Computer-Assisted De-Identification of Free-text Nursing Notes

by

Margaret Douglass

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

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2005

Author .....
Department of Electrical Engineering and Computer Science
28 January, 2005

Certified by .....
Roger G. Mark
Professor of Electrical Engineering
Distinguished Professor in Health Sciences and Technology
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Computer-Assisted De-Identification of Free-text Nursing Notes

by

Margaret Douglass

Submitted to the Department of Electrical Engineering and Computer Science
on 28 January, 2005, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

Medical researchers are legally required to protect patients' privacy by removing personally identifiable information from medical records before sharing the data with other researchers. Different computer-assisted methods are evaluated for removing and replacing protected health information (PHI) from free-text nursing notes collected in the hospital intensive care unit. A semi-automated method was developed to allow clinicians to highlight PHI on the screen of a tablet PC and to compare and combine the selections of different experts reading the same notes. Expert adjudication demonstrated that inter-human variability was high, with few false positives and many false negatives. A preliminary automated de-identification algorithm generated few false negatives but many false positives. A second automated algorithm was developed using the successful portions of the first algorithm and incorporating other heuristic methods to improve overall performance. A large de-identified collection of nursing notes was re-identified with realistic surrogate (but unprotected) dates, serial numbers, names, and phrases to form a "gold standard" reference database of over 2600 notes (approximately 340,000 words) with over 1800 labeled instances of PHI. This gold standard database of nursing notes and the Java source code used to evaluate algorithm performance will be made freely available on the Physionet web site in order to facilitate the development and validation of future de-identification algorithms.

Thesis Supervisor: Roger G. Mark
Title: Professor of Electrical Engineering
Distinguished Professor in Health Sciences and Technology

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Patients expect their personal medical data to be shared only among the clinicians and others directly concerned with their case. When using the medical data for research purposes, we must continue to respect and preserve patient confidentiality. The de-identification process removes all explicit personal health information in order to dissociate the individual from his medical record, while still preserving all the medically relevant information about the patient.

Software tools were developed in this project to facilitate human expert de-identification of free-text nursing notes. Those tools were used to create a large collection of completely de-identified free-text nursing notes for use as a "gold standard" reference database. That database was used to characterize the performance of human de-identification and to evaluate the performance of a preliminary automated algorithm. Finally, a second fully-automated de-identification algorithm was developed based on the successes and short-comings of the preliminary algorithm, and it was tested on the gold standard reference database.

This first chapter will discuss the general problem of preserving patient privacy in biomedical research and the de-identification guidelines used. The second chapter will describe the software tools developed. The third chapter will describe how those tools were used in the creation of a gold standard de-identified database and in the evaluation of the performance of different de-identifiers. The fourth will cover the results found from the human de-identification and the preliminary de-identification

algorithm. The final chapter will discuss the work done on a new, improved de-identification algorithm and the project's conclusions.

1.1 Patient Privacy

For as long as physicians have been treating patients, patient privacy has been an important concern. The Indian physician Charaka in the sixth century B.C. highly praised the trust between physician and patient, and he advocated patient confidentiality in physician-patient relationship [21]. The Hippocratic oath from 400 B.C., Greece, includes: "Whatever, in connection with my professional service, or not in connection with it, I see or hear, in the life of men, which ought not to be spoken of abroad, I will not divulge, as reckoning that all such should be kept secret" [22]. The American Medical Association's original Code of Ethics from 1847 includes in the description of the duties of physicians: "Secrecy and delicacy, when required by peculiar circumstances, should be strictly observed; and the familiar and confidential intercourse to which physicians are admitted in their professional visits, should be used with discretion, and with the most scrupulous, regard to fidelity and honor" [12]. Modern professional ethics codes and federal and state laws still insist on doctor-patient confidentiality.

The general public is, rightfully, very concerned about who has access to their personal medical data. Some uses of such data are benign and helpful to society, such as the collection of children's immunization records by state and local governments. A child's immunization history is made available in some states to local public health departments, the child's physician, school, and/or child-care facility [34]. Ensuring that everyone is properly immunized is a public health concern, and immunization registries are generally accepted as necessary.

Other types of personal medical data need to be kept strictly confidential between the patient and her physician. People with a history of serious medical problems complain of difficulties securing jobs with life and health insurance benefits once their employers become aware of their previous illnesses [24]. The Americans with

Disabilities Act of 1990 says that employers cannot make personal medical inquiries until after a job offer has been made, but insecure medical databases could allow job interviewers improper access to potential employees' medical records and prevent a survivor of childhood cancer, for example, from getting a job he would otherwise be offered. A 1996 survey of 84 Fortune 500 companies found that 35% of the companies use medical records of personnel in making employment-related decisions [27].

The right to privacy comes into conflict with the medical community's need for large collections of patient medical records for monitoring the outcome of care, evaluating treatments, and conducting follow-up studies [29]. The complications and illnesses that patients may suffer years after the initial procedure was performed provide useful information that can be used in future clinical decision making. Large collections of patient medical records are important tools in epidemiological research, retrospective studies, and observational outcome studies.

The concern for patient privacy has led to a cautious, sometimes distrustful view of medical research. According to a 1996 poll, only 57% of Americans find the use of their patient records in medical research to be either "very" (18%) or "somewhat" (39%) acceptable, with 31% saying it would be "not at all" acceptable [1]. Guaranteeing the privacy of medical records used in research is the only way we can expect to gain the cooperation and consent of patients.

1.2 Legal Guidelines

In the United States, the guidelines for protecting the confidentiality of health care information have been established in the Health Information Portability and Accountability Act (HIPAA) of 1996 [4]. Records are said to be de-identified when the risk is very small that the information can be used alone or in combination with other reasonably available information to identify who the patient is. This risk can be calculated and documented statistically for all the records, or we can use the safe harbor approach and show that every record is free of the 18 types of identifiers listed in the law. Those identifiers are:

1. Names, including that of the patient, visiting relatives, and hospital staff;

2. All geographic subdivisions smaller than a state, including street address, city, county, precinct, zip code, and their equivalent geocodes, except for the initial three digits of a zip code if, according to the current publicly available data from the Bureau of the Census:

   (a) The geographic unit formed by combining all zip codes with the same three initial digits contains more than 20,000 people; and

   (b) The initial three digits of a zip code for all such geographic units containing 20,000 or fewer people is changed to 000.

   This includes all references to which hospital the patient is being treated in;

3. All elements of dates (except year) for dates directly related to an individual, including birth date, admission date, discharge date, date of death; and all ages over 89 and all elements of dates (including year) indicative of such age, except that such ages and elements may be aggregated into a single category of age 90 or older.

   (We removed the year for all patients, though that is not required by the law);

4. Telephone numbers, including pager numbers;

5. Fax numbers;

6. Electronic mail addresses;

7. Social security numbers;

8. Medical record numbers;

9. Health plan beneficiary numbers;

10. Account numbers;

11. Certificate/license numbers;

12. Vehicle identifiers and serial numbers, including license plate numbers;

13. Device identifiers and serial numbers;

14. Web Universal Resource Locators (URLs);

15. Internet Protocol (IP) address numbers;

16. Biometric identifiers, including finger and voice prints;

17. Full face photographic images and any comparable images; and

18. Any other unique identifying number, characteristic, or code (Section 164.514 b of [4])

Such data is known as protected health information (PHI).

HIPAA came into effect April 2003, and as of September 2004, the Department of Health and Human Service's Office of Civil Rights had received and initiated reviews of over 7,577 complaints of HIPAA violations [18]. The law has already come under attack by biomedical researchers, who complain that the constraints on the use of human data and the fear of litigation have caused time delays, increased the administrative cost of studies, introduced bias towards the type of patient who understands and supports medical research, and have blocked important research from being even suggested [23].

Medical researchers must obey the US Common Rule 45CFR46, which states that all research involving human subjects requires informed consent from subjects. A study can be exempt from the rule if it is: "Research involving the collection or study of existing data, documents, records, pathological specimens, or diagnostic specimens, if these sources are publicly available or if the information is recorded by the investigator in such a manner that subjects cannot be identified, directly or through identifiers linked to the subjects" [3]. Research must also be approved by the Institutional Review Board. At MIT, the Committee on the Use of Humans as Experimental Subjects's requirements for using human medical data is the same as

HIPAA's, in that de-identified human data is permissible to be used without patient consent [2].

There are privacy laws in other countries. In Canada, there is the Personal Information Protection and Electronic Documents Act [9]. The European Union has adapted the European Union Directive on Data Privacy [7]. The Organization for Economic Co-operation and Development, which has members from 30 countries and has active relationships with another 70, has established Guidelines on the Protection of Privacy and Transborder Flows of Personal Data [8]. Like HIPAA, all these laws aim to protect sensitive personal data.

1.3 De-Identification and MIMIC II

The Laboratory for Computational Physiology (LCP) is participating in a project to create an advanced intensive care unit (ICU) patient monitoring system that would report all relevant patient data to clinicians and that would automatically generate pathophysiologic hypotheses that best explain the observed data [28]. MIMIC II is a multi-parameter database of patient medical data developed to support that ongoing research project [31]. The database now contains waveforms, trend plots, lab results, and other types of medical data for over three thousand patients from the ICUs of a local hospital, and it is growing to contain records from even more patients. Because of privacy concerns, we are unable to use that data outside our lab without first removing all personally identifiable information from the records. This necessity makes outside collaboration difficult. Eventually we would like to have the entire MIMIC II database in a fully de-identified form so that all the medical data can be made publicly available to the entire research community.

Removing PHI by hand is a time-consuming and expensive task which may be prone to serious error. Our group wants to develop algorithms to perform the de-identification task automatically. As a first step towards that goal, another student working with the LCP developed a preliminary de-identification algorithm discussed in Section 1.4.4 (see [25]). The focus of my project is the creation of methods to test

and verify the performance of de-identification algorithms on free-text nursing notes from the MIMIC II database.

1.4 De-Identification Solutions

Various methods have been developed to remove personally identifiable information from different types of medical text. None of the algorithms developed outside our group is designed to work on data as unstructured as the free-text nursing notes we are focusing on, but the different approaches provide ideas for how we can improve our own de-identification methods.

1.4.1 Latanya Sweeney's Scrub system and Datafly system

One of the leaders in the field of medical record privacy is Dr. Latanya Sweeney, now the director of the Laboratory for International Data Privacy at Carnegie Mellon University. Her Scrub system [32] is designed to remove the PHI from clinical correspondence and clinical notes. Her software looks for PHI using "common-sense" templates and look-up tables. It uses probability tables for template matching, detectors for medical terms to reduce false positives, tools to identify words that sound like other words to account for spelling variations, and detectors for re-appearing terms. Her system found 99-100% of the PHI in her test set, though the test set is not rigorously described and no statistics are given of her false positive rate.

Sweeney's Datafly system [33] adjusts medical databases to render them anonymous enough to be released. The system generates a profile for each user based on the user's access to external information and the probability that he will use the outside sources to re-identify the individual based on information from the fields in the database. The user tells the system the specific fields and records wanted from the database, and the system uses the user's profile to determine what form of the data to return to him in order to guarantee anonymity. The minimum level of anonymity is given and used to calculate value b, such that every value in each field will occur at least b times, except for one-to-one replacement values like Social Security number.

The biggest disadvantage to Sweeney's de-identification tools to us is that they are all closed-source. The Datafly system is licensed to Privacert, Inc. [6]. That company targets organizations that want to share person-specific data without revealing identities and other sensitive information about individuals, companies, or other groups. The "Privacert Appliance" claims to fully de-identify databases containing personal information, presumably using the Datafly system, but no technical details are given.

1.4.2 Natural Language Processing techniques

Some de-identification algorithms use natural language processing (NLP) tools for processing the text. Many NLP tools have been developed for non-medical texts, so they must be adjusted to deal with medical terminology, abbreviations, and the different types of numeric data to be effective on medical records.

