Scalable Natural Language De-identification based on Machine Learning Approaches

By

Muqun Li

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science

May 11, 2018

Nashville, Tennessee

Approved:

Bradley A. Malin, Ph.D.

Daniel Fabbri, Ph.D.

Douglas H. Fisher, Ph.D.

Yevgeniy Vorobeychik, Ph.D.

Lynette Hirschman, Ph.D.

Khaled El Emam, Ph.D.

ACKNOWLEDGMENTS

First and foremost, I am greatly indebted to my advisor Dr. Bradley Malin for being the best mentor I could ask for on the long and winding road to a Ph.D. An excellent example as he is, he has never failed to convey how to be a successful researcher. Most importantly, he has trusted me to become one myself (even when I was plagued with self-doubt), for which I am eternally grateful.

I would also like to acknowledge everyone else on my dissertation committee: Dr. Daniel Fabbri, Dr. Douglas Fisher, Dr. Yevgeniy Vorobeychik, Dr. Lynette Hirschman, and Dr. Khaled El Emam. In spite of their busy schedules, they generously carved out time and provided invaluable expertise and insights on this work.

My heartfelt gratitude also extends to Dr. Martin Scaiano, who contributed tremendously to this work. I am thankful for his ideas, encouragement, and mentorship during the final year before my Ph.D. completion.

I have been extremely fortunate to be surrounded by inspiring and kind people. Though I may not be able to include all the names, I would like to express my deep gratitude to the following: my TA supervisor Dr. Jerry Roth, who generously lent a helping hand during my time of need; my collaborators Dr. David Carrell, John Aberdeen, Ben Wellner, Sam Bayer, Jacqueline Kirby; my amazing colleagues from Health Information Privacy Laboratory Chao, Grayson, James, Steve, Wei, Weiyi, Wen, Yongtai, You, Zhijun, Zhiyu; my brilliant team members from Privacy Analytics Hazel, Mark, Colette, Michael, Andrew, Sarah, Niamh, Khaldoun, my sweet friends Lina, Yukun, Barbara, Coda, Giuseppe and Doris, Sam and Rose, Haley, Lijie, Man, Xianwen and Lingjun, Zhangshi, Berk, Chen, Sophia, Mabel, Xi, Ying, Siyi, Antong, Pei, and many more.

Special thanks go to Privacy Analytics for the immense support in my dissertation work for the past year. I am also especially grateful to Peking University for laying out a solid foundation for my Ph.D. research.

This work would not have been possible without the endless support from my family. I need to thank my parents and my grandparents for the love and encouragement throughout this entire journey. Moreover, I owe an enormous debt of gratitude to my husband Long Wang. Thank you for being the most awesome partner-in-crime with your wisdom, optimism, and patience. Above all, thank you for always letting me be me.

Appreciation is given to the National Library of Medicine (grants R01LM011366 and R01LM009989), the National Human Genome Research Institute (grants U01HG006385 and U01HG006378), and Team for Research in Ubiquitous Secure Technology (grant CCF-0424422) from the National Science Foundation for funding this work. Finally, I want to thank Starbucks at 2525 West End and everyone else that has helped me along the way.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

For the past several decades, electronic medical record (EMR) systems have been progressively adopted in multiple aspects of clinical care and healthcare endeavors, e.g., capturing the condition of patients, facilitate communication between healthcare providers [1, 2], and improve the quality of care [3] . Meanwhile, the importance of repurposing the data in such resources to enhance secondary use, such as public health [4, 5] and biomedical research [6, 7], has been increasingly acknowledged. To realize such programs on a large scale, it is critical to share EMR data with researchers within and beyond the healthcare organization (HCO) at which it was generated [8]. In certain instances, such as when research is sponsored by the National Institutes of Health, HCOs must have plans for sharing data [9]. However, concerns over the privacy rights of the corresponding patients [10, 11, 12] have been posed by the dissemination of such data and remain one of the primary challenges before data sharing by HCOs.

One of the most common approaches to mitigate the above concerns as recommended by the National Institutes of Health data sharing policy [9], is to ensure the removal of protected health information (PHI) in the data to be shared. This information includes explicit identifiers (e.g., patient names) and quasi-identifiers (e.g., dates). This sanitizing process, called de-identification, is usually according to a regulatory standard, such as that specified in the Privacy Rule of the Health Insurance Portability and Accountability Act of 1996, or HIPAA [13], or European Medicines Agency (EMA) policy 0070 [14].

It is readily apparent where potential identifiers reside (e.g., a column in a database table labeled as “Patient Name”) in structured data (e.g., diagnosis codes [15]) or semi-structured text [16](e.g., problem lists [17]), which makes such data relatively straightforward to de-identify. However, a substantial amount of information is documented only in the form of natural language (e.g., clinical narratives) [18], which has proven to be a great enabler of flexibility in clinical workflow [19] and decision support [20]. Trying to thoroughly identify the existence of sensitive information (i.e., achieve perfect recall) while leaving all instances of non-identifiers in place (i.e., achieve high precision) [21, 22, 23, 24, 25, 26] is improbable in practice in clinical narratives, no matter manually or automatically [27, 28].

Thus it is fair to presume that even an HCO would make a considerable investment in the natural language de-identification process, some level of identifier leaking is still possible or the shared data will hardly be useful due to redaction of non-identifiers.

It is important to recognize the above fact because evidence suggests that increasing the amount of training data provided to de-identification tools could lead to diminishing returns in improving recall and precision eventually.

Scalability is always an issue for an HCO to take into account when evaluating de-identification methods. Manual de-identification of clinical narratives can be costly, both in labor and time [29], which means to construct a corpus of more than a few hundred documents is almost infeasible. Since even moderate size health care institutions can annually accumulate millions of records in the EMR system, evidently this is not a scalable approach. This is where automated de-identification tools come into play. Such tools translate the task of de-identification into a natural language processing (NLP) problem to make the process more efficient and replicable. Roughly categorized there are three groups of automated systems [2]: 1) rules and dictionaries, 2) machine learning, and 3) a hybrid of the two.

Rule-based systems perform well when informed by local knowledge and hand-crafted rules [30, 31]. But such knowledge is not always easy to elicit and might require significant amount of time to gather, which leads to scalability and portability problems [26].

Contrarily, solutions based on machine learning tend to be more generalizable and robust [26]. For this type of approaches, de-identification models are *inferred* from numerous textual features automatically derived from annotated training data. Nevertheless, they demand a certain amount of manually-annotated narrative to start with, in order to inform the learning process, which could call the scalability into question.

Hybrid models, which strive to integrate the best of both rule-based and machine learning-based algorithms, can improve de-identification performance [25], but require not only local knowledge but also human annotated training data.

1.1 Problem Statement

This dissertation is primarily concerned with the scalability challenge in de-identification systems based on machine learning. We address this challenge by fulfilling three tasks in the

context of natural language de-identification. The overall architecture of this research is illustrated in Figure 1. Starting from a collection of unannotated natural language clinical data which is subject to the exploit of malicious attackers when shared, the ultimate aim of the system is to successfully identify and therefore protect the PHI in the dataset.

1.1.1 Task 1 - Game Based Model for Investment Allocation

The first task aims to answer the question of how to formalize the interaction between an HCO that shares its EMR data and a potential adversary that intends to exploit the private information in the shared dataset. Note that the HCO has access to limited budget and human help to start the machine learning based de-identification, which requires an optimized solution to make investments rather than simply exhausting the resource. In this dissertation we introduce a game theory based framework to model the cost and benefit relation between an HCO and an adversary in order to investigate the threat of PHI exposure and to minimize the expenditures of an HCO, namely, the amount of training data allocated for natural language de-identification. In our game model, we assume the adversary is capable of mimicking the HCO's strategy of de-identification, which enables the HCO to construct an adversarial model and simulate the behavior of the adversary before actually performing the de-identification process.

1.1.2 Task 2 - Document Clustering

The second aspect of the scalability challenge in this dissertation focuses on how to better utilize a given set of training data for machine learning based de-identification. This is based on the observation that for machine based de-identification systems, training and testing on the same document type (e.g., discharge summaries) of clinical narratives yield the best performance [32]. Yet the information of document types is not always readily available and may not always provide the best basis for grouping records due to heterogeneity in documentation practices. We then proposed and developed a feature extraction and clustering strategy to partition clinical documents into inferred types (categorized by writing complexity and clinical vocabulary usage) over which de-identification models are trained and tested. This clustering strategy is performed on both the gold standard data and the testing data after the amount of training resources needed for de-identification is determined by our game based framework in Task 1.

1.1.3 Task 3 - Active Learning

For the last part of the problem, we incorporate active learning [33] in machine learning de-identification and answer the question of whether utilizing active learning in the process of machine learning de-identification can yield better results than passive learning (i.e., randomly sampling documents for training) in terms of performance measures. The hypothesis is that if the machine learning de-identification system could actively request information that helps to create a better model, less training data will be needed to maintain (or even improve) the performance of trained models. As shown in Figure 1, after the clusters of training and testing documents are generated as in Task 2, the clusters are fed into the active learning framework as input to create the final output of protected data.

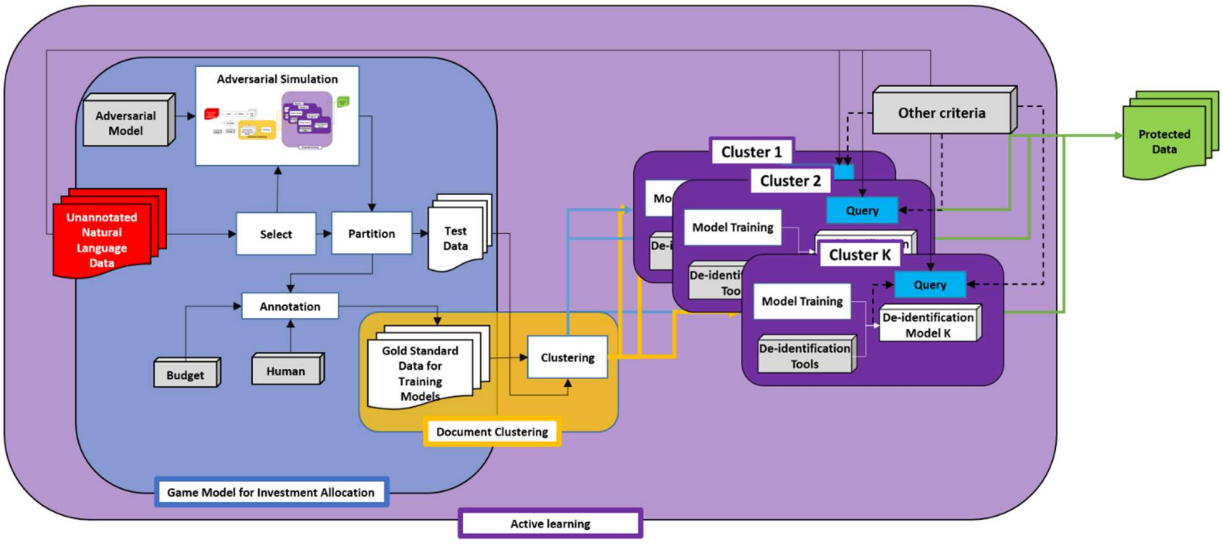


Figure 1. Natural language de-identification pipeline and how this dissertation fits in the big picture of the de-identification workflow

1.2 Dissertation Overview

This dissertation is constructed as follows: we first formalize the game theory based resource allocation framework for natural language de-identification in chapter 2 as Task 1, then in chapter 2 we present Task 2, the document clustering strategy calculated with writing complexity measures in order to enhance the training and testing of de-identification. In the last

chapter, we introduce an active learning approach for de-identification, design a pipeline to conduct experiments and assess the performance. Note that all three frameworks in this dissertation are evaluated on real world datasets.

Chapter 2 Game Based Model for Investment Allocation

To mitigate privacy risks in EMR usage, HCOs aim to remove potentially identifying patient information. Since a substantial quantity of EMR data is in the form of natural language and subject to exploits by ill-intentioned data recipients even after de-identification, HCOs have been encouraged to make as much investment as possible to detect and remove potential identifiers. However, such a strategy assumes the recipients are sufficiently incentivized and capable of exploiting leaked identifiers, which may not hold true in practice and may lead HCOs to overinvest in de-identification. The goal of Task 1 is to design a natural language de-identification framework, rooted in game theory, which enables an HCO to optimize their investments given the expected capabilities of an adversarial recipient. The answer to this problem depends largely on how much shared data is worth - to both the HCO providing the data and the potential recipients (who may exploit it maliciously, such as re-identification of patient data). Therefore, for an HCO to a make rational decision about how much to invest in de-identification, the incentives (and disincentives) of sharing data, as well as the cost-benefit model that incorporates behavior of the anticipated recipients, need to be well-defined.

We introduce a Stackelberg game to balance risk and utility in natural language de-identification. This game represents a cost-benefit model that enables an HCO with a fixed budget to minimize their investment in the de-identification process. Specifically, we model the HCO as a *defender/publisher* who has a limited budget, with a responsibility to protect patient privacy, and the malicious data recipient as a potential *attacker* who attempts to exploit it via re-identification. Under this model, the HCO incurs a cost when performing de-identification (e.g., paying readers to manually redact identifiers or annotate an EMR corpus to train an automated tool) based on which the publisher aims to achieve better protection of the data while retaining its utility. The attacker, by contrast, is incentivized to expose as much sensitive information from the published records as possible, but is bounded in capability (e.g., by a budget of their own) to perform the attack.

We formalize the interaction between the HCO and the ill-intentioned data recipient in a game theoretic framework. In this game, the publisher is a leader, who chooses whether or not to share data, and, if so, how much of their budget to spend on de-identification tasks (with the incentive to minimize spending, so that the surplus may be applied to other activities, such as

additional research studies). The attacker, by contrast, is a follower, who aims to discover leaked instances of PHI. The publisher may choose not to share the data, for example, if de-identification costs or risks from data sharing outweigh the benefits. The attacker, similarly, may opt out of attacking altogether if the benefits (e.g., from finding and exploiting leaked sensitive information) are not worth the cost of uncovering this information. One important aspect of our framework is that it explicitly models several mechanisms by which an attacker may be deterred. The first is for the publisher to manipulate the data and influence the confidence the attacker has in their claims of identifier discovery. An example of such a strategy is the "hiding in plain sight", or HIPS, approach, whereby all detected instances of identifiers are replaced with fake instances that exhibit a similar semantics (e.g., replacing the name "Rachel" with "Alice", replacing an actual date "4/12/2015" with a randomly generated date "4/25/2015" and replacing a real medical record number "12638920" with a generated medical record number "53267935") [28] which makes it difficult for an attacker to distinguish between fake and real PHI. A second deterrence mechanism is to institute data use agreements that penalize the attacker when they commit an exploit and are caught in the act. The model we introduce explicitly represents and reasons over both mechanisms.

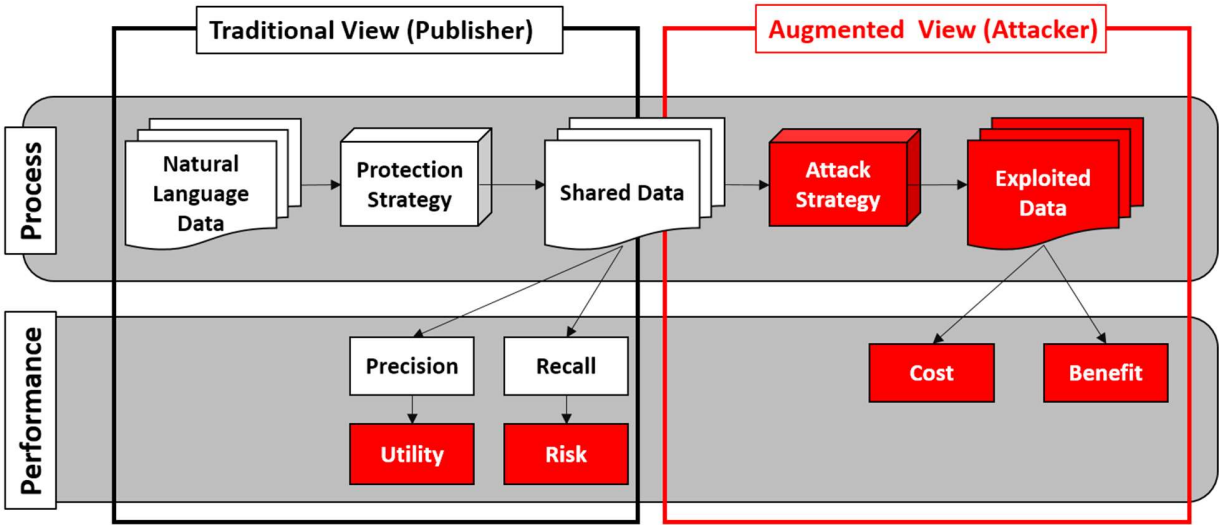


Figure 2. A depiction of the traditional view on natural language de-identification (left) and an augmented view that accounts for potential attackers (right) and translation of traditional information retrieval measures into economic factors.

We evaluate the model by assessing the overall payoff to the HCO and the adversary using 2100 clinical notes from Vanderbilt University Medical Center. We simulate several policy

alternatives using a range of parameters, including the cost of training a de-identification model and the loss in data utility due to the removal of terms that are not identifiers. In addition, we compare policy options where, when an attacker is fined for misuse, a monetary penalty is paid to the publishing HCO as opposed to a third party (e.g., a federal regulator).

Our results show that a game theoretic framework can be applied in leading HCO's to optimized decision making in natural language de-identification investments before sharing EMR data. More specifically in our study, when an HCO is forced to exhaust a limited budget (set to $2000 in the study), the precision and recall of the de-identification of the HCO are 0.86 and 0.8, respectively. A game-based approach enables a more refined cost-benefit tradeoff, improving both privacy and utility for the HCO. For example, our investigation shows that it is possible for an HCO to release the data without spending all their budget on de-identification and still deter the attacker, with a precision of 0.77 and a recall of 0.61 for the de-identification. There also exist scenarios in which the model indicates an HCO should not release any data because the risk is too great. In addition, we find that the practice of paying fines back to a HCO (an artifact of suing for breach of contract), as opposed to a third party such as a federal regulator, can induce an elevated level of data sharing risk, where the HCO is incentivized to bait the attacker to elicit compensation.

Task 1 provides three primary insights:

1) An Adversarial Model for Natural Language De-identification: The traditional view on natural language de-identification is depicted to the left of Figure 1. In this view, a publisher considers only the precision and recall of the redaction strategy. The rate of PHI discovery tends to grow logarithmically in the amount of training data supplied [22], which means that a publisher would require infinite investment to achieve perfect data protection. However, in the game view, the role of an attacker can be formalized, depicted to the right of Figure 1, as can the budgets available to both players in the system. In this augmented scenario, both sides engage in a cost-benefit analysis, which explicitly accounts for the interactions between the two agents.

2) A Stackelberg Formulation of De-identification: Based on the adversarial model, we introduce a game theoretic approach to solving this problem. This approach is based on a Stackelberg (or leader-follower) game, where the publisher can simulate the capabilities of the adversary before deciding on which strategy to implement (e.g., how much funding to invest in the de-identification process). In doing so, the publisher assumes that the adversary optimizes

their strategy and chooses a level of investment in de-identification that maximizes their benefit accounting for an attacker's response.

3) Systematic Policy Evaluation: We investigate the game under several policy designs for how penalties are paid for violations. We use a dataset of approximately 2100 real clinical notes from Vanderbilt University Medical Center to assess each policy. In doing so, we perform a sensitivity analysis on the decisions made as a function of the costs (e.g., penalties) enforced in the system. We find that there are cases in which the attacker will choose to forgo an attack while the publisher invests only a moderate amount in supporting de-identification. We also show that there are cases when the publisher should choose not to play and not share data.

This representation of the de-identification problem is notable in that it can 1) provide HCOs (or regulators, in the event that such contracts are formalized in policies) with the ability to engineer appropriate levels of penalization; 2) provide institutional review boards (IRBs) with a clearer picture of the actual (as opposed to perceived) risks of sharing natural language EMRs, and 3) explicitly model the tradeoffs between data utility and privacy risks.

2.1 Background

2.1.1 Machine Learning and Natural Language De-identification Tools

There are various machine learning approaches to de-identification that have been developed. These include strategies based on maximum entropy models [34], decisions trees [24], random forest [35], support vector machines [36], and conditional random fields [2, 23, 37] (CRF). CRFs [38], in particular, have been broadly applied by the NLP community to solve various problems, such as shallow parsing in sequence labeling tasks [39] and biomedical named entity recognition [40]. In the context of de-identification, the task is generalized to a named entity tagging problem [22], such that the goal is to identify and correctly assign type labels to each PHI instance (e.g., person names, ages, and calendar dates). As a brand of classifier designed to label words according to such types, CRFs presume that dependencies exist between these type labels, and then capture these dependencies under a first-order Markov assumption.

Various software tools have adopted CRFs for de-identification. The Health Information DE-identification (HIDE) [23] and a tool at Cincinnati Children's Hospital [2] were both developed based on the Mallet toolkit [41], while the Best-of-Breed (BoB) system [25]

incorporates a CRF implementation from the Stanford NLP group [42]. For this dissertation, we work with the MITRE Identification Scrubber Toolkit (MIST), which is based on the Carafe toolkit [43]. We use MIST because its built-in functionality addresses several useful tasks, including a web-based graphical annotation interface, a tagging module, a redaction and re-synthesis module, and an automated experiment engine.

Existing machine learning approaches to de-identification strive to maximize performance in terms of standard information retrieval measures, such as precision, recall, or a balanced F-measure [27, 44, 45, 46] at the token level (individual word) and instance level (phrase, e.g., first and last name).

However, this also implies that a recipient of the data will have the ability (or motivation) to exploit all leaked identifiers. Given that this can lead to over investment in training, our approach is substantially different in that it models the decision making process of selecting a natural language de-identification strategy as a game, where the performance measures, as well as the final payoffs (which are influenced by these scores and other considerations), determine the strategy. This reflects a more complete and principled approach to modeling the tradeoff between costs and benefits in de-identification.

2.1.2 Cost-Benefit Tradeoff in Security and Privacy Problems

A number of models have been introduced to support cost-benefit analysis in privacy problems. Here, we highlight several of the more relevant to our investigation. Specifically, it was shown that privacy valuations can be characterized by certain economic factors [47, 48], while a company's market value can be related to privacy breaches' [49]. Recently, an analytical cost model was proposed to monetize the tradeoff between privacy and data utility (according to data mining algorithms) in health data publishing [50]. Our research is similar in the sense that we aim to quantify the costs and benefits for the publisher and the attacker in our framework.

2.1.3 Game Theory in Privacy

Game theory has become an effective framework for modeling privacy (as well as security) challenges [51]. Our work specifically makes use of the Stackelberg game, in which the publisher acts first, after which the attacker makes a decision. In certain settings, this game has been modeled to enable a publisher to optimize the allocation of limited resources for better security, a number

of which [52, 53] were based on the similar assumption of our study, which is the adversary makes perfectly rational decisions to maximize the payoff.

There have been several investigations in solving privacy problems with a Stackelberg game approach, some of which exhibit a similar concept with our work in modeling a two-party interaction with a Stackelberg game [53, 54, 55, 56, 57, 58]. Liu et al. proposed a game model based on the assumption that the players have uncertainty with respect to each other's payoff function [53]. Rajbhandari et al. provided a model with a smaller strategy space [56]. Blocki et al. studied economic considerations in the design of internal audit mechanisms parameter [57]. Wan et al. designed a game theoretic framework for a data publisher to evaluate the tradeoff between re-identification risk and the value of sharing structured data [58]. While we represent a similar setting for the game by viewing the publisher as the leader and the potential attacker as the follower, our focus on natural language de-identification is novel, as well as our explicit reliance on classification performance measures. Additionally, our study considers the representation of alternative penalty-payment mechanisms, such as whether fines are paid back to the publisher or to some external regulator.

2.2 Methods

2.2.1 Game Theoretic Risk Model

We assess the risk of the re-identification attack against natural language clinical text that has been subject to the HIPS de-identification approach as a Stackelberg game. Our game is composed of two players, a publisher p and an attacker a. Both players aim to maximize their payoffs during the game. The publisher moves first by proactively deciding whether or not to share data (and if so, how much budget to allocate in the process). The attacker moves second by deciding whether or not to re-identify the data (and if so, how much budget to allocate in the process). Below we describe in greater detail the strategy space for the publisher and the attacker.

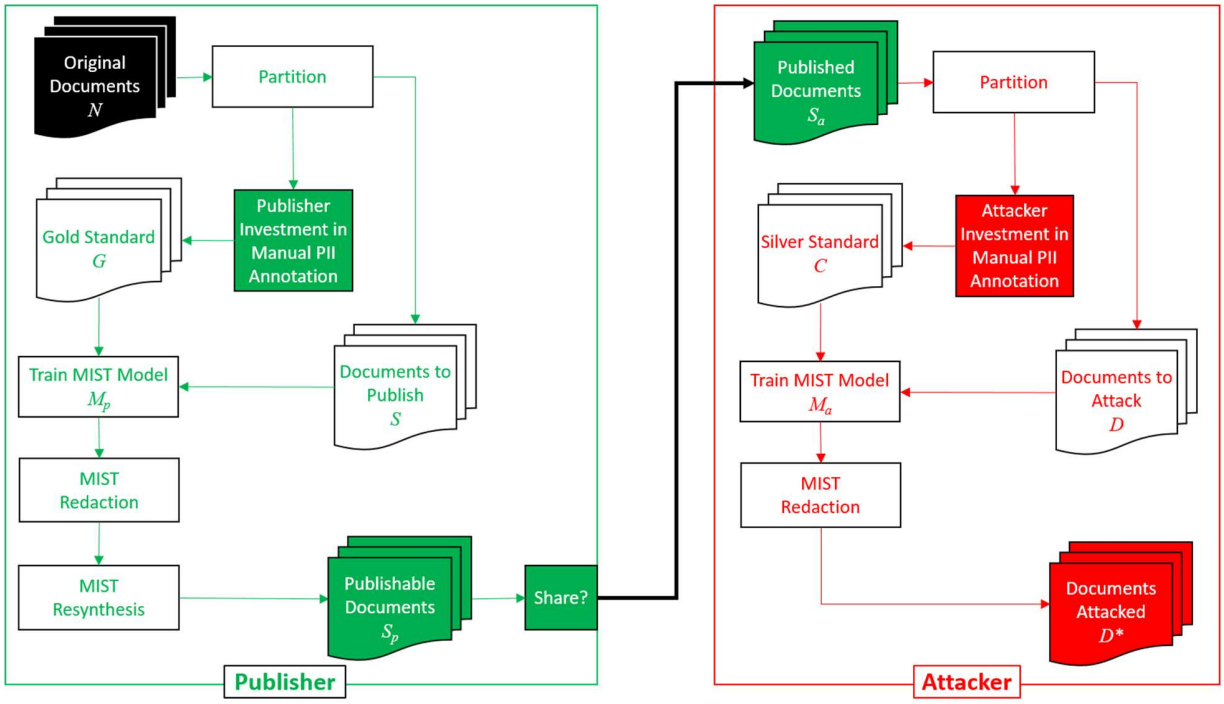


Figure 3. A natural language data de-identification and sharing pipeline from the publisher (left) to the attacker (right).

To model such an attack, we constructed a data sharing pipeline based on the MITRE Identification Scrubber Toolkit (MIST) [22] as shown to the left of Figure 3. We selected MIST for its built-in functionality of several useful tasks: 1) a tagging module, 2) an identifier redaction and resynthesis module, and 3) an automated experiment engine.

The publisher begins with a set of identified EMRs, denoted by N, to be shared. This set is split into 1) a subset $A \subset N$ for manual labeling and training of a de-identification model M_p (which in the case of MIST is a conditional random field) and 2) a subset $S \subset N$ to be de-identified by M_p, resynthesized with the HIPS method and turned into a publishable document set S_p. We assume the number of EMRs that may be shared is fixed, while the publisher can choose the size of the training data set. The publisher decides whether or not to share S_p, which turns into S_a when published to the attacker.

After receiving published documents S_a, the attacker targets the residual identifiers using the pipeline depicted in the right of Figure 3. Note that here we assume the adversary knows that the fake identifiers were distinguishable from non-identifiers in the publisher's model, thus they try to rebuild the publisher's de-identification model. If the attacker wants to exploit the

shortcomings of the model and chooses to use a different framework, then it is likely to be a suboptimal classifier since they are unlikely to replicate what the publisher did and are likely to retain more fake instances than using the same framework (or pipeline). Though, as illustrated in Figure 3, the attacker's aim is to attack the detection of PHI, this framework generalizes to the case when the attacker targets the resynthesis process itself. The attacker begins by manually labeling a subset of S_a, as $C \subset S_a$. Next, the attacker trains a model M_a under the same framework as the publisher (in our case, the attacker also utilizes MIST or some other tool that relies on conditional random fields). Finally, the attacker applies model M_a to a subset of the remainder of the published EMRs, denoted by $D \subset S_a$, to obtain a set of EMRs with tagged predicted identifiers, denoted by D^*.

2.2.2 Attacker Model

Note that prior research [28] suggests that humans cannot distinguish leaked identifiers from fake identifiers. Here our assumption is the worst case scenario in which the attacker can manually distinguish real from fake instances in the published EMRs, when the attacker can assign distinct labels to fake and real identifiers in the training data, but it takes time and effort to accomplish this task, especially when fake information is highly prevalent in the published data. This is why the attacker leverages the help of machine-based approaches to decrease the impact and distraction of the fake. As a start, the attacker chooses manually tagging either real or fake identifiers in a small batch of documents as a binary classification target.

Tag the real information. The training model is instructed to search for real identifiers, which means the machine-tagged results are actual leaks. We define precision and recall in such attacks as follows:

$$(\boldsymbol{P})recison = \frac{\text{attacker match}}{\text{attacker match + attacker spurious}}$$

$$(\boldsymbol{R})ecall = \frac{\text{attacker match}}{\text{attacker match + attacker missing}}$$

For reference purposes, the upper section of Table 1 provides explicit definitions of these variables.

Variable	**Definition**
attacker match	Residual real identifiers successfully tagged by the attacker
attacker spurious	Information tagged by the attacker that is not a real identifier
attacker missing	Residual real identifiers not detected by the attacker
"missing" reported by publisher	Total residual real identifiers in the published data
"spurious" reported by attacker	Information that is not a fake identifier redacted by the attacker (i.e., it is either a real instance or general clinical text)
publisher missing ∩ attacker spurious	Residual real identifiers redacted by the attacker
"missing" reported by attacker	Residual fake identifiers left in place by the attacker

Table 1. Variables in the Attacker's strategies.

Tag the fake information. The corresponding model trained by fake information tends to tag resynthesized information in the attacked EMRs. After redacting such machine tagged identifiers, the final residual identifiers in the resulting EMRs (which we assume that the attacker can find by careful inspection of the document – in effect annotating the residual PHI) are considered by the attacker as real (i.e., leaks). To evaluate the performance of the attack, we define the precision and the recall of the attacker in this case as follows. The lower section of Table 1 provides definitions of these variables.

$$(\boldsymbol{P})recison = \frac{\text{publisher missing} - (\text{publisher missing} \cap \text{attacker spurious})}{(\text{publisher missing} - (\text{publisher missing} \cap \text{attacker spurious})) + \text{attacker missing}}$$

$$(\boldsymbol{R})ecall = \frac{\text{publisher missing} - (\text{publisher missing} \cap \text{attacker spurious})}{\text{publisher missing}}$$

Note that in the following experiments we only applied "Tag the fake information" since 1) our preliminary test with "Tag the real information" suggests that there are not enough training data for the attacker when training with the real, 2) even under an assumption of an infinitely-sized

training dataset, there would still be problems with "Tag the real information". This is because, by definition, these real identifiers left in place by the publisher's model are the hardest instances for the CRF to find. It may be the case that what we are looking at here is the "noise" to the "signal". While it may be possible to pick up some of these instances, this is outside the scope of this paper (where we assume that the attacker attempts to mimic the publisher).

2.2.3 Cost Model Assumptions

We make the following assumptions for risk assessment purposes:

1) Both the publisher and the attacker begin with a fixed budget to spend in their activities;
2) The publisher can choose not to share any data and the attacker may choose not to attack if their payoffs are negative;
3) The publisher pays for annotating each EMR for training;
4) The publisher incurs a penalty for each false positive (that is, for each asserted leaked PHI instance that is not, in fact, a leak), resulting in utility loss due to over-redaction;
5) The publisher incurs a loss for each successful recovery of a leaked PHI instance by the attacker;
6) The attacker pays for annotating each HIPS de-identified record for model training purposes;
7) The attacker incurs a penalty for each false positive; and
8) The attacker is rewarded for a successful detection of a leaked identifier.

Variable Description	Publisher	Attacker
Average number of (real and fake) instances per EMR	i	i
Number of EMR training documents (*decision variable*)	α	γ
Number of published HIPS de-identified documents	β	β
Annotation cost per document	c_p	c_a
Loss for a false positive	l_p	l_a
Value for a true positive	0	v
Budget	B_p	B_a
Precision	P_p	P_a
Recall	R_p	R_a

Table 2. Definitions of the variables in the cost models.

2.2.4 Cost Model Basis

The variables relied upon to define the cost functions for the publisher and attacker are summarized in Table 2. We define B_a as the attacker's budget, c_a as the annotation cost for each EMR, and γ as the number of training instances annotated by the attacker. Consequentially, γc_a corresponds to the total cost of training.

Given the average number of PHI instances per EMR i, the publisher's published size β, and the publisher's recall R_p, the total number of instances leaked by the publisher is expected to be:

$$i\beta(1 - R_p)$$

Now, let R_a be the attacker's recall. Then, the number of correct guesses for leaked PHI is expected to be:

$$i\beta(1 - R_p)R_a$$

We define the value per true positive for the attacker as v. As such, the total value for a successful attack is expected to be:

$$i\beta(1 - R_p)R_a v$$

Next, let us consider the attacker's total guesses for the leaks. This can be derived by the attacker's recall and precision (P_a):

$$\frac{i\beta(1 - R_p)R_a}{P_a}$$

Based on this characterization, the number of false guesses made by the attacker is:

$$\frac{i\beta(1 - R_p)R_a}{P_a} - i\beta(1 - R_p)R_a = i\beta(1 - R_p)R_a\left(\frac{1}{P_a} - 1\right)$$

Let us define the attacker's loss per false positive as l_a. Then the loss for incorrect guesses is:

$$i\beta(1 - R_p)R_a\left(\frac{1}{P_a} - 1\right)l_a$$

The attacker's payoff for a re-identification attack, should he choose to attack, is thus defined as:

$$B_a - \gamma c_a + i\beta(1 - R_p)R_a\left(v - l_a\left(\frac{1}{P_a} - 1\right)\right) \qquad (1)$$

Therefore, the net gain, accounting for true and false leaks detected by the attacker is:

$$i\beta(1 - R_p)R_a\left(v - \left(\frac{1}{P_a} - 1\right)l_a\right) \quad (2)$$

2.2.5 Policy Alternatives

Turning now to the publisher's payoff, we consider four variations, giving rise to different instances of strategic interactions between the publisher and the attacker.

***Traditional Policy**:* This is how an HCO is currently forced to play in a data sharing ecosystem. Under this model, the publisher:

1) is required to publish the data under all circumstances,
2) is forced to exhaust the budget on training the de-identification model, which implies that $B_p - \alpha c_p = 0$, where B_p is the publisher's initial budget, c_p is the cost to annotate a document, αc_p is the total cost for training α documents to create a de-identification model, and
3) is penalized monetarily (or reputationally) by a third party for false positives (poor precision) and the attacker's successful attack.

Similar to the derivation of the attacker's payoff, let us define the publisher's precision as P_p and the loss per false positive as l_p. Then, the overall loss incurred by the publisher for poor precision is:

$$i\beta R_p\left(\frac{1}{P_p} - 1\right)l_p$$

Based on this setup, the publisher's payoff can be defined as:

$$B_p - \alpha c_p - i\beta R_p\left(\frac{1}{P_p} - 1\right)l_p - i\beta(1 - R_p)R_a v \quad (3)$$

Since, in this setting $B_p - \alpha c_p = 0$, the publisher's payoff can be simplified to:

$$- i\beta R_p\left(\frac{1}{P_p} - 1\right)l_p - i\beta(1 - R_p)R_a v. \quad (4)$$

***Safe-forward Policy**:* In this policy, the publisher's goal is to deter the attacker. If deterrence is not possible, or the publisher cannot achieve positive payoff, the publisher will

choose to publish nothing, indicating a payoff of 0. In this setting, whenever the attacker is penalized, it is assumed that the fine is collected by a third party (e.g., the U.S. Department of Health and Human Services). Since there is no penalty paid from the publisher, $i\beta(1 - R_p)R_a v = 0$. Thus, the payoff of the publisher (who makes the decision to play) can be simplified from Equation (3) to:

$$B_p - \alpha c_p - i\beta R_p \left(\frac{1}{P_p} - 1\right) l_p \qquad (5)$$

Since the publisher is not willing to tolerate a loss, this can also be thought of as the "no risk" policy.

***Attack-forward Policy**:* In this policy, the publisher's decision variables are whether to publish any data at all, and if so, how much training data to use (that is, α is a decision variable). As before, any penalty incurred by the attacker is collected by a third party. The payoff of the publisher is defined as in Equation (3).

***Attack-back Policy**:* This policy is similar to the Attack-forward policy, except now the publisher collects fines paid out by the attacker. This type of setting reflects what happens when the publisher is legally entitled to damages from the attacker (i.e., the popular "I'll sue you for violation of a contract." situation). Since the publisher basically pays for the net gain of the attacker (see Equation (2)) in this policy, the payoff of the publisher can be defined as:

$$B_p - \alpha c_p - i\beta R_p \left(\frac{1}{P_p} - 1\right) l_p - i\beta(1 - R_p)R_a(v - \left(\frac{1}{P_a} - 1\right) l_a) \qquad (6)$$

2.3 Experimental Design and Results

2.3.1 Dataset

The dataset used in Task 1 consists of a corpus of 2100 clinical encounter notes drawn from the Vanderbilt University Medical Center (VUMC) EMR system. Each file was drawn from a different patient and this set of records is composed of five document types, including i) clinical notes for history and physicals (HP), ii) discharge summaries (DS), iii) pathology notes (PATH), iv) clinical communications (CC), and v) respiratory care encounter notes (RC). The numbers of documents of each type are shown in Table 3. There are 11 PHI types in the corpus, the details, and justification, of which are provided in Appendix A1.

Document Type	Number
CC	500
DS	409
HP	398
PATH	349
RC	400

Table 3. Distribution of document types in the corpus.

2.3.2 Publisher and Attacker Model Performance Measures

We set up a data publishing pipeline (Figure 4) to evaluate the performance of both parties in this process. The size of the dataset to be published was fixed at 400 documents. The publisher trained a data de-identification model with an independent set of 200 clinical records, manually annotated as gold standard documents. From this dataset, we randomly sampled subsets of 10, 20, 50, 100, and 200 documents to train models on different quantities of training data, corresponding to different levels of de-identification accuracy. Note that these training numbers are unrealistically small, but serve an experimental purpose of generating variation in model performance, which is desirable from an experimental perspective.

