(3):151–160, 1937.

Siwei Lai, Kang Liu, Shizhu He, and Jun Zhao. How to generate a good word embedding. *IEEE Intelligent Systems*, 31(6):5–14, 2016.

Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron C Wallace. Does BERT pretrained on clinical notes reveal sensitive data? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 946–959, 2021.

Yitong Li, Timothy Baldwin, and Trevor Cohn. Towards robust and privacy-preserving text representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 25–30, 2018.

Yuhua Li, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150, 2006.

Daniyal Liaqat, Mohamed Abdalla, Pegah Abed-Esfahani, Moshe Gabel, Tatiana Son, Robert Wu, Andrea Gershon, Frank Rudzicz, and Eyal De Lara. WearBreathing: Real world respiratory rate monitoring using smartwatches. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2):1–22, 2019.

Zenobia Liendo, Guillaume De Roo, and Amitabha Karmakar. Classifying medical notes into standard disease codes. *Github.com*, 2019. URL https://github.com/zliendo/AI_MedicalNotes/blob/master/w266FinalReport_ICD_9_Classification.pdf.

Martin Lindvall and Jesper Molin. Designing for the long tail of machine learning. *arXiv preprint arXiv:2001.07455*, 2020.

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. *arXiv preprint arXiv:1907.11692*, 2019.

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, 2017.

Jordan J Louviere and George G Woodworth. Best-worst scaling: A model for the largest difference judgments. Technical report, Working paper, 1991.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human language Technologies*, pages 142–150, 2011.

Laura C Maclagan, Mohamed Abdalla, Daniel A Harris, Branson Chen, Elisa Candido, Richard H Swartz, Andrea Iaboni, Therese A Stukel, Liisa Jaakkimainen, and Susan E Bronskill. Using natural language processing to identify signs and symptoms of dementia and cognitive impairment in primary care electronic medical records (EMR). *Alzheimer's & Dementia*, 17:e054091, 2021.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, Roberto Zamparelli, et al. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2014)*, pages 216–223. Reykjavik, 2014.

Guy Martin, Paul Martin, Chris Hankin, Ara Darzi, and James Kinross. Cybersecurity and healthcare: How safe are we? *The British Medical Journal*, 358, 2017.

Andrew WE McDonald, Sadia Afroz, Aylin Caliskan, Ariel Stolerman, and Rachel Greenstadt. Use fewer instances of the letter "i": Toward writing style anonymization. In

International Symposium on Privacy Enhancing Technologies Symposium, pages 299–318. Springer, 2012.

Andrew WE McDonald, Jeffrey Ulman, Marc Barrowclift, and Rachel Greenstadt. Anonymouth revamped: Getting closer to stylometric anonymity. In *PETools: Workshop on Privacy Enhancing Tools*, volume 20, 2013.

Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg C Corrado, Ara Darzi, et al. International evaluation of an AI system for breast cancer screening. *Nature*, 577 (7788):89–94, 2020.

Stephane Meystre, Shuying Shen, Deborah Hofmann, and Adi Gundlapalli. Can physicians recognize their own patients in de-identified notes? In *e-Health–For Continuity of Care*, pages 778–782. IOS Press, 2014.

Stephane M Meystre, F Jeffrey Friedlin, Brett R South, Shuying Shen, and Matthew H Samore. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC Medical Research Methodology*, 10(1):1–16, 2010.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013b.

George A Miller and Walter G Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991.

R Miller, JK Boitnott, and GW Moore. Web-based free-text query system for surgical pathology reports with automatic case deidentification. *Archives of Pathology & Laboratory Medicine*, 125:1011, 2001.

Saif Mohammad. *Measuring semantic distance using distributional profiles of concepts*. PhD thesis, University of Toronto, 2008.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. A dataset for detecting stance in tweets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3945–3952, 2016.

Jane Morris and Graeme Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48, 1991.

Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM workshop on Cloud Computing Security Workshop*, pages 113–124, 2011.

Ishna Neamatullah, Margaret M Douglass, H Lehman Li-wei, 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, 2008.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, 2019.

Trygve Nissen and Rolf Wynn. The clinical case report: A review of its merits and limitations. *BMC Research Notes*, 7(1):1–7, 2014.