Ruch [30]'s technique uses sophisticated NLP techniques to tag words with appropriate parts of speech and a specialized MEDTAG semantic category. After the text has been tagged, it uses contextual rules based on those tags to identify PHI. It matches templates for small groups of up to five words, and it implements some "long-distance" rules as finite state machines. The technique looks for PHI around words marked as IDM (IDentity Markers). The software was developed for post-operative reports, laboratory and test results, and discharge summaries written primarily in French, though some documents were in German and English. The system found 98-99% of all PHI in their test corpus.

Taira [35] has created an algorithm to identify patient name references in clinical correspondence, discharge summaries, clinical notes, and operative/surgical reports for pediatric urology patients. He uses a lexicon with over 64,000 first and last names and a set of semantic selectional restrictions to assign probabilities for a given word being a name. It attempts to classify every sentence according to the type of logical relation it contains, then extracts the potential name based on that logical relation. For example, the sentence "John is a 5 year old male" would be classified as containing "Patient-age" and "Patient-gender" logical relations, with the patient name being "John" for both. This technique to classify names according to their

semantic use had a sensitivity of 99.2%, but it was limited only for patient names and not for any other type of PHI.

1.4.3 Other Methods

Another method to extract names from patient records was developed by Thomas [36]. This method uses a lexicon of 1.8 million proper names to identify potential names and a list of Clinical and Common Usage words from the Unified Medical Language System (UMLS) and Ispell spell-checker dictionary to reduce false positives. If a word is on both lists, there are a few simple context rules to classify the word. This method has been tested on pathology reports and found 98.7% of all names.

Berman [14] developed a technique for removing all PHI from pathology reports by removing all terms that do not appear in the UMLS. His algorithm parses sentences into coded concepts from the UMLS and stop-words, which are high-frequency structural components of sentences like prepositions and common adjectives. All other words, including names and other personally identifiable information, are replaced by blocking symbols, so the output is totally stripped of non-medical and extraneous information. This technique depends on knowing the standard structure of the input text, and the final output may not be readable if the sentence structure deviates from what is expected.

Gupta [20] recently published a de-identification system for pathology reports. It implements a set of rules and dictionaries designed to identify the presence of PHI, uses the UMLS for the identification of medical phrases not to be removed, and replaces identifiable text with de-identified but specific tags. The most interesting part of this study was the measures taken to verify and improve the quality of the de-identification software. The de-identified files were linked to the original file on a secure server, then only the de-identified versions were distributed to four pathologists with training in pathology and informatics. These examiners looked for text that should have been de-identified and was not, and for instances where the program removed clinical text that should not have been removed. The examiners did not have access to the original reports, but the labels on the tag could be checked with

the context of the removed text to look for misclassification of text. The developers used the feedback from the human reviewers to improve the software in three separate evaluations. By the final evaluation, the false positive and false negative rates had been drastically improved to have a final sensitivity of 99.1%. Systematic human reviews continue for quality assurance tests as the software continues being improved.

1.4.4 LCP Method

The other de-identification methods discussed were developed for specialized, highly structured data sets. My project focused on nursing notes, described in more detail in Section 3.1. A simple Perl automated de-identification algorithm for nursing notes was developed for in-house use by another student [25]. First it uses pattern-matching to identify potential dates, telephone numbers, social security numbers, and other protected types of identification numbers. Next it uses look-up tables to identify potential locations and patient, clinician, and hospital names. Finally the algorithm applies several simple context-based rules, such as the word following "Dr" will often be the doctor's last name. See [25] for further details. Preliminary tests showed the algorithm has a high false positive rate, but its overall sensitivity is high. We used this algorithm on our notes, as described in Section 3.3.

Chapter 2

Software Tools

Software was developed to gather the PHI selections of human experts, to combine the PHI selections of multiple de-identifiers looking at the same text in order to form an adjudicated consensus, to evaluate the performance of individual de-identification methods, and to re-identify text by replacing the PHI with authentic-looking surrogate data. Java and Perl programs were written for these tasks.

2.1 Overview of the Deid Program

The Deid program is used for human de-identification of text. It has two main modes: the De-identification Mode, in which a single clinician views and selects the PHI in each note; and the Aggregate Mode, in which the selections of multiple de-identifiers are used to create an adjudicated consensus and to evaluate individual performance compared to that consensus.

Deid is a Java-based graphical user interface. It is run from the command-line in the directory with the source code.[1] The code contains these major classes:

- *Deid.java*: Used in both the De-identification and Aggregate modes. Sets up the graphical display. Controls changes between users, modes, and tasks within

[1] *Usability Note* : The code was designed to run on Java 2 Platform, v1.4.2. To run the software:
1. Move to the directory with the source code.
2. Compile the Java code: "javac Deid.java"
3. Run the code: "java Deid"

each mode. Keeps track of the current state of what is being displayed to the user.

- *NoteManager.java*: Used in both modes. Loads the text from the raw data files. Organizes the text according to which patients the text were written about and when the text was recorded. Groups patients into sets. Returns the appropriate text when given a patient and note number.

- *FileManager.java*: Used in both modes. Reads and writes files with the locations of PHI. Reads files for lists of users and the headers used in organizing the text according to the appropriate patient and note number.

- *NoteText.java*: Used in the De-identification Mode. Returns the part of the text to be displayed by the Deid display. Returns the character index boundaries of that section of the text. Used to avoid scrollbars.

- *HighlightManager.java*: Used in both modes. Stores all the PHI selections in the text made by each user. Returns the locations of the selections made by a given user for a specified patient and note number.

- *CompareWindow.java*: Used in the Aggregate Mode. Controls the JPanel display on the top half of the display when forming and revising a consensus. Displays all the selections. Collects and processes the user input for adjudicating consensus formation. (See Section 2.3.1 for more information.)

- *EvaluationWindow.java*: Used in the Aggregate Mode. Controls the JPanel display on the top half of the display when evaluating the performance of an individual de-identifier. Displays all the selections. Collects and processes the user input for evaluating user performance. (See Section 2.3.2 for more information.)

Deid uses many smaller classes for displaying specialized dialog menus and for handling action events. Those classes are:

- *ChooseUserAndNoteGroupDialog.java*: Used in both modes. Displays the list of recognized user names, and the different groups of notes (#1-5) to be looked at. Returns the selected user name and group of notes.

- *ChooseConsensusDialog.java*: Used in the Aggregate mode. Displays the list of recognized user names and the different groups of notes (#1-5) to be looked at. Returns one or more names and a single set of notes.

- *ChooseModeDialog.java*: Displays the choice of modes, tasks, and exit. Returns the user's choice.

- *ChooseNoteDialog.java*: Used in the De-identification Mode. Displays the current patient and note number being displayed. Allows the user to input a new patient and note number to view. Returns the specified patient and note number.

- *CheckBoxListener.java*: Used in many dialog boxes and windows to track whether a check box is marked.

- *KeyboardAction.java*: Used in the De-identification Mode. Allows the user to select words using keyboard commands.

- *MouseAction.java*: Used in the De-identification Mode. Allows the user to select words using mouse clicks.

- *WindowHandler.java*: Allows user to close the program by closing the window.

Deid also depends on several data files. *Headers.txt* contains the mapping of the note headers to the corresponding patient number and note number[2]. The actual nursing notes are in text files listed in a field in the *NoteManager.java* code (see the comments in the actual code). For this project, we used notes from three text files: *note_enents1.csv*, *noteset2.csv*, and *enriched – all.txt*.

[2]*Usability Note* : The format of the file is,
1102=018-07-26 03:30:00=2018-07-26 03:52:00=Nursing/Other=1102= 1102 1
Where the first field is the complete header, 1102 is the patient number, and 1 is the note number.

2.2 De-Identification Mode

Deid's De-identification mode is used to gather the PHI selections of a single clinician. New users are added manually to the *user.txt* file[3]. The user selects her name and the set of notes she wants to de-identify in the dialog box shown in Figure 2-1. The first screen of text for the first patient in that set of notes is then displayed in the main window of the graphical display, as shown in Figure 2-2.

The user labels words as PHI by clicking once on the word or by selecting part of the word or a series of words. When selecting entire words at a time, the program automatically selects all the text between spaces, including punctuation around the text and any words that may not be properly separated from the desired text. (For example, in the text: "pt was visited by significant other wil,updated on her condition.", clicking on the name "wil" will cause "wil,updated" to be selected.) The PHI that the user selects is highlighted on the screen and is automatically saved in the HighlightManager when the user presses the "Next" button.

Every user has her own file (*username.deid*) listing the locations of PHI she selected in the notes.[4] All the selected PHI locations are written to that file every time the patient number changes.

The display on the top left of the screen tells which patient and note number is being displayed on the screen. The user can go forward or backward in the text using the "Next" and "Previous" buttons at the bottom of the screen. A dialog box will appear when she has completed the last patient that is part of the set. The user can skip to a different part of the set of notes by clicking on the "Note" button on the top of the screen. Other options include changing the color of the highlighted text, changing the user, switching modes, and showing the original text or the enriched

[3] *Usability Note* : To add a new user "Crystal", for example, to the *user.txt* file, add the line "User: Crystal" at the end of the file.

[4] Format of *username.deid*:
Patient 1001 Note 1
764 764 768
895 895 901
The first number is the character index of the beginning of the word with the PHI. The second number is the index of the beginning of the PHI selection. The third number is the index of the last character selected as PHI.

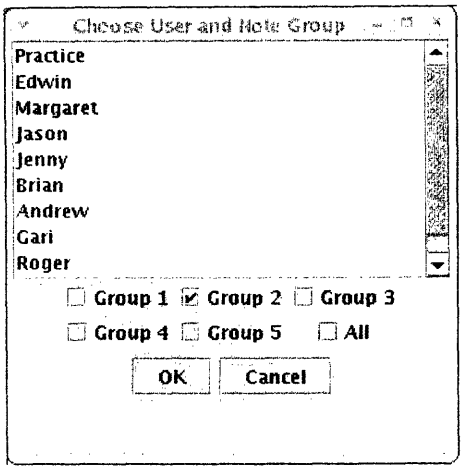

Figure 2-1: The dialog for choosing the user and set of notes in the De-identification Mode.

version (only applicable in the fourth set of notes, explained in Section 3.1).

2.3 Aggregate Mode

The user can switch to the tasks in the Aggregate Mode in the dialog shown in Figure 2-3. That dialog appears when the user presses the "Change Mode" button in the De-identification Mode or when the user completes or cancels a task in the Aggregate Mode.

The Aggregate Mode combines the selections made by the separate human de-identifiers to create and revise a consensus of what should be marked as PHI and to evaluate how well an individual de-identifier performs compared to that consensus.

2.3.1 Create and Revise Consensus

The PHI selections of multiple doctors looking at the same notes are combined in the Aggregate Mode's "Create Consensus" and "Revise Consensus" tasks. In the Java interface as shown in Figure 2-4, the selections of all clinicians for each note are displayed, and a suggestion for the correct text is generated based on the majority response. A clinician referee from our group reviewed all the selected PHI and made the final decision as to whether a word should be classified as PHI.