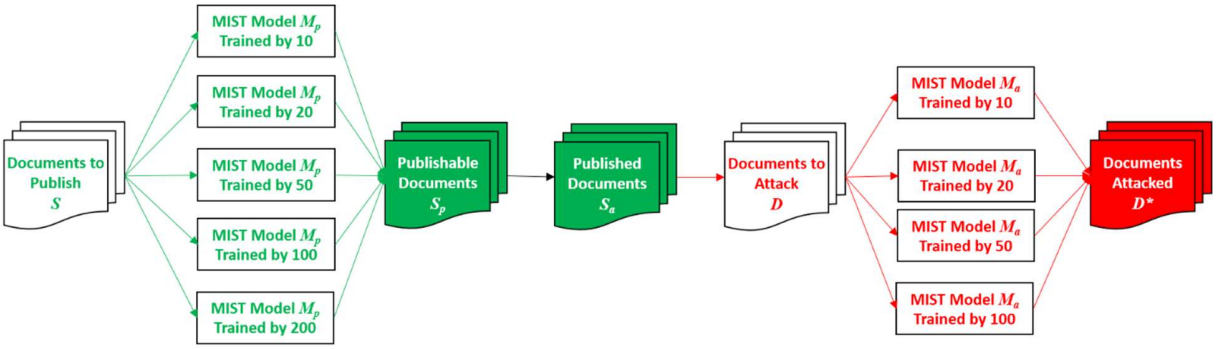


Figure 4. Experimental design for performance and cost model evaluations of the publisher and the attacker. Note that a specific experiment consists of the publisher and attacker choosing one level of training each.

The resulting model was then applied to redact the PHI instances in the 400 documents planned for publication, after which all detected instances were replaced with fake instances. Since the model cannot achieve 100% recall, the 400 records available for publication included a mix of residual leaked identifiers and fake identifiers. For each training scenario of the publisher, we simulated the adversary's strategy by annotating subsets of 10, 20, 50 and 100 EMRs using the tag fake PHI model mentioned earlier. The resulting models were applied to the remainder of the shared dataset. The entire process is shown as a simplified pipeline in Figure 4.

We assessed the performance of the publisher and attacker in terms of precision, recall, and F-measure. All the scenarios were repeated five times and we report average results. Figure 5 shows that the publisher's precision and recall improve with increasing size of the training data. The precision ranged from 0.70 (when training with 10 records) to 0.86 (when training with 200 records). The recall of the publisher sustained a more dramatic change from 0.49 (when training with 10 records) to 0.8 (when training with 200 records). This indicates that the potential leaks available for detection by the attacker drops from 51% of the instances to 20%.

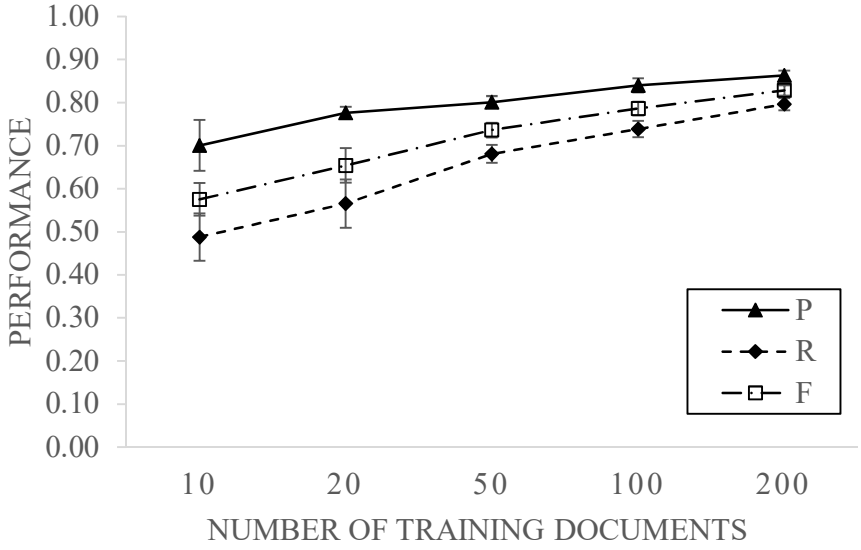


Figure 5. Performance for the publisher's de-identification model as a function of the number of documents provided for training.

Figure 6 shows the attacker's performance using the "tag fake" approach under different training scenarios of the publisher. In each plot, the horizontal axis represents the attacker's change in training size, while the vertical axis represents a performance measure. Each of the three lines shows the performance of the attacker in a specific training size of the publisher (e.g., the solid line in Figure 6(a) depicts the attacker's change in precision when the publisher trains with 10 documents). As shown in Figure 6(a), overall, the attacker's precision increases as additional data is annotated and applied for training. Specifically, for the documents de-identified by the publisher's 10-document training model, the attacker's precision increases from 0.72 to 0.85 when the attacker increases the training size from 10 to 100 documents. When the publisher increases the training size, the attacker's precision shows a decreasing trend. For example, when the attacker fixes the number of training documents to 10, the precision falls from 0.72 to 0.38 as the publisher's training size grows from 10 to 200. The attacker's recall (with respect to instances that were leaked by the publisher), on the other hand, remains steady, at around 0.99 among all scenarios, as shown in Figure 6(b). The F-measure change with respect to the training sizes followed a similar path to the precision (see Figure 6(c)).

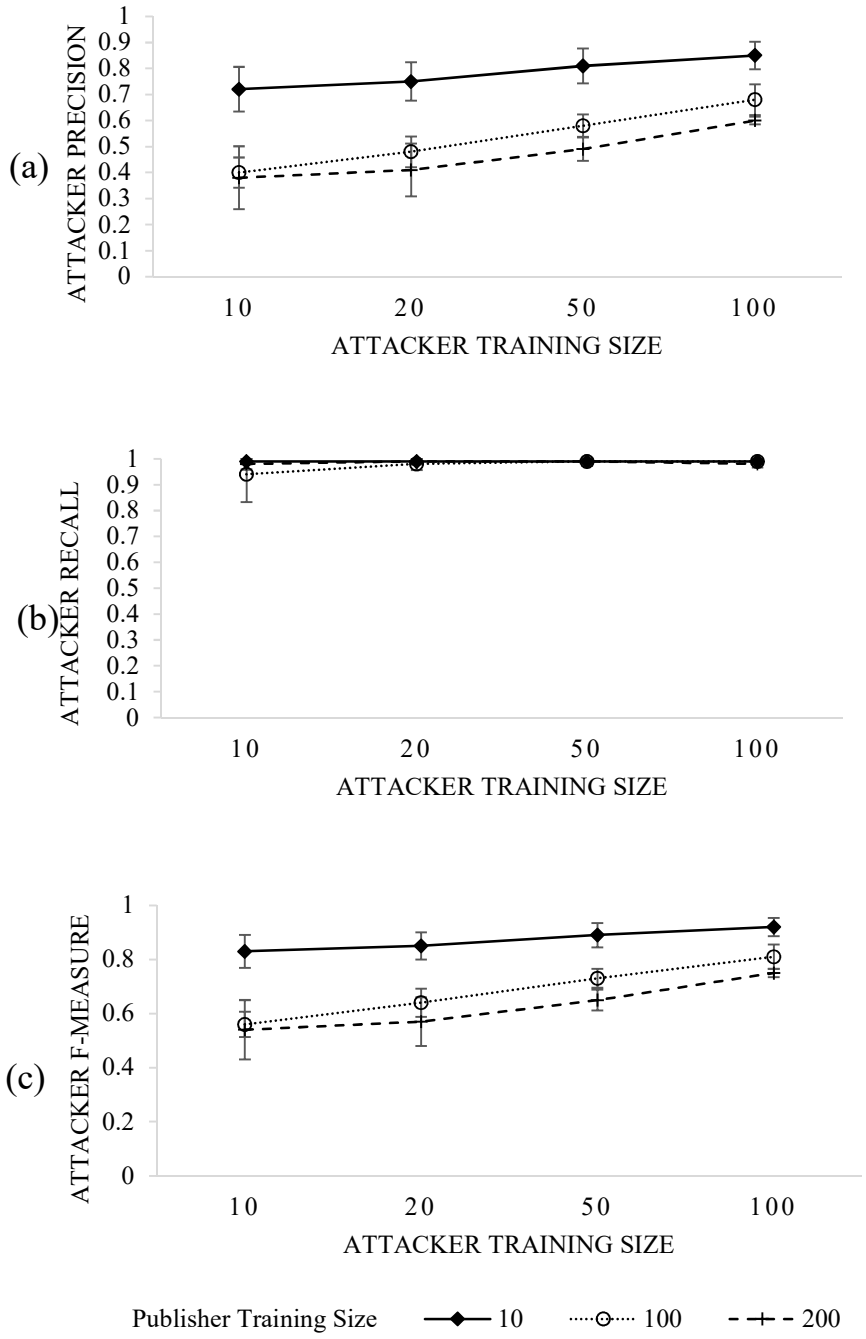


Figure 6. The influence of the publisher's training dataset size on the attacker's (a) precision, (b) recall, and (c) F-measure.

2.3.3 Publisher and Attacker Costs

We begin our case studies by assigning values to variables as summarized in Table 4. The number of total PHI instances in the EMRs available for publication is 7758, such that the average number of PHIs per EMR (i) is 19.4. For interpretation purposes, we set this value to 20. We set the annotation cost per record for the publisher (c_p) to be \$10. As stated earlier, the publisher has a limited budget, which is determined by the maximum amount of money needed for annotation. In our case the publisher's maximum training size (α) is 200, leading to an initial publisher budget (B_p) of \$2000. We chose the (reputational) penalty for removal of non-identifiers (l_p) to be \$0.1 per instance.

Case Name	Attacker's Annotation Cost Per EMR	Attacker's Loss Per False Positive	Attacker's Value Per True Positive
Low	\$1	\$0.30	\$0.10
Mid-low	\$4	\$0	\$0.10
Mid-high	\$4	\$0.50	\$0.50
High	\$10	\$0.30	\$0.50

Table 4. Names of the case studies and their corresponding variable values.

On the attacker's side, we set the budget (B_a) to \$1000. For each combination of the parameters, we first calculated the attacker's payoff. Since we modeled the interaction between the publisher and the attacker as a Stackelberg game, for each strategy available to the publisher, we computed the attacker's best response (i.e., the strategy that maximizing the attacker's payoff) and then evaluated the publisher's payoff. If the attacker's optimized payoff was negative after deducting the initial budget, we considered the attacker to be completely deterred. Similarly, if the publisher's payoffs from sharing data was negative, we assumed that no data would be shared if such an option was available.

We applied four different combinations of parameters yielding four case studies, which we refer to as *Low*, *Mid-low*, *Mid-high*, and *High*. In the *Low* case, all values of the three parameters were relatively small (c_a = \$1, l_a = \$0.3, v = \$0.1), while in the *High* case, the parameter values were high comparing to the other cases (c_a = \$10, l_a = \$0.3, v = \$0.5). For the *Mid-low* and *Mid-high* cases, the values of c_a was set to \$4, and the remaining values were l_a = 0, v = \$0.1 and l_a =

\$0.5, v = \$0.5, respectively. Note that in the *Mid-low* case, we selected 0 for the value of l_a to show an extreme case when the attacker does not suffer from incorrect prediction at all.

The results of the case studies are shown in Table 5. Each row reports the results of a specific case in terms of training strategies and payoffs for both the publisher and the attacker. Each grouping of four rows present a summary of cases in one of the four policies (i.e., Traditional, Safe-forward, Attack-forward, and Attack-back).

Traditional Policy: As noted earlier, in this policy, the publisher is obligated to spend all of the budget on training a protection model and publish the EMRs (see Table 5). There were several notable findings from this baseline analysis. First, it should be noticed that the publisher always spends money because they are forced to exhaust their budget. They are then subject to further losses from inaccurate de-identification (i.e., translation of non-identifiers to fake identifiers) and attacks from the recipient. At the same time, it should be recognized that the recipient only chooses to attack in one of the cases. Specifically, in the *Mid-low* case, the attacker chooses to train on 10 documents and achieves a return of \$116.30.

Policy	Case Name	Number of Documents in Attacker Training	Attacker Payoff	Number of Documents in Publisher Training	Publisher Payoff	Does the Publisher Share Data?
Traditional	*Low*	0	0	200	-$104.12	✓
	Mid - Low	10	$116.30	200	-$260.49	✓
	Mid - High	0	0	200	-$104.12	✓
	High	0	0	200	-$104.12	✓
Safe-forward (No risk)	*Low*	0	$0	30	$1,558.22	✓
	Mid - Low	0	$0	0	$0.00	×
	Mid - High	0	$0	180	$101.71	✓
	High	0	$0	190	$3.37	✓
Attack-forward	*Low*	0	$0	30	$1,558.22	✓
	Mid - Low	10	$361.17	10	$1,343.76	✓
	Mid - High	0	$0	180	$101.71	✓
	High	24	$563.49	40	$44.58	✓
Attack-back	*Low*	0	$0	30	$1,558.22	✓
	Mid - Low	10	$361.17	10	$1,343.76	✓
	Mid - High	58	$651.34	30	$675.68	✓
	High	26	$422.17	50	$688.63	✓

Table 5. Case study results for each policy in terms of number of training EMRs and payoffs for the publisher and attacker.

Safe-forward (No risk) Policy: In this policy, the publisher decides whether or not to play and manipulates the training dataset size to ensure that the attacker will never attack. There are several notable findings to highlight at this point. First, when the publisher cannot find a strategy to prevent the attack, s/he chooses not to share EMRs. This occurs in the *Mid-low* case, which leads to no payoff in the Safe Forward policy in contrast to the negative payoff in the Traditional policy. Second, since the attacker cannot benefit from the attacks, the attacker will never play the game, such that when the publisher shares data, the publisher always ends up with a positive payoff. In contrast to the Traditional policy, each of the *Low*, *Mid-high*, and *High* cases lead to positive payoffs for the publisher. Nonetheless, it should be noted that each of these cases leads to a different amount of investment in training and overall payoff.

Attack-forward Policy: In this policy, the publisher decides whether or not to play, but now selects a training dataset size that maximize the overall payoff. Any penalty the attacker incurs is

paid forward to a third party. Again, there are several notable findings to highlight. First, this design of the game leads to situations in which the publisher makes the same decision as in the Safe-forward policy. This means that the publisher's optimal strategy occurs when the attacker chooses not to play (see the *Low* and *Mid-high* cases). Second, there are cases when the publisher chooses to share EMRs when they would have failed to do so under the Safe-forward policy (see the *Mid-low* case). However, this policy also leads to what appears to be a negative outcome in comparison to the Safe-forward policy. Specifically, in the *High* case, the publisher chooses to share less data (40 documents instead of 190), so that they can achieve a higher payoff ($44.58 instead of $3.37). Yet, in doing so, they open the system up to attack and enable the attacker to receive a positive payoff instead of being deterred from playing entirely.

Attack-back Policy: In this policy, the publisher decides whether or not to share, but now selects a training dataset size that maximize the overall payoff. Any penalty the attacker incurs is paid back to the publisher. The results for this game imply that permitting the publisher to sue the attacker for damages could be a dangerous policy. This is because it appears to encourage the publisher to undertrain and bait the attacker, inducing more hazard in this system. A clear illustration of this finding is depicted in the *Mid-high* case, where the publisher has lowered their training from 180 to 30 EMRs to raise their payoff from $101.71 to $675.68 while enabling the attacker to move from a decision of not playing to actually playing and receiving a positive payoff of $651.34.

To provide context for how the publisher arrives at this decision, Figures 7 and 8 illustrate the process of decision making for the *Mid-high* case. The processes for the rest of the cases are presented in Appendix A2. In each figure, the optimized attacker decision (in terms of investment in training) as a function of the publisher's potential decision (in terms of investment as well) is shown in the upper plot, while the actual payoffs to the two players are shown in the bottom plot. For instance, in Figure 7, the training size options for the publisher are represented by the horizontal axes, whereas the attacker's decisions and the payoff are shown by the vertical axes. The publisher thus makes its optimized decision based on all possible payoffs to the attacker and the publisher.

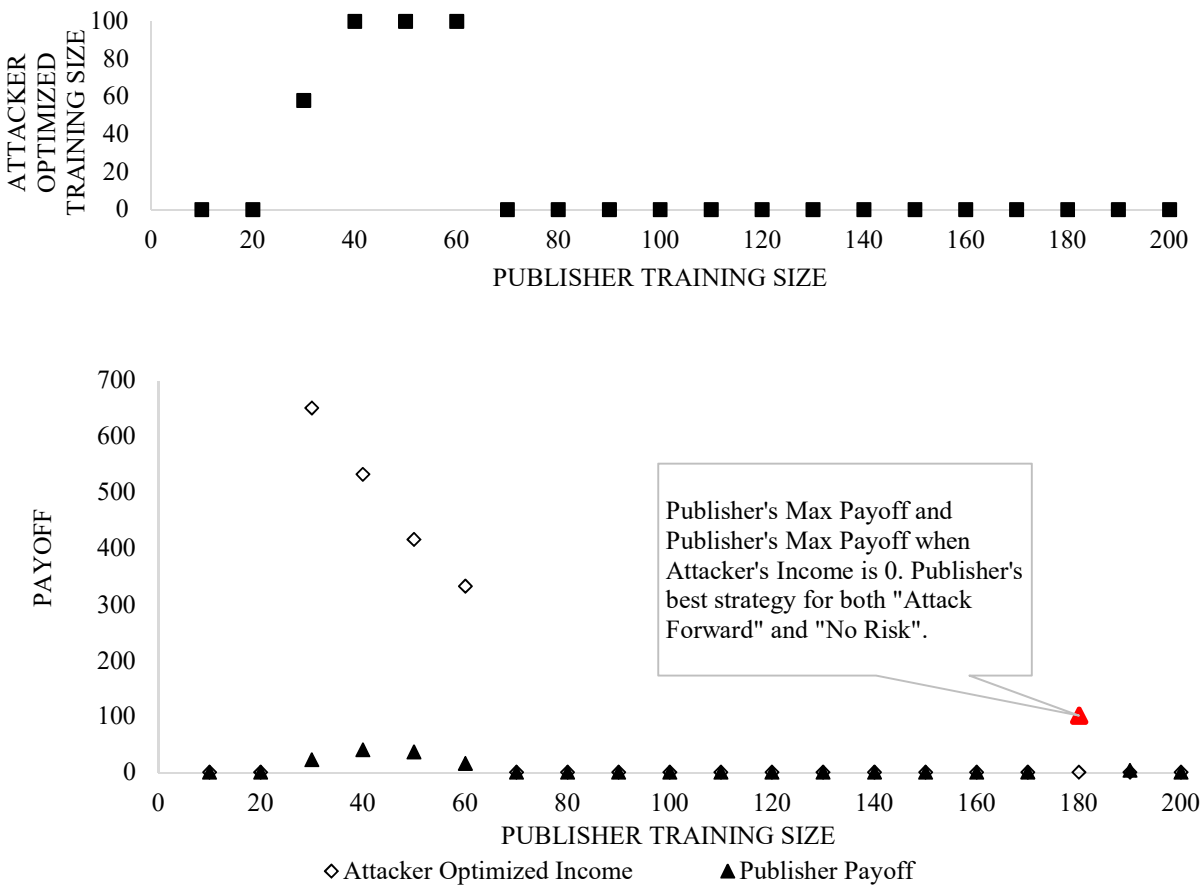


Figure 7. Decision making process of the publisher and corresponding strategies of the attacker in the *Mid-high* case when the attacker's pays the penalty forward to a third party.

As can be seen in Figure 7, when the attacker's penalty goes to a third party, if the publisher chooses to train with 50 documents, the attacker's optimal size of training data is 100, which yields the payoff of $36.52 to the publisher and $416.17 to the attacker. If the publisher aims to make the most of the budget, the publisher should pick 180 (tagged in red) as the training size, left with a payoff of $101.71. In this case, the attacker will choose not to attack, since the attacker can never obtain a positive payoff. Consequently, if the publisher attempts to always deter the attacker, the publisher's optimal training size is, again, 180. Thus, this figure illustrates that in case *Mid-high*, policies of Safe-forward and Attack-forward actually yield same decisions for the publisher.

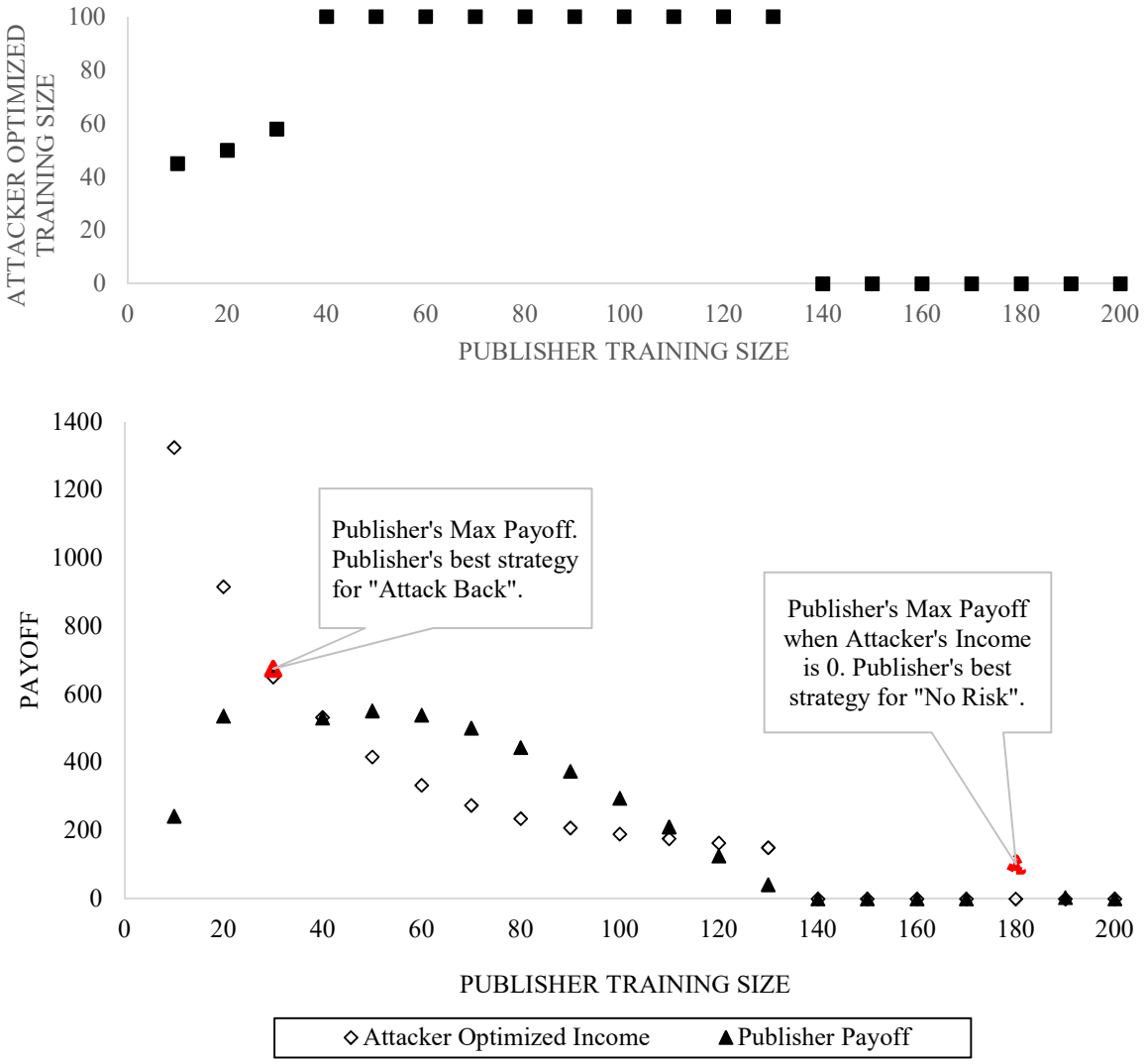


Figure 8. Decision making process of the publisher and corresponding strategies of the attacker in the *Mid-high* case when the attacker pays the penalty back to the publisher.

Similarly in Figure 8, when the publisher receives the fine paid by the attacker, the publisher's decision to ensure the attacker be suppressed stays the same with Figure 7, which is 180. However, if the publisher targets to maximize the payoff, the best option for the publisher changes to be 30, leaving a payoff of $675.68, where the attacker trains with 58 and is left with $651.34.

2.3.4 Sensitivity Analysis

Next, we performed a sensitivity analysis of the payoffs of the publisher and the attacker to investigate the relationships between the strategies of the publisher and the parameters of the attacker. For each of the policies shown in Figs 9 and 10, we fix the value of the attacker's annotation cost per document (c_a) and vary the attacker's fine for incorrect detection per instance (l_a) and the value per successfully detected instance for the attacker (v). The results are depicted as heat maps, in which the horizontal axes represent v (the value per true positive), and the vertical represent l_a (the loss per false positive). Note that the lighter the yellow, the greater the net payoff to the publisher/attacker. For brevity, we analyze the figures corresponding to the *Low* and *Mid-high* cases and provide the figures of the other cases in Appendix A3.

Figure 9 shows the sensitivity analysis for the four policies (i.e., Traditional, Safe-forward, Attacker-forward, and Attack-back) of the *Low* case, where $c_a = 1$. Note that in the Traditional policy (i.e., where the publisher always exhausts the budget to train a de-identification model), the attacker will not attack above the diagonal of the plot, where $v \leq l_a$. However, the publisher's payoff stays below 0 under all possible combinations of v and l_a. In all Safe-forward, Attack-forward and Attack-back policies the publisher's best choice is 30, while the regions in which no attack transpires appears when $v \leq 0.5 * l_a$. In both forward penalty payment policies the change of v exhibits a larger impact on the publisher's payoff than l_a (which hardly affects the payoff). The larger the v, the smaller the publisher's payoff. In the Attack-back policy, v still has a greater impact than l_a, though it should be noted that l_a shows an increased impact over the forward policies. This is expected because the fine incurred by l_a is returned to the publisher.

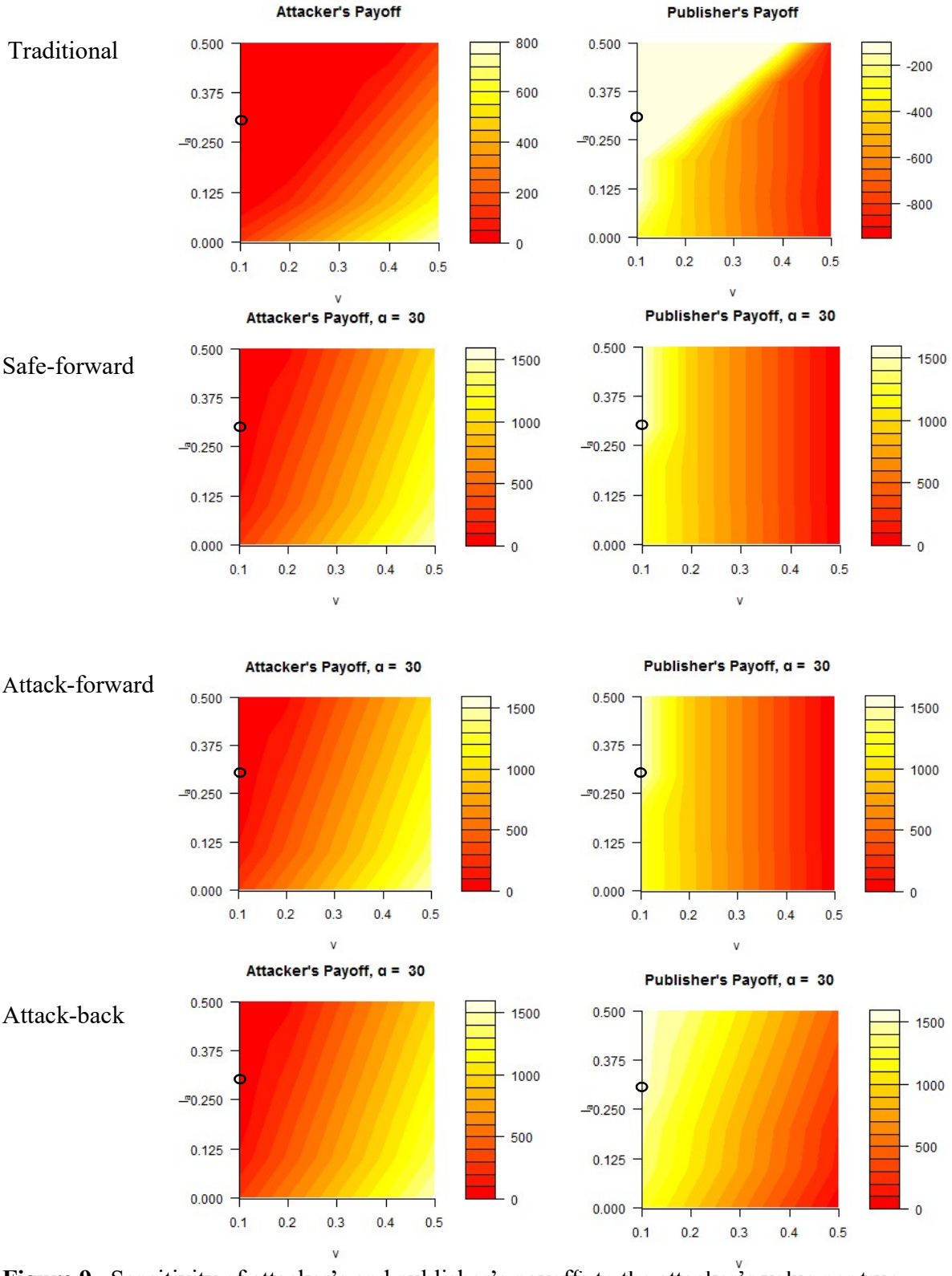


Figure 9. Sensitivity of attacker's and publisher's payoffs to the attacker's value per true positive (v) and loss per false positive (l_a) for policies Traditional, Safe-forward, Attack-forward and Attack-back when the attacker's annotation cost per EMR (c_a) is \$1. The result for the *Low* case (v = \$0.1, l_a = \$0.3) is circled in each figure.

Figure 10 depicts the policies corresponding to the *Mid-high* case, where $c_a = 4$. The region in which no attack transpires for the Traditional policy is approximately when $l_a \geq 2/3(v - 0.1)$. Again, the publisher's payoff stays below 0 under all possible combinations of v and l_a. In both of the forward policies, the publisher's best option is to train on 180 documents, while the attacker's payoff is always 0. Notably, the publisher's payoff can be divided into two general situations: 1) the region that is all red, which indicates a payoff of 0 because the publisher chooses not to play and 2) the remainder of the space where there is a positive payoff for the publisher that is devoid of attack risk. Finally, for the Attack-back policy, the attacker's payoff is 0 when $v < l_a$, leaving the publisher with a large positive payoff. Clearly, in this policy, where risk is permitted, the publisher's payoff is affected by both the attacker's value per true positive (v) and the loss per false positive for the attacker (l_a).

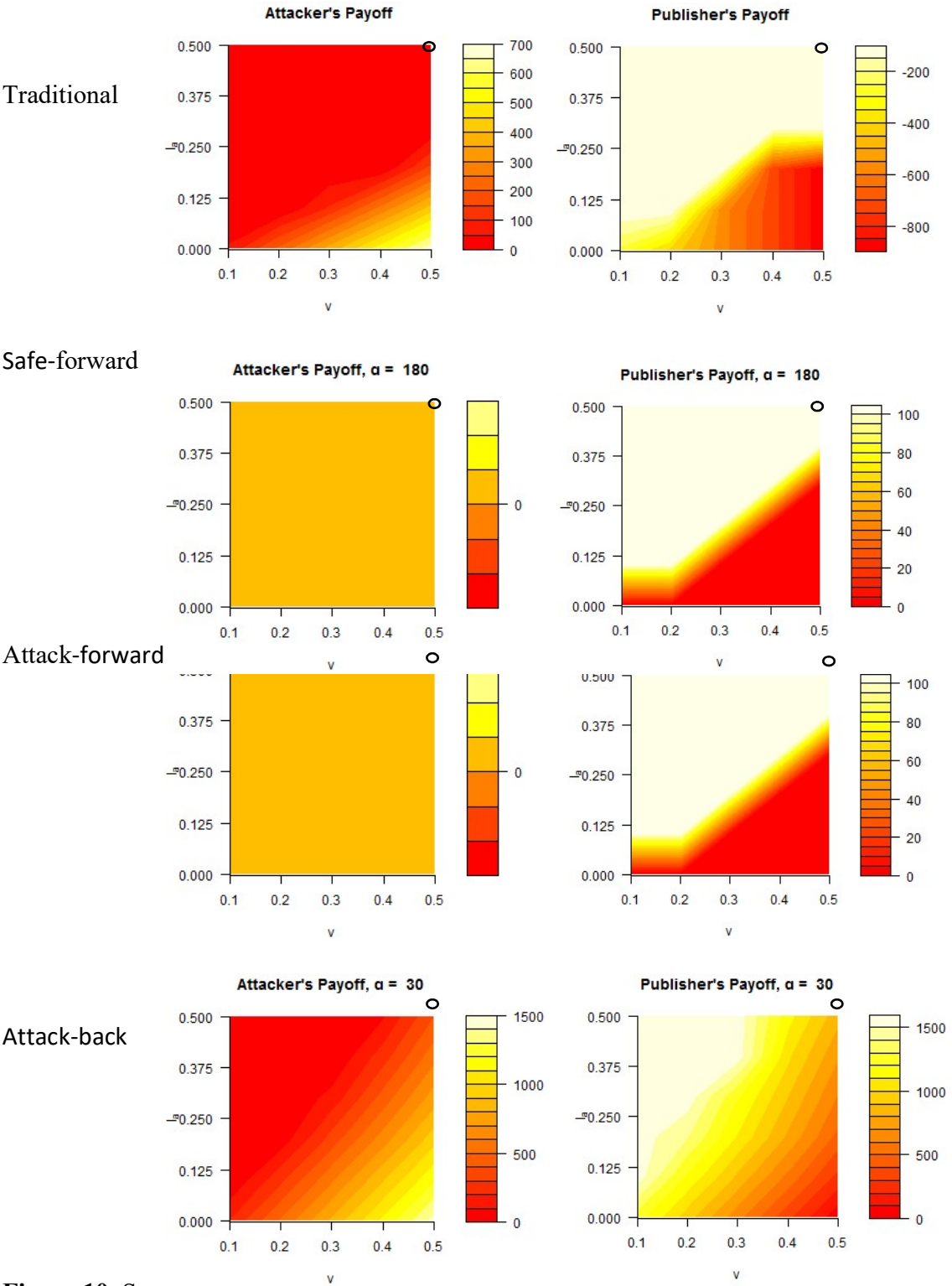


Figure 10. Sensitivity of attacker's and publisher's payoffs to the attacker's value per true positive (v) and loss per false positive (l_a) for four policies when attacker's annotation cost per EMR (c_a) is \$4. The *Mid-high* case (v = \$0.5, l_a = \$0.5) is circled.

2.4 Discussion

Task 1 demonstrates that natural language de-identification under a hiding in plain sight (HIPS) [28] framework can be mapped to a game theoretic model that formally quantifies the tradeoff between data publishing and privacy risks. In doing so, we showed traditional views [27, 44] on privacy protection can force HCOs to exhaust budgets and bear the risk of losing reputation (in the event of changing non-identifiers to redacted or fake identifiers) and money (in the event re-identification is successful), such optimal strategies still enable the HCO with a positive payoff.

By modeling data sharing as a game, we observed that, under certain circumstances, it is possible for the HCO to opt for strategies that will ensure a malicious but rational recipient of de-identified EMR data will not attempt re-identification via residual identifiers. Moreover, we discovered that the way in which penalties are paid by adversaries for violating terms of service significantly influences how data is shared, sometimes resulting in perverse outcomes. Specifically, we showed that when those sharing the data are entitled to damages for violation of a contractual agreement, they may be inappropriately incentivized to bait an attacker by publishing potentially exploitable patient data. We acknowledge that this is not a solution for all EMR data sharing scenarios, but can serve as one approach to organizing and managing the system when the main problem parameters can be defined.

2.4.1 Limitations

While this investigation suggests that a game theoretic framework for natural language de-identification can facilitate more informed and, in certain circumstances, greater amounts of data sharing, there are several limitations that need to be acknowledged.

First, the above findings are based on the assumption that we understand what the motivation is for the attacker and that it is monetary (or at least a factor is quantifiable). In other words, our model focused on economic motivated attackers, whereas reputation driven attackers are out of scope.

Second, we acknowledge that our case studies are based on a single dataset. As such, the generalizability of our findings requires further investigation with other datasets. However, it is critical that such investigations be performed on EMRs with real residual identifiers. This is challenging because, public use datasets, such as those made available by i2b2 [27, 45, 46] or

Cincinnati Children's hospital [59] have removed all identifiers, including those which machine learning methods would have a difficult time discovering in the first place [32].

Third, it is important to recognize that the definition of cost plays a key role in our framework. We performed a sensitivity analysis to determine the stability of our findings, but depending on the actual content in an EMR, the costs per record may change.

Chapter 3 Document Clustering

De-identification tools based on machine learning approaches require a considerable amount of training data for a kick-start. The model training process is usually based on either a random group of documents or a pre-existing document type designation (e.g., discharge summary). Previous studies have shown grouping clinical narratives by document types before starting the process of de-identification model training, and apply models to documents according to the designated types could improve the performance of de-identification models.

Task 2 of this dissertation is based on the observation that in practice the information of document types could be missing. Further, even if it is available, since there is heterogeneity existing in documentation practices, such information may not always be the most reliable standard for dividing medical records into training groups.

In this dissertation we investigate if inherent features can identify document subsets to enhance de-identification performance. The hypothesis is that certain characteristics of the clinical documents themselves that are mathematically calculable, can be used to enhance the training of machine learned de-identification models by grouping clinical documents into more homogeneous subsets thus improve the performance of de-identification. Specifically we explore the usage of writing complexity and richness of clinical vocabulary of clinical narratives. To assess this hypothesis, we developed a feature extraction and clustering strategy to partition clinical narratives into inferred types over which de-identification models are trained and tested. For corpora that contains highly diverse documents, the methods we discuss in this dissertation may be of particular value for de-identification.

To evaluate this hypothesis we utilized two corpora. The first consists of the 889 discharge summaries from the i2b2 challenge which is publicly available. The second corpus consists of over 4500 medical records from the Vanderbilt University Medical Center (VUMC) of different document types (e.g., discharge summaries, history and physical reports, and radiology reports). Specifically, we investigate three alternative scenarios for clustering clinical narratives: 1) EHR-assigned document type, 2) writing complexity and clinical vocabulary richness, and 3) a random process.

For scenario 2), we applied an unsupervised clustering method to group two corpora based on writing complexity measures and compared the performance in terms of recall, precision, and

F-measure of de-identification models trained on such clusters with models trained on documents grouped randomly or VUMC document type.

Our experiment results showed that for the i2b2 dataset which is a highly regularized dataset with the same document type, training and testing on the same clusters based on complexity measures (average F-score 0.966) did not significantly surpass randomly selected clusters (average F-score 0.965). For the Vanderbilt dataset, which consists a variety of clinical narratives, de-identification models trained on the same writing complexity measures (with the average F-measure of 0.917) are better than models trained on random groups (with an average F-measure of 0.881). Moreover, increasing the size of a training subset sampled from a specific cluster could yield improved results (e.g., for subsets from a certain stylometric cluster, the F-measure raised from 0.743 to 0.841 when training size increased from 10 to 50 documents, and when training with 200 documents the F-measure reached 0.901). It was also observed that in some cases training on the same stylometric features tended to surpass training on the same document types.