Michel Oleynik, Amila Kugic, Zdenko Kasáč, and Markus Kreuzthaler. Evaluating shallow and deep learning strategies for the 2018 n2c2 shared task on clinical text classification. *Journal of the American Medical Informatics Association*, 26(11):1247–1254, 2019.

Bryan Orme. MaxDiff analysis: Simple counting, individual-level logit, and Hb. *Sawtooth Software*, 2009.

Serguei Pakhomov, Bridget McInnes, Terrence Adam, Ying Liu, Ted Pedersen, and Genevieve B Melton. Semantic similarity and relatedness between clinical terms: An experimental study. In *AMIA annual symposium proceedings*, volume 2010, page 572. American Medical Informatics Association, 2010.

Serguei VS Pakhomov, Ted Pedersen, Bridget McInnes, Genevieve B Melton, Alexander Ruggieri, and Christopher G Chute. Towards a framework for developing semantic relatedness reference standards. *Journal of Biomedical Informatics*, 44(2):251–265, 2011.

Jagadeesh Patchala and Raj Bhatnagar. Authorship attribution by consensus among multiple features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2766–2777, 2018.

Ted Pedersen, Serguei VS Pakhomov, Siddharth Patwardhan, and Christopher G Chute. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics*, 40(3):288–299, 2007.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. *arXiv preprint arXiv:1906.05474*, 2019.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

Karl R Popper. Science as falsification. *Conjectures and Refutations*, 1(1963):33–39, 1963.

Stanley Presser and Howard Schuman. *Questions and answers in attitude surveys: Experiments on question form, wording, and context*. SAGE Publications, Inc, 1996.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. A word at a time: Computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th International Conference on World Wide Web*, pages 337–346, 2011.

Sudha Rao and Joel Tetreault. Dear sir or madam, May i introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, 2018.

P Buddha Reddy, T Raghunadha Reddy, M Gopi Chand, and A Venkannababu. A new approach for authorship attribution. In *Information and Decision Sciences*, pages 1–9. Springer, 2018.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL http://arxiv.org/abs/1908.10084.

Peter Mark Roget. *Roget's thesaurus of English words and phrases...* TY Crowell Company, 1911.

Herbert Rubenstein and John B Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

Magnus Sahlgren. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1): 33–54, 2008.

Martin Scaiano, Grant Middleton, Luk Arbuckle, Varada Kolhatkar, Liam Peyton, Moira Dowling, Debbie S Gipson, and Khaled El Emam. A unified framework for evaluating the risk of re-identification of text de-identification tools. *Journal of Biomedical Informatics*, 63:174–183, 2016.

Adriaan MJ Schakel and Benjamin J Wilson. Measuring word significance using distributed representations of words. *arXiv preprint arXiv:1508.02297*, 2015.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. *arXiv preprint arXiv:1705.09655*, 2017.

Rakshith Shetty, Bernt Schiele, and Mario Fritz. A4NT: author attribute anonymity by adversarial training of neural machine translation. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1633–1650, 2018.

Timothy Shimeall and Jonathan Spring. *Introduction to information security: A strategic-based approach*. Newnes, 2013.

Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, 26(11):1297–1304, 2019.

Tawanda Carleton Sibanda. *Was the patient cured?: Understanding semantic categories and their relationship in patient records*. PhD thesis, Massachusetts Institute of Technology, 2006.

Jordi Soria-Comas, Josep Domingo-Ferrer, David Sánchez, and David Megías. Individual differential privacy: A utility-preserving formulation of differential privacy guarantees. *IEEE Transactions on Information Forensics and Security*, 12(6):1418–1429, 2017.

Jackson M Steinkamp, Taylor Pomeranz, Jason Adleberg, Charles E Kahn Jr, and Tessa S Cook. Evaluation of automated public de-identification tools on a corpus of radiology reports. *Radiology: Artificial Intelligence*, 2(6):e190137, 2020.

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, 2015.

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, 2017.

Latanya Sweeney. Replacing personally-identifying information in medical records, the Scrub system. In *Proceedings of the AMIA Annual Fall Symposium*, page 333. American Medical Informatics Association, 1996.