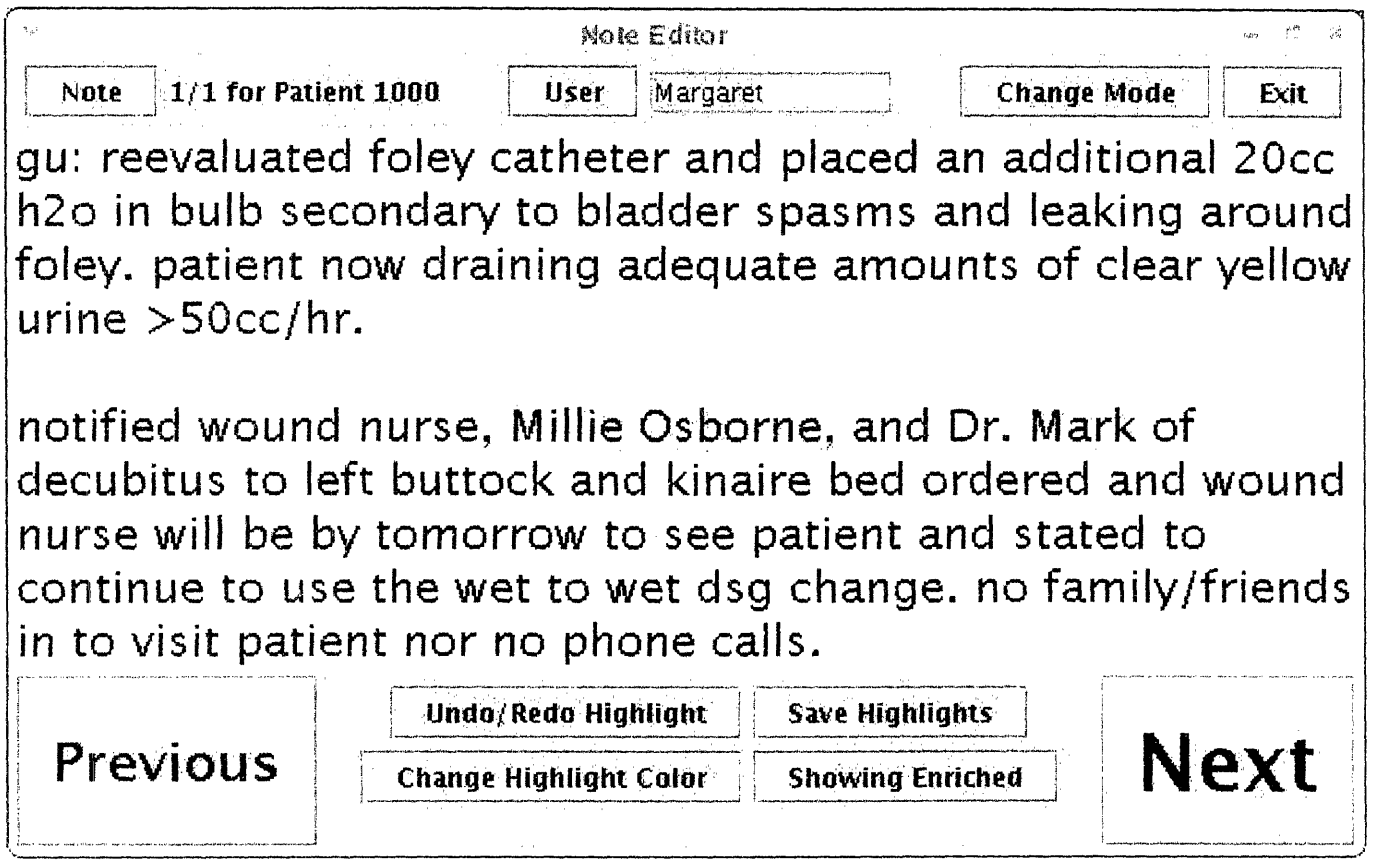

Figure 2-2: The display for the De-identification Mode.

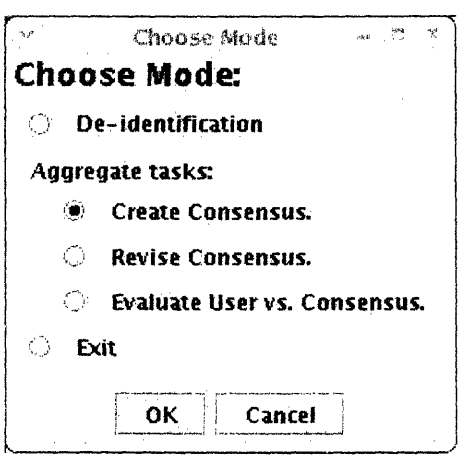

Figure 2-3: The dialog for switching between modes and tasks.

The user selects a set of notes and the individual de-identifiers who reviewed that section in a dialog box very similar to that shown in Figure 2-1 in the De-identification Mode. The selections for all the individuals chosen for the specified group of notes are combined and displayed. The first column is the location of the potential instance of PHI (Patient number, Note number, the character index of the start of the word). Clicking on the button changes the text on the bottom half of the screen to that specific note with the potential PHI instance highlighted, so the adjudicator can see the original context to evaluate whether the instance should be counted as PHI. The next column of check boxes indicates whether that instance should be counted as PHI. When the consensus is being first created, the box is checked only if a majority of the human de-identifiers selected the word as PHI. The "Correct" text column is an edit-able text field of the exact text that should be labeled as PHI. The leading and trailing punctuation is automatically removed. The "Correct" text is by default the text selected by the majority of human de-identifiers. If the majority did not select the text or if different parts of the word were selected by different de-identifiers, that text field is empty and must be manually filled in by the adjudicator. The "Correct" text field's background is colored white if all the de-identifiers agree, gray if a majority agree on a selection, and pink if less than a majority made the selection. The remaining columns display the complete text selected by each human de-identifier at that location. If no selection was made, that field is left blank. If the potential instance of PHI is not true PHI, the confirmation check box can be left unchecked and it will not be saved in the consensus file.

The instances of PHI that have been confirmed with a mark in the check box and have the "Correct" text field filled in will have the locations saved in a file (by default *Consensus.deid*). The consensus file can be altered in the "Revise Consensus" display, with all the confirmation boxes and text field filled out according to what has already been saved, or the user can directly alter the Consensus file in the De-identification Mode with the user "Consensus".

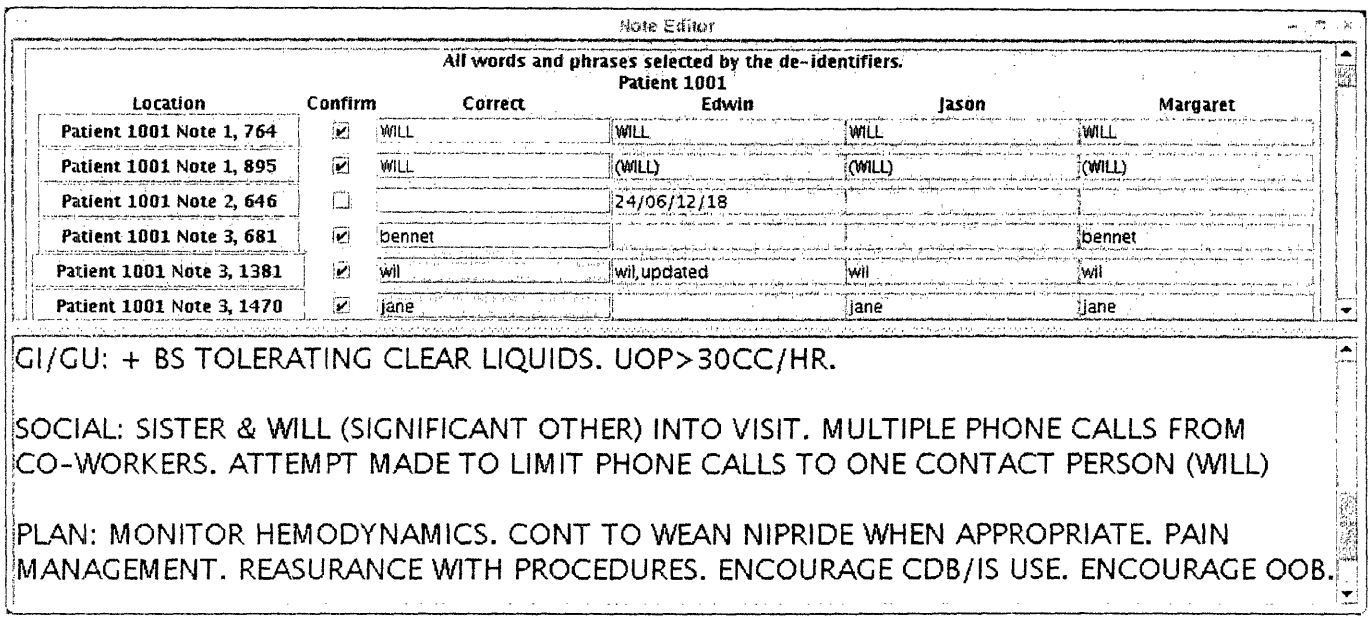

Figure 2-4: The display for creating a consensus in the Aggregate Mode. The top half of the screen has a listing for every instance of identified PHI. At the bottom of the list, not shown in the figure, is an "OK" button that is pressed after the user has finished classifying all the selections.

2.3.2 Evaluate User

The selections of a single de-identifier are compared to the completely de-identified gold standard to calculate the sensitivity and positive predictive value of that de-identifier's performance. We adjudicated the evaluation to decide when to count agreements and disagreements as separate instances. The software initially parses every word as a separate instance of PHI.

The user selects which de-identifier she is evaluating and the set of notes the de-identifier read through. All the selections made by the user and all the selections in the Consensus file for that set of notes are displayed in the interface shown in Figure 2-5. The first column displays the location of the selection. As with the Creating and Revising Consensus tasks, clicking on the button changes the text in the bottom half of the screen to that note text with the selected PHI highlighted. This allows the adjudicator to see the context of where the PHI appears when she is deciding whether multiple selections should be considered as one or multiple instances of PHI. The next three columns of radio buttons are the classifications for the PHI. By

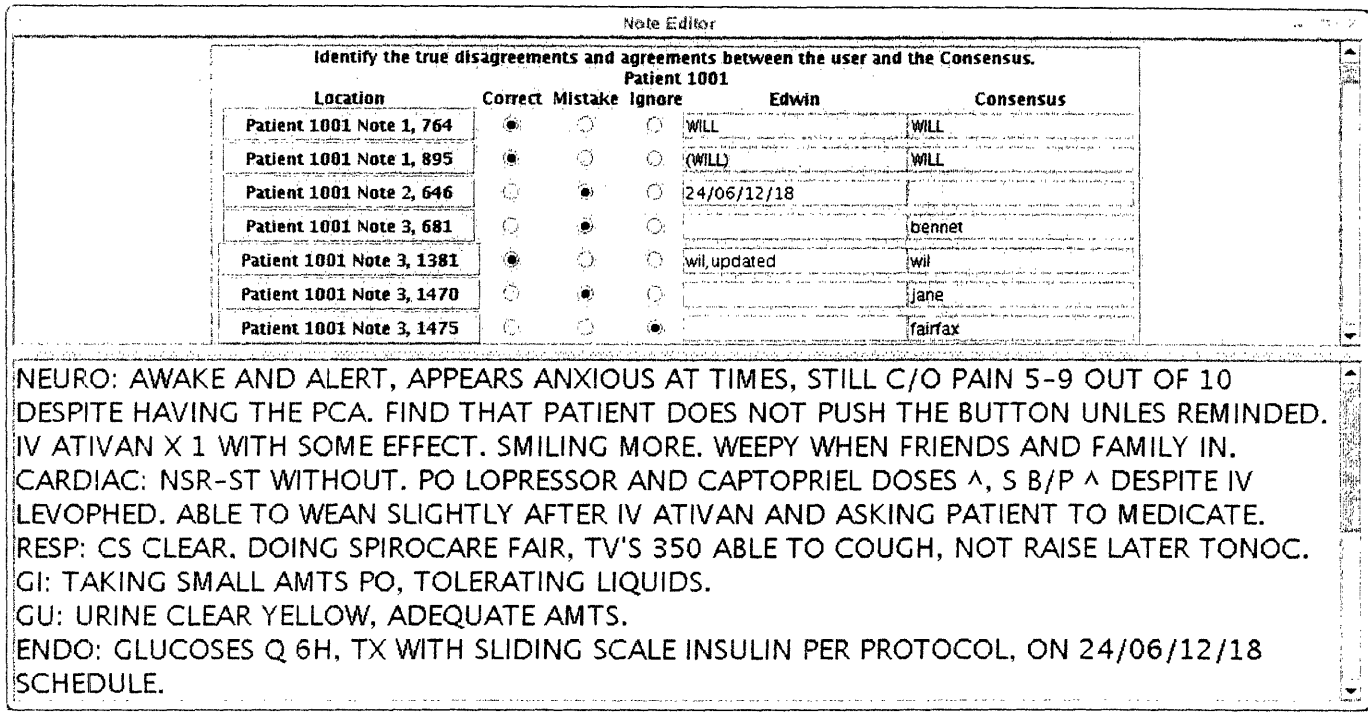

Figure 2-5: The display for evaluating user performance in the Aggregate Mode. The top half of the screen has a listing for every instance of PHI. At the bottom of the list, not shown in the figure, is an "OK" button that is pressed after the user has finished classifying all the selections.

default, perfect agreements (except for leading and trailing punctuation) between the user and selection are counted as "Correct", disagreements (including different parts of the same word being selected) are counted as "Incorrect", and nothing is classified "Ignore". The next column is the text selected by the de-identifier at that location. The text field is colored pink if there is a disagreement with the consensus. The last column displays the text labeled as true PHI by the consensus at that location.