3.1 Background

3.1.1 Writing Styles

Writing styles are author-specific literary patterns [60], the formal characterization of which can be traced back to at least the 19th century [61]. A wide array of assessment functions have been proposed to quantitatively represent personal writing style in the form of stylometric features. These measurements vary in their nature; e.g., lexical features are based on word-tokens [61], while another feature class is derived from syntactic information, approximated by part-of-speech [62].

In this work, we utilize two general types of writing complexity. The first type corresponds to *readability*. A readability formula is a measure designed to provide quantitative estimates of the difficulty in writing style [63]. Such formulae tend to count language variables in a document and estimate the reading difficulty level based on the associated statistics [64]. The earliest formulation of this concept [65] is attributed to Lively and Pressey for children's readability,[3] which measured the vocabulary difficulty with specific indicators. This work influenced later readability studies, notably the Flesch's Reading Ease formula [66] (one of the most widely used readability measurements in the world [63]), which predicts the ease of reading based on ratios of

syllables per word and words per sentence. Over 200 readability formulas have been published [65], and in this work we focus on several representative formulae, which are presented in Table 7.

Our second type of writing complexity is *lexical richness* (or *diversity*). This is a subclass of lexical features based on vocabulary size, frequency distribution, and other variations [60]. Lexical richness includes a simple type token ratio (TTR):

$$TTR(N) = \frac{V(N)}{N},$$

where N denotes a document's word count and $V(N)$ denotes the number of unique words.

Some measures are variations of TTR, such as Carroll's Corrected TTR [67]. Others, such as Dugast's Uber Index [68], incorporate simple transformations of $V(N)$ and N.

3.2 Methods

3.2.1 Materials

In this study, we applied our methodology to two datasets. For the first dataset, we selected patient records from an existing de-identified version of StarChart, the VUMC EHR [69]. The system contains information dating back to 1984 and continues to receive feeds from a diverse set of sources, including lab results, radiology reports, and external transcription companies. In lieu of a large human annotated corpus, we leveraged a locally-specialized version of DE-ID, which is a commercially available rules-based de-identification software tool [31], to indicate the position and syntactic type (e.g., name versus date) of each patient identifier. DE-ID replaces identifiers with generic placeholders for the syntactic type. This decision was based on the fact that currently DE-ID, in conjunction with several pre- and post-processing modules developed at the VUMC, remains one of the core technologies by which the VUMC de-identifies its medical records (over 1.7 million) for local investigator-initiated research projects [70]. In addition, in a demonstration project conducted at the VUMC for its institutional review board, it was shown that the software in place at the VUMC exhibited a recall of over 99.9% for HIPAA Safe Harbor identifiers.

Document Type	Documents	Average No. of PHI Instances	Precision	Recall	F-measure
History & Physical (*HP*)	550	23.8	0.882	0.897	0.889
General Report (*REP*)	550	24.64	0.941	0.922	0.931
Discharge Summary (*DS*)	550	47.29	0.937	0.941	0.939
Radiology Note (*RAD*)	547	5.64	0.954	0.914	0.933
Pathology Note (*PATH*)	550	16.93	0.913	0.910	0.912
Family History (*FH*)	200	1.12	0.949	0.768	0.842
Clinical Communication (*CC*)	550	23.67	0.949	0.979	0.964
Letters (*LET*)	550	44.2	0.887	0.905	0.896
Rehabilitation Note (*REHAB*)	550	44.2	0.888	0.905	0.897
		Average	0.922	0.904	0.911
		St. Dev.	0.030	0.057	0.035

Table 6. VUMC EMR document types in the study and corresponding performances of de-identification training models.

The corpus used for this study was randomly selected across dates and portions of the clinical enterprise. It consists of 4,597 clinical documents of 9 EHR types, including discharge summaries (DS), history & physical assessments (HP), radiology notes (RAD), and pathology notes (PATH). The frequency distribution of documents by EHR type is shown in Table 6. There are approximately 550 documents of each type, except for family histories (FH) which consisted of 200.

The second dataset corresponds to the i2b2 de-identification challenge corpus [22]. It contains 889 annotated clinical notes in the form of hospital discharge summaries from the Partners Healthcare System, within which the real identifiers were replaced by synthetic information.

3.2.2 Document Pre-processing and Clustering Based on Complexity and Richness Measures

The process by which documents were pre-processed and clustered is depicted in Figure 11, which also provides an overview of the experimental design. In general, the process of

stylometric clustering corresponds to three stages: 1) Data Preprocessing (resynthesis of DE-IDed documents), 2) Feature Extraction, and 3) Unsupervised Document Clustering.

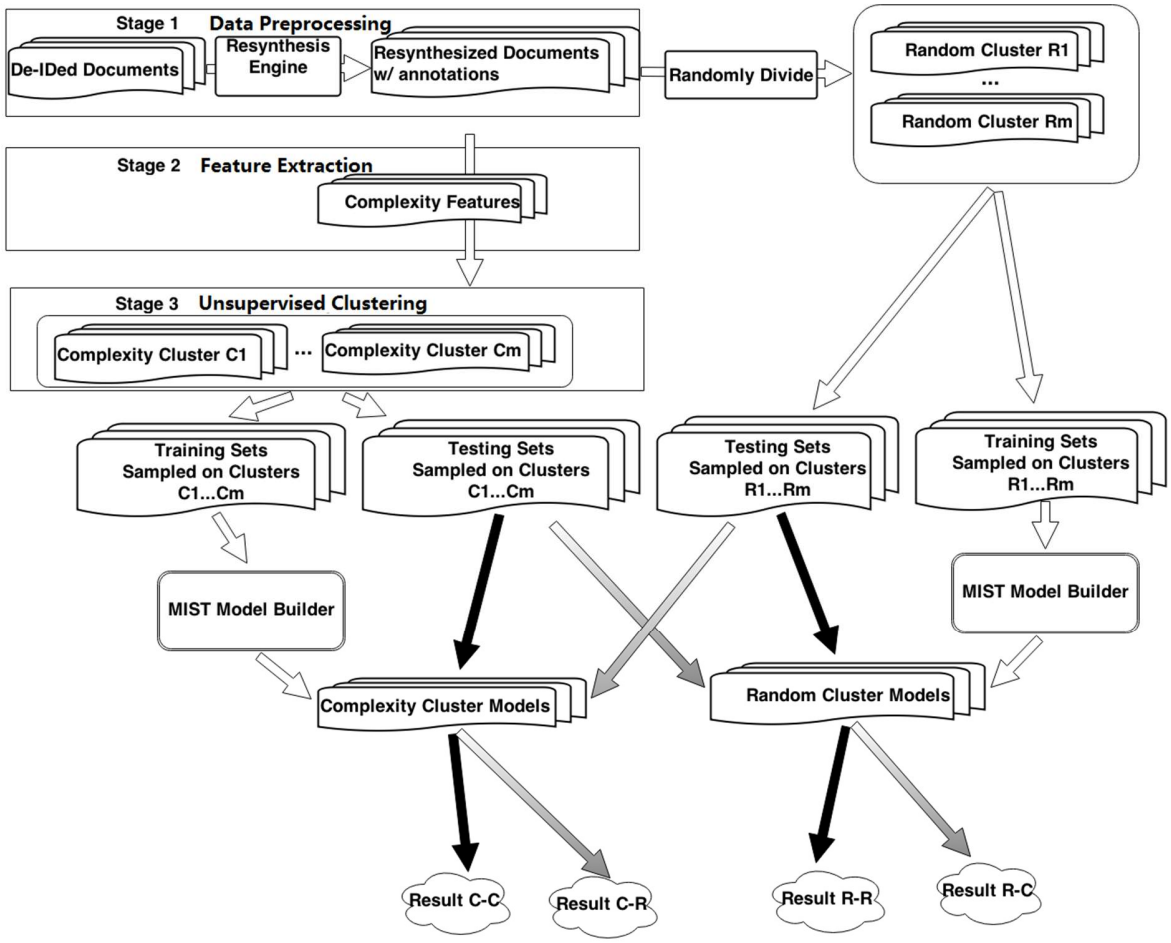


Figure 11. Framework for building, training, and testing de-identification models based on complexity measures and random processes.

3.2.2.1 Preprocessing

The DE-ID software tool in use at VUMC replaces instances of PHI with fillers of the corresponding type (e.g. "John Smith could be replaced by "**NAME[XXX WWW]"). To correctly calculate the writing style features and compare results with the i2B2 corpus, we transformed DE-ID'ed documents into realistic text because the fillers might interfere with the syntax or semantics of the sentences, and thus impact the complexity score computations. Realistic

text was also needed for training the machine-learned models that would subsequently be used to evaluate the impact of alternative clustering strategies on de-identification performance. The conversion from de-identified documents was accomplished using the MIST Resynthesis Engine [22].This software engine creates plausible replacements for the PHI fillers, generating documents looking realistic to users [28]. A fictional example of a clinical document pre- and post-resynthesis is depicted in Figure 12.

De-identified Record	**Resynthesized Record**
NAME*[ZZZ, YYY M] ***DATE*[Sep 08 2010] 13:18: **NAME[XXX WWW] (school nurse) calling needing to clear up some things like when pt has seizures at school when pt can be sent home. Pls call 615-PHONE* .	Martin, Jessie C. Dec 15 2010 13:18: Ashley Johnson (school nurse) calling needing to clear up some things like when pt has seizures at school when pt can be sent home. Pls call 615-331-7755.

Figure 12. An example of a de-identified and resynthesized clinical narrative.

3.2.2.2 Complexity and Richness Feature Extraction

The goal of complexity and richness feature extraction is to construct an appropriate feature set for each clinical document. This set is then translated into a corresponding feature matrix supplied to a machine learning algorithm, where each row and column of the matrix corresponds to an individual document and complexity feature value, respectively. For this work we used three types of measures to characterize document writing style. The first two correspond to the readability and lexical richness measures mentioned earlier. The implementation for the reading complexity measures and lexical richness measures were accomplished using the Korpus package in the R programming environment [71].

The third writing style measure we refer to as *clinical term richness*. Here, we justify this measure and describe its computation. While the two types of writing complexity we use have been in use for decades, they only characterize the complexity of a document's syntax. Our documents all derive from the clinical domain, but a wide range of healthcare workers compose

them (e.g., nurses, technicians, physicians from various medical specialties). We hypothesized that the characteristics of PHI in documents may correlate with the quantity of clinical language they contain, and that clinical language "richness" would be an important factor to consider when clustering documents for de-identification purposes. The clinical term richness measure was thus designed to represent the ratio of the sum of all clinical terms in a document to the sum of all of its terms, as shown in Table 7. To identify clinical terms we processed each document with MetaMap [72], which maps words and phrases to the Unified Medical Language System's (UMLS) Metathesaurus [73]. This clinical richness ratio was included with the writing complexity measures in each document's feature vector.

Name	Formula	Type	Dataset
Danielson.Bryan	DB = 1.0364 CPSp + 0.194 CPSt – 0.6059 (CPSp = characters per space, CPSt = characters per sentence)	Readability	Vanderbilt, i2b2
Dickes.Steiwer Shortcut	V* = 235.95993-(Var2 * 73.021) – (Var1 * 12.56438) – (Var3 * 50.03293) Var1 = ln((number of words/number of sentences)+1) Var2 = ln((number of letters/number of words)+1) Var3 = TTR(Type-Token-Ratio)	Readability	Vanderbilt
Flesch Reading Ease	RE = 206.835 – 0.846wl – 1.015sl (wl = number of syllables per 100 words; sl = average number of words per sentence)	Readability	Vanderbilt
Gunning FOG Index	FOG = 0.4(average sentence length + percentage of words of 3 or more syllables)	Readability	Vanderbilt
Linsear Write Index	Result = (number of easy words) + (3 * number of hard words)/number of sentences If result is > 20, divide by 2 for grade level. If number is < 20, subtract 2, then divide by 2 for grade level. Easy words: two syllables or less Hard words : three syllables or more	Readability	Vanderbilt
Carroll's corrected TTR	$CTTR(N) = \frac{V(N)}{2\sqrt{N}}$	Lexical Richness	Vanderbilt
Dugast's Uber Index	$U = \frac{log^2 N}{logN - logV(N)}$	Lexical Richness	Vanderbilt
Clinical Term Ratio	CTR = $\frac{CTN}{N}$ CTN : clinical terms number	Clinical Terms Richness	Vanderbilt, i2b2

(a) Formulas and types of complexity feature (part 1).

Coleman-Liau Index	**First estimates cloze percentage, then calculates grade equivalent:** $\boldsymbol{CL_{ECP} = 141.8401 - 0.214590 \times \frac{100 \times number\ of\ clozes}{number\ of\ words} + 1.079812 \times \frac{100 \times number\ of\ sentences}{number\ of\ words}}$ $\boldsymbol{CL_{grade} = -27.4004 \times \frac{CL_{ECP}}{100} + 23.06395}$	**Readability**	**i2b2**
Flesch-Kincaid Grade Level	$FK_{grade} = 0.39 \times \frac{number\ of\ words}{number\ of\ sentences} + 11.8 \times \frac{number\ of\ syllables}{number\ of\ words} - 15.59$	Readability	i2b2
FORCAST Readability Formula	$\text{FORCAST} = 20 - \frac{number\ of\ 1\ syllable\ words \times \frac{150}{number\ of\ words}}{10}$	Readability	i2b2
Lasbarhetsindex(LIX)	$\text{LIX} = \frac{number\ of\ words}{number\ of\ sentences} + \frac{number\ of\ words\ longer\ than\ 6\ characters \times 100}{number\ of\ words}$	Readability	i2b2
Kuntzsch's Text-Redundanz-Index(TRI)	$\text{TRI} = (0.449 \times number\ of\ 1\ syllable\ words) - (2.467 \times number\ of\ punctuation\ marks) - (0.937 \times number\ of\ foreign\ words) - 14.417$	Readability	i2b2
Yule's K (K.ld)	$\text{K} = 10^4 \times \frac{\left(\sum_{i=1}^{V(N)} V(i,N)\ \left(\frac{i}{N}\right)^2\right) - number\ of\ tokens}{(number\ of\ tokens)^2}$ V(i, N) : the numbers of word types occurring i times in a sample of length N	Lexical Richness	i2b2
Measure of Textual Lexical Diversity (MTLD)	MTLD = The mean length of sequential word strings in a text that maintain a given TTR value.	Lexical Richness	i2b2

(b) Formulas and types of complexity feature (part 2).

Table 7. Formulas and types of complexity feature and their associated formulaic representation. [63, 64, 68, 71, 74, 75, 76]

We started with 19 readability formulas and 12 lexical richness measures. To eliminate the effect caused by correlations between measures, we performed a feature selection process based on Regularized Random Forest (RRF) on the 31 features [77]. The process was performed independently for each of the corpora. The final feature sets for the VUMC and i2b2 datasets are shown in Table 7. The feature set for the VUMC corpus was composed of 8 measures: 5 associated with readability measurement, 2 with lexical richness assessment, and the clinical term richness measure. The fact that the latter measure was retained indicated that this feature was sufficiently distinct from standard writing complexity measures. The feature set for the i2b2 dataset was composed of 9 measures, again including the clinical term richness measure, among which 6 were related to readability measures, 2 with lexical richness.

3.2.2.3 Unsupervised Document Clustering

Clustering was performed in an unsupervised manner according to a hierarchical model, with Ward's coefficient [78] as the measure of goodness of fit. The number of clusters was determined by evaluating the quality of the clusters based on the Dunn index [79]. Using this criterion, we assigned each of the documents in our study corpus of the VUMC dataset to one of 13 clusters, which ranged in size from 60 to 672 documents. The i2b2 data was partitioned into 2 clusters, the sizes of which were 413 and 476, respectively.

3.3 Experiment Design and Results

3.3.1 Training and Testing Data Preparation

To perform our experiments, we defined the following sets of clusters for the Vanderbilt data:

1) *VUMC EHR-assigned Document Types.* This set of clusters was based on document type assigned to each document as it was composed in the EHR by a VUMC employee (Table 6). We refer to these clusters as T_1, …, T_9.

2) *Complexity Measures.* This set of clusters was based on the process described in the previous section. We refer to these clusters as C_1, …, C_{13}.

3) *Complexity Measure-based subsets.* These sets were randomly selected from clusters C_1, …, C_{13}, with varying size over {10, 20, 30, 50, 75, 100, 150, 200}, denoted as $Cs_{i,j}$, from which

i ranges from 1 to m and j corresponds to the number of randomly selected documents (e.g., $Cs_{i,10}$ corresponds to a set of 10 randomly selected documents from the i^{th} cluster).

4) *Random Processes*. We generated random groups of equal size for the stylometric clusters to ensure this type of cluster contains a comparable amount of data with the complexity-based clusters. In doing so, we derived a third set of clusters $R_1, \ldots, R_{13}$, such that $|C_i| = |R_i|$ for all clusters.

5) *Large random clusters.* After generating clusters from complexity measures, for each cluster C_i, we gather all the remaining clusters (i.e., $C_1, \ldots, C_{i-1}, C_{i+1}, \ldots, C_{13}$) to create a large random document set. We derived several random subsets from such large clusters, with varying size over {100, 200, 500, 1000}. These random subsets are denoted as $D_{i,j}$, where i ranges from 1 to m and j corresponds to the number of randomly selected documents (e.g., $D_{i,100}$ corresponds to 100 randomly selected documents not in the i^{th} cluster). The goal of experiments based on these subsets is to create a baseline for the evaluation of stylometric clusters.

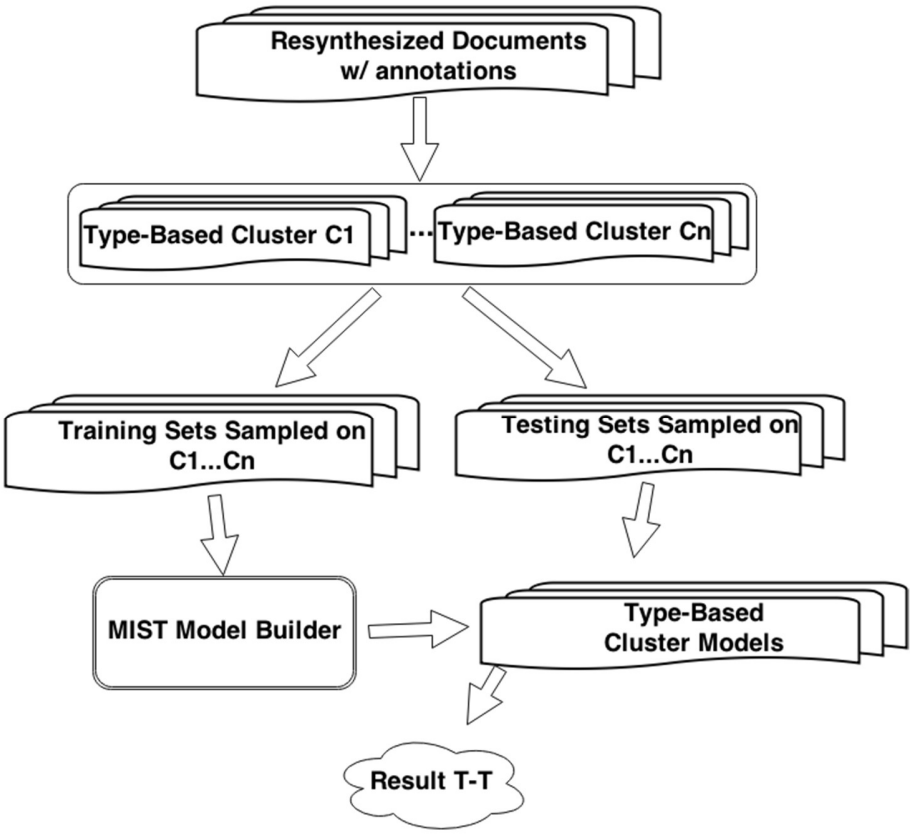


Figure 13. Framework for building, training, and testing de-identification models based on VUMC type designation.

For evaluation, we performed five sets of experiments using 10-fold cross-validation in each scenario. First, we trained and tested over each of the datasets, $\{C_1, \ldots, C_{13}\}$, $\{R_1, \ldots, R_{13}\}$, and $\{T_1, \ldots, T_9\}$ (as shown in Figure 11 and Figure 13). Second, within the stylometric and random clusters, we evaluated the performance of the learned models in a cross-dataset manner (as shown in Figure 11). For insta1nce, we train a model over C_i and test it over $C_1, \ldots, C_{i-1}, C_{i+1}, \ldots, C_m$. Third, we trained on each stylometric cluster and tested with the corresponding same-size random set. Fourth, we evaluated the training performance of the large random clusters when testing on the stylometric clusters (as shown in Figure B1 of Appendix B1). Specifically, we trained on each of the random subsets $D_{i,1}, \ldots, D_{i,s}$ and tested against C_i. For the final set of experiments, which aim to evaluate the training sensitivity of the stylometric clusters, we trained and tested over all the subsets of such clusters, $\{Cs_{1,10}, \ldots, Cs_{m,k}\}$ (as shown in Figure B2 of Appendix B1).

3.3.2 De-identification Model Training

Separate de-identification models were trained for each cluster of documents using default MIST settings. The resulting models were used in their respective experiments to evaluate de-identification performance.

3.3.3 Evaluation Measures

We report our results for each of the experiments using standard information retrieval measures. In the context of this study, these measures are defined as follows:

$$(p)\text{recision} = \frac{number\ of\ PHI\ instances\ correctly\ labeled}{number\ of\ PHI\ instances\ labeled\ by\ the\ system}$$

$$(r)\text{ecall} = \frac{number\ of\ PHI\ instances\ correctly\ labeled}{number\ of\ PHI\ instances\ labeled\ in\ the\ gold\ standard}$$

$$(F)\text{-measure} = \frac{2p}{p+r}$$

3.3.4 Results

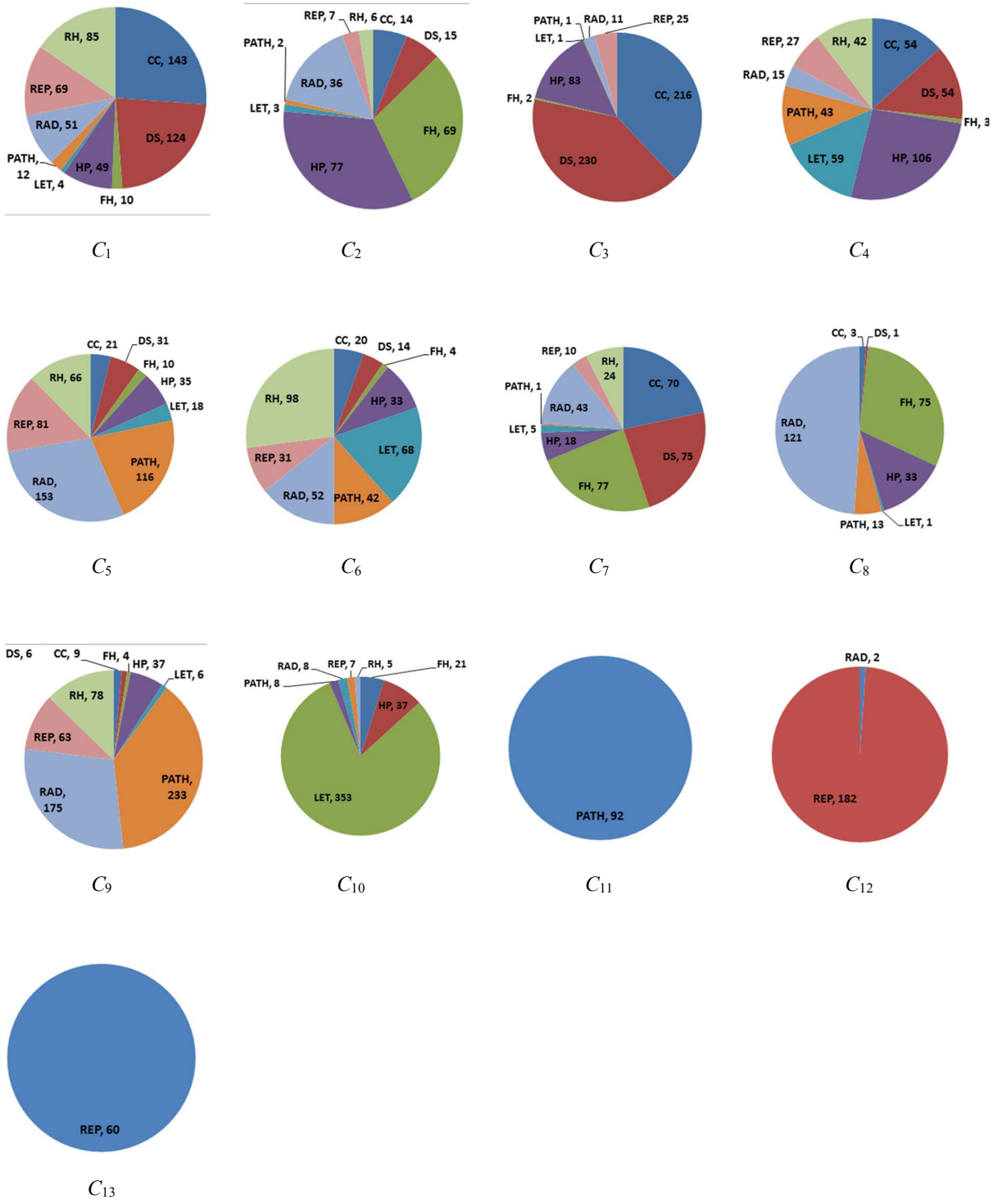


Figure 14. VUMC document type distribution for stylometric clusters.

To provide insight into the composition of stylometric clusters, Figure 14 reports on the distribution of document types for each cluster. These charts show that 9 out of 13 stylometric clusters are highly heterogeneous mixtures of VUMC document types, which confirms that our clustering method is not merely replicating the grouping of the document types. The clusters that were relatively homogenous in document type consisted of general reports (C_{12} and C_{13}), letters (C_{10}), and pathology notes (C_{11}).

Test Cluster	Min	Max	Self	Min Cluster	Max Cluster (if not self)
Stylometric Clusters					
C_1 (n = 547)	0.172	0.921	0.921	C_{13}	---
C_2 (n = 229)	0.157	0.881	0.851	C_{13}	C_3
C_3 (n = 569)	0.198	0.944	0.944	C_{13}	---
C_4 (n = 403)	0.176	0.872	0.872	C_{13}	---
C_5 (n = 531)	0.15	0.925	0.915	C_{13}	C_6
C_6 (n = 362)	0.13	0.884	0.878	C_{13}	C_{10}
C_7 (n = 323)	0.169	0.941	0.922	C_{13}	C_4
C_8 (n = 247)	0.106	0.897	0.896	C_{13}	C_6
C_9 (n = 611)	0.125	0.906	0.906	C_{13}	---
C_{10} (n = 439)	0.148	0.889	0.889	C_{13}	---
C_{11} (n = 92)	0.492	0.993	0.96	C_{13}	C_4
C_{12} (n = 184)	0.246	0.977	0.977	C_{13}	---
C_{13} (n = 60)	0.101	0.99	0.99	C_{12}	---
Random Clusters					
R_1 (n = 547)	0.791	0.894	0.891	R_{13}	R_7
R_2 (n = 229)	0.809	0.903	0.886	R_{13}	R_7
R_3 (n = 569)	0.81	0.916	0.896	R_{13}	R_7
R_4 (n = 403)	0.793	0.899	0.892	R_{13}	R_7
R_5 (n = 531)	0.801	0.903	0.888	R_{13}	R_7
R_6 (n = 362)	0.786	0.89	0.869	R_{13}	R_7
R_7 (n = 323)	0.805	0.902	0.902	R_{13}	---
R_8 (n = 247)	0.809	0.906	0.903	R_{13}	R_7
R_9 (n = 611)	0.803	0.899	0.89	R_{13}	R_7
R_{10} (n = 439)	0.799	0.913	0.858	R_{13}	R_5
R_{11} (n = 92)	0.783	0.9	0.88	R_{13}	R_2
R_{12} (n = 184)	0.775	0.896	0.836	R_{13}	R_8
R_{13} (n = 60)	0.869	0.959	0.869	R_{13}	R_5

Table 8. VUMC de-identification performance (in terms of F-measure) based on clusters derived from the complexity measures and random process.

Clusters Compared	C ⇒ R	R ⇒ C
C_1 vs R_1	0.866	0.914
C_2 vs R_2	0.756	0.893
C_3 vs R_3	0.835	0.937
C_4 vs R_4	0.884	0.858
C_5 vs R_5	0.844	0.929
C_6 vs R_6	0.877	0.872
C_7 vs R_7	0.800	0.936
C_8 vs R_8	0.831	0.863
C_9 vs R_9	0.841	0.881
C_{10} vs R_{10}	0.860	0.790
C_{11} vs R_{11}	0.434	0.967
C_{12} vs R_{12}	0.746	0.835
C_{13} vs R_{13}	0.172	0.551
Average	*0.750*	*0.864*
St. Dev.	*0.210*	*0.105*

Table 9. VUMC de-identification performance, in terms of F-measure, for cross-cluster experiments. (Training Cluster ⇒ Test Cluster)

Tables 8 and 9 summarize the evaluation results for the various de-identification models with the Vanderbilt dataset. For brevity, tables 8 and 9 focus on results of the Vanderbilt data with respect to the F-measure. In general, recall and precision were balanced (the full results including recall and precision are presented in Appendix B2). We would like to note that our evaluation used the DE-ID processed documents as a gold standard, which indicates the impact of the DE-ID performance on our results. Table 10 presents the total number of PHI types and instances in training de-identification models for stylometric clusters.

Cluster	Number of PHI Types in Training	Average Number Of PHI Instances in Training
C_1 (n = 547)	13	12912
C_2 (n = 229)	9	1036
C_3 (n = 569)	14	17303
C_4 (n = 403)	14	9688
C_5 (n = 531)	13	6207
C_6 (n = 362)	13	8613
C_7 (n = 323)	11	3974
C_8 (n = 247)	11	7944
C_9 (n = 611)	13	9145
C_{10} (n = 439)	12	17455
C_{11} (n = 92)	4	281.7
C_{12} (n = 184)	10	3648.6
C_{13} (n = 60)	3	3231
Cluster (i2b2 dataset)	**No. of PHI Types in Training**	**Average No. of PHI Instances in Training**
IC_1 (n = 413)	8	24259.5
IC_2 (n = 476)	8	23817.6

Table 10. Number of PHI types and instances in each of the stylometric clusters in the VUMC corpus.

3.3.4.1 De-identification Training on Complexity and Random Clusters

In Table 8, each row represents a test dataset. Since we tested models trained on each of the clusters, for each column, we report the best (*max*) and worst (*min*) F-measures with corresponding training datasets. We also report the F-measure of the model trained on the corresponding testing cluster (*self*). The results presented in the first half of the table were generated by the scenario in which training and testing was performed with clusters derived from the complexity measures, while the second half correspond to random clusters.

Regarding the stylometric experiment, generally speaking, in seven of the clusters (C_1, C_3, C_4, C_9, C_{10}, C_{12}, C_{13}), training and testing on the same cluster (i.e., C_i vs. C_i) performed better than testing with models based on different clusters (i.e., C_i vs. C_j, $i \neq j$). For example, when testing on

C_{13}, which is the smallest (60 documents), the model trained on documents from this cluster achieved an F-measure of 0.990, but the F-measure never reached higher than 0.808 for C_3 with models based on other clusters, in which cases the F-measure dropped as far as 0.101 for C_{12}. The possible explanation could be found in the limited number of PHI types and relatively sufficient number of PHI instances in the cluster, as shown in Table 10. It also suggests that training and testing on the same cluster might eliminate the limitation of insufficient training data. Specifically, training on smaller clusters (e.g., C_{12}, C_{13}) performed poorly when evaluating on clusters other than themselves, but exceptionally well when testing on their own.

The pattern found in the previous tests, however, did not hold true in the scenario where training and testing was performed with clusters derived from a random process. The results found in these experiments confirmed our hypothesis that training and testing on the same random clusters would not necessarily yield the best performance comparing with training and testing on different clusters based on random process. Rather, de-identification models trained on different clusters showed similar variations in testing, despite the fact that they were testing on their own or on other random clusters. Specifically, models created by R_{13} always performed worse than other training models (with the F-measure ranging from 0.775 to 0.869) including testing on itself. R_7 on the other hand was the best training set, whose F-measures dominated 9 out of all 13 testing sets, including testing on its own. Unlike the high variation in experiments for complexity-measure-based clusters, the F-measure was bounded between 0.775 and 0.959 in this half of Table 8.

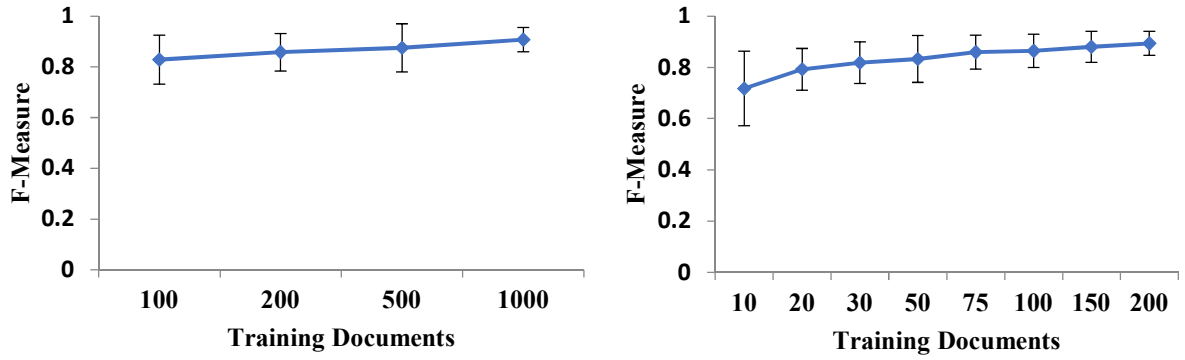
3.3.4.2 De-identification Training and Testing Crossing Cluster Types

Table 9 reports on the evaluation of experiments crossing cluster types. In the columns under $C \Rightarrow R$ we trained on stylometric clusters and tested on random clusters of the same size, while in $R \Rightarrow C$ we reversed the process. Comparing Tables 8 and 9, it can be seen that the stylometric models performed worse than random models on the random clusters, with an F-measure between 0.172 and 0.884. When testing on stylometric clusters, nine of the models trained on the testing clusters themselves performed better than random training models. R_2, R_5, R_7 and R_{11} achieved better models than their stylometric counterparts C_2, C_5, C_7 and C_{11}.

However, recalling the fact that six of the stylometric clusters showed best testing results with training models of clusters other than themselves, the four clusters (C_2, C_5, C_7 and C_{11}) were

all included. When testing on C_7 and C_{11}, models trained on C_4 yielded the best performance over all stylometric clusters, with F-measures of 0.941 and 0.993 respectively, which also surpassed the best performances of models by random clusters R_7 and R_{11}. For testing on C_2 and C_5, models trained on C_3 and C_6 yielded the best F-measures (0.881 and 0.925, respectively), both on par with the random models R_2 and R_5 (F-measures of 0.893 and 0.929, respectively).

3.3.4.3 De-identification Training on Large Random Clusters



(a) Random Process Clusters (b) Stylometric Clusters

Figure 15. Average F-measure (+/- 1 standard deviation) of de-identification models as a function of the training (subset) set size for left) large random mixture clusters and right) stylometric clusters.

The average performance for de-identification models trained on different sized subsets of large random-process based clusters and tested on complexity-measure based clusters is depicted in Figure 15 left). Figures depicting the performance of each of the large random clusters are included in Appendix B3. Overall, larger size training sets performed better than smaller size sets, as shown in the left of Figure 15. For instance, the F-measure of 100 documents for $D_{10,100}$ was 0.733, which increased with the sample size, such that by 1000 documents, the F-measure was 0.804.

In comparison to the results from Table 8, the F-measures for training and testing on a particular stylometric cluster C_i generally performed better than a sample size of 200 documents from D_i. It can be seen that 8 out of the 13 complexity clusters outperformed subsets of size 500 and five (C_4, C_9, C_{10}, C_{12}, and C_{13}) yielded better F-measures than the random subsets of 1000 documents. Specifically, C_{10}, C_{12} and C_{13} contain the least number of documents of all clusters -

each with less than 200 documents. These findings indicate that for smaller clusters, training and testing on their own may achieve better performance than training models over a much larger dataset of randomly selected documents (while C_{11} was an exception to this trend).

3.3.4.4 De-identification Training on Subsets of Complexity Based Clusters

The experimental results for de-identification models trained and tested on varying sizes of subsets from the complexity-measure based clusters are depicted in online Appendix B3, for evaluating the scalability of such clusters. The right of Figure 15 plots the average performance of stylometric clusters. Similar to the large random datasets, increasing sample size seems to improve de-identification model training.

Recalling the results from large mixed random clusters, a subset of C_{10} with the size of 75 already outperformed the corresponding random dataset of 1000 documents in terms of F-measure - the former achieved 0.813 while the latter achieved 0.804. With C_{12} and $C_{13,}$ the 50-document subset yielded better F-measures than the 1000-document random clusters, a subset of 10 documents could achieve an F-measure of 0.961, while a random dataset of 1000 documents only achieved 0.859. Such findings suggest that smaller clusters tend to be homogeneous, for which training on a limited number of documents of their own group would surpass training on much larger random clusters.

3.3.4.5 De-identification Training on Clusters based on VUMC Designated Type

Beyond the statistics for VUMC-designated document type, Table 6 reports the testing results of training de-identification models based on these types. The overall performance was worse than the complexity-measures based clusters scenario. Here, it can be seen that documents of type CC generated the highest scores in all the measurements, with an F-measure of 0.964, and type FH produced the lowest F-measure of 0.842. Comparing all scenarios, the average F-measures of training and testing on the same cluster are ordered as follows: complexity measures clustering (0.917) > document type clustering (0.911) > random clustering (0.881). The F-scores of the complexity-based clusters were significantly higher than those with random clusters ($t = 2.79$, $p = 0.006$); meanwhile, their F-score difference between the complexity clustering and document type based clustering were not statistically significant.

3.3.4.6 De-identification Training on i2b2 Data

Table 11 summarizes the experiment results on the i2b2 dataset. The training and testing procedures were consistent with the Vanderbilt dataset, except that the type-based clustering was not possible because the i2b2 data only contains discharge summaries.

Train ⇒	**IC_1**			**IC_2**		
⇓Test	**P**	**R**	**F**	**P**	**R**	**F**
IC_1 (*n* = 413)	0.964	0.955	0.959	0.963	0.948	0.955
IC_2 (*n* = 476)	0.961	0.936	0.948	0.974	0.970	0.972

Train ⇒	**IR_1**			**IR_2**		
⇓Test	**P**	**R**	**F**	**P**	**R**	**F**
IR_1 (*n* = 413)	0.970	0.960	0.965	0.966	0.962	0.964
IR_2 (*n* = 476)	0.969	0.960	0.965	0.967	0.962	0.965

Table 11. i2b2 de-identification performance (in terms of Precision, Recall, and F-measure) based on complexity clusters and random clusters

For comparison, we randomly divided the dataset into 2 clusters (IR_1 and IR_2) with the same sizes as the stylometric clusters (413 and 476 documents).

It can be seen that, for the stylometric clusters, training and testing on the same cluster always yielded better F-scores than training and testing on different clusters. IC_2 performed best with an F-measure of 0.972. However, the general performance of both stylometric clusters were on par, with the random clusters. Specific, the stylometric and random cluster F-scores were 0.966 and 0.965, respectively. As such, there was no statistical significant difference.