György Szarvas, Richárd Farkas, and Róbert Busa-Fekete. State-of-the-art anonymization of medical records using an iterative machine learning framework. *Journal of the American Medical Informatics Association*, 14(5):574–580, 2007.

Ricky K Taira, Alex AT Bui, and Hooshang Kangarloo. Identification of patient name references within medical documents using semantic selectional restrictions. In *Proceedings of the AMIA Symposium*, page 757. American Medical Informatics Association, 2002.

Jonathan Taylor and John Fenner. The challenge of clinical adoption—the insurmountable obstacle that will stop machine learning? *British Journal of Radiology Open*, 1, 2018.

Patricia Thaine and Gerald Penn. Reasoning about unstructured data de-identification. *Journal of Data Protection & Privacy*, 3(3):299–309, 2020.

Sean M Thomas, Burke Mamlin, Gunther Schadow, and Clement McDonald. A successful technique for removing names in pathology reports using an augmented search and replace method. In *Proceedings of the AMIA Symposium*, page 777. American Medical Informatics Association, 2002.

Louis L Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273, 1927.

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, 2007.

Özlem Uzuner, Tawanda C Sibanda, Yuan Luo, and Peter Szolovits. A de-identifier for medical discharge summaries. *Artificial Intelligence in Medicine*, 42(1):13–35, 2008.

Mengting Wan and Julian J. McAuley. Item recommendation on monotonic behavior chains. In Sole Pera, Michael D. Ekstrand, Xavier Amatriain, and John O'Donovan, editors, *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*, pages 86–94. ACM, 2018. doi: 10.1145/3240323.3240369. URL https://doi.org/10.1145/3240323.3240369.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. Fine-grained spoiler detection from large-scale review corpora. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2605–2610. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1248. URL https://doi.org/10.18653/v1/p19-1248.

Yanshan Wang, Sijia Liu, Naveed Afzal, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Paul Kingsbury, and Hongfang Liu. A comparison of word embeddings for the biomedical natural language processing. *Journal of Biomedical Informatics*, 87:12–20, 2018.

Yu-Xiang Wang. Per-instance differential privacy. *Journal of Privacy and Confidentiality*, 9(1), 2019.

Benjamin Weggenmann and Florian Kerschbaum. Syntf: Synthetic and differentially private term frequency vectors for privacy-preserving text mining. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 305–314, 2018.

John Wieting and Kevin Gimpel. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1042. URL https://aclanthology.org/P18-1042.

Shweta Yadav, Asif Ekbal, Sriparna Saha, and Pushpak Bhattacharyya. Deep learning architecture for patient data de-identification in clinical records. In *Proceedings of the Clinical Natural Language Processing Workshop (ClinicalNLP)*, pages 32–41, 2016.

Xi Yang, Tianchen Lyu, Qian Li, Chih-Yin Lee, Jiang Bian, William R Hogan, and Yonghui

Wu. A study of deep learning methods for de-identification of clinical notes in cross-institute settings. *BMC Medical Informatics and Decision Making*, 19(5):1–9, 2019.

Xiang Yue and Shuang Zhou. PHICON: Improving generalization of clinical text de-identification models via data augmentation. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, 2020.

Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman SM Chow. Differential privacy for text analytics via natural text sanitization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3853–3866, 2021.

Haoran Zhang, Elisa Candido, Andrew S Wilton, Raquel Duchen, Liisa Jaakkimainen, Walter Wodchis, and Quaid Morris. Identifying transitional high cost users from unstructured patient profiles written by primary care physicians. In *Pacific Symposium on Biocomputing 2020*, pages 127–138. World Scientific, 2019.

Hongchao Zhou and Gregory Wornell. Efficient homomorphic encryption on integer vectors and its applications. In *2014 Information Theory and Applications Workshop (ITA)*, pages 1–9. IEEE, 2014.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *arXiv e-prints*, pages arXiv–2103, 2021.

Guido Zuccon, Daniel Kotzur, Anthony Nguyen, and Anton Bergheim. De-identification of health records using Anonym: Effectiveness and robustness across datasets. *Artificial Intelligence in Medicine*, 61(3):145–151, 2014.