After all the selections have been classified, the user clicks the "OK" button and the number of true and false positives and false negatives are counted and displayed in a dialog box. Those statistics are also written to the file *username.stats*. Those values are then used in calculating the sensitivity and positive predictive values for the de-identifier.

2.4 Re-Identification Software

The tools used for re-identifying the text were separate from the Deid package. The main components of the Re-Identification software are:

- *suggest_reid.pl*: Perl script that uses the list of locations of PHI to make suggestions for text to replace the PHI. Calculates the random offsets for dates to be shifted.
- *ReidDialog.java*: Java graphical user interface that displays the suggestions to the user and allows changes to those suggestions to be made.
- *Reidentifier.java*: Java code that reads the initial suggestions from the file created by *suggest_reid.pl* and saves the user's final selections in a different record file.
- *ReidFileDialog.java*: Java dialog for entering the file names needed for the ReidDialog software.
- *NoteManager.java* and *FileManager.java*: Same as for the Deid software described in Section 2.1.
- *replace_reid.pl*: Perl script that uses the output from the Java program to remove the PHI and replace it with the surrogate data, as well as record the new locations of the surrogate PHI.

2.4.1 Making Suggestions

The Perl script *suggest_reid.pl* takes as arguments the file with the locations of the PHI and the output file that records the suggestions of surrogate text to replace the PHI[5]. The script uses the *Headers.txt* file and all the raw data files (*note_events1.csv*,

[5]*Usability Note*: The code was designed to run on Perl v5.8.1. To run *suggest_reid.pl*, assuming that the file with PHI locations is *Consensus.deid* and the file with the record output is *record.txt*:
1. Move to the directory with the script.
2. Type into the command-line: "perl suggest_reid.pl Consensus.deid record.txt"
3. The output is written on *record.txt*, and you can use that file for running the ReidDialog Java program.

noteset2.csv, *enriched – all.txt*) to locate the notes with PHI and to extract the text for each PHI instance.

The process for classifying each instance is summarized in Figure 2-6. For instances containing numbers, the script uses pattern-matching to look for dates and telephone numbers. If it does not match those patterns, it is given the label "UNKNOWN". For instances comprised of only letters, the script groups instances that appear next to each other as complete phrases and uses look-up tables of common first and last names, locations, local hospital names, and hospital-specific terms. If the instance does not appear in any of the tables, it looks at the word preceding the PHI instance to see whether it is preceded by a title (i.e. "Dr") or a first name or an initial. That allows the code to properly label unusual last names. If a multi-word phrase is not able to be labeled, the individual words in the phrase are classified separately.

Each instance is labeled as a type (ex. "Month/Day", "Last Name") and given a replacement. The dates in each patient's record are shifted by a random number of days calculated using code taken from [25]. The date is shifted by a random, non-zero number of years between -25 and 25. The year offset is converted to days and rounded to the nearest multiple of 7 to preserve the day of the week. Then the date is shifted by a random, non-zero number of weeks between -2 and 2. By shifting all the dates by the same random offset, the patient's age is preserved, as are the time of year and the day of the week.

The replacement names are randomly selected from lists generated based on lists of names of Boston and Chicago residents with the first and last names randomly switched. This way unusual or oddly spelled names can appear in the re-identified notes. Each name can only be used once in the entire collection of notes, even if only a first or last name is used. There are separate lists for female and male names, and when only a last name is needed, both lists are used for choosing a random name. The references to local areas are replaced with references to small towns in the Baltimore, MD area. The original hospital names are replaced with hospitals in Maryland. The direct references to parts of the hospital where the patient is being

treated at are replaced by references to fictitious buildings and wards in a fictitious hospital ("General Hospital", "GH", "Quartermain building", "floor q2"). Other types of PHI are not given suggestions. Capitalization of the suggested data is based on the capitalization of the original text.

Text that appears multiple times in the same patient's notes will always be given the same replacement text. So if a patient's friend Jennifer visits and the algorithm replaces the name "Jennifer" with "Eva", future occurrences of "Jennifer" will also be replaced with "Eva". However, if her name appears as "Jenny" or misspelled as "Jenifer", the algorithm will treat it as a new name and assign it a different replacement.

The output record file lists the start and end indices of the original PHI, the suggested replacement text, the original text, and the category the original text was given.

The *suggest_reid.pl* script shifts the dates in the headers for all the notes, whether the note contained PHI or not. The new dates are recorded in the file *offsets.txt* and are used in the *replace_reid.pl* script.

2.4.2 Human Approval

The ReidDialog Java graphical user interface[6] allows the user to look over and edit the suggestions made by *suggest_reid.pl*. The ReidDialog software requires the user to input the names for the file with the locations of the PHI and suggestions for replacements, which can be the output of *suggest_reid.pl* or the normal *username.deid* file generated by the Deid software with no suggestions made for replacements; and for the output file to record the approved suggestions. The ReidDialog main window, shown in Figure 2-7, displays the location of the PHI, an edit-able text field with the suggested replacement text from *suggest_reid.pl*'s output (or blank if the *suggest_reid.pl* script was not used), the original text of the PHI instance, and the

[6]To run the ReidDialog software:
1. Move to the directory with the source code.
2. Compile the Java code: "javac ReidDialog.java"
3. Run the code: "java ReidDialog"

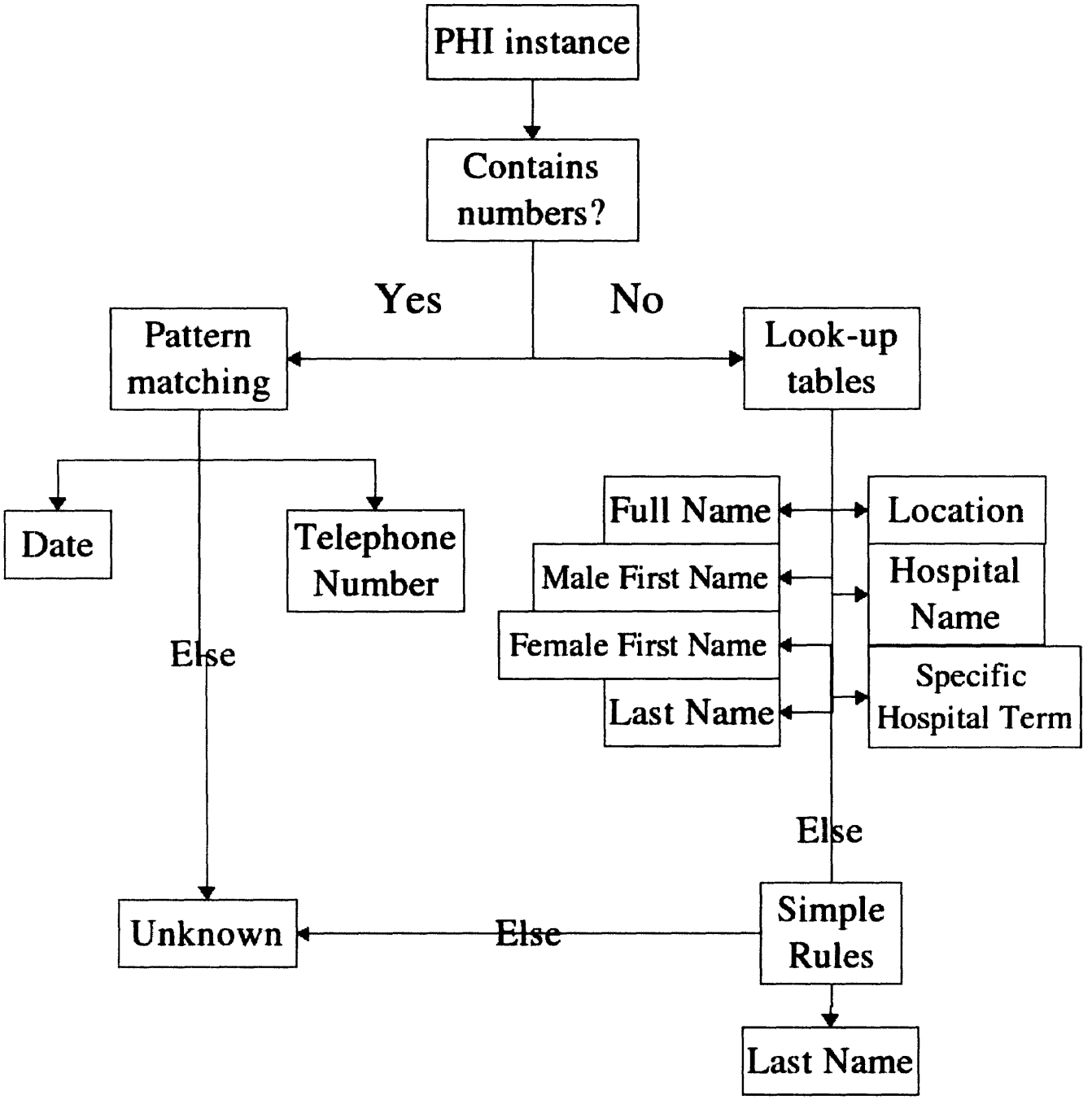

Figure 2-6: Classification of a single instance of PHI in the *suggest_reid.pl* script.

category of the text that was assigned by *suggest_reid.pl*. If the category was "UNKNOWN" from *suggest_reid.pl*, the background of the test field with the suggested surrogate data is colored pink and the text field is left blank. The suggested surrogate data can be directly edited by the user to insert spelling mistakes or to use different terms than those suggested. Clicking on the location of the PHI displays the original note text on the bottom half of the display and highlights the PHI text. Users can then check that the surrogate text is reasonable in context. The approved suggestions will be incorporated in the re-identified version of the text.

The approved replacements are written to a file (by default $temp - replace.txt$[7]) after the user presses the "OK" button.

2.4.3 Replacements

The suggestions made by *suggest_reid.pl* and approved in the ReidDialog software are inserted by *replace_reid.pl*. The script can be run from the command-line[8]. The text to be re-identified is specified in the *patients.txt* file. Each patient and the number of notes for that patient is listed in that file.[9] The original PHI is removed from all the notes specified, the new surrogate data is inserted in the text, and the locations of the new PHI are recorded in the same format as a usual $Consensus.deidfile$. The dates in each note's header are replaced with the shifted dates from $offsets.txt$.

The final re-identified database has the characteristics of the original nursing note

---

[7]Format of $temp - replace.txt$:
Patient 1003 Note 4
27 32 GH
36 40 7/11
The first number is the character index of the beginning of the original PHI text. The second number is the index of the last character selected as PHI. All the text between those indices will be removed and replaced by the text in the third column.

[8]*Usability Note* : To run *replace_reid.pl*, assuming that the file with PHI locations and suggestions is *suggestions.txt*, the file that will have the re-identified text is *reid_notes.txt*, and the file that will have the locations of the surrogate PHI in the re-identified text is *reid_locs.deid*:
1. Move to the directory with the script.
2. Type into the command-line: "perl replace_reid.pl suggestions.txt reid_notes.txt reid_locs.deid"

[9]Format of *patients.txt*:
Patient 1000: 1
Patient 1001: 5
The patient number is followed by the number of separate notes in that patient's record.