However, the results are not entirely unsurprising. The findings suggest that the i2b2 dataset is a highly homogenous corpus and lacks the variance in writing styles that is amenable to

our stylometric clustering strategy. This is consistent with the fact that the dataset consists entirely of discharge summaries from Partners Healthcare [27].

3.4 Discussion

There are several notable aspects of the experimental analysis to highlight. First, the results confirm the hypothesis that training on a stylometric cluster yields better de-identification performance when testing on the same cluster. Second, it was further confirmed that stylometric de-identification models yield results that are better than models based on random collections of documents (which is akin to training de-identification models on mixtures of many document types as is common done). One possible explanation of this observation is the writing style features lead to models that account for grammar and phrasing, which are critical in natural language processing. As a result, the models learned over these features could yield better performance than the ones based on random features. Third, the performance of stylometric models is, in many instances, better than those derived from VUMC-designated document types. In combination, these findings suggest that higher fidelity de-identification models can be composed with less training data and institutional knowledge. This is critical because there are potentially hundreds to thousands of document types generated in healthcare settings. Though the proposed strategy does not always yield the best de-identification model, we believe it is more scalable because of its hierarchical clustering strategy, which minimizes the number of clusters and thus de-identification models we apply. We note that hierarchical grouping could also be performed on document types to reduce the number of total documents types, but this will still yield random groups due to the inability to partition predefined types.

Despite the positive nature of the results, we recognize that our study is limited in its scope for several reasons. First, we used a VUMC-specialized version of DE-ID as a proxy for a human-annotated corpus. This technology is oriented toward PHI recall and, thus may over-redact, which could explain why our F-measures are a little lower than those observed in gold standard environments. Second, our data was mainly based on the VUMC, a single healthcare organization. While there is data available from other sites, such as the i2b2 corpus, they tend to be composed solely of a single type (e.g., discharge summaries) and are relatively small in size (e.g., ~1500). The clustering experiment on the i2b2 data showed limited improvement on de-identification model training, indicating the homogeneity of i2b2 data obstructed the writing stylometric

clustering. Third, the VUMC EMR contains over 1000 document types because EMR users are allowed to create custom document types, but we chose to focus on only nine of the larger size types in this setting to report on interpretable results and to avoid an overly complex analysis.

Finally, while our findings lend credibility to the claim that features based on readability and lexical richness can lead to better de-identification models than random groups of documents, we acknowledge that such features may be correlated with other unknown factors. For instance, though unlikely, the MIST resynthesis engine may embed behind a certain pattern in the new PHI instances. This pattern could, in turn, influence the grammar and phrasing. Second, the DE-ID process might induce a pattern in the documents as well. For instance, DE-ID obscures the distribution of character length of PHI instances, which could, in turn, bias the resynthesis process. Third, our findings are mainly based on documents from a single institution; i.e., the VUMC. It is possible that data from a different institution might not exhibit as robust a clustering or distinction in performance.

Chapter 4 Active Learning for De-identification

Given that modern EMR systems manage data on millions of patients, it is critical to develop de-identification routines for such data in a manner that are both effective and efficient [21]. However, creating a gold standard corpus for training de-identification models can be excessively costly in practice [80, 81], such that incorporating an active learning in the process may reduce the overall cost for annotation and, thus, support the establishment of a more scalable de-identification pipeline. Instead of randomly sampling a dataset for training purposes (or what is called passive learning), active learning works by allowing the machine learning system to select the data to be annotated by a human oracle and added to the set of training data iteratively [82]. The system is expected to learn and improve its accuracy throughout the process, while eventually fewer instances annotated by humans are required than passive learning [83]. For this chapter of the dissertation, we report on an active learning pipeline for de-identification based on machine learning. In doing so, we assess the extent to which active learning can lead to better results than passive learning (i.e., randomly selecting documents to be annotated by humans for the purposes of training a classifier). Our hypothesis is that, with machine learning based de-identification systems actively requesting more informative data that helps to create a better model from human annotators, less training data will be needed in the machine learning process to maintain (or even improve) the performance of trained models for de-identification.

This chapter is organized as follows. We first review existing active learning applications in NLP (especially for named entity recognition tasks) and specifically with clinical documents. Next, we establish an active learning workflow for natural language de-identification and introduce several new heuristics for solving the key problem of active learning (i.e., choosing the most informative data for annotation), which is what makes our work notable. We then conduct a series of controlled and systematic experiments on a real world dataset of clinical study reports (CSRs) for a clinical trial (company name of which withheld for business confidentiality reasons) and the publicly accessible i2b2 dataset [27] on the active learning pipeline, while evaluating the performance of the heuristics. We show that, in general, active learning can yield a comparable and, at times, better performance, with less training data than passive learning.

4.1 Background

In this section, we review 1) query strategies in related research of active learning, 2) applications of active learning specifically with clinical documents.

4.1.1 Active learning query strategies

Several query strategies for active learning in sequence labeling tasks with CRFs were investigated and compared by Settles [82], including uncertainty sampling, information density, Fisher information, and query-by-committee, some of which are more computationally costly in practice.

Kapoor [84] introduced a decision-theoretic active learning approach with Gaussian process classifiers and evaluated the framework with voice messages classification. The work utilized the expected value of information (VOI) that accounts for the cost of misclassification.

Also, the concept of return on investment (ROI) was implemented in active learning to account for the cost of annotation [85], which was assessed on a part of speech tagging task. In our work, we introduce and assess query strategies based on uncertainty sampling and the notion of return on investment (ROI) for de-identification, as these approaches are more practical in real world.

4.1.2 Active learning with clinical documents

Active learning has been shown to be an effective tool in named entity recognition tasks in clinical text [86]. The study simulated several selection strategies including both uncertainty and diversity sampling, these findings suggest that active learning is more efficient than passive learning in most cases. They further suggested that uncertainty sampling was the best strategy for reducing the annotation cost. The results implied that human annotation cost should be taken into account when evaluating the performance of active learning.

In the context of de-identification, Bostrom and colleagues [87] proposed an active learning approach that relied upon a random forest classifier. They evaluated the approach with a dataset of 100 Swedish EMRs. In their framework, the query strategy to determine which documents humans should annotate next focused on entropy-based uncertainty sampling. However, this investigation was limited in several notable ways. First, entropy-based uncertainty sampling does

not explicitly account for the human annotation cost, such that we introduce and implement several query strategies in our system beyond. Second, we perform an expanded investigation and conduct controlled experiments using real, as well as a publicly accessible resynthesized, EMR data.

Recently, Fong [88] developed an active learning workflow for Patient Safety Events (PSE) identification with SVM and showed that active learning helped in identifying health information technology (HIT) - related events.

4.2 Methods

We adopt a pool-based active learning framework [82]. In our scenario, this means that there is a limited amount of annotated data and a large pool of unannotated data the framework could select from.

The pipeline for the active learning framework for natural language de-identification is illustrated in Figure 16. Initially, we use a small batch of data that is selected randomly from the dataset as the starting point of the active learning. The human annotators then manually tag the PHI in the initial batch of data to create a gold standard dataset for de-identification model training.

Since human annotation is costly, the goal of active learning is to reduce the total amount of human annotation needed in the process while maintaining (or even improving) the performance of de-identification model training. In reality, the human effort involved in the framework can be viewed as two parts: 1) the human annotation effort in gold standard creation and 2) the human correction effort that is needed to fix incorrect labels generated in the previous round when the de-identification model is applied to unannotated data (because no reasonable existing automatic de-identification approaches yield a recall of 100%). After the first batch of gold standard data is created by human annotators, we train a de-identification model, which is applied to the remainder of the unannotated data. The active learning pipeline then queries for more informative data to be corrected by humans based on the performance of previous models and/or additional criteria. This information is expected to assist in better de-identification model development. Another way to view this strategy is, instead of randomly selecting a fixed amount of unannotated data for training data (passive learning), the system actively queries for the data that potentially contributes more information in model training.

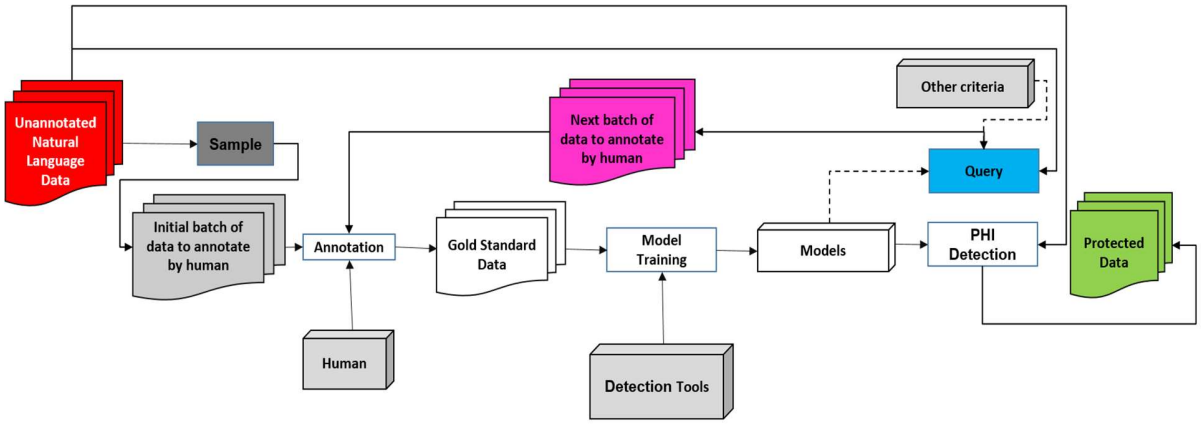


Figure 16. Pipeline of the active learning framework for natural language de-identification.

This begs the question: how should we select the data that is more informative? Since the query for active learning is based on a heuristic, we propose and develop several query strategies for our active learning de-identification framework and compare the performances with our simulation. Before providing the details of each strategy, we formalize the problem statement.

4.2.1 Problem Formulation

Let D be a set of documents, D_L and D_U be the set of annotated and unannotated documents, where $D = D_L \cup D_U$. D_U consists of n documents, $d_1, d_2, \ldots, d_n$.

Let $\mathrm{Q}(d_i)$ be the query strategy that the active learning framework utilizes to select additional documents for human annotation. Note that, we choose documents rather than tokens as the unit for selection, as it is impractical for a human annotator to annotate tokens out of context, also selecting tokens could lead to a significant computational overhead since it takes time to retrain the model each time the training dataset is updated.

The goal of the query step is to choose the data d_c that maximizes $\mathrm{Q}(d_i)$,

$$d_c = arg\max_{d_i \in D_U} (\mathrm{Q}(d_i)).$$

At this point, let D_S be the selected batch of k documents for a human to annotate, In essence, this consists of the k documents that maximize $\mathrm{Q}(d_i)$. Note that the value of k could depend on the learning rate of the framework, as well as the time that it takes to retrain and reannotate.

Once D_S has been corrected by the human, it is removed from the unannotated document set D_U, while D_L is updated to include the annotated batch of documents:

$D_L' = D_L \cup D_S,\ D_U' = D_U \backslash D_S$.

Each time D_S is annotated and added to training, the de-identification model needs to be retrained using D_L'. Additionally, the unannotated documents will become part of the annotated set, based on the updated model. Next, we introduce several options for query strategies that we utilize in our system.

4.2.2 Uncertainty Sampling

One of the more prevalent query strategies for active learning is uncertainty sampling [82]. In this model, it is assumed that the active learning system picks the data that the current model is most uncertain of when making predictions.

4.2.2.1 Least confidence (LC)

For a CRF model, given a token x, let y be the most likely predicted label of x (e.g., a patient's name or a date) and let $P(y|x)$ be the posterior probability. Then $P(y|x)$ is the confidence score of x given the current model. We next define the uncertainty of token x as $1 - P(y|x)$. Note that, we aim to find the document for which the current model has the least confidence. Upon doing so, we could either use the summation of the LC-based uncertainty of all tokens in a document d_i:,

$$UC(d_i) = \sum_t (1 - P(y_t|x_t)) \tag{7}$$

or the mean of all token uncertainty based on LC:

$$UC(d_i) = \frac{\sum_t (1 - P(y_t|x_t))}{l_i} \tag{8}$$

where l_i is the total number of tokens in d_i.

The problem with adopting the mean of all token uncertainty is that it neglects the length of the documents in the selection process, which may not be optimal.

One of the initial findings of employing the sum of token uncertainty approach is that the predicted non-PHI tokens are more likely to produce a prediction confidence score of 0.95 or higher, while the confidence of the tokens that are predicted as PHI are, in most cases, lower. Also, since the selection aims for documents with a higher sum of token uncertainty, it tends to be biased towards documents that contain a larger amount of tokens, even though the PHI density in the selected set of documents could be low (which is not desirable for model training).

For a simple illustration, imagine we have the following two documents d_1 and d_2, the information of the two documents is summarized in Table 12.

Statistic	**Document d_1**	**Document d_2**
Total number of non-PHI token	1000	100
Average non-PHI token confidence	0.99	0.99
Total number of PHI token	5	10
Average PHI token confidence	0.6	0.6
PHI density	0.5%	9.1%
Sum of token uncertainty	12	5

Table 12. An example of how the number of tokens and PHI density influence the sum of token uncertainty of documents.

Note that, document $\boldsymbol{d_2}$ consists of a much higher PHI density than document $\boldsymbol{d_1}$ and might provide more information in de-identification model training. Nonetheless, uncertainty sampling will more likely choose document $\boldsymbol{d_1}$ over document $\boldsymbol{d_2}$ due to a higher sum of token uncertainty.

To mitigate this problem, we introduce a modified version of the least confidence approach, which we refer to as least confidence with upper bound (LCUB). In this variation, instead of summing the uncertainty of all tokens, the framework calculates the sum of uncertainty of tokens, when $P(y_t|x_t) < \theta$, where θ is a cutoff value for uncertainty sampling.

Now, let $\sum f(x_t, \theta)$be the modified sum of token uncertainty with cutoff value θ:

$$f(x_t, \theta) = \begin{cases} (1 - P(y_t|x_t), P(y_t|x_t) < \theta \\ 0, \quad P(y_t|x_t) \geq \theta \end{cases} \tag{9}$$

4.2.2.2 Entropy

Entropy measures the potential amount of discriminative information available. Given a token x, its entropy $\mathrm{H}(x)$ is computed as:

$$\mathrm{H}(x) = -\sum_{j}^{m} P(y_j|x) log P(y_j|x) \tag{10}$$

where m corresponds to the number of most probable labels of x, as predicted by the current classification model (e.g., the CRF). Here $P(y_j|x)$ is the probability that x's label is y_j.

Again, for a document d_i that contains t tokens, the total entropy-based uncertainty of d_i can be calculated as:

$$UC(d_i) = -\sum_{t}\sum_{j} P(y_{tj}|x_t) log P(y_{tj}|x_t) \tag{11}$$

Similar to the LC approach, entropy-based uncertainty also tends to suffer from the problem of low PHI density documents. To mitigate this issue, we introduce an entropy with lower bound (ELB) approach. In this approach, we set a minimum threshold ρ for token entropy.

Thus, let $\sum g(x_t, \rho)$ be the modified sum of token entropy with minimum value ρ :

$$g(x_t, \rho) = \begin{cases} \mathrm{H}(x_t), & \mathrm{H}(x_t) > \rho \\ 0, & \mathrm{H}(x_t) \leq \rho \end{cases} \tag{12}$$

4.2.3 Return on Investment

The goal of active learning is to reduce the human effort needed in the machine learning process. Both the least confidence and the entropy-based uncertainty sampling methods seek to solve the problem by minimizing the training data required. However, this implicitly assumes that the cost for human annotation is fixed and is not explicitly modeled during the query step. In reality, we need to acknowledge that the effort that a human annotator spends is more complex than the above assumption. Consider, it is likely that the cost varies based on PHI types, error types, human fatigue (due to the number, or length, of documents), among other factors. Additionally, the contribution of human correction towards a better model can also vary according to various factors, such as PHI types and error types. Thus, we designed a query strategy that accounts for both the cost and the contribution of human correction.

We assume there is a reading cost for the human annotator that is proportional to the length of the document that is being annotated. The average reading cost per token is denoted by ct_r, which implies that the total reading cost for a document d_i of length l_i is $ct_r \times l_i$.

Again, we start formalizing the problem considering a given token x. Let y be the most likely label of x. $P(y|x)$ is the probability that the active learning system assigns y as the label of x, while $P'(y|x)$ is the true probability that x is of class y.

Without loss of generality, here, we consider only a two-class of the problem, PHI versus non-PHI. This indicates that we assume the annotation cost and the human contribution of correcting a PHI instance classified as the wrong PHI type cancel each other out. Let ct_n and ct_p be the human correction cost of correcting a false negative (or FN) instance (i.e., a token mistakenly labeled by the learned model as non-PHI) and a false positive (or FP) instance (i.e., a non-PHI token mistakenly labeled by the current model as PHI), respectively. Similarly, cn_n and cn_p represent the human correction contribution of correcting an FN instance and correcting an FP instance, respectively.

Thus, the expected total contribution of human correction for token x when y is a non-PHI instance can be defined as $TCCN(x)$ and calculated as:

$$TCCN(x) = cn_n \times P(y|x) \times \big(1 - P'(y|x)\big) + cn_p \times (1 - P(y|x)) \times P'(y|x) \quad (13)$$

The expected total cost of human correction for token x when y is a non-PHI instance is represented by $TCCT(x)$, then

$$TCCT(x) = ct_n \times P(y|x) \times \big(1 - P'(y|x)\big) + ct_p \times (1 - P(y|x)) \times P'(y|x) \quad (14)$$

Then, we have the expected return on investment (ROI) of token x labeled as non-PHI:

$$ROI(x) = (cn_n - ct_n) \times P(y|x) \times \big(1 - P'(y|x)\big) + \big(cn_p - ct_p\big) \times \big(1 - P(y|x)\big) \times P'^{(y|x)} - ct_r \quad (15)$$

Similarly, the expected ROI of token x labeled as PHI:

$$ROI(x) = \left(cn_p - ct_p\right) \times P(y|x) \times \left(1 - P'(y|x)\right) + (cn_n - ct_n) \times \left(1 - P(y|x)\right) \times P'(y|x) - ct_r \quad (16)$$

Note that $cn_n - ct_n$ and $cn_p - ct_p$ represent the net contribution of correcting an FN instance and an FP instance, respectively. At this point, let NC_n and NC_p denote the net contribution of an FN instance correction and an FP instance correction, respectively. Thus,

$$ROI(x) = \begin{cases} NC_n \times P(y|x) \times \left(1 - P'^{(y|x)}\right) + \\ NC_p \times \left(1 - P(y|x)\right) \times P'(y|x) - ct_r, \quad y\ is\ non - PHI \\ NC_p \times P(y|x) \times \left(1 - P'^{(y|x)}\right) + \\ NC_n \times \left(1 - P(y|x)\right) \times P'(y|x) - ct_r, \quad y\ is\ PHI \end{cases} \quad (17)$$

Consequently, the total expected ROI of unannotated document d_i is:

$$ROI(d_i) = \sum_t ROI(x_t) \quad (18)$$

Finally, it is desirable that the active learning pipeline picks documents that could maximize ROI.

4.3 Experiment Design and results

4.3.1 Dataset

After constructing the active learning pipeline, we utilized two datasets for simulation and evaluation: 1) a dataset drawn from a healthcare organization (anonymized due to contractual agreements), which we refer to as dataset A and 2) a publicly available dataset from the i2b2 de-identification challenge [27].

Dataset A consists of 370 documents with a total number of 312991 tokens, and 7098 PHI instances from 12 PHI types. Note that certain PHI types contain no more than a couple of instances in this corpus (e.g., COUNTRY, DOB, MIDDLE_NAME, and SITE_ID), which makes it challenging to train an effective model. This, in turn, leads to considerably lower recall scores in the experiments.

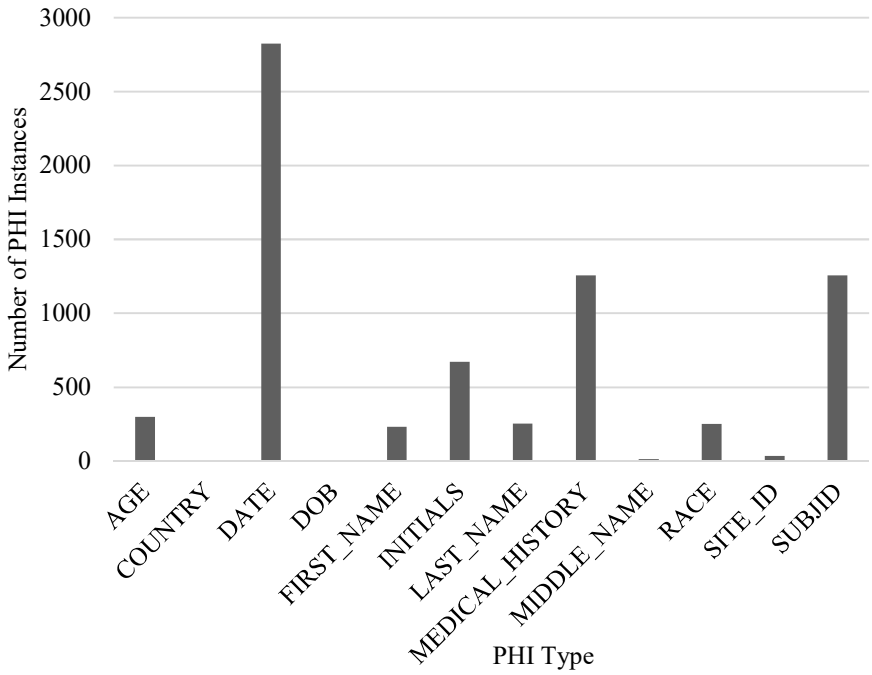
To assess the de-identification performance of this whole dataset, we conducted a 10-fold cross-validation on this corpus before simulating the active learning process. Each time we trained

a de-identification model with 90% of the documents and tested the model on the remaining 10% of the data. The average precision, recall and F-measure for the PHI types in this dataset are summarized in Table 13.

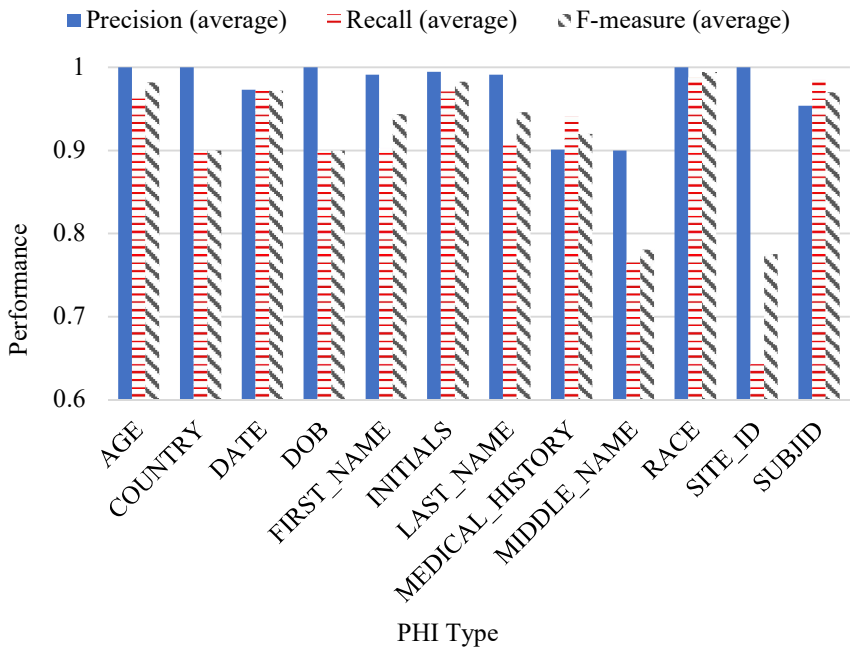
Note that the FIRST_NAME, LAST_NAME and MEDICAL_HISTORY types appear to be the most difficult to detect among all types (Figure 17), the possible reasons of which could be that FIRST_NAME and LAST_NAME have only a few hundred instances, and EDICAL_HISTORY lacks sufficient contextual cues in the documents.

PHI Type	Total Number of Instances	Precision (average)	Recall (average)	F-measure (average)
AGE	298	1.000	0.965	0.982
COUNTRY	1	1.000	0.900	0.900
DATE	2824	0.973	0.972	0.972
DOB	1	1.000	0.900	0.900
FIRST_NAME	234	0.991	0.903	0.944
INITIALS	671	0.995	0.971	0.983
LAST_NAME	254	0.991	0.906	0.946
MEDICAL_HISTORY	1257	0.901	0.941	0.920
MIDDLE_NAME	14	0.900	0.767	0.780
RACE	253	1.000	0.988	0.994
SITE_ID	36	1.000	0.643	0.775
SUBJID	1255	0.954	0.986	0.970
Overall	7098	0.962	0.962	0.962

Table 13. The average number of precision, recall and F-measure overall and for specific PHI types included in dataset A



(a) Total number of PHI instances by PHI type



(b) De-identification performance by PHI type

Figure 17. Dataset A overall statistics: (a) Total number of PHI instances and (b) De-identification performance by PHI type.

The i2b2 dataset contains 889 annotated discharge summaries drawn from Partners Healthcare, with the real identifiers replaced by synthetic information [27].

4.3.2 Experimental Design and Evaluation

We simulated the active learning query strategies with dataset A. For uncertainty sampling strategies, namely least confidence and entropy, we varied the parameters to evaluate the original and modified models (i.e., LC with a cutoff value and entropy with a minimum value). Specifically, the cutoff value θ of LC ranged from 0.1 to 0.9, with a step size of 0.1 and the minimum entropy score was selected from a set of four values: {0.001, 0.01, 0.1, 1}.

For the ROI model, the parameters include: 1) the costs of correcting an FP instance and an FN instance, 2) the contributions of the two types of correction, and 3) $P'(y|x)$, which is the true probability that token x is of class y. Since $P'(y|x)$ is unknown in this scenario, we approximate $P'(y|x)$ with $P(y|x)$,, the predicted probability based on the current model. In doing so, the ROI can be reduced to:

$$ROI(x) = \left(NC_n + NC_p\right) \times P(y|x) \times \left(1 - P(y|x)\right) - ct_r \tag{19}$$

We vary NC_n, NC_p and ct_r to investigate how these parameters influence the active learning performance. The parameter settings for LCUB, ELB and ROI are listed in Table 14.

Case Name	LCUB_1	LCUB_2	LCUB_3	LCUB_4	LCUB_5	LCUB_6	LCUB_7	LCUB_8	LCUB_9	LCUB_10
$\boldsymbol{\theta}$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0

(a) Simulation case names and corresponding parameter settings for query strategy LCUB

Case Name	ELB_1	ELB_2	ELB_3	ELB_4	ELB_5
$\boldsymbol{\rho}$	0	1	0.1	0.01	0.001

(b) Simulation case names and corresponding parameter settings for query strategy ELB

Case Name	NC	$cost_r$
ROI_1	0.1	0
ROI_2	0.1	0.01
ROI_3	0.5	0.01

(c) Simulation case names and corresponding parameter settings for query strategy ROI

Table 14. The simulation case names and their corresponding parameter settings: (a) LCUB, (b) ELB, and (c) ROI.

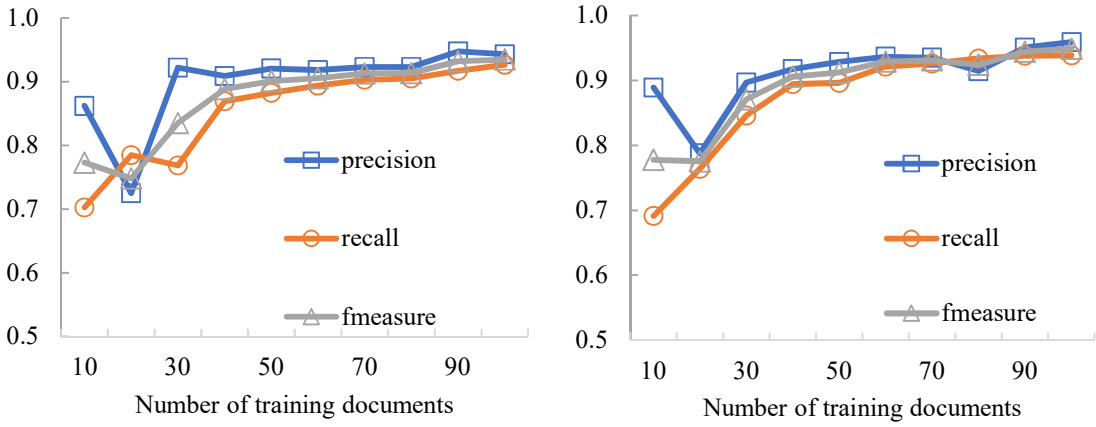
4.3.3 Simulation Results and Analysis

We initially evaluated the approaches with a randomly selected initial batch of 10 documents. For each learning iteration, the simulation selects an additional batch of documents to add to the training set and learns a new de-identification model. The process proceeds for 10 iterations. For a baseline comparison, we use a random selection of the next batch of documents to be added to training, which we refer to as Random. All results are based on an average of 3 runs. We evaluate the framework with a batch size of 10, 5 and 1.

4.3.3.1 Dataset A Results

4.3.3.1.1 Batch size 10

Table 15 compares the overall performance in terms of precision, recall and F-measure for the various query strategies, with the best score for each training level highlighted in bold. The learning curves are reported in Figure 18. For brevity, we choose the setting with the best performance to plot for each strategy. Note that in LCUB_10, the upper bound of LC is set to 1, which implies that all confidence scores are taken into account. Also, ELB_1 is equal to using entropy without a minimum value.



(a) Random overall performance

(b) LCUB_6 overall performance

(c) ELB_3 overall performance

(d) ROI_2 overall performance

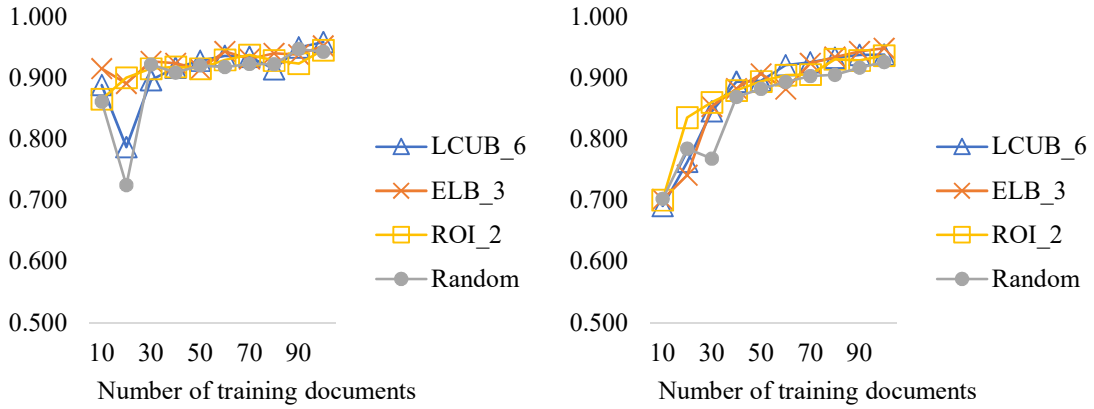
Figure 18. Learning curves for the active learning strategies and passive learning (with a batch size of 10 documents) for dataset A. (Note: Each curve corresponds to the simulation case that achieved the highest performance.)

It can be seen that there is a general increasing trend for all query strategies (including random selection) as additional training data is added. This is particularly pronounced when there is less than 50 training documents, at which point most of the selection approaches yield an F-measure of over 0.9. The observed growth in performance slows down from training 50 to 100 documents. Additionally, the classification model generally favors precision over recall in all testing scenarios, especially at early training stages (i.e., when the number of training documents is smaller than 50).

In terms of recall, ELB_3 outperforms all other selection methods at the final training stages, achieving 0.949 at 100 training documents, while LCUB_6 achieves the best precision at the final training stage. However, ROI_2 learns faster at the beginning than all other strategies. Also, it provides the most steady growth in all three performance measures among all participants in comparison, which is preferable in reducing human correction effort.

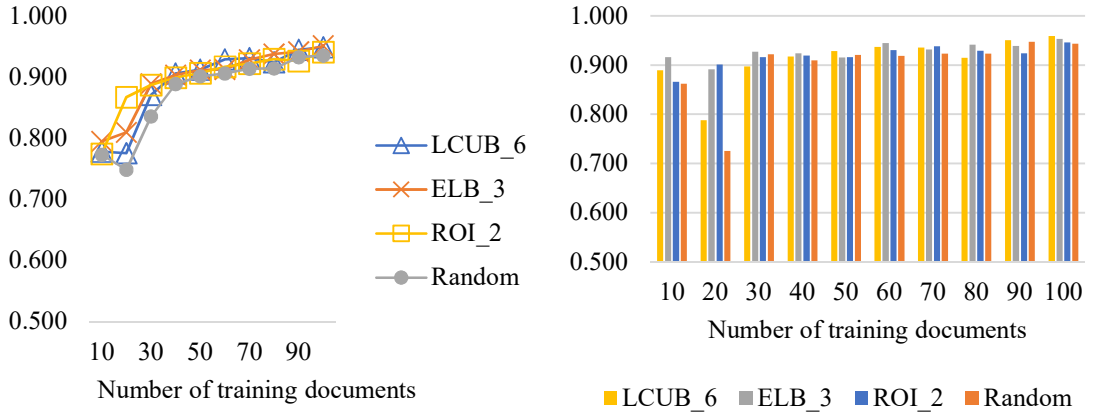
Number of Training Documents	Precision				Recall				F-measure			
	LCUB_6	ELB_3	ROI_2	Random	LCUB_6	ELB_3	ROI_2	Random	LCUB_6	ELB_3	ROI_2	Random
10	0.889	0.916	0.866	0.862	0.691	0.702	0.701	0.703	0.778	0.795	0.774	0.773
20	0.788	0.891	**0.901**	0.725	0.764	0.742	**0.836**	0.785	0.775	0.810	**0.867**	0.748
30	0.897	**0.927**	0.916	0.922	0.846	0.853	**0.861**	0.768	0.871	**0.889**	0.887	0.835
40	0.917	**0.924**	0.919	0.909	**0.894**	0.883	0.880	0.869	**0.906**	0.903	0.899	0.888
50	**0.928**	0.915	0.916	0.921	0.896	**0.907**	0.895	0.883	**0.912**	0.911	0.906	0.901
60	0.937	**0.944**	0.930	0.918	**0.921**	0.882	0.905	0.894	**0.929**	0.912	0.917	0.906
70	0.935	0.932	**0.938**	0.923	**0.926**	0.925	0.906	0.903	**0.931**	0.928	0.922	0.913
80	0.915	**0.941**	0.929	0.923	0.934	**0.935**	0.931	0.905	0.924	**0.938**	0.929	0.914
90	**0.950**	0.939	0.924	0.948	0.938	**0.944**	0.929	0.917	**0.944**	0.941	0.926	0.932
100	**0.959**	0.953	0.946	0.943	0.939	**0.949**	0.937	0.927	0.949	**0.951**	0.941	0.935

Table 15. Performance of various active learning strategies and passive learning (with a batch size of 10 documents) for dataset A.



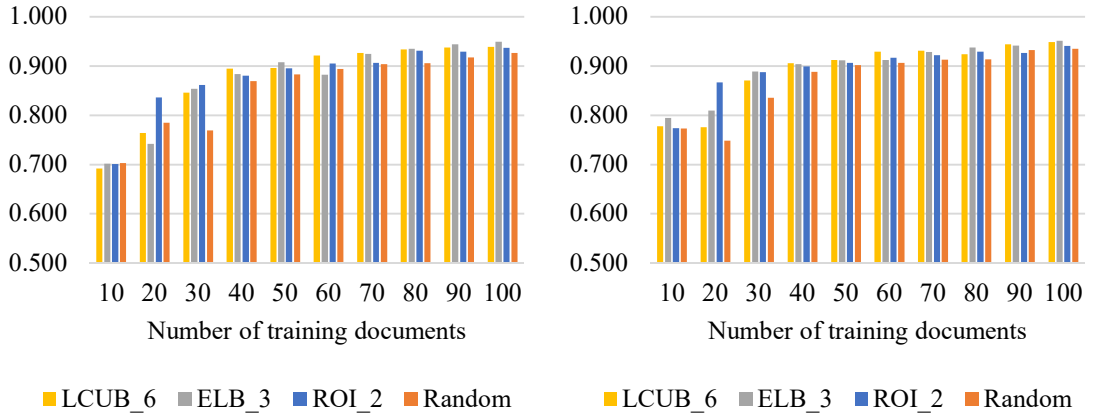
(a) Overall precision learning curves

(b) Overall recall learning curves



(c) Overall F-measure learning curves

(d) Overall precision comparison

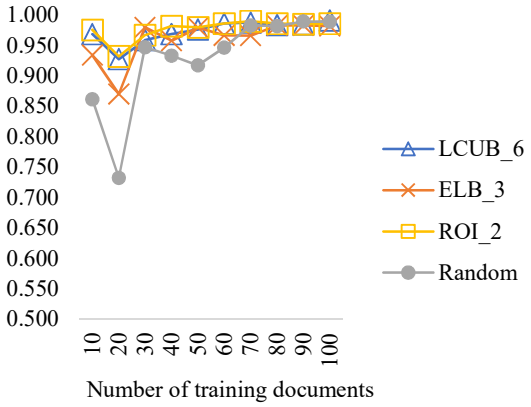


(e) Overall recall comparison

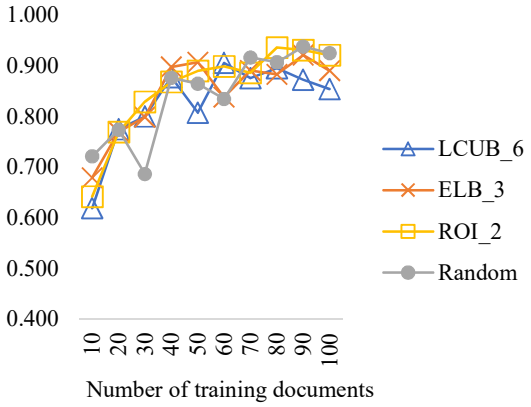
(f) Overall F-measure comparison

Figure 19. Overall performance comparison for the active learning strategies and passive learning (with a batch size of 10 documents) for dataset A.

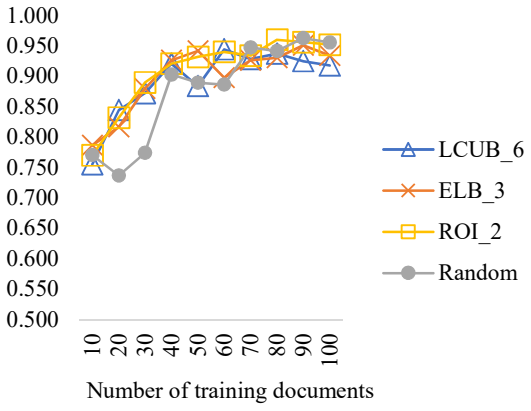
However, the previous observation is not guaranteed to hold true for each PHI type. Shown in Figures 20 through 22, among the three types of PHI that we focus on (FIRST_NAME, LAST_NAME and MEDICAL_HISTORY), there is no obvious trend for precision for FIRST_NAME or LAST_NAME. By contrast, the precision for MEDICAL_HISTORY increases with additional training data (Figure 22). Both the recall and F-measure exhibits more clear increasing trends for all three PHI types, in both the active and passive learning scenarios. For FIRST_NAME (Figure 20) and LAST_NAME (Figure 21), the active learning approaches are more stable in growth than Random, but do not necessarily outperform Random in the final iterations. For MEDICAL_HISTORY, the selection based on active learning exceeds Random selection, especially for ROI_2 (Figure 22).