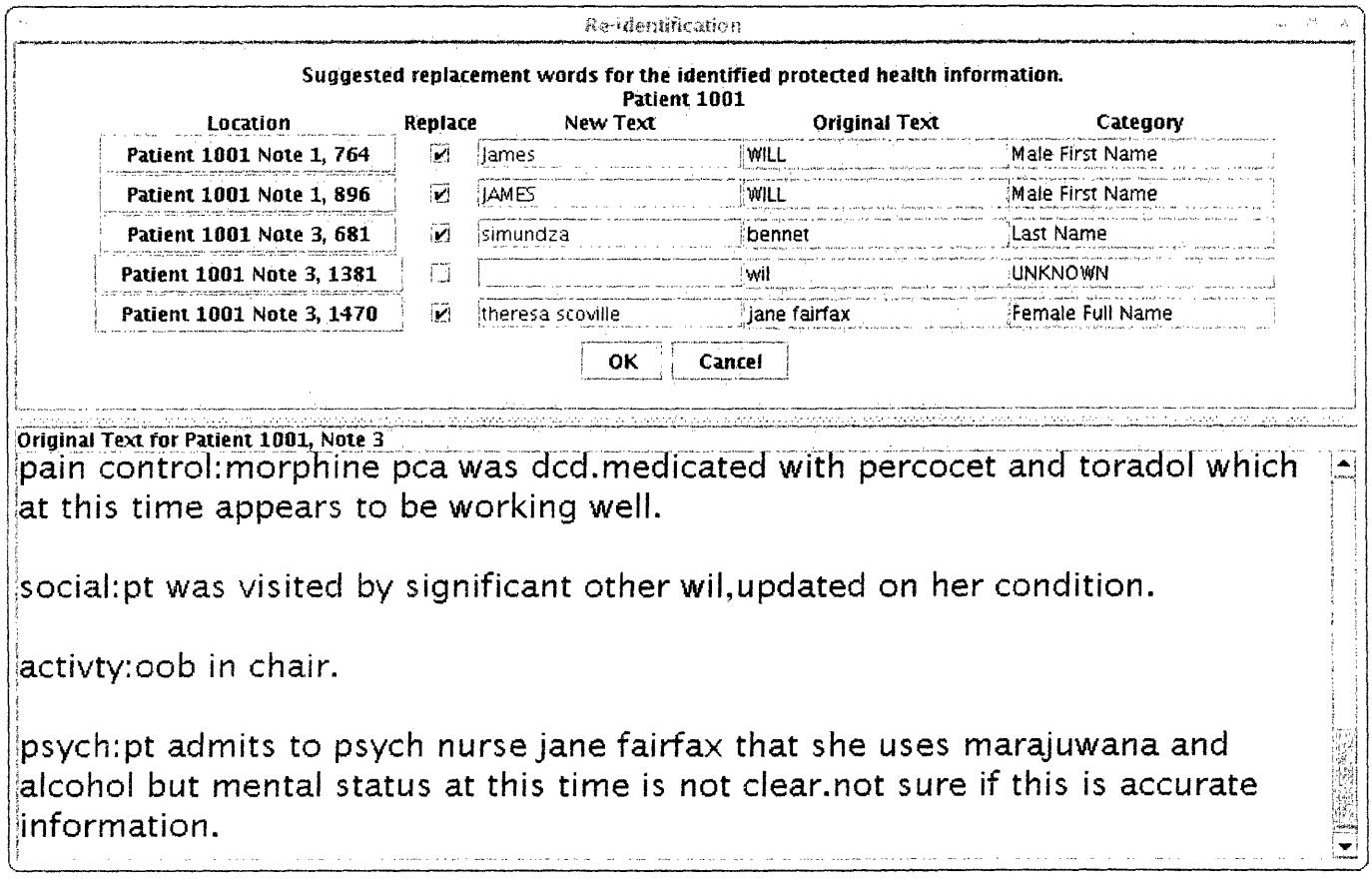

Figure 2-7: The display for reviewing and changing the suggested surrogate data for re-identification.

text, but all the protected health information has been removed and replaced by authentic-looking surrogate data. The new *Consensus.deid* file generated by the script can be used for evaluating de-identification methods running on the re-identified version of the database. An algorithm that performs well on the re-identified database will also perform well on the original data.

Chapter 3

Development of "Gold Standard" and Evaluating Performance

We need a large collection of nursing notes with many instances of different types of PHI for testing the performance of different methods of de-identification. Our "gold standard" reference database would have the locations of all its PHI recorded. To find the PHI we used both human and algorithmic methods of de-identification, as summarized in Figure 3-1. We used that gold standard database to evaluate the performance of different methods of de-identification.

3.1 Corpus

Medical data is collected as part of the MIMIC II project from patients admitted to the intensive care units of a local hospital [31]. The nursing progress notes are unstructured free text typed by the nurses at least twice a day, and include observations about the patient's medical history, his current physical and psychological state, medications being administered, laboratory test results, notes about visitors, and other information about the patient's state. In these notes, the nurses frequently employ technical terminology, non-standard abbreviations, ungrammatical statements, misspellings, and incorrect punctuation and capitalization. Some sample notes are included in Appendix B.

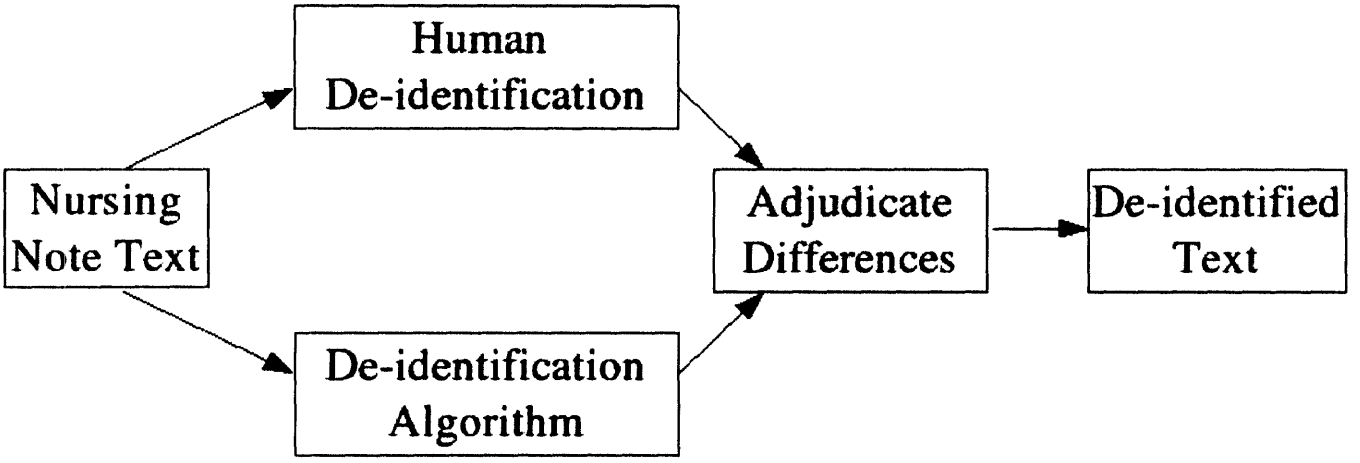

Figure 3-1: Overview of the creation of the "Gold Standard" reference database.

The corpus we used includes notes from 166 randomly selected patients. There are a total of 2,785 notes, with a total word count of 356,103. Of those notes, 119 have been manually "enriched" to include PHI that is especially difficult to identify (such as "foley catheter" and "Parkinson's disease") and to include more instances of infrequently appearing types of PHI.

To determine the approximate corpus size needed, a standard sample size estimate [17] can be used.

$$N = p(1-p)\left(\frac{Z(1-\frac{\alpha}{2})}{E}\right) \tag{3.1}$$

where E is the margin of error, p is the population proportion, and $Z(1-\frac{\alpha}{2})$ reflects the desired level of confidence. Since we wish to distinguish between a 90% and 93% accuracy level, $E = 0.03$ and $Z(1-\frac{\alpha}{2}) = 1.96$ (from tables). A conservative value for p is 0.5, which maximizes the value of N in equation 3.1 (see [17]). Following this formula, at least 1068 instances of PHI are required in our testing database.

3.2 Human De-Identification

Medical house officers from local hospitals were recruited to locate and label the PHI in the nursing note corpus. Every clinician came in for a 3 to 6 hour time block, including breaks. They were paid $50 per hour, with the additional incentive of a

$200 bonus for the best performer in a group of 6 de-identifiers.

Each clinician was given a text definition and examples of what is defined by HIPAA as PHI. They were encouraged to make a best guess for ambiguous cases. A Java application (described in Section 2.1) displayed the nursing note text in an easily readable format and recorded the locations of the PHI identified by each clinician. The software was run on a tablet PC, and clinicians located PHI by tapping the word on the screen with the tablet's pen. The locations of the PHI in every note were written to a file.

The entire process of human de-identification is summarized in Figure 3-2. The nursing notes corpus was separated into four sets approximately equal in size, and each set of notes was de-identified by three clinicians independently. A subset of the data was de-identified by four clinicians, but no advantage was found by adding the fourth person.

For comparison purposes, consensuses without an outsider adjudicator were created for two clinician subsets and for three clinicians subsets using the simple *createUnions.pl* Perl script. The unadjudicated consensuses were created by taking the union of all selections. Most of the errors made during human de-identification are false negatives, so taking the union minimizes the number of missed false negatives.

In the Java interface described in Section 2.3.1, the selections of all clinicians for each note are combined and displayed, and a suggestion for the correct text is generated based on the majority response. A clinician referee from our group reviewed the selected PHI and checked the context of each selection in the original note text in order to make the final decision as to whether a word should be classified as PHI.

3.3 Algorithm

The algorithm described in Section 1.4.4 [25] was used to locate PHI in the entire collection of nursing notes. The output of the algorithm was not in the format of what the Deid code accepted as input, so extra processing had to be done to the algorithm's output so that it would be compatible with the testing software. The

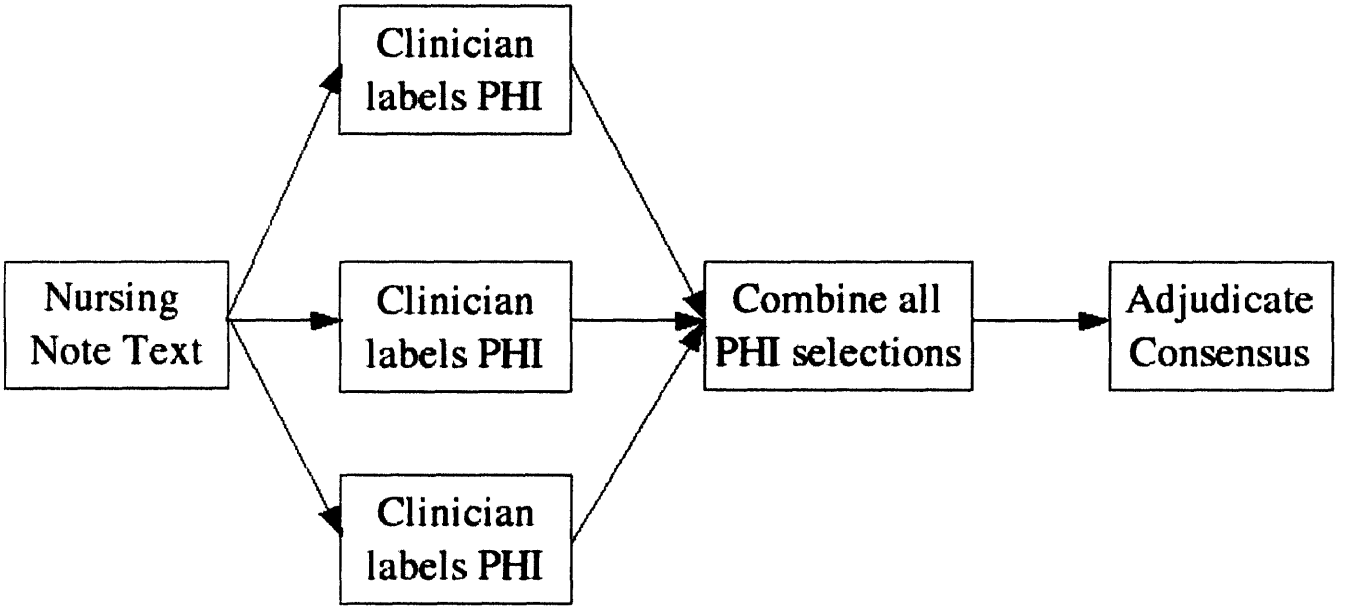

Figure 3-2: The human de-identification process.

get_indices.pl Perl script was able to extract the indices of most of the labeled PHI, then I manually went through the list of errors detected by the script to add those indices to the list of found PHI by the algorithm. Once the algorithm's results were converted into the same format as the selections made by humans using the rest of my software, the Deid software could be used to compare its PHI with that of the humans.

Of the PHI that only the algorithm found, I removed obvious false positives from the list and had a clinician verify all the reasonable-looking new PHI found only by the algorithm and not by the human de-identifiers.

3.4 Evaluation of Performance

The selections made by a given de-identifier are compared with the "gold standard" selections. (The software to do so is described in Section 2.3.2.) An adjudicator must decide when to count agreements and disagreements as separate instances. By default every word is counted as a separate instance, so finding the name "Dr. Everard van

Tijlen" would weigh more heavily than the name "Charity Dawson", and locating the place "New York City" would count more than finding "Newton".

The de-identifier is given credit or penalized only once for each instance of PHI, no matter how many words that instance consists of. If a de-identifier found part of a multi-word PHI phrase, she was not penalized for missing the rest of it. The PHI instance is counted as successfully located. If a de-identifier selects additional words around the actual PHI, like the title "Dr" before "Everard van Tijlen" or "Medical Center" after "Baystate", she is not penalized.

If the de-identifier systematically marks the same text incorrectly as PHI, such as the name of a ward, she is only penalized once even if she continues to mark subsequent appearances of the term. The additional occurrences are ignored. Finally, there were instances when the de-identifier was uncertain and asked about how to classify a term. If we later change our mind about that classification of that ambiguous case, we do not penalize them for marking it as we told them when they asked.

3.5 Concerns

By the time a note is pronounced completely de-identified, four different clinicians and one algorithm have looked at the text.

The evaluation process was very time-consuming and subjective. We were consistent in our enforcement of what to count as separate instances for all de-identification schemes for comparison purposes, but we may have been overly lenient in excusing missed parts of phrases.

Chapter 4

Results for De-Identification

4.1 Human Performance

A total of 11 different clinicians independently scored 20.8% to 43.3% of the corpus. Most could read through about 80,000 words in a 4-5 hour session. Clinicians were encouraged to take breaks whenever they needed, but many chose instead to do the entire task in one sitting. To give them a goal to work towards, they were given lists of patients whose notes we wanted them to get through, but if the list turned out to be too long or if the clinician was a slow reader, we let her stop before she had made it through the list.

Feedback was requested about the software and how it could be improved, though no feedback was gathered about their actual de-identification strategy. From in-house tests, clinicians said they skimmed mostly and looked for names and dates without fully reading the text, though some said that they read the text to make the task more interesting.

Those who read the fastest tended to have the most false negatives. More mistakes were made closer to the end of the session, as the clinicians became more tired and more eager to get through the assigned set of notes. No matter how much we pay and how we may try to make it as comfortable and painless an experience as possible, it is still very difficult to keep a human de-identifier motivated and attentive to the boring de-identification task for a very long period of time.

Table 4.1: De-identification Performance for humans and for an automated algorithm. The "gold standard" is the adjudicated union of the algorithm and three independent human experts. PPV = Positive Predictive Value.

		Min	Max	Mean
1 person	Sensitivity	0.63	0.94	0.81
	PPV	0.95	1.0	0.98
2 people	Sensitivity	0.89	0.98	0.94
	PPV	0.95	0.99	0.97
3 people	Sensitivity	0.98	0.99	0.98
	PPV	0.95	0.99	0.97
Algorithm	Sensitivity	-	-	0.85
	PPV	-	-	0.37

We documented the performance of single clinicians' selections, the union of two clinicians' selections, and the union of the selections of three clinicians reading through the corpus. The individual statistics are given in Appendix A, and a summary is shown in Table 4.1. Individual performance varied greatly, with the sensitivity ranging from 0.63 to 0.94. When combining all the selections made by two people, the sensitivity increased to an average of 0.94 without seriously affecting the positive predictive value. The union of three had an even higher sensitivity. The number of false negatives (FN) for an individual is high and the number of false positives (FP) is low. Having more people look at the notes reduces the number of combined FNs while adding only a small number of FPs.

The most common type of mistake for people is missing dates. Text written in all capital letters is also more difficult for selecting PHI. Clinicians remarked that it took a longer time to read notes in all capital letters, whereas reading notes in all lower-case were not much worse than reading notes with proper capitalization. Humans could find most names and places, even if the spelling was incorrect. The few false positives were mostly ambiguous measurements mistaken as dates.

The software simplified the process of collecting all the locations of PHI in the text and combining the selections of multiple clinicians. The software was run on a tablet PC, so in the beginning clinicians had to adapt to using the tablet pen/screen

interface. The pen tends to be overly sensitive to taps, so users have to make certain that the pen has not clicked twice when only one click is wanted. This is especially important when clicking the buttons to change between notes. The first version of the software displayed one note at a time, and long notes would have scrollbars on the side of the screen. Users complained that they sometimes did not notice the scrollbars and would miss reading the end of notes. The scrollbars also slowed down the reading process. The software was changed for the six sessions with the third and fourth groups so that only one screen of text was displayed at a time, thereby removing the need for scrollbars.

The 11 clinicians spent a total of 53 hours de-identifying our gold standard database, and two were awarded $200 bonuses for their good performance. The adjudication took about 3 hours and was done with clinicians in our group, though we could have hired more clinicians to perform the task for an additional $50/hr. The total cost of human de-identification was thus $3200 and took 56 hours, not counting all the time I spent recruiting and arranging time for the clinicians to take part in our study and the administrative burden of dealing with the paperwork to see that everyone was paid.

4.2 Algorithm Performance

The algorithm had a sensitivity of 0.85, which is better than the average human although less than the union of two humans, but it had a very low positive predictive value of 0.37 since it identifies many FPs. The algorithm does detect most PHI, and it even detected PHI not found by any of the human de-identifiers.

The most common errors made by the algorithm were the misclassification of numbers as dates or identification numbers. Any group of numbers that were in the format "##/##" or "# – #" with values that could be valid months and days were classified as dates. This lead to the algorithm, for example, always tagging as PHI the CPAP (Continuous Positive Airway Pressure) setting, which often appears as "##/5". Those misclassified dates could be reduced by simple context rules.

Another common error was the misclassification of valid medical terms as names. The list of common first and last names used by the algorithm contained some unusual entries, such as "Cardiology", "Deal", and "Vent", that have a very low probability of appearing in the nursing notes as a name. There are also some common names that are also used in medical terms (ex. "Foley"), and names that are also common words (ex. "Black"). Context rules should be used to better classify those words.

The algorithm missed the first initial when a name was given as an initial followed by the last name (ex. "M. Amis"). Those errors can be reduced by adding a simple rule looking to see if the word before a found last name is a single letter. The algorithm lacked rules to find the date if the month was written out (ex. "Sept 26"). The reliance on look-up tables meant that unusual or misspelled names were always missed.

Because of the simple nature of the algorithm, we can be assured that every occurrence of a number that could look like a date was found, and every time a title like "dr." appeared in the text, the algorithm will have found it. Of course that simplicity lead to the high false positive rate, which made going through all the PHI selected by the algorithm a very tedious, time consuming task. The reliance on a human to filter through all the unreasonable selections allows for greater potential for human error.

4.3 Re-Identified Nursing Note Collection

All the instances of PHI in the "gold standard" nursing note collection were removed and replaced with fake but authentic-looking surrogate data. As described in Section 2.4, replacement text for the PHI were suggested by the *suggest_reid.pl* Perl script and approved by a clinician and me using the *Reid_Dialog.java* Java software. Based on the known limitations of the *suggest_reid.pl* script, we knew to check that:

1. The dates were correctly shifted. Sometimes a date that was in the "Month/Year" format was given a replacement that was in the "Month/Day" format. We would have to calculate the correctly shifted "Month/Year" string to replace the date

in the note text. Years appearing by themselves also had to be shifted by the human referee. The proper offset could be calculated by comparing the time stamps in the old and new headers of the notes.

2. The hospital names were replaced by valid local hospitals. The protected portion of a hospital name – such as *Johns Hopkins* Hospital or *Cleveland* Clinic – were often classified as names and locations. The replacement text was changed to be a Maryland hospital name. Logical abbreviations, such as "MD" instead of writing out "Maryland" in a hospital name, were occasionally used.

3. The names were replaced reasonably and consistently. First we checked that the names were different from the original names and that the first and last name combinations were realistic. (For example, we deemed the name "Giuseppe LeBlanc" to be unreasonable looking.) Sometimes a location or a first name would be incorrectly labeled as a first or last name, so more appropriate replacement text would be provided. Every time a name appears in a patient's set of notes, it should be replaced by the same name. We checked that even when misspelled, the same names were used, often with misspellings similar to those that occurred in the original PHI.

Every item of PHI and its replacement text were reviewed, with extra attention given to those known concerns. During the course of the re-identification process, we found misclassified PHI, including both false negatives and false positives. The re-identification process became in practice another adjudication process, though the changes to the "gold standard"'s list of PHI had to be manually made because of the different software used in the re-identification process compared to that used in the adjudication process.

4.4 Discussion

The results show the limitations of human de-identification of medical data. The combined efforts of four clinicians were needed to completely de-identify the test

corpus of the nursing notes to a level of 98% (100% included adjudicated algorithm results combined with the human results). The simple algorithm therefore found another 2%. Tools have been developed to facilitate the process of using a team of humans to perform the task, but human de-identification is still a very time- and manpower-intensive process. There is a clear need for accurate, fully automated de-identification algorithms.

The simple preliminary algorithm evaluated here is an early draft and is far from perfect, but it already has a higher sensitivity than the average human de-identifier. Its high FP rate limits its practical usefulness at present. Important data is being tagged and removed as PHI. The algorithm was successful in following simple, common-sense-based rules to identify clear, obvious instances of PHI, like a doctor's name when the last name is preceded by "Dr.". The algorithm failed when misspellings, incorrect punctuation, or unusual spacing made the target text no longer fit the expected template in the simple rules or the entries in the look-up table. If spelling mistakes can be identified and corrected automatically, the simple rules can be implemented more effectively. In order to reduce false positives, we must rely less on look-up tables and pattern-matching and instead base our approach more on context-based rules.

The most difficult type of PHI for both people and for the algorithm to correctly identify was dates. There is a huge variation in how the dates are written, whether the numbers are divided with "/", "-", or "."'s, whether the months are spelled out, how the spelled-out months are abbreviated, and so on. Our notes have headers that say when the notes were written, so we should be able to use that knowledge in distinguishing what is most likely a date. Even without the header, we should be able to see that most of the dates are around a certain time and a date for a completely different month or year would be less likely to be genuine. Improvements to the algorithm are considered in the discussion of the strategy of the improved de-identification algorithm in Section 5.1.

The re-identified reference database will be publicly available on Physionet [19, 5] for the use of the research community. The corpus contains nursing notes from 166

patients, a total of 2,785 notes, a total word count of 356,103, and the corpus includes 1,802 instances of PHI. All the source code for the software used in this project will also be placed on Physionet.

4.5 Future Work for Reference Databases

It would be beneficial to have a larger, more diverse "gold standard" reference database of nursing notes for testing purposes. All the notes in the current gold standard database were taken from the same hospital, and many of the notes were written by the same nurses. It would be helpful to get nursing notes from other institutions to make certain that the notes' style is not too specific to that hospital's guidelines and practices. Notes from additional patients could include descriptions of medical problems and tests not included in our current collection.

Though we went to great efforts to fully de-identify our corpus, it is possible that some PHI have been missed. We should gather feedback from the users of our reference database about any PHI they have been able to find. It could be interesting to conduct a more detailed evaluation of how well the database truly is de-identified. Even if we perfectly remove all the explicit identifiers required by HIPAA, there remain other types of personal information that could be used to identify the patient. For example, the statement "The patient is the daughter of the governor of Montana" does not use any words that would be removed as PHI, though the information can be used to identify who she is. Depending on the results of such an evaluation, we may want to alter our requirements for de-identification and remove additional types of information that could be used to identify a patient.