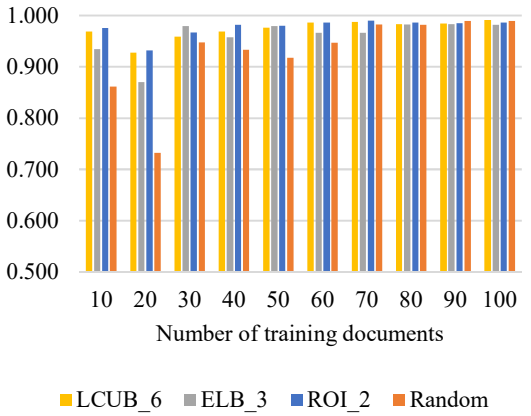
(a) FIRST_NAME precision learning curves



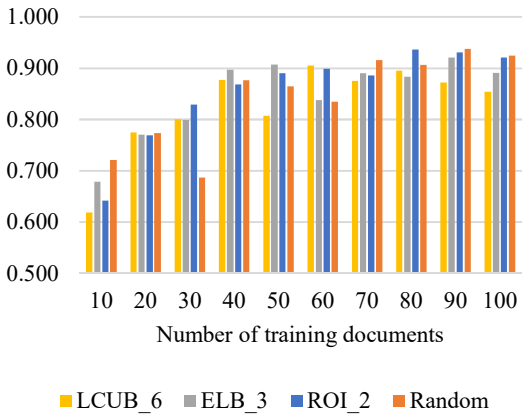
(b) FIRST_NAME recall learning curves



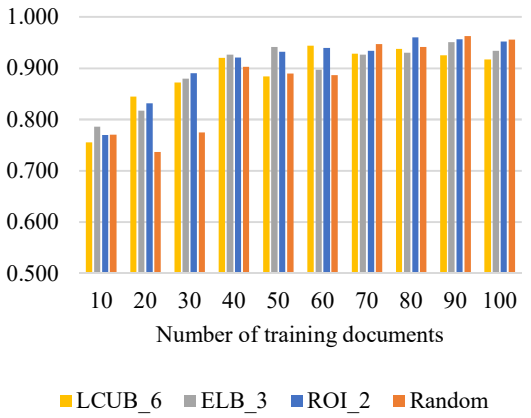
(c) FIRST_NAME F-measure learning curves



(d) FIRST_NAME precision comparison

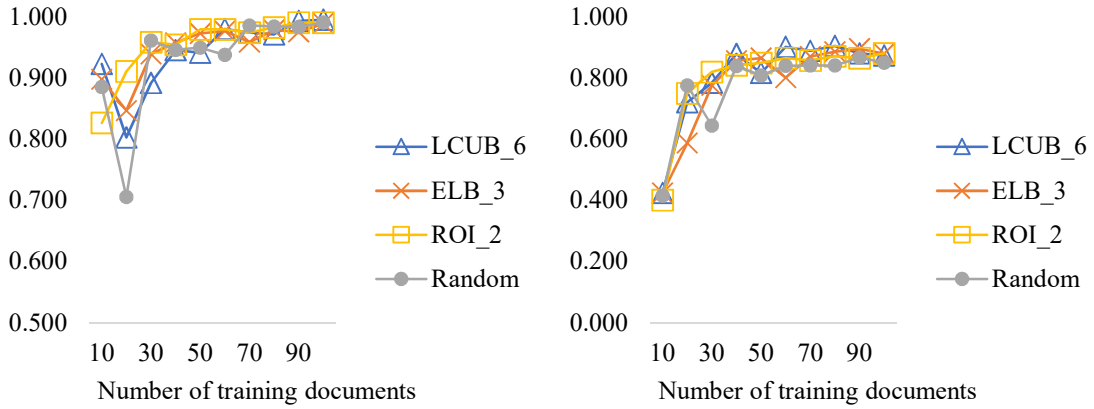


(e) FIRST_NAME recall comparison



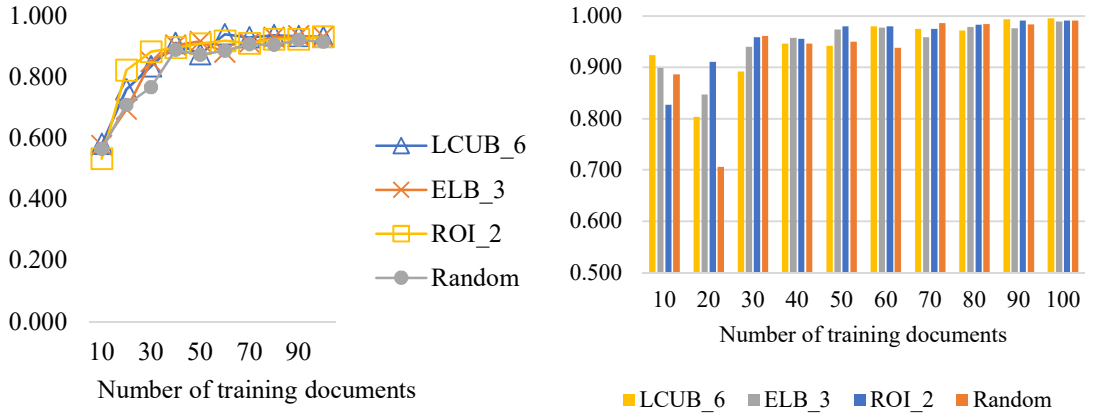
(f) FIRST_NAME F-measure comparison

Figure 20. FIRST_NAME performance comparison for the active learning strategies and passive learning (with a batch size of 10 documents) for dataset A.



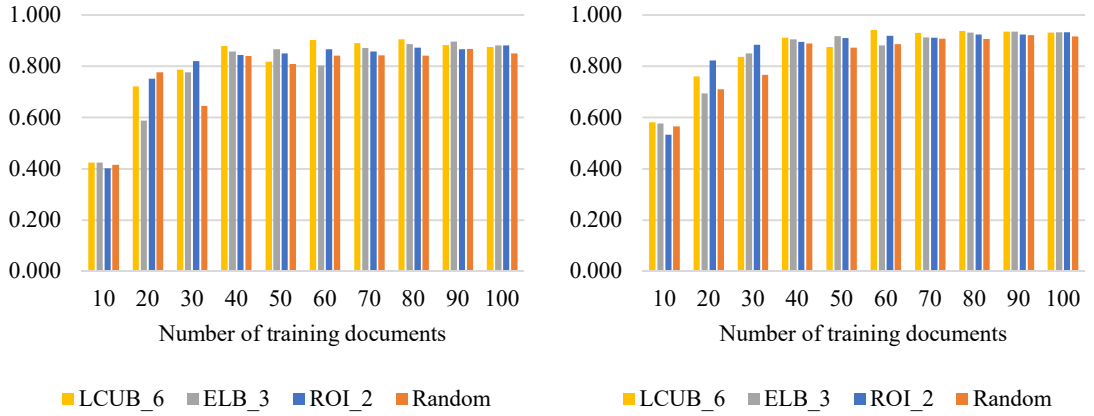
(a) LAST_NAME precision learning curves

(b) LAST_NAME recall learning curves



(c) LAST_NAME F-measure learning curves

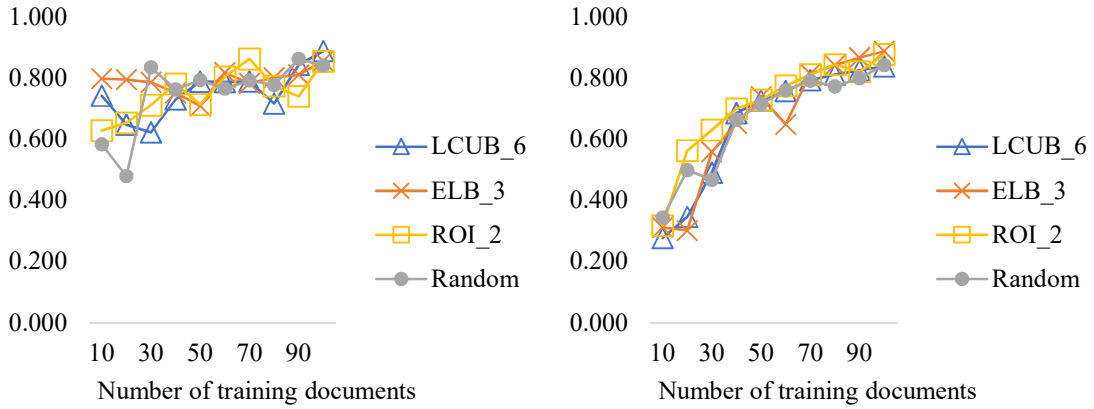
(d) LAST_NAME precision comparison



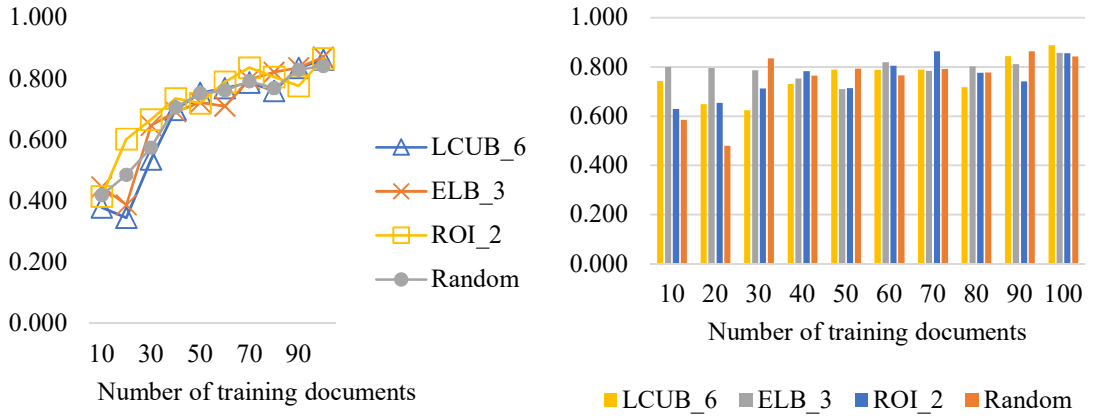
(e) LAST_NAME recall comparison

(f) LAST_NAME F-measure comparison

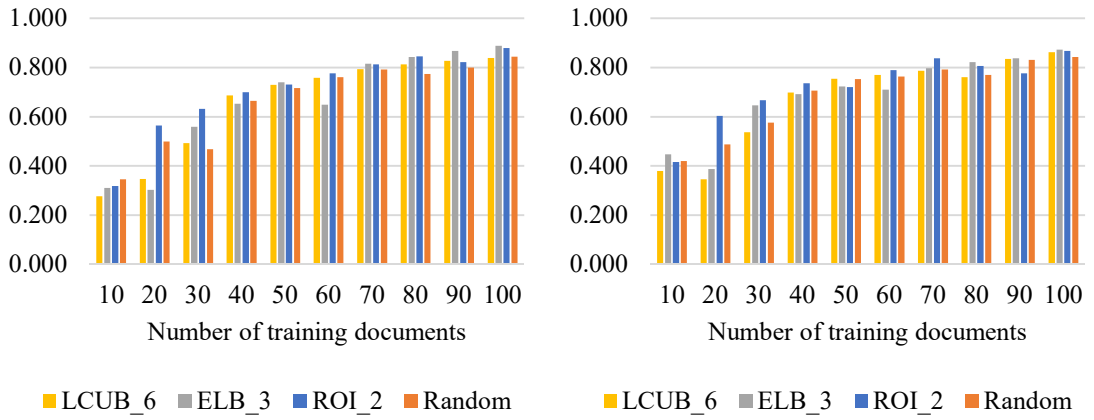
Figure 21. LAST_NAME performance comparison for the active learning strategies and passive learning (with a batch size of 10 documents) for dataset A.



(a) MEDICAL_HISTORY precision learning curves (b) MEDICAL_HISTORY recall learning curves



(c) MEDICAL_HISTORY F-measure learning curves (d) MEDICAL_HISTORY precision comparison



(e) MEDICAL_HISTORY recall comparison (f) MEDICAL_HISTORY F-measure comparison

Figure 22. MEDICAL_HISTORY performance comparison for the active learning strategies and passive learning (with a batch size of 10 documents) for dataset A.

4.3.3.1.2 Batch Size 5

When the batch size is reduced to 5, the advantage of active learning versus passive learning (random selection) becomes more apparent (Table 16, Figure 23). Generally, ELB_3 and ROI_3 (the best performed settings of ELB and ROI) exceeds Random on all three performance measures (P, R, and F). Additionally, LCUB_6 (the best of LCUB) shows less stable growth than the other two active learning approaches, but is still better than Random most of the time.

Similar to a batch size of 10, the overall recall and F-measure improves with higher training quantities of training documents for both of the active and passive approaches. The increasing trend does not hold for precision. Rather, precision fluctuates around 0.9 for active learning and 0.85 for random. As for recall, to yield a score 0.9, ROI_3 requires around 35 training documents, LCUB_6 and ELB_3 need 50 documents, and for Random the best score within 10 iterations is only 0.876. At a training level of 30 documents, ROI_3 could reach an F-measure of over 90%, while the highest F-score for random is 0.887.

Regarding performance scores of the three individual PHI types, the advantage of adopting active learning over passive learning remains for ROI_3 (Figures 24-26). Specifically for MEDICAL_HISTORY (as in Figure 26), the recall of ROI_3 sees a steady growth and arrives at 0.787 after 10 iterations. By contrast, the random approach manifests a much more unstable trend and never reaches above 0.7. Comparing to the overall F-measure, although the F-measure of MEDICAL_HISTORY is generally worse, it shows much more drastic growth within the 10 iterations.

Number of Training Documents	Precision				Recall				F-measure			
	LCUB_6	ELB_3	ROI_3	Random	LCUB_6	ELB_3	ROI_3	Random	LCUB_6	ELB_3	ROI_3	Random
10	0.907	0.890	0.891	0.903	0.722	0.704	0.7	0.699	0.803	0.786	0.784	0.788
15	**0.915**	0.877	0.915	0.886	0.747	0.766	**0.766**	0.722	0.822	0.818	**0.834**	0.795
20	0.838	0.836	**0.882**	0.805	**0.840**	0.833	0.828	0.818	0.839	0.831	**0.854**	0.809
25	0.901	0.917	**0.917**	0.822	0.729	0.827	**0.852**	0.834	0.792	0.868	**0.884**	0.825
30	0.865	0.872	**0.932**	0.727	**0.877**	0.864	0.874	0.821	0.870	0.868	**0.902**	0.755
35	**0.898**	0.880	0.883	0.892	0.865	0.818	**0.899**	0.860	0.881	0.839	**0.89**	0.875
40	0.869	**0.929**	0.919	0.892	0.894	0.868	**0.902**	0.841	0.880	0.897	**0.91**	0.863
45	0.911	**0.936**	0.92	0.869	0.850	0.876	**0.898**	0.876	0.876	0.905	**0.908**	0.871
50	0.934	0.829	0.918	**0.938**	**0.908**	0.898	0.907	0.822	**0.920**	0.857	0.912	0.873
55	**0.947**	0.928	0.937	0.915	0.905	0.904	**0.907**	0.861	**0.926**	0.916	0.922	0.887

Table 16. Performance of the active and passive learning strategies, using a batch size of 5 documents for dataset A.

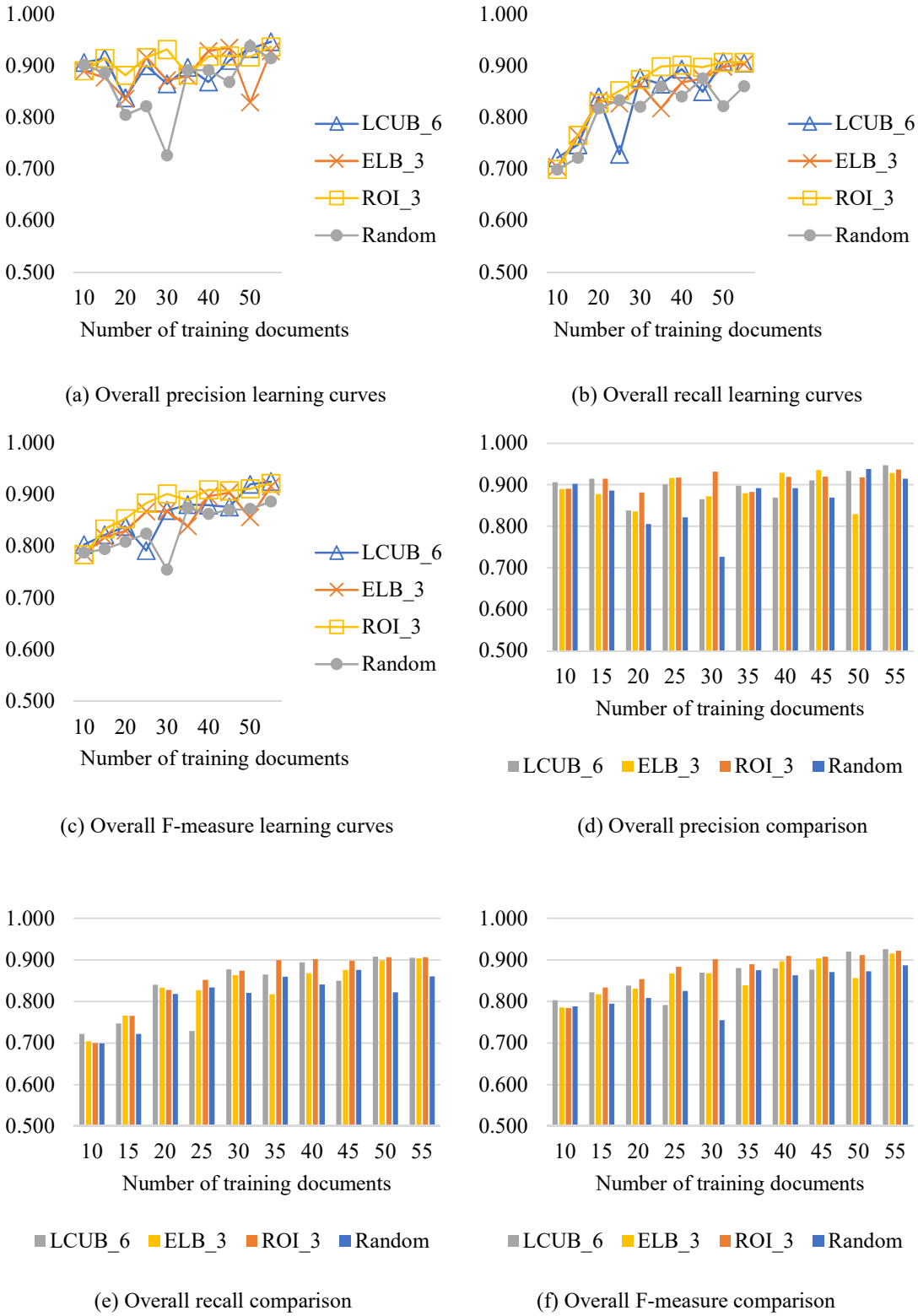
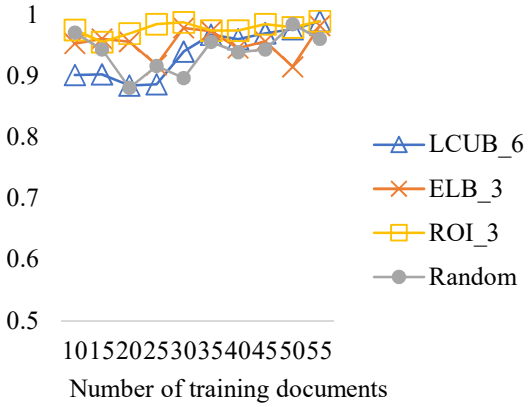
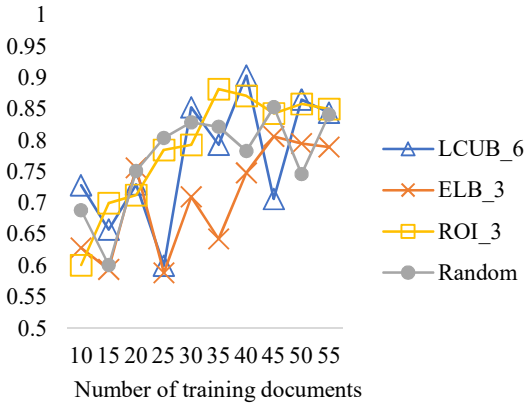


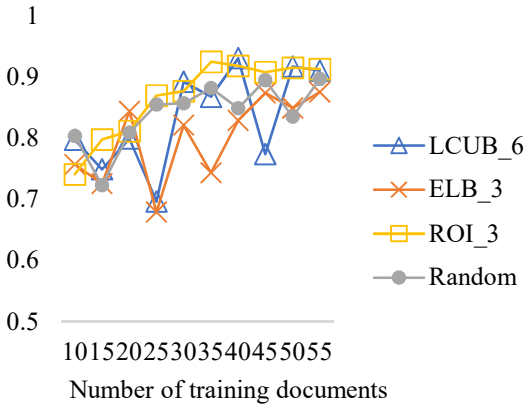
Figure 23. Overall performance comparison for the active learning strategies and passive learning (with a batch size of 5 documents) for dataset A.



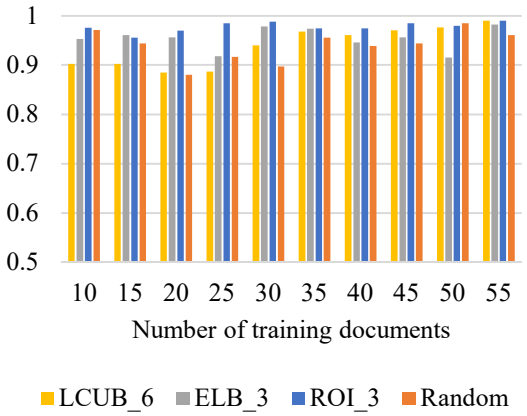
(a) FIRST_NAME precision learning curves



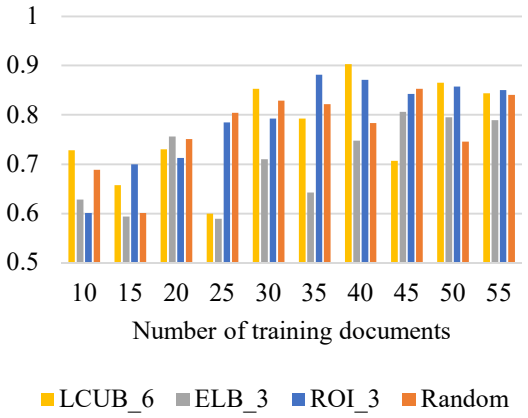
(b) FIRST_NAME recall learning curves



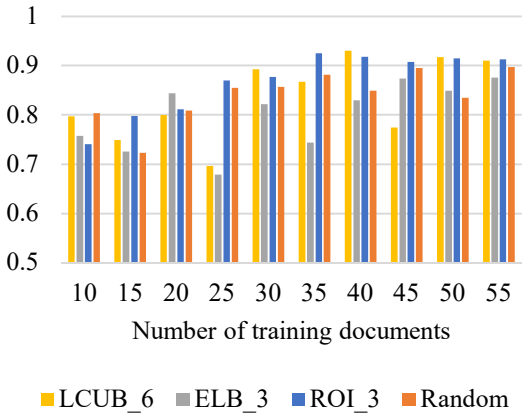
(c) FIRST_NAME F-measure learning curves



(d) FIRST_NAME precision comparison

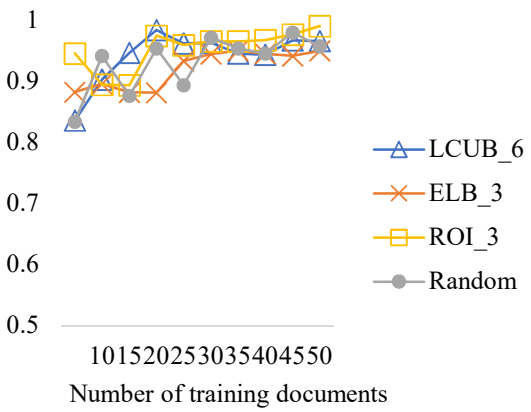


(e) FIRST_NAME recall comparison

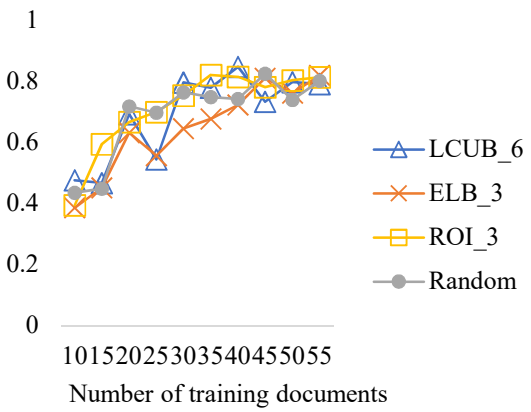


(f) FIRST_NAME F-measure comparison

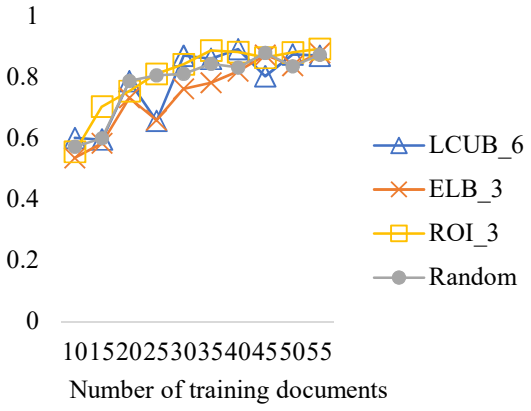
Figure 24. FIRST_NAME performance comparison for the active learning strategies and passive learning (with a batch size of 5 documents) for dataset A.



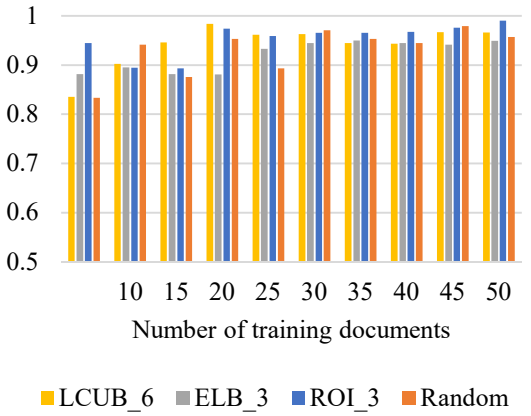
(a) LAST_NAME precision learning curves



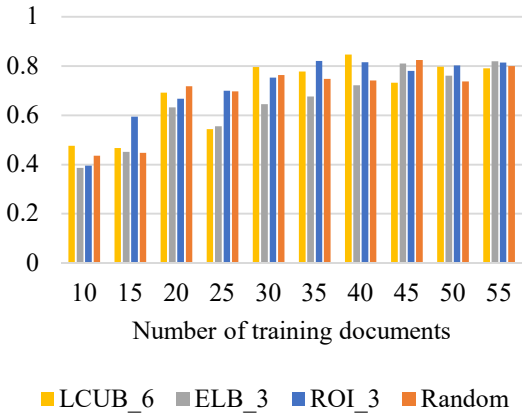
(b) LAST_NAME recall learning curves



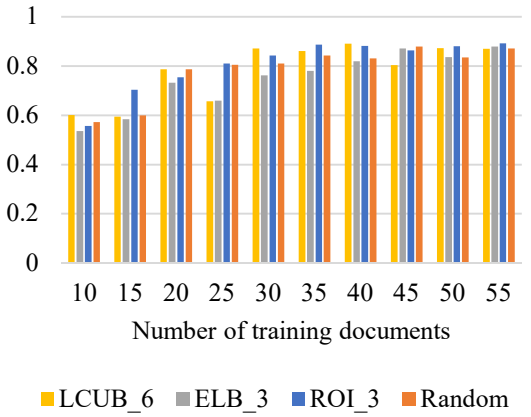
(c) LAST_NAME F-measure learning curves



(d) LAST_NAME precision comparison

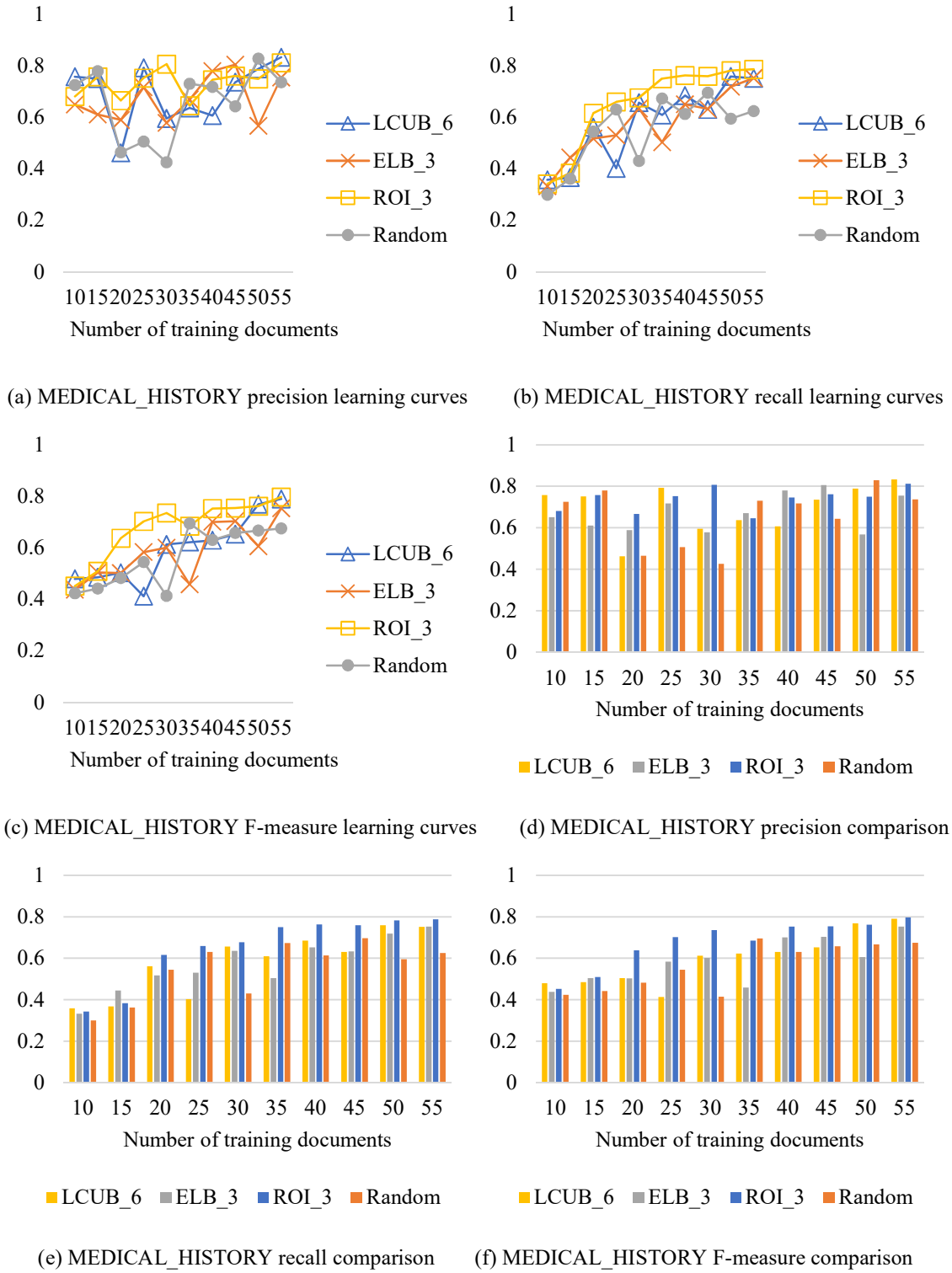


(e) LAST_NAME recall comparison



(f) LAST_NAME F-measure comparison

Figure 25. LAST_NAME performance comparison for the active learning strategies and passive learning (with a batch size of 5 documents) for dataset A.



(a) MEDICAL_HISTORY precision learning curves

(b) MEDICAL_HISTORY recall learning curves

(c) MEDICAL_HISTORY F-measure learning curves

(d) MEDICAL_HISTORY precision comparison

(e) MEDICAL_HISTORY recall comparison

(f) MEDICAL_HISTORY F-measure comparison

Figure 26. MEDICAL_HISTORY performance comparison for the active learning strategies and passive learning (with a batch size of 5 documents) for dataset A.

4.3.3.1.3 Batch Size 1

The simulation with a batch size of 1 document led to more complex results than above scenarios (Table 17, Figure 27). ROI_1 generally outperformed random selection as well as other active learning approaches within 10 iterations, reaching an F-measure of 0.87 with 19 training documents. LCUB_6 and ELB_3 generated higher performance scores than random in most cases, but the learning was less stable than ROI_1. When considering the performance by PHI types (Figures 27-30), ROI_1 demonstrated a greater advantage than the other strategies, particularly in terms of recall and F-measure.

Number of Training Documents	Precision				Recall				F-measure			
	LCUB_6	ELB_3	ROI_1	Random	LCUB_6	ELB_3	ROI_1	Random	LCUB_6	ELB_3	ROI_1	Random
10	0.895	0.904	0.918	0.840	0.696	0.696	0.683	0.719	0.783	0.786	0.783	0.772
11	0.896	0.823	0.877	**0.905**	0.701	**0.734**	0.722	0.655	0.785	0.775	**0.790**	0.753
12	0.880	0.677	0.856	**0.899**	0.734	0.714	**0.792**	0.714	0.799	0.652	**0.818**	0.795
13	0.903	0.813	**0.907**	0.862	0.743	**0.791**	0.719	0.741	**0.815**	0.801	0.802	0.796
14	0.822	**0.891**	0.880	0.841	0.729	**0.792**	0.782	0.765	0.773	**0.839**	0.827	0.798
15	0.868	0.877	**0.906**	0.836	0.780	**0.792**	0.737	0.777	0.816	**0.832**	0.810	0.804
16	0.901	0.842	0.905	**0.907**	**0.795**	0.770	0.790	0.743	**0.844**	0.797	0.843	0.815
17	**0.907**	0.857	0.882	0.815	**0.824**	0.806	0.816	0.798	**0.863**	0.828	0.847	0.806
18	**0.916**	0.868	0.880	0.840	**0.794**	0.790	0.828	0.801	0.849	0.824	**0.852**	0.819
19	0.913	**0.918**	0.889	0.868	0.772	0.818	**0.861**	0.802	0.835	0.864	**0.874**	0.833

Table 17. Performance of various active and passive learning strategies with a batch size of 1 training document for dataset A.

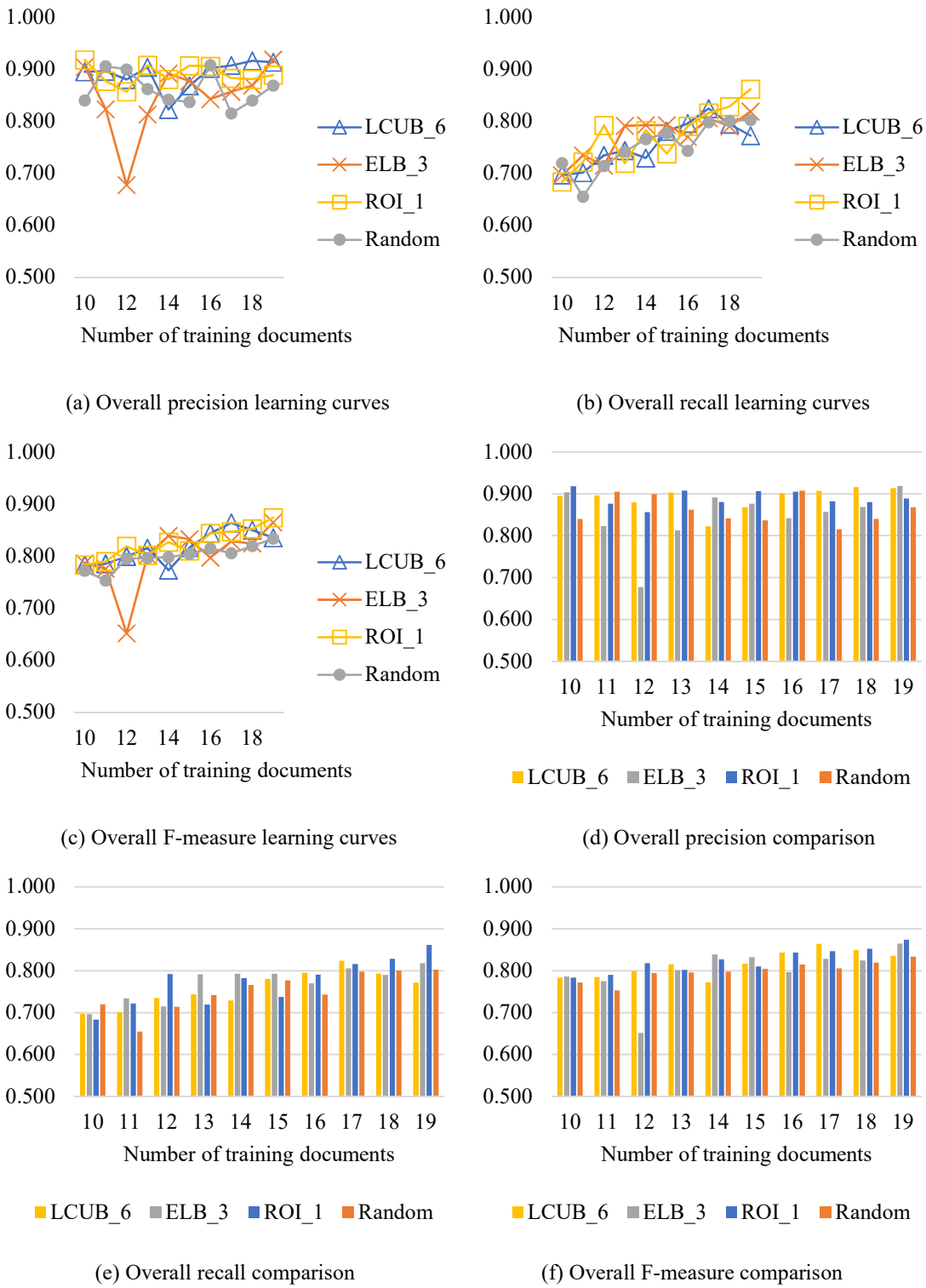


Figure 27. Overall performance comparison for the active learning strategies and passive learning (with a batch size of 1 document) for dataset A.