The algorithms we are still developing will be tested on our "gold standard" reference database. We hope to have an algorithm soon that reliably performs better than humans that we can use on the MIMIC II database.

Chapter 5

Next Step: New De-Identification Algorithm

The next task in this project is the development of an improved automated de-identification algorithm. Using the lessons learned from the evaluation of the preliminary algorithm, I developed a new de-identification algorithm written in Perl.

5.1 Strategy of De-Identification Algorithm

The new de-identification method finds instances of PHI in text based on pattern matching, look-up lists, and common sense heuristics. There is not enough training data to be able to implement statistical natural language processing techniques like hidden Markov Models, and the nursing note text is too unstructured and ungrammatical to be able to rely on existing part-of-speech tagging techniques or any of the other common natural language processing approaches.

The algorithm assumes that the text is a medical record, and there are many specific rules that are based on what I have seen in my nursing note corpus, but the algorithm does not depend on the inputted text being nursing notes in the format found in the MIMIC II database. The algorithm also does not depend on the availability of any other information about the patients or the hospital staff that can be found in our database.

5.1.1 Finding Names

The most important type of data we need to remove with 100% accuracy is the patient's name. A single mention of the patient's name in publicly released data would be an unacceptable violation of privacy. We could get the patient's name from tables in the MIMIC II database, but in the nursing note text the name could be spelled incorrectly ("Willaim") or the patient may use a nickname ("Bill"), so the algorithm cannot rely on being provided with the name information. We also want to identify and remove the names of other specific people mentioned in the notes, including visiting relatives and the attending clinicians.

A relatively small number of first names are used in America. According to the 1990 Census [15], 59.5% of men have a first name that is found on the list of 100 most common male first names, and 43.1% of women have a first name that is found on the list of 100 most common female first names. The list of 1,219 male first names and 4,275 female first names covers 90% of the population. In contrast to that, only 18.8% of people have a last name that is found on the list of 100 most common last names in America, and nearly 90,000 last names are listed to cover 90% of the population. Based on the way names are distributed in the United States, it is reasonable to use look-up tables for common first names but not for last names. Since the top 100 male and female names cover such a large portion of the population, the algorithm also looks for misspellings of those names using the approximate matching capabilities of Perl.

The algorithm does not rely solely on look-up tables for identifying names; several heuristics have been implemented. In the nursing note texts, the last names are always preceded by a first name, the individual's initials, or a title. First names are usually found before a last name or close to a word like "wife", "friend", or "nurse", that identifies who the person is. First names can occasionally be found alone, without a last name or any sort of explanation of who the name refers to, especially if the person has been mentioned elsewhere by that name. For example, the first time that the patient's brother Philip visits, the nurse may write out "Philip (brother) visited."

For later visits the nurse may recognize the brother and only write "Phil visited".

Heuristics are used to look for last names after titles, though not all words that look like titles may be functioning as titles. For example, "MS" could be a title, or it could stand for milliseconds or multiple sclerosis. Single letters followed by a "." are not always initials, and not all initials are followed by ".". The heuristics are used to identify potential first and last names, then the words are compared to lists of common English words and medical terms. The biggest problems come from names like "Will", "Ray", "Eve", "May", and "Mae". They are words that often come up in the notes, and they are common names. The algorithm is designed to tag even ambiguous PHI as PHI to be removed, because it's better to remove too much than to skip over a name or other information that could identify the patient.

5.1.2 Finding Locations

The names of locations smaller than a state are found often in names of hospitals, where the patient comes from, and where the patient's visitors are from. The algorithm uses a list of local hospital names to locate occurrences of hospital names, so the locations found as part of hospital names would also be tagged that way. Since most patients will be coming from the area around the hospital, the algorithm uses lists of towns and cities in the area to locate the names of local places. The patients' visitors can come from anywhere around the world, so the algorithm uses lists of major cities in the US and the world, and it uses simple heuristics to try to pick out cities that are not on the lists or that are misspelled. The algorithm looks for phrases like "comes from", "visiting from", and "returns to".

5.1.3 Finding Numeric PHI

Finding numeric PHI requires using regular expressions that are flexible enough to accommodate all the reasonable variations in how the data may be expressed while not being so flexible that numeric data that is not PHI is removed. Finding dates and being able to identify when the date is in the form "Month/Day" or "Month/Year"

is important because dates are automatically shifted and the replacement text would only make sense if the original PHI had been correctly identified.

Telephone numbers are found by looking for the pattern "###-####" or "###-###-####". The punctuation could vary, spaces could be inserted between the groups of numbers, and the area code could be in parentheses. The regular expressions have to be able to find telephone numbers in all the reasonable variations. Simple heuristics look for indicators that the numbers are telephone numbers, like the word "Phone" or "Telephone", or the name of the person whose number it is. Pager numbers are usually written just as a string of random numbers. The algorithm looks for an indicator, like the word "Pager" or "Pg", that show that the following number is a pager number. Social security numbers and other types of identification numbers are also located by looking for words around strings of numbers that could indicate what the numbers are.

Dates are written in many different ways in the nursing notes. Sometimes the date is given as "Month/Day/Year", or else it is just "Month/Day". Sometimes the month's name is written out. Sometimes the day is written as "the 1st". The algorithm looks for patterns of numbers that look like dates, and it specifically looks for the months to look for the days and years around the month.

A year by itself often appears in the patient medical history (ex. "cholecystectomy, 1953"). We tried many different methods of locating isolated years, but none worked well. In the end, we decided to allow the years to remain. HIPAA does not require the removal of years unless they are indicative of an age above 89 [4]. Our nursing notes never mention date of birth, so we can safely leave in all years. The major disadvantage in not being able to locate and remove the years is that we will be unable to automatically shift those years to correspond to the time shift in the other dates in the notes. For example, the year of the note may be shifted back to be 1985, but there could be references in the note text to a myocardial infarction in 2000. Readers would not know how far in the past the patient had the MI.

5.1.4 Checking for Repeated Occurrences of PHI

The same names often reappear in the notes for a single patient. The patient's son may visit often, or the same clinicians may see the patient during her stay. The algorithm looks for repeated PHI instances within the collection of notes for a single patient.

First all the non-numeric PHI instances – the names, locations, and hospital names – are collected from all the notes for the patient. Then the algorithm compares the list of unique PHI instances with a list of common English words (from the Spell Checking Oriented Word Lists at size 10 [13]). PHI instances that are on the list of common words are removed from the list. The resulting list is used by the algorithm to identify other occurrences of already found PHI in the patients' notes.

5.1.5 Data Sources

The algorithm uses many look-up tables that are based on lists found in Table 5.1 for names and Table 5.2 for the other types of non-numeric data. The look-up tables came from many different online sources. The name lists came from the U.S. Census name lists [15], the location lists come from the U.S. Census's lists of urbanized areas and clusters [16] and from lists of the 100 most populous cities and the capitals of all the countries in the world [10], and the last name prefixes come from a list online [11].

Because of the separation of the contents of the look-up tables from the algorithm itself, changing and supplementing the contents of look-up tables is easy. If the notes are from a new local area, for example, the contents of the files with the names of local places can be changed.

5.2 Performance

The algorithm is still being developed, but some preliminary tests of its performance have been conducted. First I looked at its performance on the easier task of removing

Table 5.1: File names, number of entries, and description of the data files needed by the de-identification algorithm. All the files are available in a large archive file, and they should be put in their own directory when running the algorithm. UMLS = Unified Medical Language System [26]. List of common English words come from Spell Checking Oriented Word Lists at size 35 [13].

File Name	Count	Description
female_names_unambig.txt	3631	Common female first names that are not also common English words and that are not medical terminology listed in the UMLS
female_names_ambig.tx	644	Common female first names that are common English words or that are medical terminology listed in the UMLS
female_names_popular.txt	126	100 most popular names along with those names' common nicknames and spelling variations (Manually removed: Eve, Mae, May)
male_names_unambig.txt	800	Common male first names that are not also common English words and that are not medical terminology listed in the UMLS
male_names_ambig.txt	419	Common male names that are common English words or that are medical terminology listed in the UMLS
male_names_popular.txt	134	100 most popular names along with those names' common nicknames and spelling variations (Manually removed: Will, Ed, Ray)
last_names_unambig.txt	81,495	Common last names that are not also common English words and that are not medical terminology listed in the UMLS
last_names_ambig.txt	7,289	Common last names that are common English words or that are medical terminology listed in the UMLS
last_names_popular.txt	93	100 most common popular names that are not also common English words

Table 5.2: File names, number of entries, and description of the data files needed by the de-identification algorithm. All the files are available in a large archive file, and they should be put in their own directory when running the algorithm.

File Name	Count	Description
local_places_unambig.txt	337	All the towns and cities in the area around the hospital (Massachusetts for original text, Maryland for the re-identified text) that are not also common English words and that are not medical terminology listed in the UMLS
local_places_ambig.txt	14	All the towns and cities in the area around the hospital that are common English words or are medical terminology listed in the UMLS
locations_unambig.txt	3128	Cities from around the US and the largest cities around the world that are not common English words and that are not medical terminology listed in the UMLS
locations_ambig.txt	127	Cities around the US and world that are also common English words or are medical terminology listed in the UMLS
last_name_prefixes.txt	148	Prefixes (ex. O', von, Al-) that may appear before a last name

Table 5.3: Results for initial tests for the algorithm on structured, non-medical data. TP = True Positive, FP = False Positive, FN = False Negative. PPV = Positive Predictive Value

Text Source	TP	FP	FN	Sensitivity	PPV
Austen's *Persuasion*	85	6	2	0.98	0.93
Joyce's "The Dead"	59	4	2	0.97	0.94
Dosteovsky's *Brothers Karamazov*	13	6	19	0.41	0.68
Quinn's *Minx*	44	4	0	1.00	0.92
Overall	201	20	23	0.90	0.95

the PHI in structured, non-medical text. Then the algorithm was tested on nursing notes from our gold standard reference database.

5.2.1 Performance on Non-Medical Texts

Several fiction excerpts were used: the first chapter of Jane Austen's *Persuasion* for perfect, formal, grammatically correct English containing many instances of PHI; the first few pages of "The Dead" from James Joyce's *Dubliners* for less formal, grammatically correct prose; the first chapter of Fyodor Dosteovsky's *The Brothers Karamazov* (translated into English) for structured English with non-standard names; and the opening of Julia Quinn's popular romance novel *Minx* for informal, colloquial English. The texts were chosen because of their very different styles of writing. The total length of the excerpts was 5,955 words.

The performance statistics of the algorithm on each text and on the entire collection are given in Table 5.3. The algorithm performed very poorly on the Dosteovsky text. None of the long Russian names in the text appeared in the look-up tables the algorithm uses for identifying names, no titles like "Mr." were used, and the name indicators did not appear immediately next to the name. The poor performance on that corpus shows how strongly the algorithm depends on titles and look-up tables to find names. The traditional English names in the other texts were found with very high specificity. There were few false positives in the structured text. A few false positives did appear when contractions were used.

Table 5.4: Results for initial tests for the algorithm on a collection of 747 nursing notes, containing 99,443 words. TP = True Positive, FP = False Positive, FN = False Negative. PPV = Positive Predictive Value.

Type of PHI	TP	FP	FN	Sensitivity	PPV
Names	139	178	3	0.98	0.44
Dates	160	132	6	0.96	0.55
Overall	378	490	33	0.92	0.44

5.2.2 Performance on Nursing Notes

The algorithm was tested on nursing notes from the gold standard reference data base. We used 747 notes taken from 22 patients, containing 99,443 words. The results from the test are shown in Table 5.4. The high false positive rate comes from the overly general rules for identifying dates and names. The names of drugs and common abbreviations for medical terminology often are tagged as names. The reference lists of common words and medical terms should be expanded to contain drug names and manufacturers, as well as include more abbreviations. In addition to medical abbreviations, like "gtt" for "drops" and "bid" for "twice a day", the reference lists should also contain abbreviations for common words like "cont" for "continue" and "prev" for "previous".