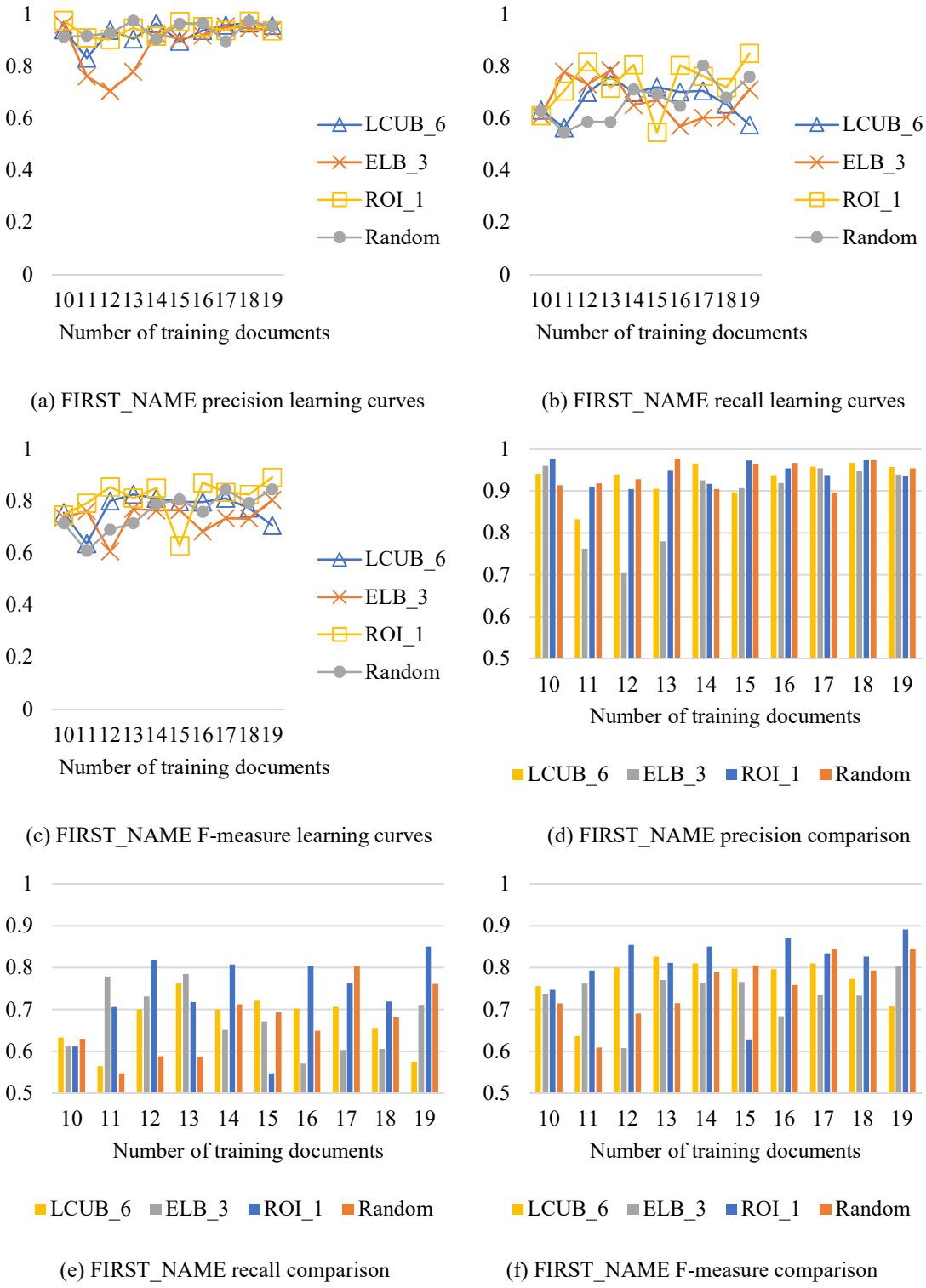
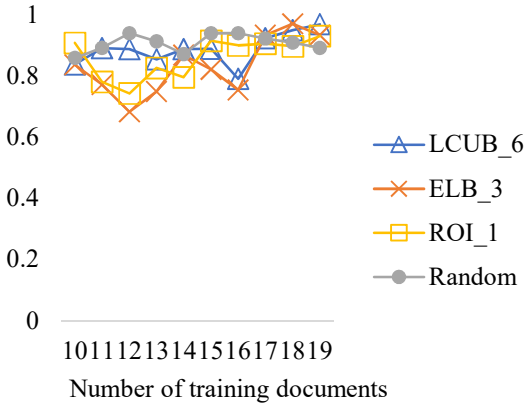
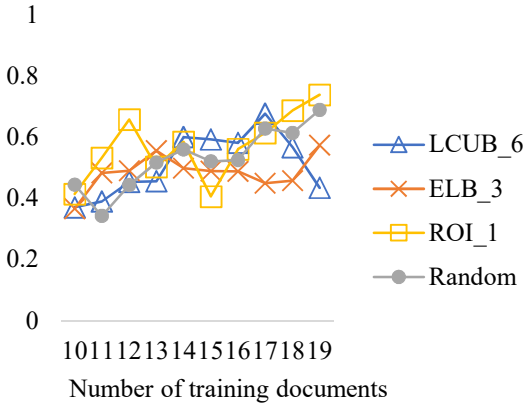


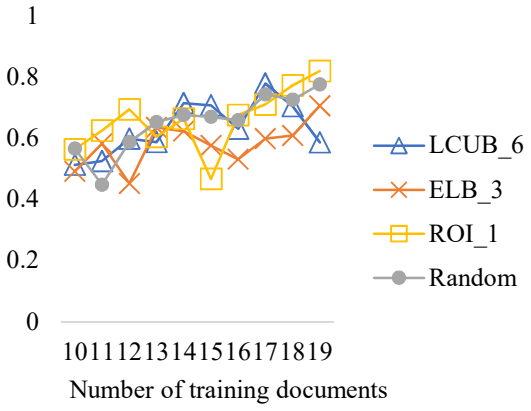
Figure 28. FIRST_NAME performance comparison for the active learning strategies and passive learning (with a batch size of 1 document) for dataset A.



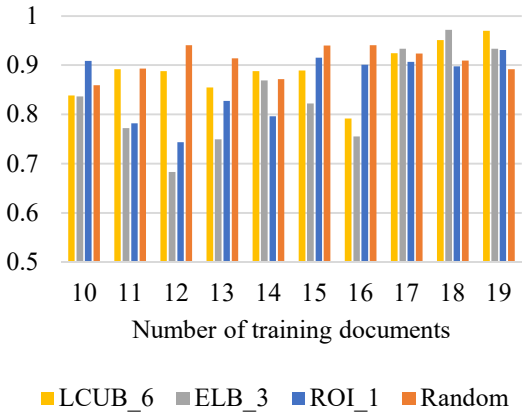
(a) LAST_NAME precision learning curves



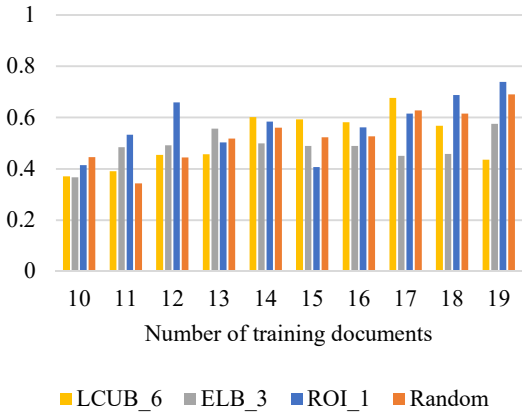
(b) LAST_NAME recall learning curves



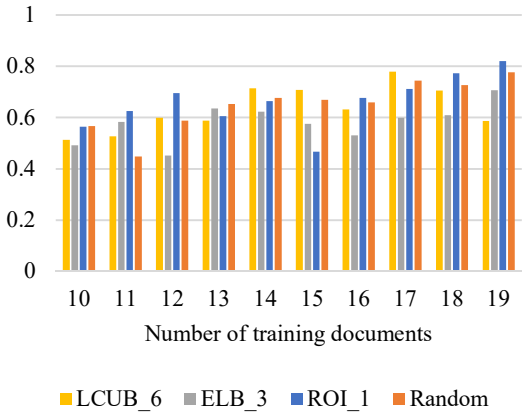
(c) LAST_NAME F-measure learning curves



(d) LAST_NAME precision comparison

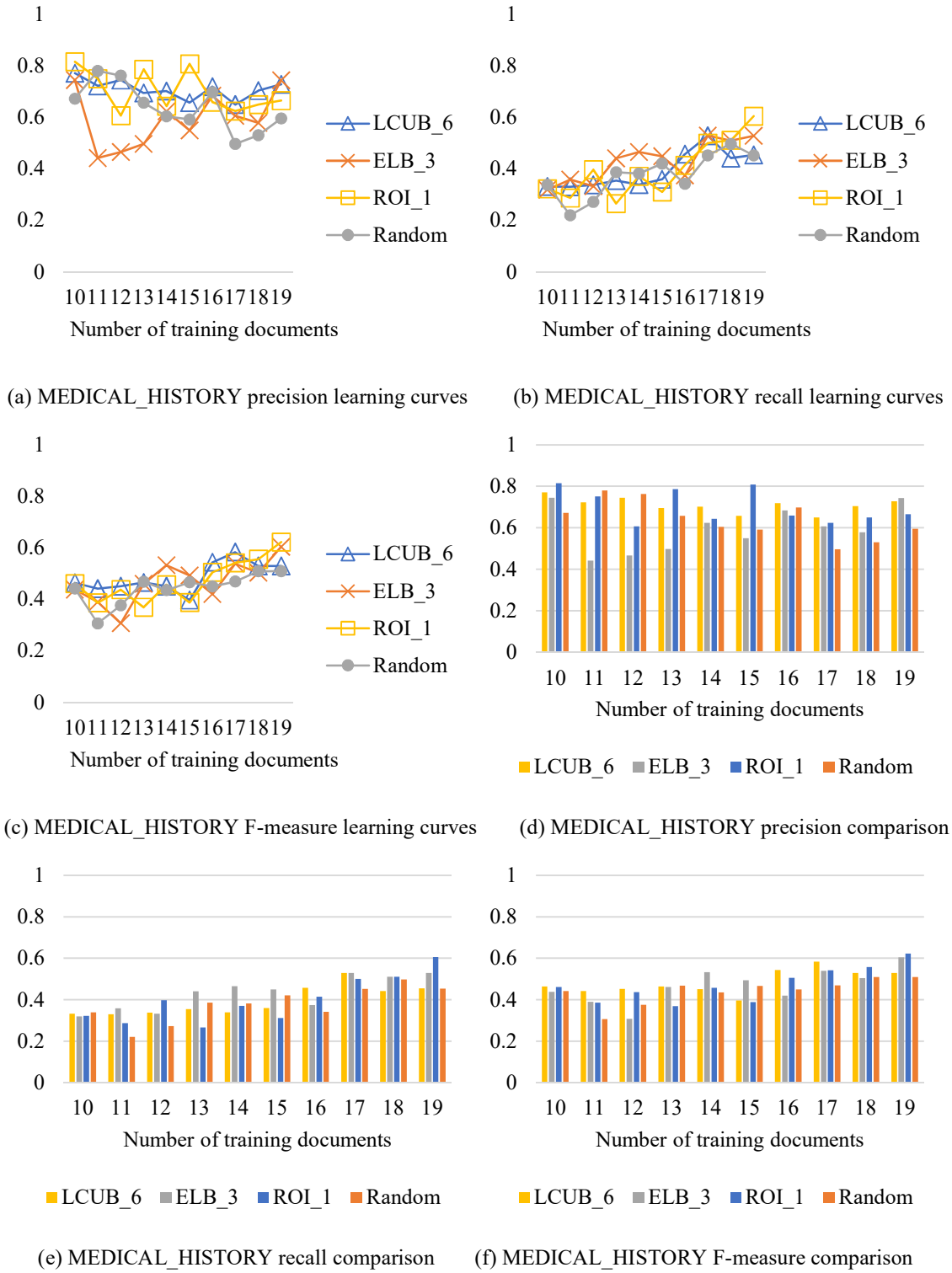


(e) LAST_NAME recall comparison



(f) LAST_NAME F-measure comparison

Figure 29. LAST_NAME performance comparison for the active learning strategies and passive learning (with a batch size of 1 document) for dataset A.



(a) MEDICAL_HISTORY precision learning curves

(b) MEDICAL_HISTORY recall learning curves

(c) MEDICAL_HISTORY F-measure learning curves

(d) MEDICAL_HISTORY precision comparison

(e) MEDICAL_HISTORY recall comparison

(f) MEDICAL_HISTORY F-measure comparison

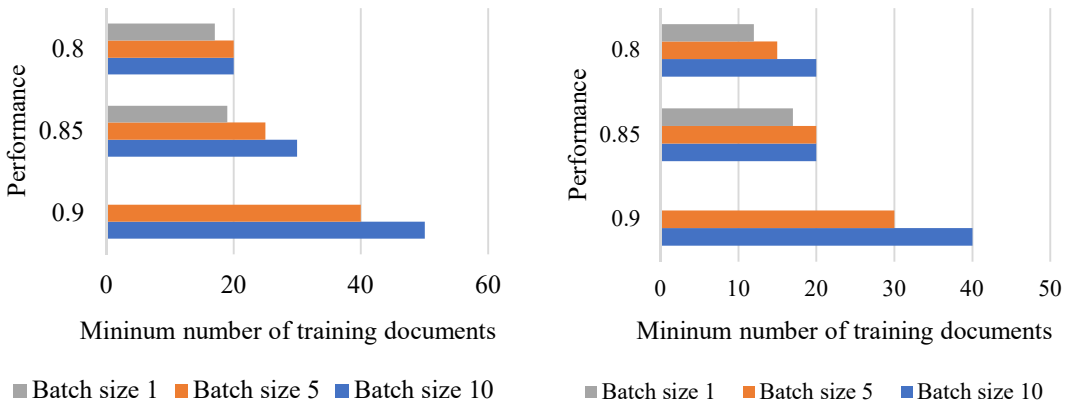
Figure 30. MEDICAL_HISTORY performance comparison for the active learning strategies an d passive learning (with a batch size of 1 document) for dataset A.

4.3.3.1.4 Batch size comparison

To compare the performance of models trained with different batch sizes, we plot the levels of performance scores against the size of training datasets needed for each of the batch sizes (i.e., the minimum number of training documents the active learning was provided to reach a certain level of performance). Since the precision across all PHI types could yield over 0.9 at the beginning 10 random documents and did not exhibit a clear trend as more training data was supplied, we focus on the recall and F-measure for this comparison. Overall, when the active learning proceeds with smaller batch sizes, the performance scores, in terms of recall and F-measure, could attain a certain level with less training data involved.

As shown on Figure 31, a recall of 0.8 required 20 training documents for batch sizes 10 and 5, while only 17 documents were needed for batch size 1. A recall of 0.85 required at least 30 documents for a batch size of 10, while 25 were needed for a batch size 5, and only 19 for batch size 1. Finally, we observed that a batch size of 10 could reach a recall of 0.9 with 50 documents, while a batch size of 5 could yield the same level of recall with 40 documents in training. Since all batch sizes were tested with 10 iterations, the maximum training number for batch size 1 was 19, the highest recall of which (0.86) was less than 0.9 and thus not included in the plot.

Generally, in comparison to recall, it took less training data to achieve the same level of F-measure. Starting from an F-measure of 0.8, a batch size of 10 required 20 documents to build the model, and a batch size 5 and 1 needed 15 and 12, respectively. When the goal of F-measure was set to 0.85, 20 documents were needed for both a batch size of 10 and 5, whereas 17 documents were required for a batch size of 1. Again, a batch size of 1 could never reach an F-measure of over 0.9 within 10 iterations, while a minimum of 30 or 40 documents was needed with respect to training for batch size 5 or 10.



(a) Overall recall comparison

(b) Overall F-measure comparison

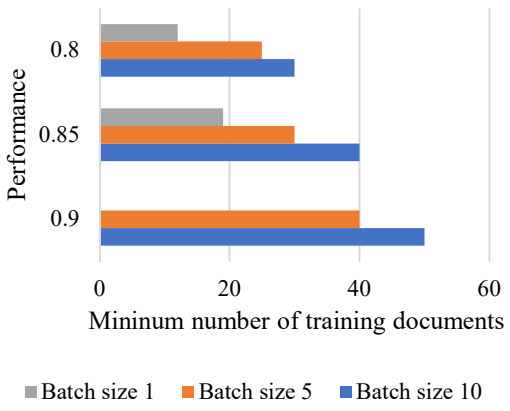
Figure 31. Overall comparison of different active learning batch sizes for the minimum number of training documents that was provided to reach a certain level of performance for dataset A.

Next, we compared different batch sizes for specific PHI types, particularly FIRST_NAME, LAST_NAME, and MEDICAL_HISTORY (as shown in Figure 32).

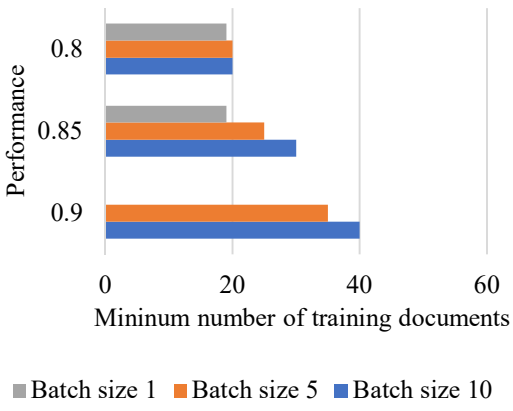
For FIRST_NAME, the comparison showed similar results with the overall comparison. Specifically, a batch size of 1 could reach a recall or an F-measure of 0.8 or 0.85 faster than batch sizes of 5 and 10, and a batch size of 5 required no greater than the same amount of training data than a batch size of 10 for the performance scores (recall and F-measure) to reach scores of 0.8, 0.85 or 0.9.

For LAST_NAME and MEDICAL_HISTORY, more performance levels were considered in the comparison because the growth was more drastic than FIRST_NAME. The findings based on overall and FIRST_NAME did not hold for LAST_NAME. A batch size of 5 almost always performed worse than a batch size of 10 or a batch size of 1, for both recall and F-measure. In other words, it took more training data for a batch size of 5 to attain a certain performance level. MEDICAL_HISTORY continued to be the most difficult PHI type to classify and using a smaller batch size in active learning helped to learn faster.

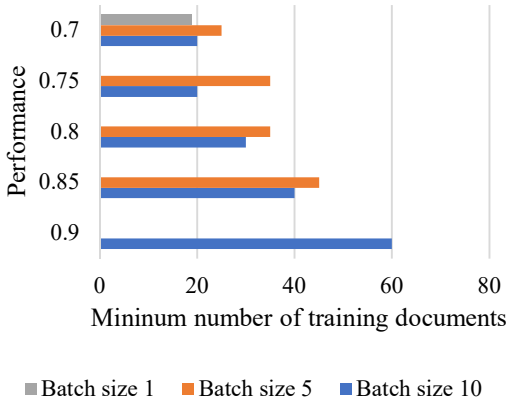
It should be noted that, though opting for a smaller batch size could benefit model training (especially in the early stages of active learning), it also requires more time for retraining the entire process, which could be substantial depending on the size of the dataset.



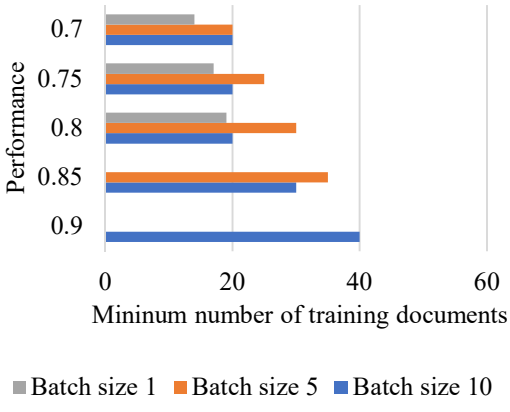
(a) FIRST_NAME recall comparison



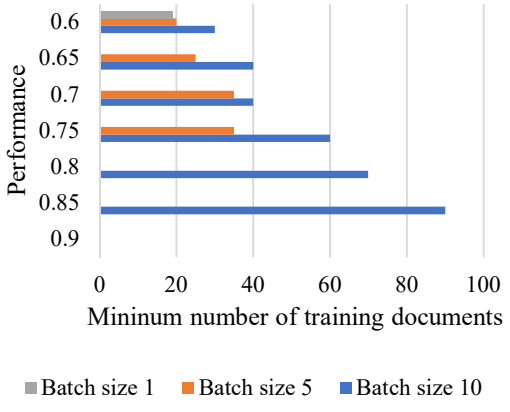
(b) FIRST_NAME F-measure comparison



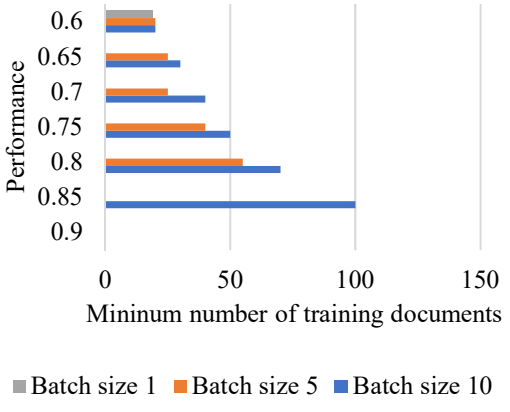
(c) LAST_NAME recall comparison



(d) LAST_NAME F-measure comparison



(e) MEDICAL_HISTORY recall comparison



(f) MEDICAL_HISTORY F-measure comparison

Figure 32. Comparison of different active learning batch sizes for the minimum number of training documents that was provided to reach a certain level of performance for a PHI type for dataset A.

4.3.3.2 I2b2 Results

When experimenting on the i2b2 dataset, the active learning approaches surpass passive learning for all three batch sizes (10, 5 and 1), especially in terms of recall and F-measure (Tables 18-20). As shown in Figure 33, similar to previous findings with dataset A, smaller batch sizes lead to more significant difference between active learning and Random. Meanwhile, unlike in dataset A, LCUB performs better than other active strategies with batch size of 10 and 5, and ROI is the best query strategy when the batch size is 1.

4.3.3.2.1 Batch Size 10

The overall performance (precision, recall and F-measure) comparison of a batch size of 10 documents for different query strategies for the i2b2 dataset is reported in Table 18. Again, the winning score is highlighted in bold for every training level.

The general increasing trend with more data included in training also exists for all query strategies in comparison for the i2b2 dataset, as well as the better performance of precision comparing with recall. Further, the growth exhibited with the i2b2 dataset is more stable than with dataset A. Overall, all active learning approaches manifest higher performance scores than passive learning.

For the later learning stages, LCUB attains the best scores among all selection strategies in terms of all three performance measures (precision, recall and F-measure), and reaches 0.926, 0.910 and 0.918, respectively, when training with 100 documents.

Number of Training Documents	Precision				Recall				F-measure			
	LCUB_4	ELB_2	ROI_3	Random	LCUB_4	ELB_2	ROI_3	Random	LCUB_4	ELB_2	ROI_3	Random
10	0.835	0.824	0.818	0.834	0.649	0.640	0.664	0.666	0.730	0.719	0.733	0.740
20	0.873	**0.882**	0.867	0.870	**0.779**	0.765	0.774	0.746	**0.823**	0.819	0.818	0.803
30	0.891	0.894	**0.894**	0.873	0.825	**0.829**	0.820	0.802	0.857	**0.860**	0.855	0.836
40	0.897	0.897	**0.899**	0.877	0.843	**0.848**	0.836	0.836	0.869	**0.872**	0.866	0.856
50	0.903	**0.906**	0.903	0.899	0.860	**0.862**	0.852	0.850	0.881	**0.883**	0.876	0.874
60	0.911	**0.911**	0.906	0.906	**0.874**	0.869	0.858	0.862	**0.892**	0.889	0.881	0.883
70	**0.915**	0.911	0.906	0.906	**0.887**	0.880	0.865	0.869	**0.901**	0.895	0.885	0.887
80	0.919	**0.919**	0.907	0.905	**0.895**	0.890	0.872	0.874	**0.907**	0.904	0.889	0.889
90	**0.923**	0.922	0.910	0.909	**0.904**	0.897	0.880	0.882	**0.913**	0.910	0.895	0.895
100	**0.926**	0.923	0.911	0.915	**0.910**	0.902	0.886	0.885	**0.918**	0.912	0.898	0.900

Table 18. Performance of various active learning strategies and passive learning of the i2b2 dataset (with a batch size of 10 documents).

4.3.3.2.2 Batch Size 5

When it comes to a batch size of 5 documents (Table 19), among all strategies, it is still LCUB that dominates at later learning stages, which ends up with 0.912, 0.874 and 0.892 for precision, recall, and F-measure after 10 iterations. Active learning performs better than Random at all training levels for recall and F-measure, and at the majority of training levels for precision.

Number of Training Documents	Precision				Recall				F-measure			
	LCUB_7	ELB_2	ROI_3	Random	LCUB_7	ELB_2	ROI_3	Random	LCUB_7	ELB_2	ROI_3	Random
10	0.818	0.842	0.842	0.842	0.680	0.667	0.635	0.648	0.742	0.744	0.723	0.732
15	0.867	0.855	0.860	**0.867**	0.723	**0.735**	0.731	0.718	0.788	**0.791**	0.790	0.785
20	**0.876**	0.867	0.867	0.869	**0.795**	0.794	0.793	0.779	**0.834**	0.829	0.828	0.821
25	0.886	0.884	**0.889**	0.873	0.805	0.813	**0.821**	0.799	0.844	0.847	**0.853**	0.834
30	0.895	**0.896**	0.895	0.891	0.825	0.822	**0.826**	0.818	0.859	0.857	**0.859**	0.852
35	**0.898**	0.894	0.895	0.888	**0.842**	0.841	0.836	0.835	**0.869**	0.867	0.864	0.860
40	**0.904**	0.896	0.902	0.895	**0.850**	0.848	0.843	0.846	**0.876**	0.871	0.871	0.870
45	**0.906**	0.897	0.902	0.899	**0.863**	0.854	0.848	0.848	**0.884**	0.875	0.875	0.873
50	0.906	0.901	0.903	**0.908**	**0.869**	0.859	0.851	0.858	**0.887**	0.879	0.877	0.882
55	**0.912**	0.904	0.902	0.906	**0.874**	0.866	0.854	0.864	**0.892**	0.884	0.878	0.884

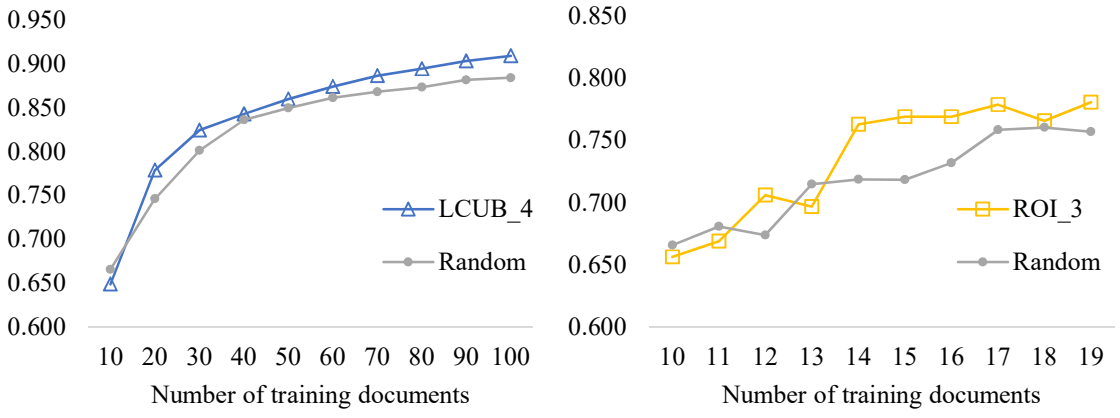
Table 19. Performance of various active learning strategies and passive learning of the i2b2 dataset (with a batch size of 5 documents).

4.3.3.2.3 Batch Size 1

Finally, for the results of learning with a batch size of 1 document (Table 20), active learning manifests more evident advantage over passive learning, especially in terms of recall and f-measure. Unlike previous findings with larger batch sizes, ROI is the best performing strategy in general. Also, overall learning is not as stable as learning with a larger batch size.

Number of Training Documents	Precision				Recall				F-measure			
	LCUB_8	ELB_4	ROI_3	Random	LCUB_8	ELB_4	ROI_3	Random	LCUB_8	ELB_4	ROI_3	Random
10	0.845	0.844	0.833	0.829	0.650	0.642	0.656	0.666	0.735	0.729	0.733	0.738
11	0.825	0.824	0.820	**0.834**	**0.715**	0.672	0.669	0.681	**0.766**	0.739	0.735	0.750
12	**0.856**	0.818	0.843	0.838	0.678	0.703	**0.706**	0.674	0.757	0.754	**0.768**	0.747
13	0.865	0.868	**0.878**	0.832	**0.718**	0.699	0.697	0.715	**0.785**	0.774	0.777	0.769
14	0.832	0.844	0.841	**0.855**	0.742	0.752	**0.763**	0.719	0.783	0.793	**0.800**	0.780
15	**0.866**	0.863	0.842	0.847	0.730	0.752	**0.769**	0.719	0.792	0.804	**0.804**	0.777
16	**0.872**	0.868	0.864	0.865	0.751	0.747	**0.769**	0.732	0.807	0.803	**0.814**	0.793
17	0.857	**0.870**	0.866	0.853	0.768	0.771	**0.779**	0.759	0.810	0.817	**0.820**	0.803
18	0.865	0.866	**0.872**	0.863	0.774	**0.776**	0.766	0.761	0.817	**0.818**	0.815	0.808
19	0.857	0.863	**0.876**	0.867	**0.797**	0.792	0.781	0.757	0.825	0.825	**0.826**	0.808

Table 20. Performance of various active learning strategies and passive learning of the i2b2 dataset (with a batch size of 1 document).

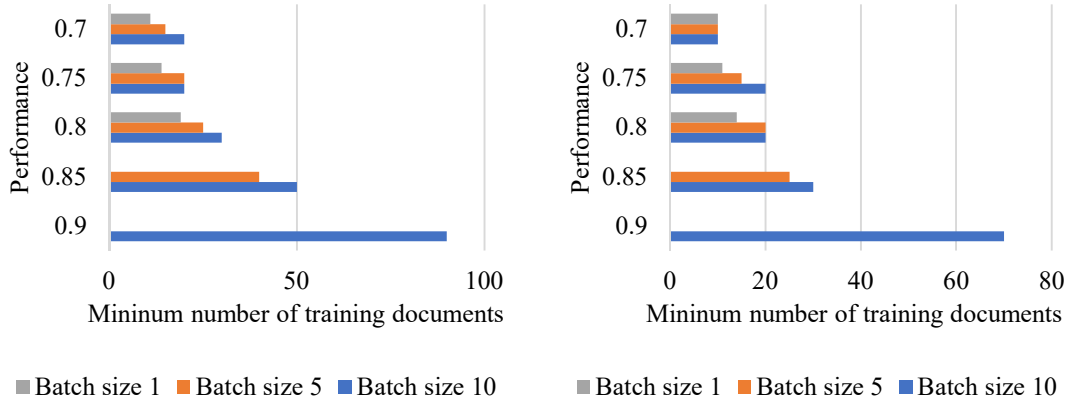


(a) Learning curves for overall recall (batch size = 10) (b) Learning curves for overall recall (batch size = 1)

Figure 33. Recall comparison for active and passive learning (with a batch size of 10 documents or 1 document) for the i2b2 dataset.

4.3.3.2.4 Batch Size Comparison

Again, to investigate the impact of batch sizes on active learning, we plot the minimum number of documents that the system trains on to reach a certain level of recall or F-measure for the i2b2 dataset (Figure 34). Generally, we could reach a similar finding with the dataset A, that is, it requires less documents for the training to attain a certain level of performance when opting for a smaller batch size. For example, batch size of 10 needs 30 documents for a recall of 0.8, while batch size of 5 requires 25 documents, and bath size 1 needs 19.



(a) Overall recall comparison (i2b2) (b) Overall F-measure comparison (i2b2)

Figure 34. Comparison of active learning batch sizes for a given number of training documents that was provided to reach a certain level of performance for the i2b2 dataset.

4.4 Discussion

There are several notable findings that we wish to highlight from this investigation.

First and foremost, the simulation results for both dataset A and the i2b2 dataset lend credibility to the hypothesis that adopting active learning in training data selection for natural language de-identification could generally result in more efficient learning than selecting data randomly (passive learning). It is reasonable to conclude that active learning could lead to comparable or higher performance scores with less amount of training data needed than passive learning.

Additionally, it is worth mentioning that various query strategies in active learning exhibited different trends in learning. Depending on the specific learning goals (e.g., focusing on

overall performance or an individual PHI type) the decision for which active learning query strategy should be adopted could vary. ROI often generated a more stable learning curve than other strategies for dataset A, but the finding did not hold for the i2b2 dataset, which may be due to the fact that the i2b2 data is highly regularized and the results with it do not always transfer to real datasets [27].

Finally, the choice of batch size could play a non-trivial part in the learning process, as smaller batch sizes could aid faster learning, yet result in a considerable increase in re-training time. For dataset A, when batch size is 1, overall precision of active learning might be exceeded by random selection, while active learning remained the advantage over passive learning in recall and F-measure.

Chapter 5 Conclusion

5.1 Summary of results and contributions

This dissertation focused on addressing the scalability challenge in machine learning based natural language de-identification through three tasks.

Task 1 formally quantified the tradeoff between data publishing and privacy risks existing in a natural language de-identification framework that incorporates HIPS strategy with a game theoretical framework. We modeled the data sharing process as a game and conducted case studies to show that, it is possible in some cases for the HCO to make decisions without exhausting its budget to ensure a malicious but rational recipient of de-identified EMR data not to attempt re-identification. Also, we demonstrated that how adversaries pay their penalties for violating terms of service significantly influences how data is shared, and could even result in perverse outcomes. Specifically, we showed that if the HCO that shares the data is entitled to damages for violation of a contractual agreement, it may be incentivized inappropriately to bait an attacker by publishing patient data that is potentially exploitable.

In Task 2, we constructed a feature extraction and clustering strategy based on writing complexity and clinical vocabulary usage to partition clinical documents into inferred types in order to better utilize a given set of data for de-identification model training. We conducted experiments on the clustering framework to show that, 1) training on a stylometric cluster yields better de-identification performance when testing on the same cluster, 2) the stylometric de-identification models yield better results than random documents generated models, 3) the performance of stylometric models is, in many instances, better than models created by documents grouped according to VUMC-designated document types. The above findings suggest that it is possible to achieve higher fidelity de-identification models with less training data and institutional knowledge.

For the last task of this dissertation, we developed an active learning pipeline for natural language de-identification, and evaluated the pipeline on a real-world clinical trials dataset through simulations. The results from the experiments confirmed that utilizing active learning in training data sampling could generate models with comparable or better performances with less data than

passive learning. Additionally, we discovered that adopting different query strategies in active learning could lead to different trends in learning. Finally, we showed that the choice of batch sizes could influence the performance of active learning in de-identification, more specifically, smaller batch sizes could lead to faster learning while increasing re-training time.

5.2 Limitations of the work and future directions

For the game theory based resource allocation framework, we recognize that our study is limited in its scope for several reasons. First, our model focused on economic motivated attackers, whereas reputation-driven attackers are out of scope of this dissertation. Also, we acknowledge that our case studies are based on a single dataset. For a future direction, we could further investigate the generalizability of the game based framework experimenting with other datasets. It is challenging to conduct such investigations on EMRs with real residual identifiers, since public use datasets (e.g., i2b2 dataset [27] or Cincinnati Children's hospital dataset [59]) have replaced all real identifiers with synthetic identifiers. Finally, it should be recognized that the definition of cost plays a key role in our framework. Although a sensitivity analysis was performed to determine the stability of the findings, the costs per record (depending on the actual content in an EMR) may change and needs more extensive study in the future.

As for task 2, we used a VUMC-specialized version of DE-ID as a proxy for a human-annotated corpus which might result in lower F-measures than those observed in gold standard environments. Again, our data was mainly based on single healthcare organization, which is the VUMC, and it is possible that data from a different institution might not exhibit the same in performance. While acquiring substantial amount data from other sites that is composed of several document types could be challenging in practice, it could be considered for future extensions of this research. Next, although the VUMC EMR contains over 1000 document types, we only chose to focus on nine of the larger size types to report on interpretable results and to avoid an overly complex analysis. Lastly for this task, we acknowledge that features based on readability and lexical richness may be correlated with other unknown factors, such that, the new PHI instances introduced by the MIST resynthesis module might influence the grammar and phrasing.

Finally, despite the merits of our investigation, there is opportunity for improvement in active learning in our setting. Here we provide several such opportunities. First, in the ROI model, we integrated human annotation cost and contribution and provided results based on simulations,

but the actual costs and contributions in real-world problems need to be measured through user studies. Second, instead of fixing a batch size over the entire active learning process, an adaptive batch sizing strategy might lead to better training performance. Lastly, deep learning methods (e.g., recurrent neural networks) might be considered for the active learning system, as they have recently shown promising results in de-identificaiton [89]. Note that, while the proposed pipeline will likely still be applicable by simply switching the machine learning basis, it is possible that doing so might influence the performance.

Appendices

Appendix A1: Summary of PHI Types in the VUMC Dataset

The corpus used in Chapter 2 consists of 11 types of PHI. The specific types and details for each are listed in Table A1. Note that there are several reasons for why we selected this set of PHI types.

First, this research is based on a collaboration between a team at the Vanderbilt University Medical Center (VUMC) and a team at the Group Health Research Institute (GHRI). At the VUMC, EMR data is currently de-identified as 15 types of PHI. By contrast, GHRI generated a dataset that consisted of 19 types of PHI for a project sponsored under the ONC SHARP program. Though there are less PHI types at the VUMC, these are not a proper subset of the GHRI set, a harmonization of the two sets was performed. Table A2 provided an alignment of the various PHI types in the de-identification projects upon which the HIPS evaluation was based. More specifically, the PT_NAME and DOC_NAME in the SHARP dataset could translate to be one PHI type as NAME in the VUMC dataset; the SHARP PHI type ADDRESS was represented by STREET-ADDRESS, PLACE and ZIP-CODE; the PHONE and FAX in the SHARP corpus corresponded to the PHONE in the VUMC corpus; the SHARP types of VEHICLE_ID, CERT_NUM, ACCT_NUM, PLAN_BENF_NUM, OTHER_ID, MED_REC_NUM, and SSN were all covered by ID-NUM in the VUMC dataset; the SHARP dataset had the type of IP_ADDR which was not included in the VUMC set, while the VUMC provided PATH-NUMBER and INITIALS which were not part of the SHARP dataset. After the harmonization, the PHI types colored in Table A2 are the ones that were utilized in the HIPS evaluation.

The second reason is that, at the time of this study, four of the of PHI types 1) IP addresses, 2) URLs, 3) room numbers, and 4) doctor's initials did not have plausible resynthesis modules in place. As such these types of PHI were removed from consideration under the HIPS model. Since this paper only considers the capability of the HIPS approach, we did not investigate these PHI types.

#	**PHI Type**	**Description**
1	Address	A street address, possibly including the city, state, and ZIP code
2	Age	The age of the entity
3	Date	A date, including the year
4	Email	An email address
5	Medical Record Number	A medical record number for a patient
6	Organization Name	The name of any organization associated with the patient or their healthcare.
7	Other ID	An ID number other than the ones identified above, such as a medical device ID or run number of an assay in a laboratory.
8	Patient Name	The personal name of a patient or relative.
9	Provider Name	The name of a medical provider
10	Phone	A telephone or fax number
11	Social Security Number	A Social Security Number for a patient

Table A1. Summary of PHI types in the studied corpus.

SHARP PHI Type	SHARP description	Vanderbilt PHI Type	Vanderbilt description
ORG_NAME	A medical facility	INSTITUTION	A medical facility
PT_NAME	The name of a patient	NAME	The name of a patient or medical provider
DOC_NAME	The name of a medical provider		
DATE	A date, including the year	DATE	A date, including the year
ADDRESS	A street address, possibly including the city, state, and ZIP code	STREET-ADDRESS	A street address, not including the city, state, or ZIP code
		PLACE	A city and/or state
		ZIP-CODE	A ZIP code
		ROOM	A hospital room number
PHONE	A telephone number	PHONE	A telephone number
FAX	A fax telephone number		
AGE	An age	AGE	An age
EMAIL	An email address	EMAIL	An email address
URL	A URL	WEB-LOC	A URL
VEHICLE_ID	An identifying number for a vehicle	ID-NUM	Any identification number
CERT_NUM	Certificate number		
ACCT_NUM	A patient account number		
PLAN_BENF_NUM	Plan beneficiary number		
OTHER_ID	An ID number other than the ones identified above		
MED_REC_NUM	A medical record number		
SSN	A Social Security number		
DEV_ID	A device ID serial number	DEVICE-ID	A device ID number
		PATH-NUMBER	A pathology number
IP_ADDR	An Internet IP address		
		INITIALS	The initials of a medical provider

Table A2. Alignment of the various PHI types in the de-identification projects upon which the HIPS evaluation was based.

Appendix A2: The Processes of Decision Making for Cases *Low*, *Mid-low* and *High*

Figures A1 through A5 illustrate results of the game for the *Low*, *Mid-low* and *High* cases under the various policies. The upper plots in these figures depict the attacker's optimized decisions for training as a function of the publisher's potential training decisions. The lower plots in these figures depict the actual payoffs to the two players.

When the attacker's income is forced to \$0, which indicates the Safe-forward (No risk) policy, the *Low* (Figure A1 and A2) and *High* (Figure A4 and A5) cases lead to positive payoffs for the publisher rather than the \$0 payoffs in the Traditional policy. Figures A1 and A2 suggest that, in the *Low* case, no matter where the attacker's penalty payment is sent, the publisher will reach the same decision in all three policies (i.e., Safe-forward, Attack-forward, and Attack-back), yielding a payoff of \$1558.22.

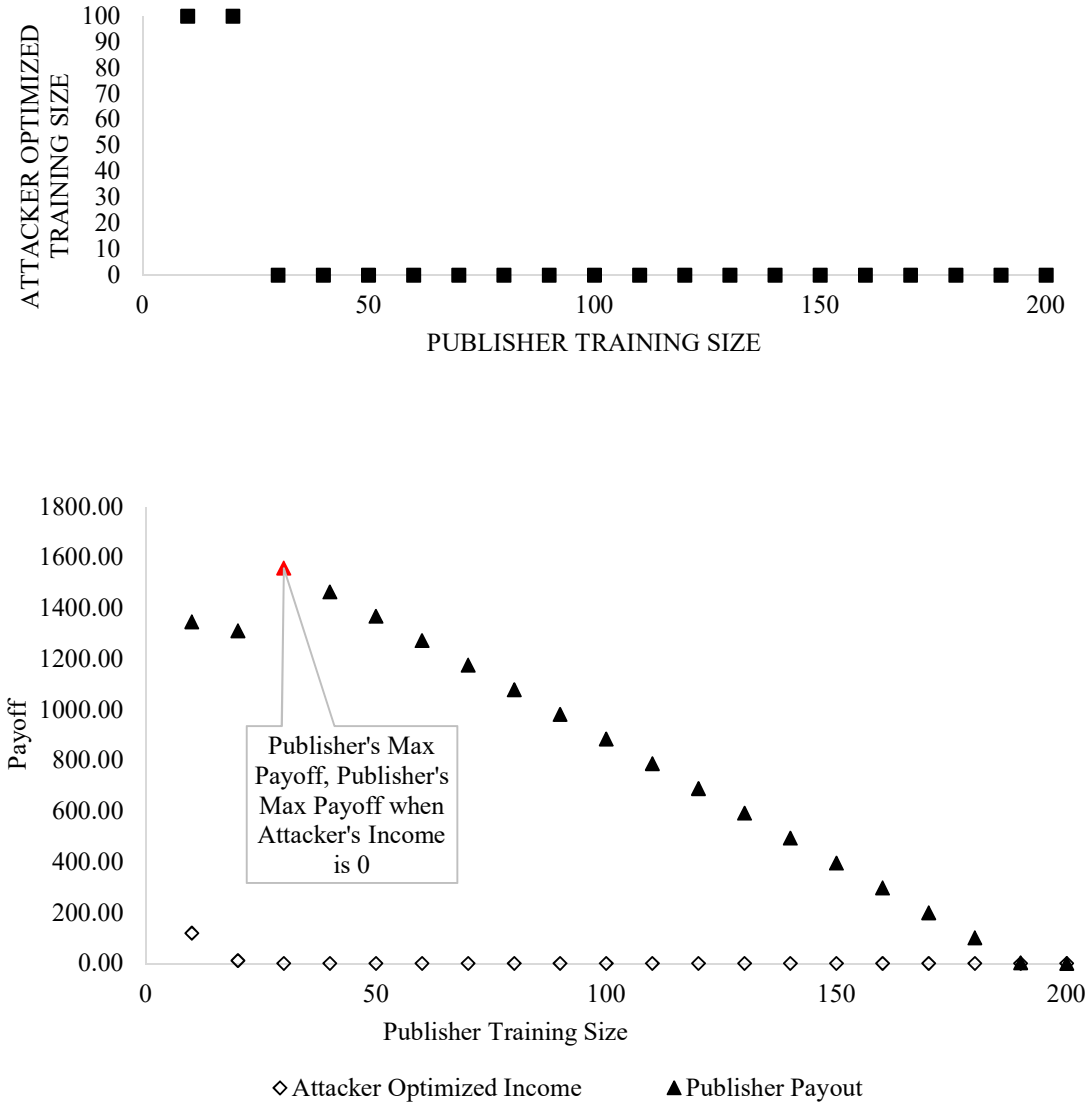


Figure A1. Decision making process of the publisher and corresponding strategies of the attacker in the *Low* case when the attacker pays its penalty forward to a third party.

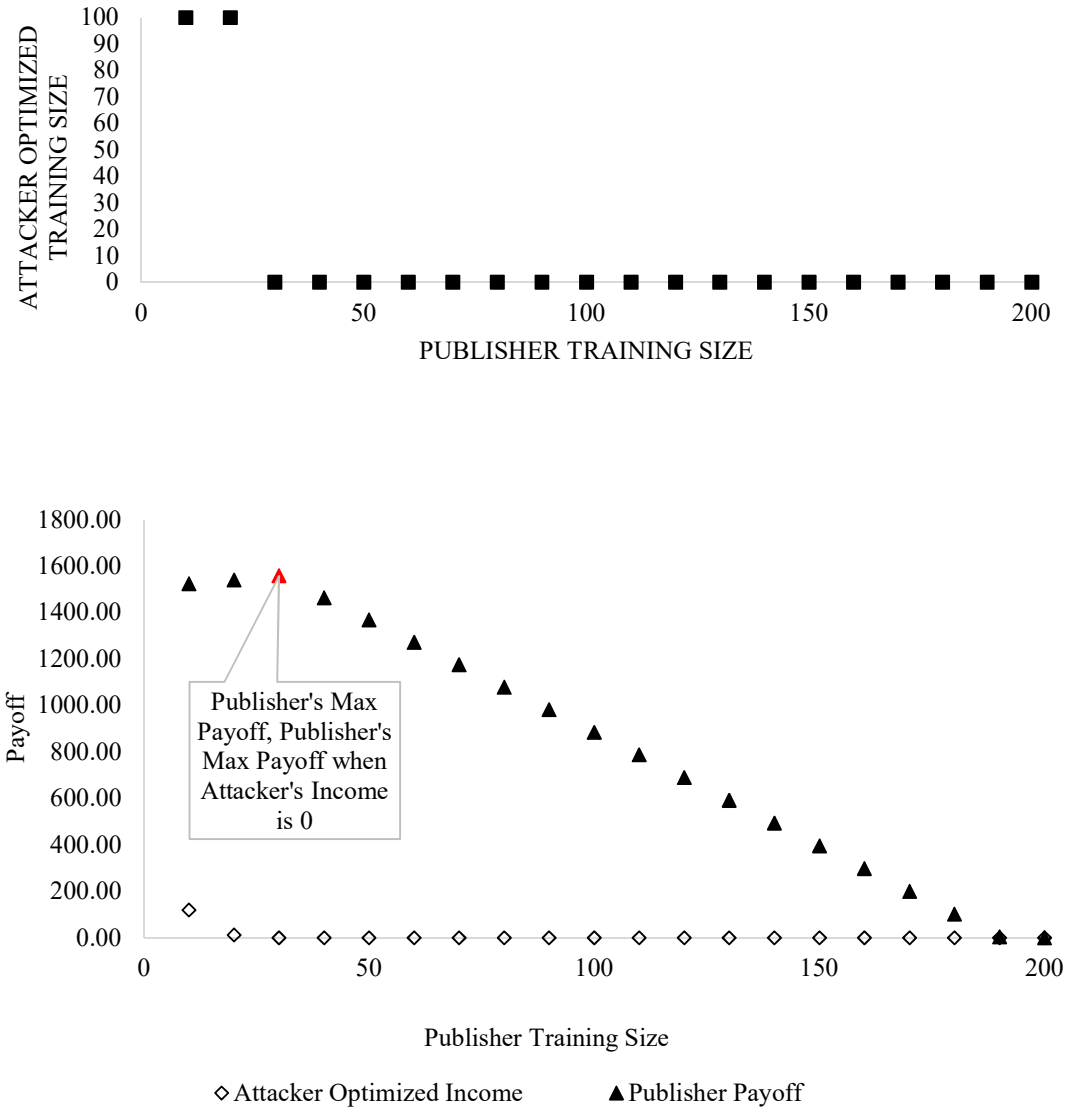


Figure A2. Decision making process of the publisher and corresponding strategies of the attacker in the *Low* case when the attacker pays its penalty back to the publisher.

Figure A3 corresponds to the *Mid-low* case. In this situation, when the publisher's aim is to suppress the attacker, its best option yields a payoff of $0. This means the publisher chooses not to share the data. Figure A3 also indicates that if the publisher aims for the maximum payoff, it tends to undertrain and bait the attacker. This suggests the case in which the attacker bears $0 penalty could put the records of patients in a hazardous situation.

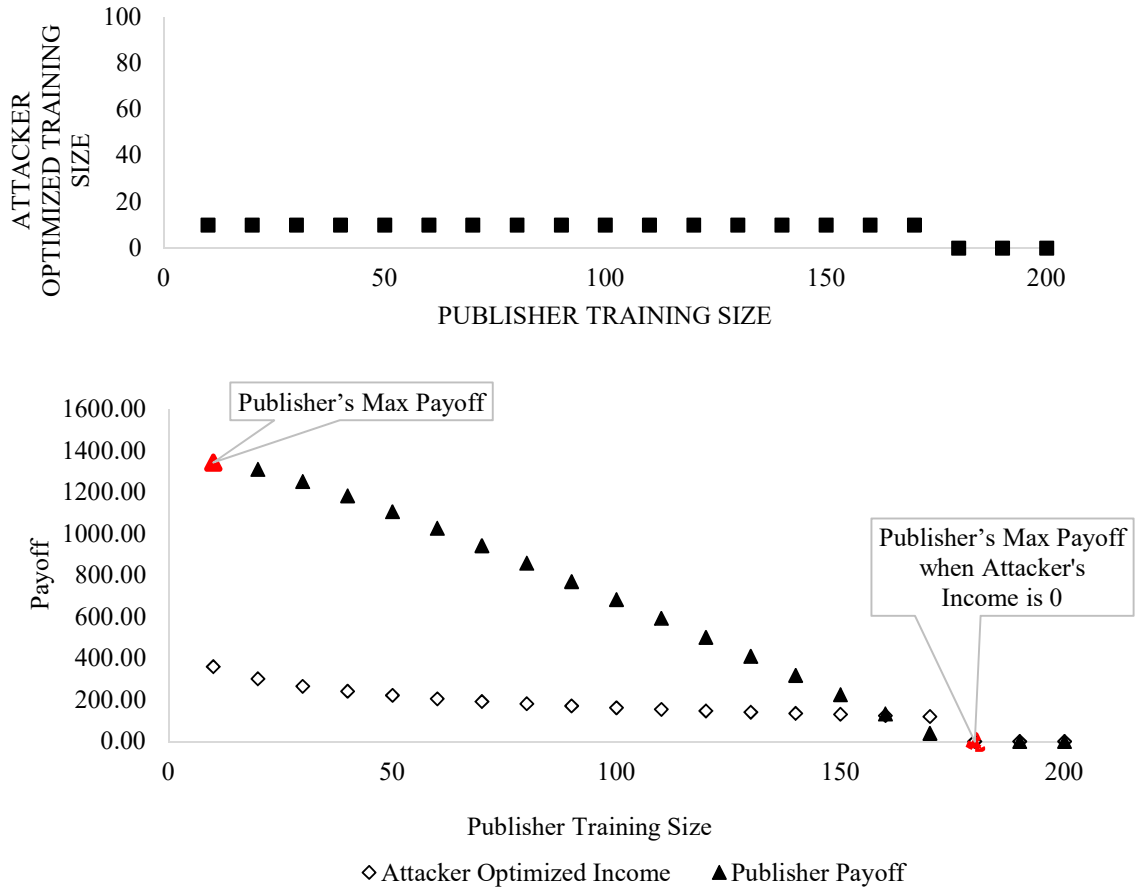


Figure A3. Decision making process of the publisher and corresponding strategies of the attacker in the *Mid-low* case.

Figures A4 and A5 correspond to the *High* case when the attacker's penalty is sent forward to a third party and back to the publisher, respectively. Notably, the Safe-forward policy reaches the same decision wherever the attacker's penalty goes, in which the publisher trains with 190 documents, redacts and shares the data, and yields a payoff slightly larger than $0. The publisher's decisions differ in the Attack-forward and Attack-back policies. Specifically, the latter encourages the publisher to train with a relatively larger dataset and leads to less payoff to the attacker.

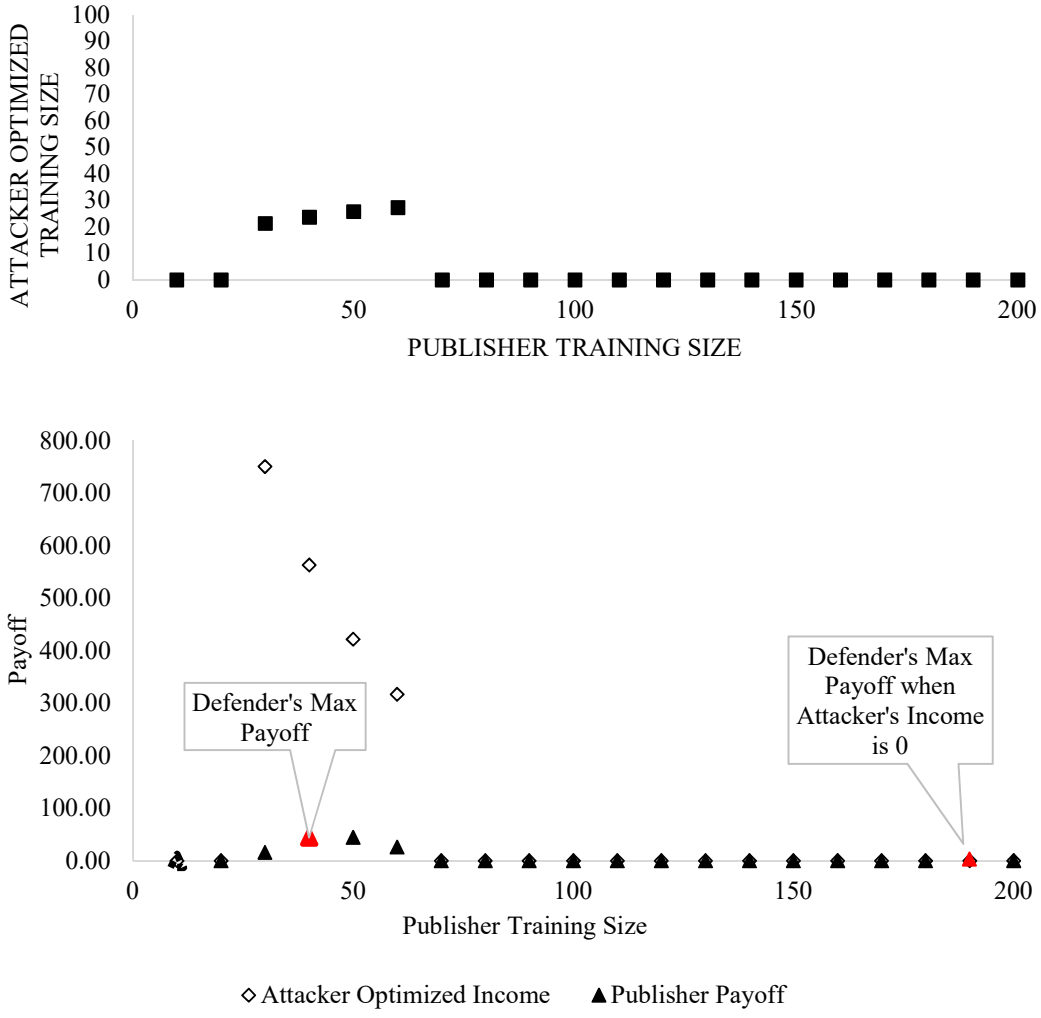


Figure A4. Decision making process of the publisher and corresponding strategies of the attacker in the *High* case when the attacker's pays its penalty forward to a third party.

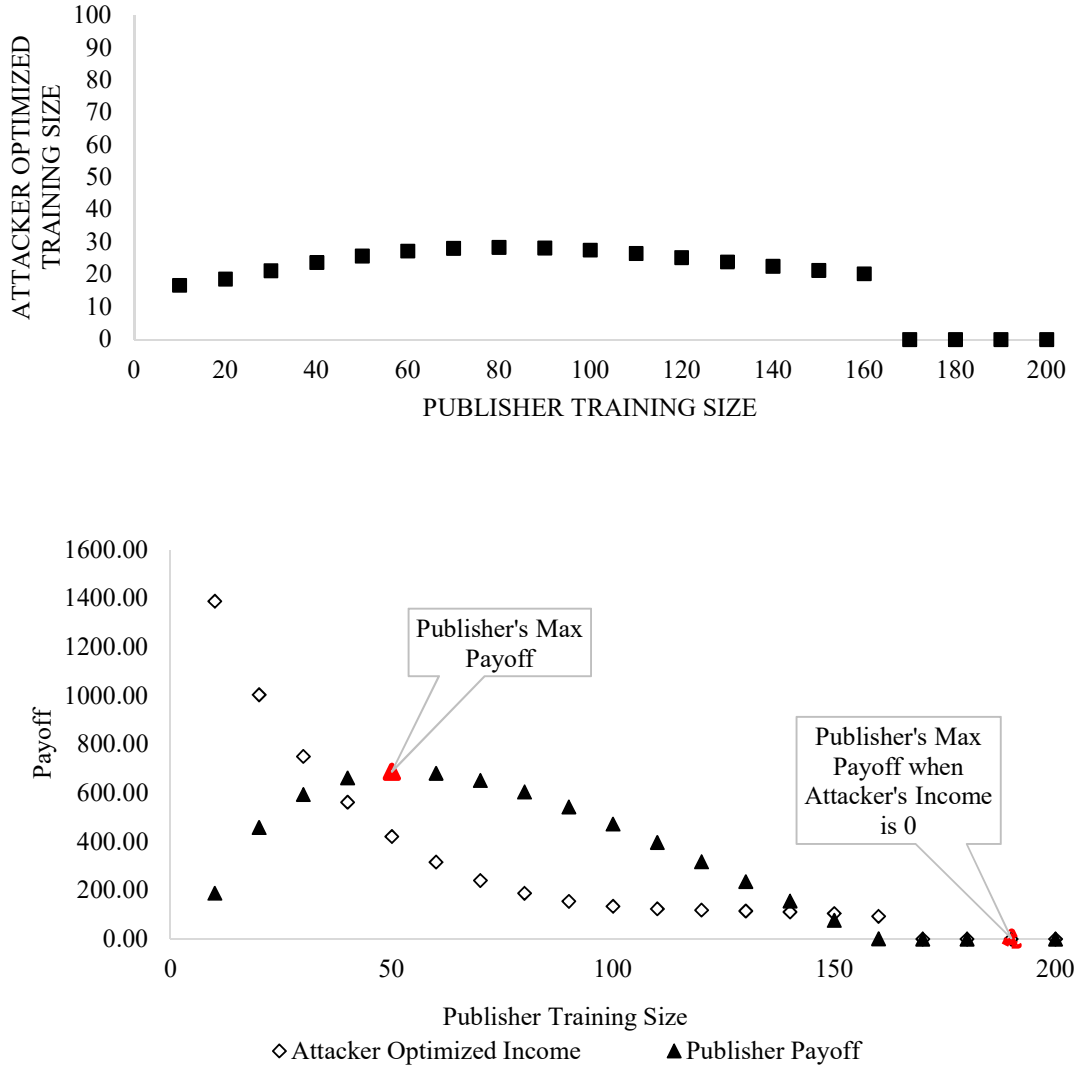


Figure A5. Decision making process of the publisher and corresponding strategies of the attacker in the *High* case when the attacker's pays its penalty back to the publisher.

Appendix A3: Sensitivity Analysis for Cases *Mid-low* and *High*

Figures A6 and A7 depict the sensitivity analysis for the *Mid-low* and the *High* cases, respectively. In each figure, the value of the attacker's annotation cost per document (c_a) is fixed. The horizontal axis represent the attacker's value per true guess (v), while the vertical axis represents its fine for false guesses per instance (l_a). The heat maps show the change of payoffs for both players, such that the lighter the yellow, the more payoff the publisher / attacker is left with.

Figure A6 shows the sensitivity analysis for the four policies (i.e., Traditional, Safe-forward, Attacker-forward, and Attack-back) for the *Mid-low* case, where c_a = \$4. In the Traditional policy, for all possible combinations of v and l_a, the publisher's payoff is negative, while the "no attack" region is realized when $l_a \geq 2/3(v - 0.1)$. In the forward penalty payment policies varying v incurs a larger impact on the publisher's payoff than varying l_a. The larger the v, the smaller the publisher's payoff. In the Attack-back policy, l_a exhibits an increased impact over the forward policies because the fine of the attacker is paid back to the publisher; however, it v continues to exhibit a major impact.

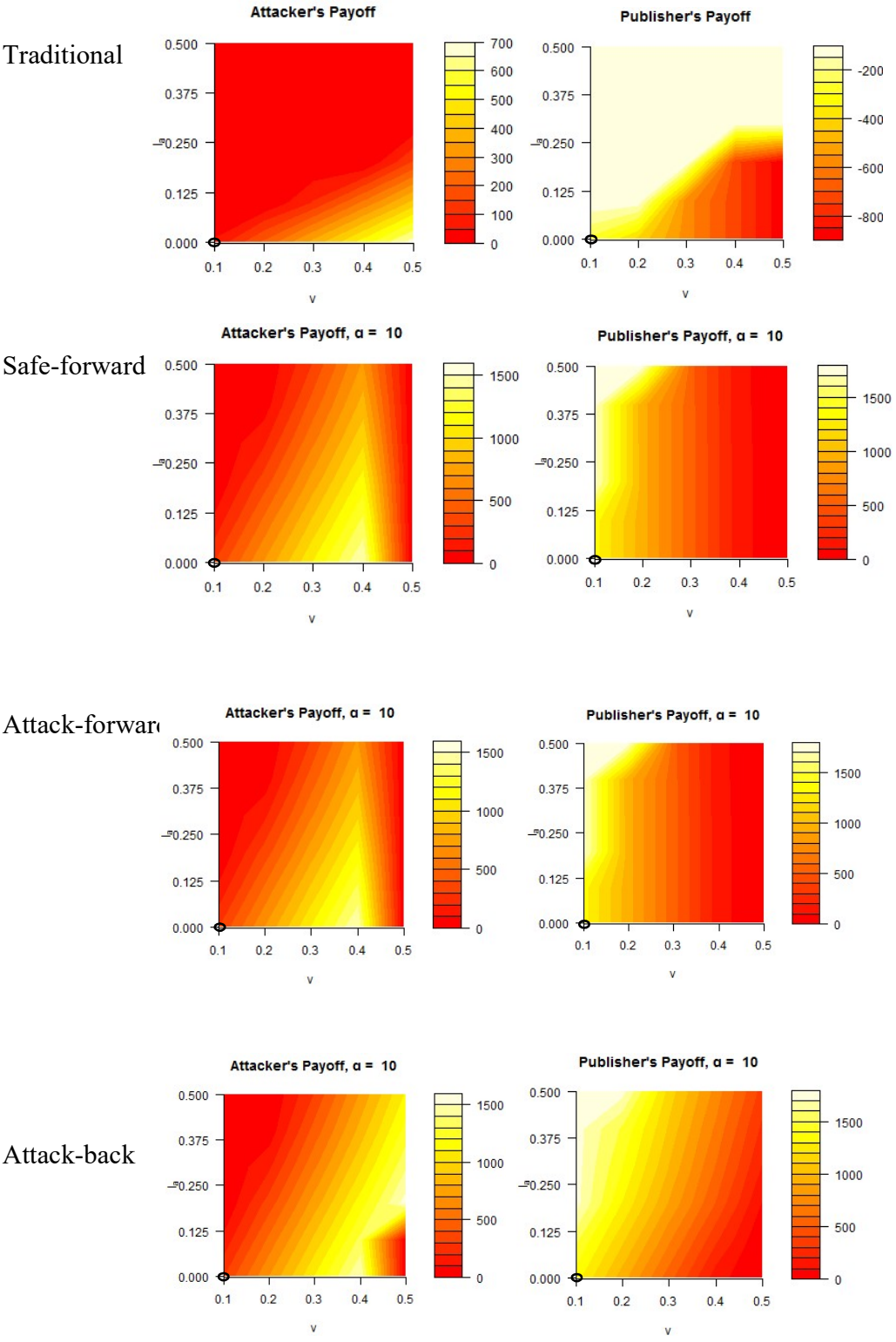


Figure A6. Sensitivity of attacker's and publisher's payoffs to the attacker's value per true positive (v) and loss per false positive (l_a) for the four policies when the attacker's annotation cost per EMR (c_a) is \$4. The result for the *Mid-low* case (v = \$0.1 , l_a = 0) is circled in each figure.

Figure A7 shows the sensitivity analysis for the four policies in the *High* case, where c_a = \$10. Again, the publisher's payoff remains below \$0 in the Traditional policy, while the "no attack" region is realized when $l_a \geq 2/3(v - 0.1)$. In the Safe-forward policy, the publisher's payoff is divided into two parts: 1) an all red region, which indicates \$0 payoff because the publisher fails to share any data and 2) the remainder of the space where there is a positive payoff for the publisher because it is devoid of any committed attacks. For both the Attack-back and the Attack-forward polices, the attacker's payoff is \$0 when $v < l_a$. In the Attack-forward policy, the publisher's payoff is affected more by v than l_a, while in the Attack-back policy both v and l_a exhibit a considerable degree of impact on the publisher's payoff.

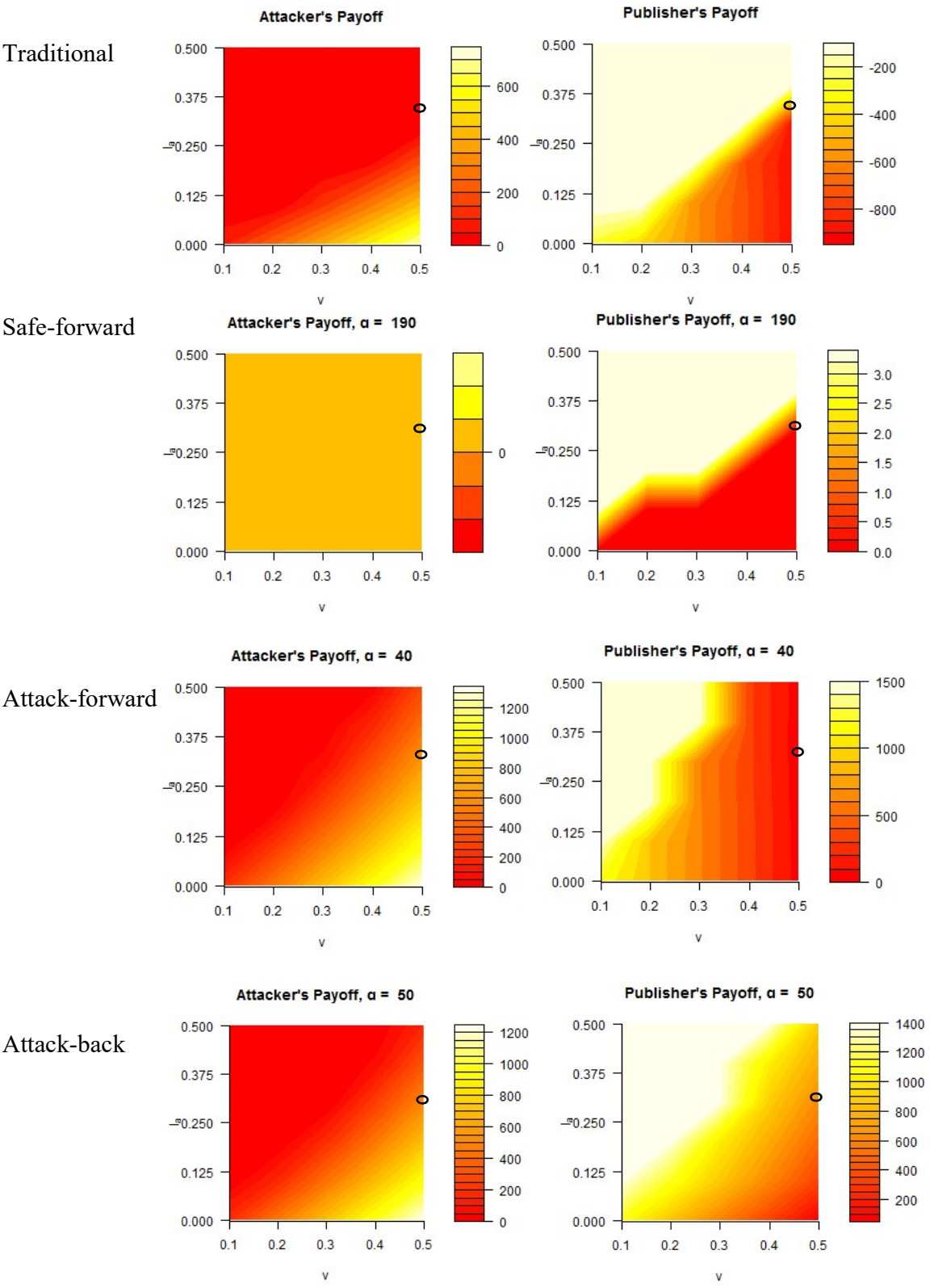


Figure A7. Sensitivity of attacker's and publisher's payoffs to the attacker's value per true positive (v) and loss per false positive (l_a) for the four policies when the attacker's annotation cost per EMR (c_a) is \$10. The result for the *High* case (v = \$0.5, l_a = \$0.3) is circled in each figure.

Appendix B1

This appendix provides an illustration of the experimental design for the scalability tests reported in the main manuscript.

The scenario depicted in Figure B1 corresponds to the experimental setting in which de-identification models are trained on large random mixture clusters and tested on stylometric clusters. In this setting, we randomly sampled from the large mixture of random clusters, with sizes ranging from 100 to 1000. The models were trained on such random subsets and tested on complexity based clusters.

The scenario depicted in Figure B2 corresponds to the experimental setting in which de-identification models are both trained and tested on stylometric clusters. In this case, de-identification models were trained and tested on subsets of stylometric clusters, sizing from 10 to 200.

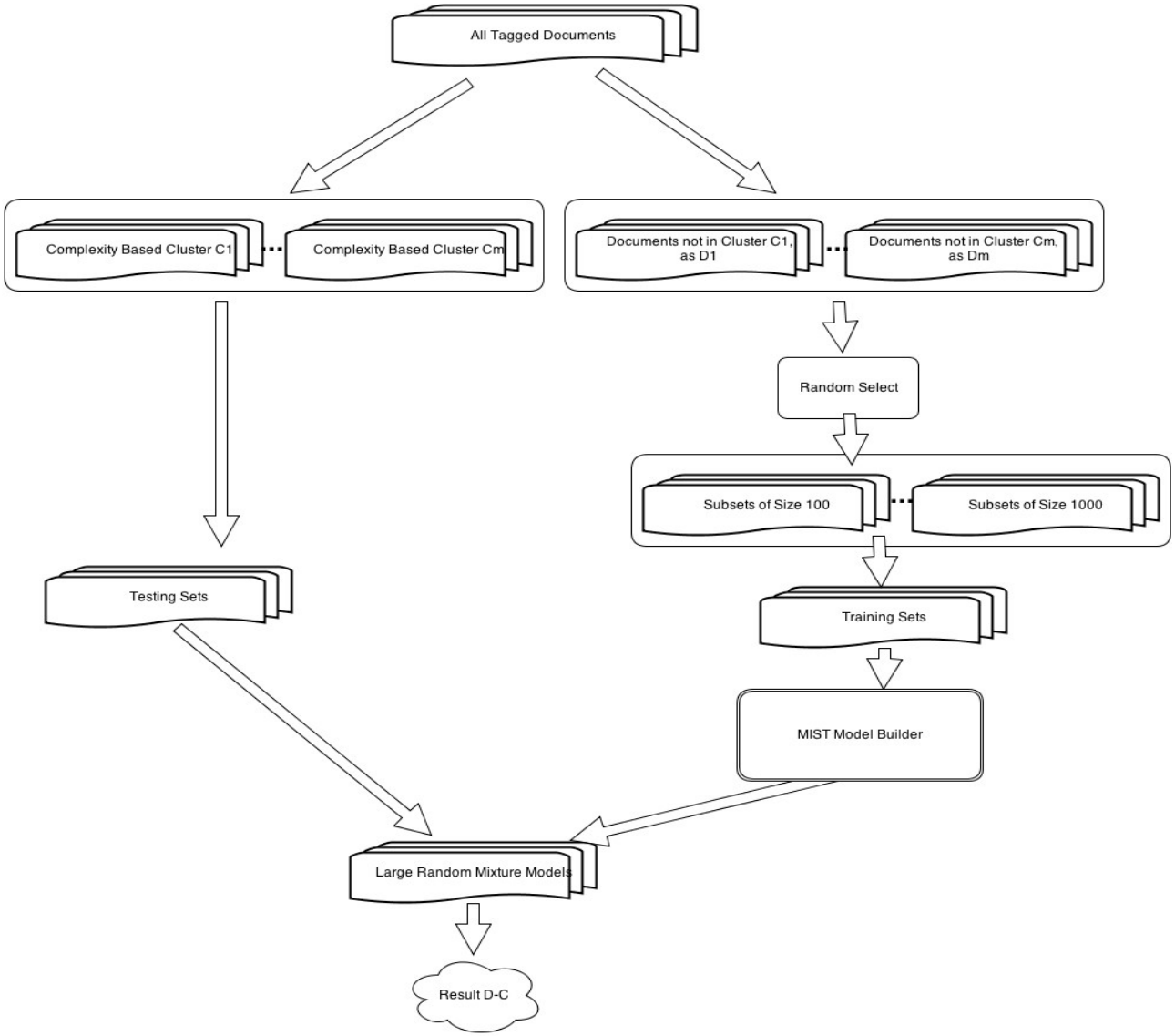


Figure B1. Experimental design of the scalability test for large random mixture clusters.

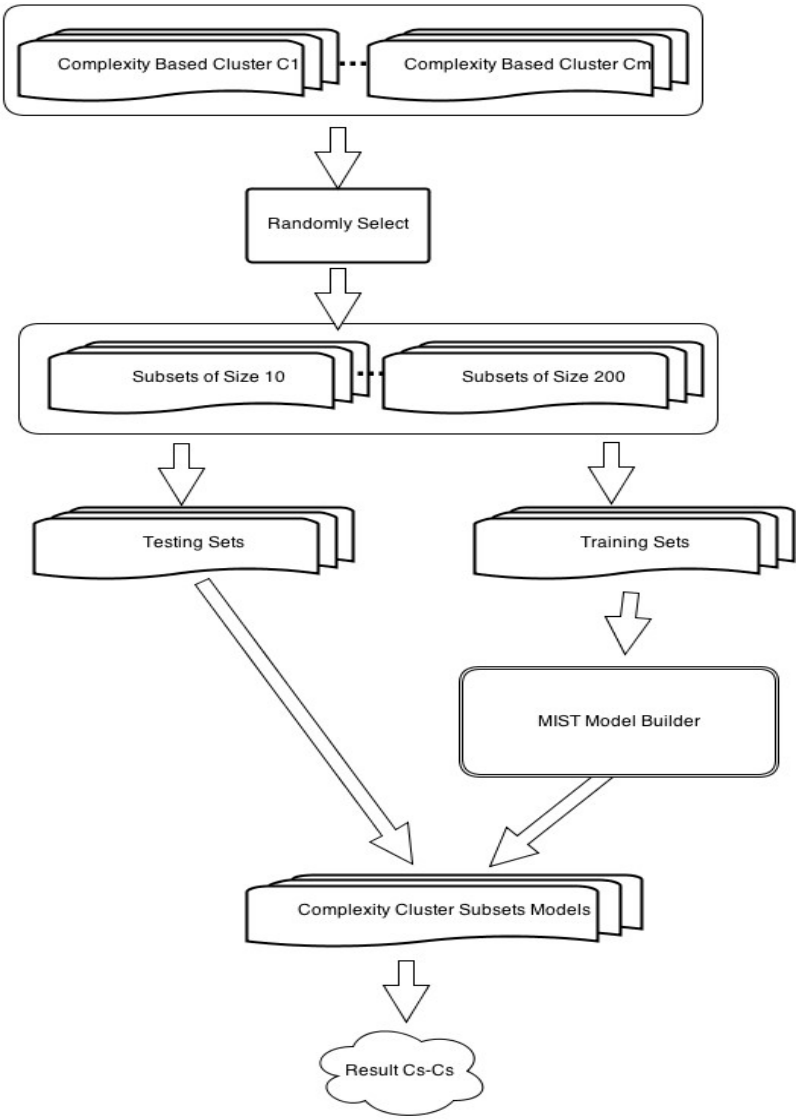


Figure B2. Experimental design of the scalability test for complexity based clusters.

Appendix B2

This section contains complete results (in terms of precision, recall and F-measure) of the de-identification model training and testing experiments.

Due to page width limitations, results for complexity based clusters and random clusters are split into 3 tables each. Tables B1-a to B1-c correspond to clusters created by complexity measures. Tables B2-a to B2-c are showing the results for random process generated clusters.

Table B1-a. De-identification performance (in terms of Precision, Recall, and F-measure) based on clusters derived from the complexity measures.

Train ⇒	C_1			C_2			C_3			C_4		
⇓Test	P	R	F	P	R	F	P	R	F	P	R	F
C_1 (n = 547)	0.922	0.921	0.921	0.839	0.806	0.821	0.899	0.913	0.906	0.902	0.918	0.910
C_2 (n = 229)	0.877	0.886	0.879	0.892	0.815	0.851	0.873	0.891	0.881	0.868	0.874	0.871
C_3 (n = 569)	0.943	0.935	0.939	0.873	0.842	0.857	0.940	0.947	0.944	0.937	0.939	0.938
C_4 (n = 403)	0.842	0.845	0.843	0.740	0.674	0.705	0.830	0.840	0.835	0.867	0.876	0.872
C_5 (n = 531)	0.923	0.925	0.924	0.870	0.805	0.836	0.897	0.918	0.907	0.914	0.927	0.921
C_6 (n = 362)	0.849	0.852	0.851	0.759	0.689	0.721	0.826	0.859	0.842	0.860	0.883	0.871
C_7 (n = 323)	0.937	0.942	0.940	0.891	0.828	0.858	0.936	0.939	0.937	0.935	0.947	0.941
C_8 (n = 247)	0.881	0.876	0.878	0.809	0.725	0.764	0.852	0.867	0.859	0.881	0.892	0.887
C_9 (n = 611)	0.872	0.858	0.865	0.790	0.749	0.768	0.819	0.860	0.839	0.846	0.878	0.862
C_{10} (n = 439)	0.766	0.772	0.769	0.674	0.586	0.625	0.749	0.766	0.758	0.815	0.832	0.823
C_{11} (n = 92)	0.988	0.970	0.978	0.947	0.878	0.910	0.976	0.985	0.980	0.989	0.997	0.993
C_{12} (n = 184)	0.935	0.834	0.882	0.757	0.758	0.757	0.844	0.837	0.840	0.839	0.873	0.855
C_{13} (n = 60)	0.994	0.569	0.722	0.978	0.542	0.687	0.993	0.681	0.808	0.997	0.610	0.755

Table B1-b. De-identification performance (in terms of Precision, Recall, and F-measure) based on clusters derived from the complexity measures.

Train ⇒	C_5			C_6			C_7			C_8		
⇓Test	P	R	F	P	R	F	P	R	F	P	R	F
C_1 (n = 547)	0.905	0.887	0.896	0.906	0.895	0.901	0.892	0.879	0.885	0.892	0.869	0.881
C_2 (n = 229)	0.862	0.814	0.837	0.878	0.851	0.864	0.812	0.792	0.801	0.842	0.806	0.822
C_3 (n = 569)	0.931	0.905	0.918	0.934	0.907	0.920	0.930	0.920	0.925	0.925	0.884	0.904
C_4 (n = 403)	0.849	0.818	0.833	0.859	0.856	0.857	0.817	0.802	0.809	0.847	0.833	0.840
C_5 (n = 531)	0.928	0.902	0.915	0.934	0.917	0.925	0.886	0.886	0.886	0.920	0.895	0.908
C_6 (n = 362)	0.865	0.826	0.845	0.879	0.877	0.878	0.816	0.809	0.812	0.868	0.869	0.868
C_7 (n = 323)	0.934	0.912	0.922	0.934	0.913	0.923	0.929	0.914	0.922	0.926	0.892	0.908
C_8 (n = 247)	0.885	0.849	0.866	0.902	0.893	0.897	0.838	0.819	0.828	0.901	0.891	0.896
C_9 (n = 611)	0.880	0.846	0.863	0.901	0.879	0.890	0.798	0.827	0.812	0.879	0.847	0.863
C_{10} (n = 439)	0.785	0.753	0.768	0.826	0.832	0.829	0.732	0.722	0.726	0.801	0.804	0.803
C_{11} (n = 92)	0.986	0.957	0.970	0.980	0.980	0.980	0.983	0.972	0.977	0.947	0.962	0.955
C_{12} (n = 184)	0.975	0.897	0.934	0.957	0.794	0.867	0.894	0.781	0.833	0.948	0.748	0.836
C_{13} (n = 60)	0.983	0.197	0.326	0.949	0.286	0.438	0.984	0.310	0.469	0.963	0.088	0.160

Table B1-c. De-identification performance (in terms of Precision, Recall, and F-measure) based on clusters derived from the complexity measures.

Train ⇒	**C_9**			**C_{10}**			**C_{11}**			**C_{12}**			**C_{13}**		
⇓Test	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
C_1 (n = 547)	0.894	0.865	0.879	0.877	0.856	0.866	0.344	0.601	0.433	0.796	0.732	0.762	0.115	0.347	0.172
C_2 (n = 229)	0.855	0.803	0.827	0.852	0.818	0.834	0.337	0.513	0.398	0.752	0.669	0.706	0.103	0.340	0.157
C_3 (n = 569)	0.925	0.878	0.901	0.911	0.855	0.882	0.382	0.629	0.466	0.819	0.768	0.792	0.136	0.367	0.198
C_4 (n = 403)	0.834	0.818	0.826	0.853	0.853	0.853	0.393	0.579	0.457	0.761	0.705	0.731	0.127	0.293	0.176
C_5 (n = 531)	0.929	0.907	0.918	0.905	0.908	0.907	0.418	0.653	0.495	0.847	0.794	0.819	0.095	0.358	0.150
C_6 (n = 362)	0.875	0.849	0.862	0.872	0.895	0.884	0.293	0.546	0.376	0.770	0.712	0.739	0.086	0.267	0.130
C_7 (n = 323)	0.929	0.890	0.909	0.906	0.875	0.890	0.417	0.614	0.492	0.814	0.796	0.805	0.114	0.345	0.169
C_8 (n = 247)	0.892	0.860	0.875	0.883	0.897	0.890	0.280	0.495	0.352	0.769	0.701	0.731	0.068	0.254	0.106
C_9 (n = 611)	0.912	0.899	0.906	0.854	0.871	0.862	0.350	0.639	0.446	0.793	0.745	0.768	0.079	0.302	0.125
C_{10} (n = 439)	0.780	0.764	0.772	0.883	0.895	0.889	0.290	0.524	0.364	0.667	0.613	0.638	0.105	0.256	0.148
C_{11} (n = 92)	0.973	0.986	0.979	0.972	0.980	0.976	0.963	0.957	0.960	0.922	0.921	0.920	0.378	0.718	0.492
C_{12} (n = 184)	0.925	0.860	0.891	0.820	0.787	0.803	0.459	0.557	0.492	0.973	0.981	0.977	0.199	0.331	0.246
C_{13} (n = 60)	0.551	0.156	0.243	0.965	0.194	0.319	0.394	0.062	0.107	0.897	0.054	0.101	0.994	0.987	0.990

Table B2-a. De-identification performance (in terms of Precision, Recall, and F-measure) based on random clusters

Train ⇒	**R_1**			**R_2**			**R_3**			**R_4**		
⇓Test	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
R_1 (n = 547)	0.893	0.888	0.891	0.886	0.877	0.881	0.884	0.882	0.883	0.892	0.885	0.888
R_2 (n = 229)	0.900	0.894	0.897	0.891	0.881	0.886	0.891	0.885	0.888	0.900	0.894	0.897
R_3 (n = 569)	0.902	0.903	0.903	0.903	0.896	0.899	0.896	0.896	0.896	0.914	0.907	0.910
R_4 (n = 403)	0.888	0.884	0.886	0.885	0.878	0.881	0.888	0.889	0.888	0.892	0.892	0.892
R_5 (n = 531)	0.895	0.888	0.892	0.881	0.886	0.883	0.887	0.887	0.887	0.901	0.893	0.897
R_6 (n = 362)	0.889	0.870	0.879	0.887	0.864	0.875	0.888	0.874	0.881	0.898	0.875	0.886
R_7 (n = 323)	0.889	0.884	0.886	0.889	0.884	0.886	0.893	0.892	0.892	0.903	0.896	0.900
R_8 (n = 247)	0.899	0.900	0.899	0.897	0.889	0.893	0.892	0.891	0.891	0.903	0.897	0.900
R_9 (n = 611)	0.892	0.885	0.888	0.883	0.880	0.881	0.889	0.884	0.887	0.897	0.888	0.892
R_{10} (n = 439)	0.886	0.897	0.891	0.901	0.900	0.901	0.903	0.908	0.906	0.909	0.913	0.911
R_{11} (n = 92)	0.891	0.889	0.890	0.903	0.896	0.900	0.891	0.892	0.891	0.900	0.896	0.898
R_{12} (n = 184)	0.896	0.885	0.890	0.895	0.883	0.888	0.886	0.870	0.878	0.901	0.887	0.893
R_{13} (n = 60)	0.944	0.947	0.945	0.941	0.954	0.947	0.946	0.958	0.952	0.948	0.959	0.953

Table B2-b. De-identification performance (in terms of Precision, Recall, and F-measure) based on random clusters

Train ⇒	**R_5**			**R_6**			**R_7**			**R_8**		
⇓Test	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
R_1 (n = 547)	0.885	0.882	0.883	0.881	0.882	0.881	0.894	0.895	0.894	0.890	0.877	0.883
R_2 (n = 229)	0.898	0.890	0.894	0.885	0.888	0.887	0.904	0.902	0.903	0.903	0.895	0.899
R_3 (n = 569)	0.910	0.902	0.906	0.906	0.905	0.905	0.912	0.920	0.916	0.905	0.903	0.904
R_4 (n = 403)	0.889	0.891	0.890	0.885	0.890	0.887	0.897	0.901	0.899	0.885	0.886	0.885
R_5 (n = 531)	0.893	0.883	0.888	0.890	0.884	0.887	0.902	0.904	0.903	0.894	0.888	0.891
R_6 (n = 362)	0.892	0.872	0.882	0.878	0.862	0.869	0.894	0.885	0.890	0.898	0.878	0.887
R_7 (n = 323)	0.892	0.885	0.889	0.888	0.895	0.891	0.901	0.903	0.902	0.894	0.889	0.891
R_8 (n = 247)	0.895	0.894	0.895	0.894	0.894	0.894	0.903	0.910	0.906	0.902	0.904	0.903
R_9 (n = 611)	0.886	0.883	0.884	0.891	0.884	0.887	0.900	0.897	0.899	0.895	0.886	0.890
R_{10} (n = 439)	0.912	0.913	0.913	0.886	0.895	0.890	0.910	0.910	0.910	0.909	0.901	0.905
R_{11} (n = 92)	0.890	0.887	0.889	0.887	0.888	0.887	0.895	0.896	0.896	0.902	0.894	0.898
R_{12} (n = 184)	0.897	0.879	0.887	0.895	0.876	0.884	0.894	0.893	0.893	0.905	0.888	0.896
R_{13} (n = 60)	0.954	0.964	0.959	0.939	0.953	0.945	0.952	0.966	0.959	0.949	0.960	0.954

-

Table B2-c. De-identification performance (in terms of Precision, Recall, and F-measure) based on random clusters

Train ⇒	**R_9**			**R_{10}**			**R_{11}**			**R_{12}**			**R_{13}**		
⇓Test	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
R_1 (n = 547)	0.888	0.882	0.885	0.869	0.842	0.855	0.866	0.852	0.859	0.853	0.830	0.841	0.850	0.741	0.791
R_2 (n = 229)	0.900	0.896	0.898	0.886	0.861	0.873	0.885	0.872	0.878	0.867	0.845	0.856	0.862	0.762	0.809
R_3 (n = 569)	0.908	0.904	0.906	0.895	0.875	0.885	0.888	0.881	0.884	0.879	0.846	0.862	0.867	0.764	0.810
R_4 (n = 403)	0.887	0.888	0.887	0.873	0.855	0.863	0.871	0.859	0.865	0.830	0.836	0.832	0.854	0.743	0.793
R_5 (n = 531)	0.888	0.887	0.887	0.879	0.853	0.866	0.880	0.861	0.870	0.860	0.822	0.840	0.856	0.756	0.801
R_6 (n = 362)	0.893	0.878	0.886	0.878	0.837	0.857	0.874	0.841	0.857	0.840	0.792	0.815	0.859	0.728	0.786
R_7 (n = 323)	0.892	0.892	0.892	0.873	0.843	0.857	0.870	0.852	0.861	0.867	0.827	0.846	0.862	0.756	0.805
R_8 (n = 247)	0.901	0.899	0.900	0.887	0.857	0.872	0.877	0.864	0.871	0.871	0.833	0.851	0.868	0.759	0.809
R_9 (n = 611)	0.891	0.888	0.890	0.884	0.849	0.866	0.883	0.856	0.869	0.865	0.825	0.844	0.873	0.743	0.803
R_{10} (n = 439)	0.899	0.898	0.898	0.872	0.844	0.858	0.889	0.883	0.886	0.864	0.824	0.843	0.864	0.744	0.799
R_{11} (n = 92)	0.892	0.891	0.891	0.879	0.856	0.867	0.890	0.872	0.880	0.867	0.848	0.857	0.855	0.724	0.783
R_{12} (n = 184)	0.896	0.891	0.893	0.875	0.841	0.857	0.887	0.865	0.876	0.855	0.819	0.836	0.855	0.718	0.775
R_{13} (n = 60)	0.944	0.969	0.956	0.926	0.926	0.926	0.943	0.952	0.947	0.913	0.878	0.894	0.931	0.818	0.869

Appendix B3

In this appendix, we provide the scalability results for large random mixture clusters (Figure B3) and complexity measure-based clusters (Figure B4). For each subplot, the x-axis represents the size of subset invoked for trained, while the y-axis corresponds to the F-measure of the result.

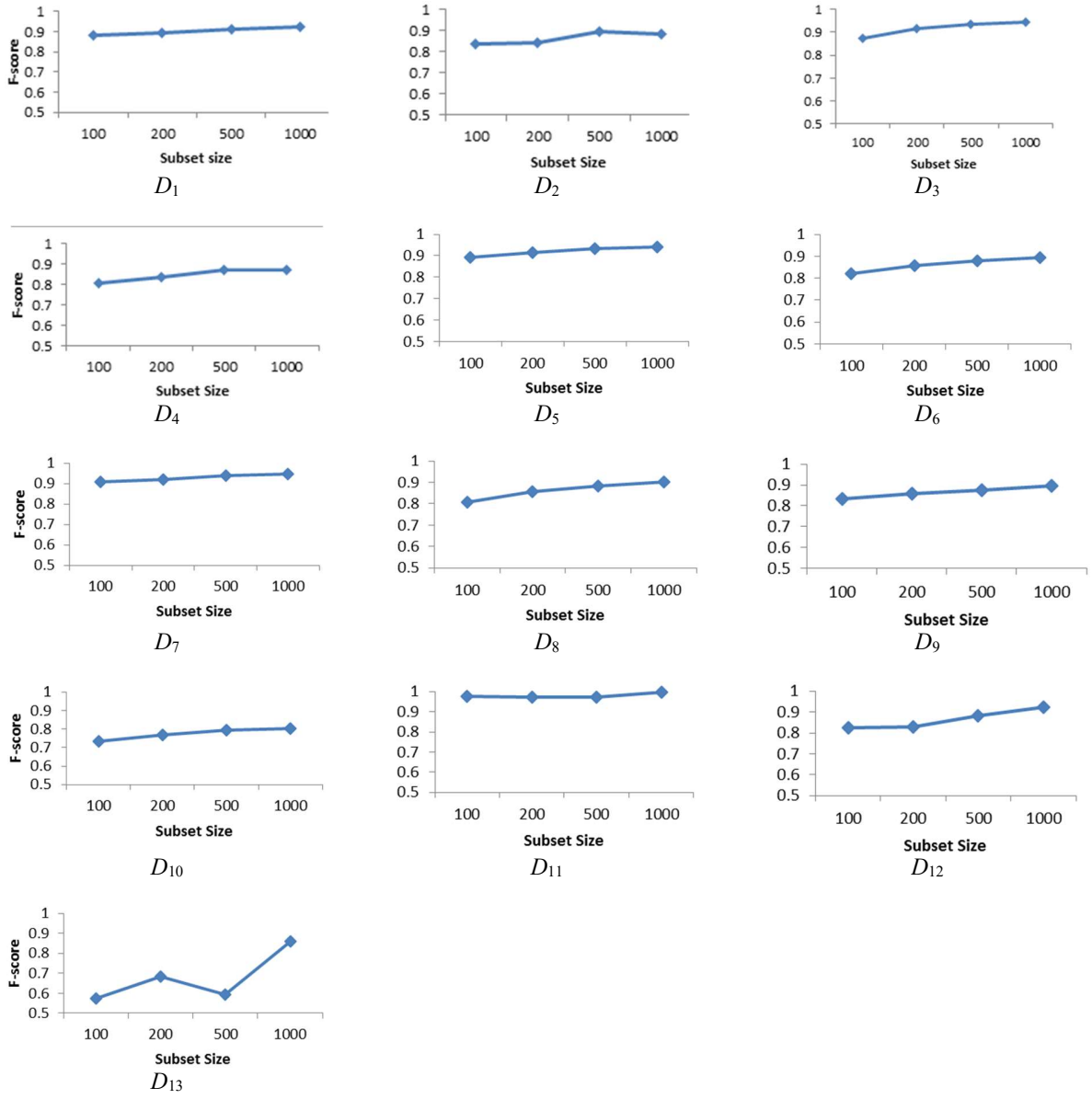


Figure B3. Scalability analysis on large mixed random subsets.

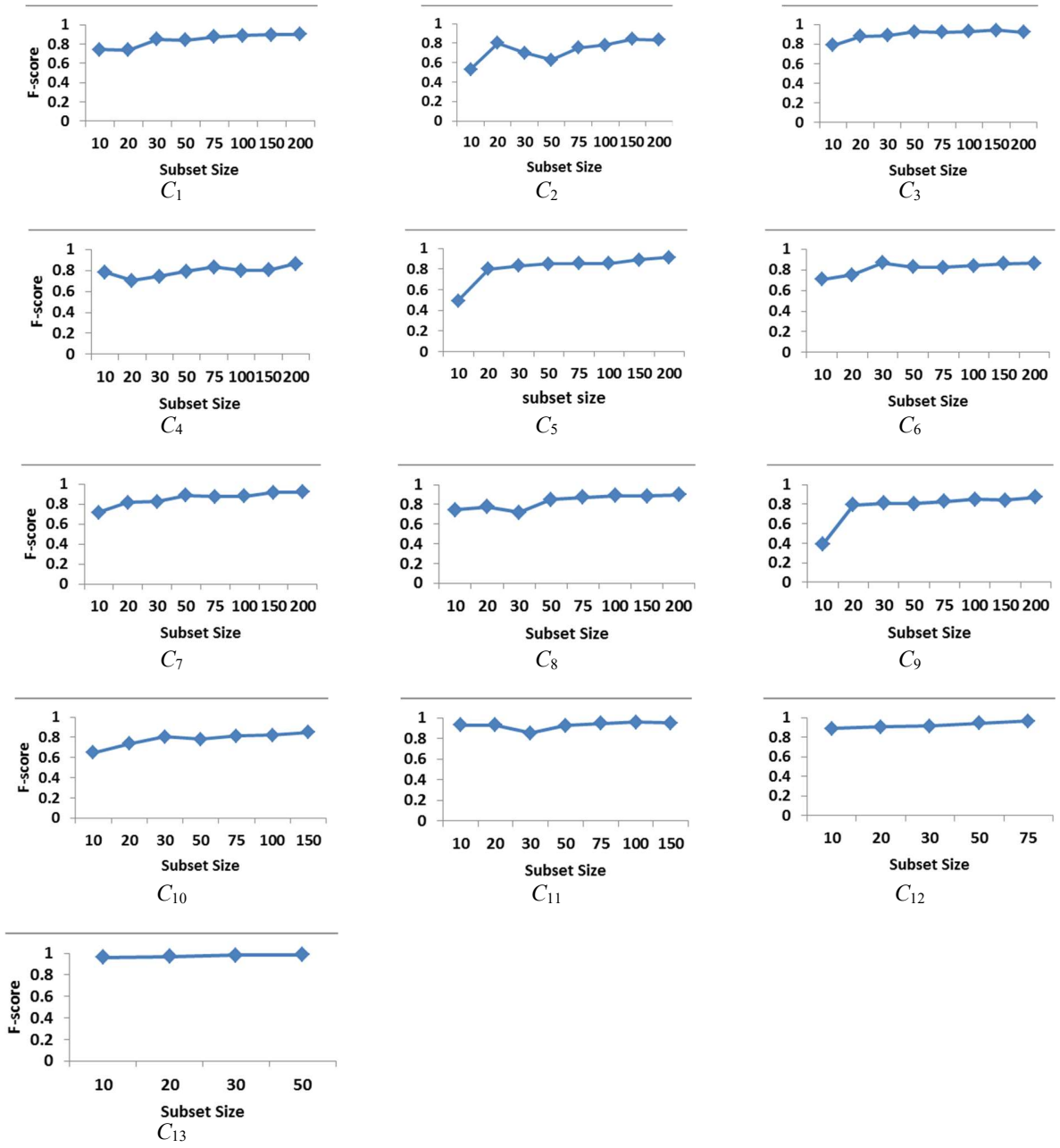


Figure B4. Scalability analysis on stylometric clusters.

References

[1] A. Jha, M. Burke, C. DesRoches, M. Joshi, P. Kralovec, E. Campbell and M. Buntin, "Progress toward meaningful use: hospitals' adoption of electronic health records," *The American journal of managed care,* vol. 17, no. 12, pp. 117 - 24, 2011.

[2] L. Deleger, K. Molnar, G. Savova, F. Xia, T. Lingren, Q. Li, K. Marsolo and e. al., "Large-scale evaluation of automated clinical note de-identification and its impact on information extraction," *Journal of the American Medical Informatics Association ,* vol. 20, no. 1, pp. 84-94, 2013.

[3] W. Stead and H. Lin, Eds., Computational technology for effective health care: immediate steps and strategic directions, Washington, DC: National Academies Press, 2009.

[4] R. R. German, L. M. Lee, J. M. Horan, R. Milstein, C. Pertowski and M. Waller, "Updated guidelines for evaluating public health surveillance systems: recommendations from the Guidelines Working Group," *MMWR. Recommendations and reports: Morbidity and mortality weekly report. Recommendations and reports/Centers for Disease Control.,* vol. 50, no. (RR-13), pp. 1-35, 2001.

[5] W. W. Chapman, J. N. Dowling and M. M. Wagner, "Fever detection from free-text clinical records for biosurveillance," *Journal of Biomedical Informatics,* vol. 37, no. 2, pp. 120-27, 2004.

[6] M. Jiang, Y. Chen, M. Liu, S. T. Rosenbloom, S. Mani, J. C. Denny and H. Xu, "A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries," *Journal of the American Medical Informatics Association ,* vol. 18, no. 5, pp. 60-06, 2011.

[7] K. M. Newton, P. L. Peissig, A. N. Kho, S. J. Bielinski, R. L. Berg, C. Vidhu, M. Basford, C. G. Chute, I. J. Kullo, R. Li, J. A. Pacheco, L. V. Rasmussen, L. Spangler and J. C. Denny, "Validation of electronic medical record-based phenotyping algorithms: results and lessons learned from the eMERGE network," *Journal of the American Medical Informatics Association,* vol. 20, no. e1, pp. e147-e154, 2013.

[8] C. A. McCarty, R. L. Chisholm, C. G. Chute, I. J. Kullo, G. P. Jarvik, E. B. Larson, R. Li and e. al., "The eMERGE Network: a consortium of biorepositories linked to electronic medical records data for conducting genomic studies," *BMC Medical Genomics,* vol. 4, no. 1, p. 13, 2011.

[9] National Institutes of Health, "Final NIH Statement on Sharing Research Data. NOT-OD-03-032," February 26, 2003.

[10] E. Emam, K. E. Jonker, L. Arbuckle and B. Malin, "A systematic review of re-identification attacks on health data," *PloS One,* vol. 6, no. 12, p. e28071, 2011.

[11] D. A. Ludwick and J. Doucette, "Primary care physicians' experience with electronic medical records: barriers to implementation in a fee-for-service environment," *International Journal of Telemedicine and Applications,* vol. 2009, no. 2, pp. 1-9, 2009.

[12] B. Malin, K. El Emam and C. O'Keefe, "Biomedical data privacy: problems, perspectives, and recent advances," *Journal of the American medical informatics association,* vol. 20, no. 1, pp. 2 - 6, 2013.

[13] U.S. Department of Health and Human Services, "Standards for privacy and individually identifiable health information. Final rule," vol. 67, no. 157, pp. 53181 - 53273, 2002.

[14] E. M. Agency, "External guidance on the implementation of the European," 22 September 2017. [Online]. Available: http://www.ema.europa.eu/docs/en_GB/document_library/Regulatory_and_procedural_guideline/2017/09/WC500235371.pdf.

[15] T. Van Vleck, A. Wilcox, P. Stetson, S. Johnson and N. Elhadad, "Content and structure of clinical problem lists: a corpus analysis," *AMIA Annual Symposium Proceedings,* pp. 753 - 7, 2008.

[16] S. Johnson, S. Bakken, D. Dine, S. Hyun, E. Mendonca, F. Morrison, T. Bright, T. Van Vleck, J. Wrenn and P. Stetson, "An electronic health record based on structured narrative," *Journal of the American Medical Informatics Association : JAMIA,* vol. 15, no. 1, pp. 54 - 64, 2008.

[17] X. Zhou, K. Zheng, M. Ackerman and D. Hanauer, "Cooperative documentation: the patient problem list as a nexus in electronic health records," *Proceedings of ACM Conference on Computer Supported Cooperative Work,* pp. 911- 20, 2012.

[18] W. W. Chapman, P. M. Nadkarni, L. Hirschman, L. W. D'Avolio, G. K. Savova and O. Uzuner, "Overcoming barriers to NLP for clinical text: the role of shared tasks and the need for additional creative solutions," *Journal of the American Medical Informatics Association : JAMIA,* vol. 18, no. 5, pp. 540-3, 2011.

[19] S. Rosenbloom, J. Denny, H. Xu, N. Lorenzi, W. Stead and K. Johnson, "Data from clinical notes: a perspective on the tension between structure and flexible documentation," *Journal of the American Medical Informatics Association : JAMIA,* vol. 18, no. 2, pp. 181 - 6, 2011.

[20] D. Demner-Fushman, W. Chapman and C. McDonald, "What can natural language processing do for clinical decision support?," *Journal of Biomedical Informatics,* vol. 42, no. 5, pp. 760-72, 2009.

[21] D. Dorr, W. Phillips, S. Phansalkar, S. Sims and J. Hurdle, "Assessing the difficulty and time cost of de-identification in clinical narratives," *Methods of Information in Medicine,* vol. 45, no. 3, pp. 246-52, 2006.

[22] J. Aberdeen, S. Bayer, R. Yeniterzi, B. Wellner, C. Clark, D. Hanauer, B. Malin and L. Hirschman, "The MITRE Identification Scrubber Toolkit: design, training, and assessment," *International Journal of Medical Informatics,* vol. 79, no. 12, pp. 849-59, 2010.

[23] J. Gardner and L. Xiong, "An integrated framework for de-identifying unstructured medical data," *Data & Knowledge Engineering,* vol. 68, no. 12, pp. 1441-51, 2009.

[24] G. Szarvas, R. Farkas and R. Busa-Fekete, "State-of-the-art anonymisation of medical records using an iterative machine learning framework," *Journal of the American Medical Informatics Association,* vol. 14, no. 5, pp. 574-80, 2007.

[25] O. Ferrández, B. R. South, S. Shen, F. J. Friedlin, M. H. Samore and S. M. Meystre, "BoB, a best-of-breed automated text de-identification system for VHA clinical documents," *Journal of the American Medical Informatics Association,* vol. 20, no. 1, pp. 77-83, 2013.

[26] S. M. Meystre, F. J. Friedlin, B. R. South, S. Shen and M. H. Samore, "Automatic de-identification of textual documents in the electronic health record: a review of recent research," *BMC Medical Research Methodology,* vol. 10, no. 1, p. 70, 2010.

[27] Ö. Uzuner, Y. Luo and P. Szolovits, "Evaluating the state-of-the-art in automatic de-identification," *Journal of the American Medical Informatics Association ,* vol. 14, no. 5, pp. 550-63, 2007.

[28] D. Carrell, B. Malin, J. Aberdeen, S. Bayer, C. Clark, B. Wellner and L. 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 ,* vol. 20, no. 2, pp. 342-48, 2013.

[29] S. Velupillai, H. Dalianis, M. Hassel and G. Nilsson, "Developing a standard for de-identifying electronic patient records written in Swedish: precision, recall and F-measure in a manual and computerized annotation trial," *International journal of medical informatics,* vol. 78, no. 12, pp. e19 - 26, 2009.

[30] I. Neamatullah, M. L. L. Douglass, A. Reisner, M. Villarroel, W. Long, P. Szolovits, G. Moody, R. Mark and G. Clifford, "Automated de-identification of free-text medical records," *BMC medical informatics and decision making,* vol. 8, no. 1, p. 1, 2008.

[31] D. Gupta, M. Saul and J. Gilbertson, "Evaluation of a de-identification software engine: Progress towards sharing clinical documents and pathology reports," *American journal of clinical pathology,* vol. 121, no. 2, pp. 176-86, 2004.

[32] R. Yeniterzi, J. Aberdeen, S. Bayer, B. Wellner, L. Hirschman and B. Malin, "Effects of personal identifier resynthesis on clinical text de-identification," *Journal of the American Medical Informatics Association,* vol. 17, no. 2, pp. 159-68, 2010.

[33] T. Dietterich, "Ensemble methods in machine learning," *International workshop on multiple classifier systems,* pp. 1 - 15, 2000.

[34] R. K. Taira, A. A. Bui and H. Kangarloo, "Identification of patient name references within medical documents using semantic selectional restrictions," *AMIA Annual Symposium Proceedings,* pp. 757-61, 2002.

[35] H. Dalianis and H. Boström, "Releasing a swedish clinical corpus after removing all words–de-identification experiments with conditional random fields and random forests," *Proceedings of the Third Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM 2012) held in conjunction with LREC 2012,* pp. 45 - 8.

[36] Ö. Uzuner, T. C. Sibanda, Y. Luo and P. Szolovits, "A de-identifier for medical discharge summaries," *Artificial Intelligence in Medicine,* vol. 42, no. 1, pp. 13-35, 2008.

[37] B. Wellner, M. Huyck, S. Mardis, J. Aberdeen, A. Morgan, L. Peshkin, A. Yeh, J. Hitzeman and L. Hirschman, "Rapidly retargetable approaches to de-identification in medical records," *Journal of the American Medical Informatics Association,* vol. 14, no. 5, pp. 564-73, 2007.

[38] J. Lafferty, A. McCallum and F. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," *Proceedings of 18th International Conference on Machine Learning,* pp. 282 - 9, 2001.

[39] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology,* pp. 134 - 41, 2003.

[40] B. Settles, "Biomedical named entity recognition using conditional random fields and rich feature sets.," *Proceedings of International Joint Workshop on Natural Language Processing in Biomedicine and its Applications,* pp. 104 - 7, 2004.

[41] A. McCallum, "MALLET: A Machine Learning for Language Toolkit," 2002. [Online]. Available: http://mallet.cs.umass.edu.

[42] J. Finkel, T. Grenager and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," *Proceedings of 43rd Annual Meeting on Association for Computational Linguistics,* pp. 363-70, 2005.

[43] B. Wellner, "Sequence models and ranking methods for discourse parsing," *Doctoral Dissertation, Brandeis University, Waltham, MA,* 2009.

[44] W. Sun, R. Anna and O. and Uzuner, "Normalization of relative and incomplete temporal expressions in clinical narratives," *Journal of the American Medical Informatics Association,* vol. 22, no. 5, pp. 1001-08, 2015.

[45] A. Stubbs, C. Kotfila and O. Uzuner, "Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/UTHealth shared task Track 1.," *Journal of Biomedical Informatics,* vol. 58, no. Supplement, pp. S11-9, 2015.

[46] A. Stubbs and O. Uzuner, "Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus," *Journal of Biomedical Informatics,* vol. 58, no. Supplement, pp. S20-9, 2015.

[47] I.-H. Hann, K.-L. Hui, T. Lee and I. Png, "Online information privacy: Measuring the cost-benefit trade-off," *Proceedings of the 23rd International Conference on Information Systems,* vol. 1, p. 1, 2002.

[48] A. Acquisti, L. K. John and G. Loewenstein, "What is privacy worth?," *The Journal of Legal Studies,* vol. 42, no. 2, pp. 249-74, 2013.

[49] A. Acquisti, A. Friedman and R. Telang, "Is there a cost to privacy breaches? An event study," *Proceedings of the 27th International Conference on Information Systems,* p. 94, 2006.

[50] R. H. Khokhar, R. Chen, B. C. Fung and S. M. Lui, "Quantifying the costs and benefits of privacy-preserving health data publishing," *Journal of biomedical informatics,* vol. 50, pp. 107-21, 2014.

[51] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Computing Surveys,* vol. 45, no. 3, p. 25, 2013.

[52] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez and S. Kraus, "Playing games for security: an efficient exact algorithm for solving Bayesian Stackelberg games," *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems,* vol. 2, pp. 895-902, 2008.

[53] W. Liu and S. Chawla, "A game theoretical model for adversarial learning," *Proceedings of the Workshops of the IEEE International Conference on Data Mining,* pp. 25-30, 2009.

[54] M. Kantarcioglu, A. Bensoussan and S. C. Hoe, "When do firms invest in privacy-preserving technologies?," *Decision and Game Theory for Security,* pp. 72-86, 2010.

[55] M. Brückner and T. Scheffer, "Stackelberg games for adversarial prediction problems," *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM,* pp. 547-55, 2011.

[56] L. Rajbhandari and E. A. Snekkenes, "Using game theory to analyze risk to privacy: An initial insight," in *Privacy and Identity Management for Life*, Springer Berlin Heidelberg, 2011, pp. 41-51.

[57] J. Blocki, N. Christin, A. Datta, A. D. Procaccia and A. Sinha, "Audit games," *Proceedings of the Twenty-Third international Joint Conference on Artificial Intelligence, Francesca Rossi (Ed.),* pp. 41-7, 2013.

[58] Z. Wan, Y. Vorobeychik, W. Xia, E. W. Clayton, M. Kantarcioglu, R. Ganta, R. Heatherly and B. A. Malin, "A Game Theoretic Framework for Analyzing Re-Identification Risk," *PloS One,* vol. 10, no. 3, p. e0120592, 2015.

[59] L. Deleger, T. Lingren1, Y. Ni, M. Kaiser, L. Stoutenborough, K. Marsolo, M. Kouril, K. Molnar and I. Solti, "Preparing an annotated gold standard corpus to share with extramural investigators for de-identification research," *Journal of Biomedical Informatics,* vol. 50, pp. 173-83, 2014.

[60] D. Holmes, "Authorship attribution," *Computers and the Humanities,* vol. 28, no. 2, pp. 87 - 106, 1994.

[61] E. Stamatatos, " A survey of modern authorship attribution methods," *Journal of the American Society for Information Science and Technology,* vol. 60, no. 3, pp. 1 - 28, 1998.

[62] M. Koppel and J. Schler, "Computational methods in authorship attribution," *Journal of the American Society for Information Science and Technology,* vol. 60, no. 1, pp. 9 - 26, 2009.

[63] G. a. o. Klare, Measurement of readability, Iowa State University Press, 1963.

[64] G. Klare, "Assessing Readability," *Reading Research Quarterly,* vol. 10, no. 1, pp. 62 - 102, 1974.

[65] W. DuBay, "The principles of readability," *available at: www. impact-information. com/impactinfo/read-ability02. pdf (accessed 29 June 2010),* 2004.

[66] R. Flesch, "A new readability yardstick," *The Journal of applied psychology,* vol. 32, no. 3, pp. 221-33, 1948.

[67] G. Aston, S. Bernardini and D. Stewart, Eds., Corpora and language learners, vol. 17, John Benjamins Publishing, 2004.

[68] F. Tweedie and R. Baayen, "How Variable May a Constant be? Measures of Lexical Richness in Perspective," *Computers and the Humanities,* vol. 32, no. 5, pp. 323 - 52, 1998.

[69] D. Giuse, "Supporting communication in an integrated patient record system," *AMIA Annual Symposium Proceedings,* p. 1065, 2003.

[70] D. Roden, J. Pulley, B. MA, G. Bernard, E. Clayton, J. Balser and e. al., "Development of a large-scale de-identified DNA biobank to enable personalized medicine," *Clinical pharmacology and therapeutics,* vol. 84, no. 3, pp. 362 - 9, 2008.

[71] M. Michalke, "koRpus: An R Package for Text Analysis," 2017. [Online]. Available: https://reaktanz.de/?c=hacking&s=koRpus.

[72] A. Aronson, "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program," *AMIA Annual Symposium Proceedings,* pp. 17 - 21, 2001.

[73] O. Bodenreider, "The Unified Medical Language System (UMLS): integrating biomedical terminology," *Nucleic acids research,* vol. 32(Database issue), pp. D267 - 70, 2004.

[74] F. Lehner, "Quality control in software documentation based on measurement of text comprehension and text comprehensibility," *Information Processing & Management,* vol. 29, no. 5, pp. 551-68, 1993.

[75] P. McCarthy and S. Jarvis, "MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment," *Behavior Research Methods,* vol. 42, no. 2, pp. 381 - 92, 2010.

[76] J. Anderson, "Lix and Rix: variations on a little-known readability index," *Journal of Reading,* vol. 26, no. 6, pp. 490 - 6, 1983.

[77] H. Deng and G. Runger, "Feature selection via regularized trees," *Proceedings of International Joint Conference on Neural Networks,* pp. 1-8, 2012.

[78] R. Mojena, "Hierarchical grouping methods and stopping rules: an evaluation," *The Computer Journal,* vol. 20, no. 4, pp. 359 - 63, 1977.

[79] J. Handl, J. Knowles and D. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics,* vol. 21, no. 15, pp. 3201-12, 2005.

[80] D. Hanauer, J. Aberdeen, S. Bayer, B. Wellner, C. Clark, K. Zheng and L. Hirschman, "Bootstrapping a de-identification system for narrative patient records: cost-performance tradeoffs.," *International journal of medical informatics,* vol. 82, no. 9, pp. 821 - 31, 2013.

[81] M. Douglass, G. D. Clifford, A. Reisner, G. B. Moody and R. G. Mark, "Computer-assisted de-identification of free text in the MIMIC II database," *Computers in Cardiology,* pp. 341-4, 2004.

[82] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning,* vol. 6, no. 1, pp. 1-114, 2012.

[83] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," *Proceedings of the conference on empirical methods in natural language processing (EMNLP),* pp. 1069-78, 2008.

[84] A. Kapoor, E. Horvitz and S. Basu, "Selective Supervision: Guiding Supervised Learning with Decision-Theoretic Active Learning," *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI),* vol. 7, pp. 877-82, 2007.

[85] R. A. Haertel, K. D. Seppi, E. K. Ringger and J. L. Carroll, "Return on investment for active learning," in *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, 2008.

[86] Y. Chen, T. A. Lasko, Q. Mei, J. C. Denny and H. Xu, "A study of active learning methods for named entity recognition in clinical text," *Journal of biomedical informatics,* vol. 58, pp. 11-18, 2015.

[87] H. Boström and H. Dalianis, "De-identifying health records by means of active learning," *Recall (micro),* vol. 97, no. 97.55, pp. 90-97, 2012 .

[88] A. Fong, J. L. Howe, K. T. Adams and R. M. Ratwani, "Using active learning to identify health information technology related patient safety events," *Applied clinical informatics,* vol. 8, no. 1, p. 35, 2017.

[89] F. Dernoncourt, J. Y. Lee, O. Uzuner and P. Szolovits, "De-identification of patient notes with recurrent neural networks," *Journal of the American Medical Informatics Association,* vol. 24, no. 3, pp. 596-606, 2017.