A major source of false positives in the algorithm came from the part of the code that looked for repeated occurrences of already found PHI. Because of the high false positive rate when identifying potential names, many common words are currently being tagged as PHI, and then the code looks for every other occurrence of the word in the other notes for that patient. So if "cont" is tagged once as a name, dozens more instances of "cont" would be removed from the other notes. The algorithm currently checks to see whether the found PHI is a commonly used word, but the reference lists of common words are based on which words are the most commonly used in normal, correctly spelled, grammatically correct English texts. None of the common nursing terminology or abbreviations are found in those lists. Errors related to repeated occurrences of incorrectly identified PHI account for 148 of the false positives. This

part of the code is not being taken out because it does find missed names, locations, and other correctly identified PHI.

The false negatives were most frequently names of other local hospitals that were either not included in the list of hospitals used by the algorithm or the names were abbreviated in ways not included in the list. The few missed dates were mostly in the form MM/DD/YYYY, because I had forgotten to include a rule in the algorithm for dates written in that pattern.

As mentioned earlier, the most important type of PHI to identify is names. The algorithm's ability to locate names is very good: only 3 were missed in all the notes used. Two of those names were of hospital employees, and we have access to lists of all the hospital employee names in MIMIC II. The algorithm does not use that extra information now, but it can easily be altered to use those name lists in finding PHI. The other undetected name was a misspelling of "Patrick", which should be findable if approximate matching had been used.

The tests we have performed have suggested simple rules that we should add to the algorithm, like look for dates in MM/DD/YYYY format, and the tests have exposed the shortcomings in our look-up tables, such as the lack common abbreviations and drug names. Even in its current imperfect form, the algorithm's performance is better than the average single person and is nearly as good as two people de-identifying the text. (The performance statistics cannot be compared exactly because the algorithm does not look for single years found alone. The human de-identifiers were looking for more types of PHI than the algorithm currently identifies.)

5.3 Future Work for De-Identification Algorithm

The LCP's old de-identification algorithm and this new one have very high false positive rates. We can either create more finely tuned rules to reduce the false positive rate, or we can find a way to involve humans to the de-identification process. Dates in particular can appear in so many different contexts that encoding all the valid possibilities is very difficult, but clinicians reading the text can usually easily distinguish

what is a date and what is a measurement. A graphical user interface could display the results of the de-identification algorithm and allow a clinician to approve, alter, or delete the locations of PHI in the text. The process will be less time consuming and be more reliable than human manual de-identification since the algorithms have such high sensitivity. The final goal is to have a completely automated algorithm, but if that is not yet possible, we can have a method that combines the algorithm and human input in order to allow us to reliably de-identify lots of patient data quickly.

The performance of the new algorithm on nursing notes from MIMIC II can be improved by using the lists of patient names and hospital employee names we have access to in the database. We can also use the timestamps in the headers to help identify what is a reasonable date.

The algorithm's list of common words should be increased to include drug names and common abbreviations. The search for repeated occurrences of PHI resulted in many false positives because medical terms or abbreviations that often appear in notes were not recognized as frequently used words. Instead of deciding what is a commonly occurring word based on general English texts, we could look at the nursing note texts we have and see which are the most frequently appearing words in our corpus.

5.4 Conclusions

There are no generally accepted standards for evaluating the performance of automated de-identification methods. HIPAA says in Section 164.528: "the Department believes that it is impracticable to account for incidental disclosures, which by their very nature, may be uncertain or unknown to the covered entity at the time they occur. Incidental disclosures are permitted as long as reasonable safeguards and minimum necessary standards have been observed for the underlying communication" [4]. The law does not explicitly define what constitutes "reasonable safeguards" and what those "minimum necessary standards" are, so the developer must interpret what is meant by the law and demonstrate that her tools meet the vague legal requirements.

My work is a step towards defining more concrete standards for evaluating de-identification performance. Using the tools I have developed, the accuracy and specificity of de-identification methods can now be calculated based on their performance on the "gold standard" reference database of nursing notes. Those statistics can be compared to those of single humans and teams of two and three humans.

The long-term goal is to create automated de-identification methods that find every instance of PHI in free-text nursing notes. As a first step towards that goal, the LCP has developed a preliminary algorithm that has been demonstrated to be more sensitive than an average human clinician. The new, improved de-identification algorithm is still a work in progress, but preliminary test results are promising. We hope to create future de-identification methods with a sensitivity exceeding that of teams of two and three humans and a false positive rate comparable to that of human de-identifiers.

In this project I have created tools to be used for the evaluation of different methods of de-identification of ICU nursing notes from the LCP's MIMIC II database. The software developed for recording and combining the selections from manual de-identification of text allows a team of clinicians to collaborate to completely de-identify medical text. The "gold standard" reference database of re-identified nursing notes along with the locations of the known PHI in the corpus can be used for testing and evaluating automated de-identification algorithms.

Automated de-identification algorithms will almost certainly become critical tools for researchers preparing to share text-based medical records with the research community. Just as important will be the methods to demonstrate the efficacy of those de-identification algorithms.

Appendix A

Data from Human De-Identification

A.1 Single Clinicians

Table A.1: Performance statistics for a single clinician de-identifying the text. The results for clinicians 4 and 9 are actually from two separate sessions for the same clinician.

Clinician	Sensitivity	PPV
1	0.765	0.986
2	0.936	0.993
3	0.704	0.960
4	0.852	0.974
5	0.828	0.992
6	0.634	0.950
7	0.851	1.000
8	0.847	0.993
9	0.864	0.99
10	0.939	0.984
11	0.810	0.975
12	0.683	0.978
Average	0.809	0.981
Algorithm	0.845	0.369

A.2 Teams of 2 Clinicians

Table A.2: Performance statistics for two clinicians de-identifying the text.

Clinicians	Sensitivity	PPV
1, 2	0.973	0.983
1, 3	0.889	0.954
2, 3	0.961	0.965
4, 5	0.964	0.977
4, 6	0.918	0.956
5, 6	0.898	0.958
7, 8	0.952	0.994
7, 9	0.958	0.994
8, 9	0.943	0.990
10, 11	0.982	0.969
10, 12	0.959	0.971
11, 12	0.896	0.961
Average	0.941	0.973

A.3 Teams of 3 Clinicians

Table A.3: Performance statistics for three clinicians de-identifying the text.

Clinicians	Sensitivity	PPV
1, 2, 3	0.977	0.959
4, 5, 6	0.977	0.952
7, 8, 9	0.976	0.991
10, 11, 12	0.987	0.960
Average	0.979	0.965

Appendix B

Sample Nursing Notes

(de-identified and replaced with surrogate data)

—

ccu nsg admission note: 12 am– pt is a 57yo f who is followed at gh by dr healey. she arrived a&ox3 via amb from kernan ew for further eval/monitoring. today pt was at home and states that her legs felt weak and she fell to the ground striking her head on the kitchen floor. pt states that she did not have loc. 911 was called and pt taken to kernan hosp. she sustained a lac to the back of her head that was sutured. she did rec tet tox per rn. per report she has been a&ox3. she had labs drawn which showed inr to be 24, hct 25.4, na 132, k 5.1, dig 2.4 w/elevated bun/creat. she had head ct done which was reported to be neg. she was also noted to have bp that dropped to 70's–pt cont'd a&ox3, she was started on dopa up to 8mcg, she was given 1 unit ffp. she was transfered to ccu for further monitoring. pt states that for the past few weeks she hasn't been feeling well. states that she has been having swelling in her abd that has caused her to lose her appetite. she has not been eating/drinking that well, also notes decreased u/o over the past few weeks. she has also had increased swelling to her lower ext which she states makes it harder to amb. she did fall 1 noc ago but did not sustain any injury at that time. she has been having problems w/loose stools for the past few weeks as well and states that she has had several tests done on stool which have been neg, she was taking imodium for diarrhea but it has not been working and has started a new med which she can't recall. states that she

has been having her inr followed and has been taking coumadin as instructed. she has only noted bleeding from hemrroids.

ros– neuro–a&ox3, mae, skin w&d, c/o pain to back of head. head w/sutures, no bleeding from site at this time

resp–ls w/crackles at bases, cta in upper lobes, sat on 5l 98%, rr 16 not labored, no c/o sob

cardiac–hr 70's av paced, arrived on 7mcg of dopa, bp 90-100/40's, no c/o cp

gi–abd obese, firm/distened, (+)bs, did pass sm amt of brown stool, no c/o abd pain at this time

gu–pt states no void since 3pm, feels like she has to void, foley placed for 50cc dark yellow urine

skin–area of ecchymosis to r shoulder/upper arm, does also have other areas of bruising to arms/legs, skin to back/buttocks intact

access–arrived w/2 #22 iv's to r arm, #18 ac placed and bloods resent

social–pt married, lives in catonsville, husband did not come to gh w/pt, he is aware that she is here

—

NSR, no ectopy. BP stable. Lungs clear, 6L/NC with good sats. Urine output marginal. MIVF started. NPO except sips with meds. Pt reports no CP. Heparin, Ntg, and aggrestat infusing. Heparin titrated per orders for PTT.

—

0730: PT AXOX3. VSS. AFEB. PT REMAINS ON IV NTG, AGGRESTAT, AND HEPARIN. PT SENT TO CATH LAC HOLDING AREA AT 0730 FOR CARDIAC CATHETERIZATION. PROCEDURE EXPLAINED TO PT BY RN. PT VERBALIZED UNDERSTANDING. TRANSPORTED WITH TRANSPORT AND RN. PT STABLE.

Bibliography

[1] 1996 equifax/harris consumer privacy survey. http://www.mindspring.com/ mdeeb/equifax/cc/parchive/svry96/docs/summary.html.

[2] Committee on the use of humans as experimental subjects. http://web.mit.edu/committees/couhes/.

[3] Common rule, 46.101 b.

[4] Health insurance portability and accountability act of 1996.

[5] http://www.physionet.org/.

[6] http://www.privacert.com.

[7] Legislative documents on personal privacy from the european commission. http://europa.eu.int/comm/internal_market/privacy/law_en.htm.

[8] Organization for economic co-operation and development guidelines on the protection of privacy and transborder flows of personal data. http://www.oecd.org/document/18/0,2340,en_2649_34255_1815186_1_1_1_1,00.html.

[9] The personal information protection and electronic documents act. http://www.privcom.gc.ca/legislation/02_06_01_e.asp.

[10] Worldatlas.com. http://worldatlas.com.

[11] X00's alphabetic list of surname prefixes. http://www.itsmarc.com/crs/Auth0718.htm.

[12] American Medical Association. Original code of medical ethics, 1847. http://www.ama-assn.org/ama/upload/mm/369/1847code.pdf.

[13] Kevin Atkinson. Spell checking oriented word lists, revision 6, 2004. http://prdownloads.sourceforge.net/wordlist/scowl-6.tar.gz.

[14] J. J. Berman. Concept-match medical data scrubbing. *Arch Pathol Lab Med*, 127:680–686, June 2003.

[15] U.S. Census Bureau. 1990 census name files, 1999. http://www.census.gov/genealogy/names/.

[16] U.S. Census Bureau. Census 2000 urbanized area and urban cluster information, 2004. http://www.census.gov/geo/www/ua/uaucinfo.html#lists.

[17] R. B. D'Angostino. *Introductory Applied Biostatistics.* Houghton Mifflin College Div., 2001.

[18] A. S. Goldberg. Hipaa summit ix boot camp, 2004. http://www.ehcca.com/presentations/HIPAA9/goldberg.pdf.

[19] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulations*, 101(23):e215–e220, 2000.

[20] D. Gupta, M. Saul, and J. Gilbertson. Evaluation of a de-identification software engine: Progress towards sharing clinical documents and pathology reports. *Am J Clin Pathol*, 121(2):176–186, 2004.

[21] N. H. Hebbar. Text and history of ayurveda, 2004. http://www.boloji.com/ayurveda/av024b.htm.

[22] Hippocrates. *Harvard Classics, Vol. 38.* P.F. Collier and Son, Boston, 1910.

[23] J. Kaiser. Privacy rule creates bottleneck for u.s. biomedical researchers. *Science*, 305(5681):168–9, 2004.

[24] N. Keene, W. Hobbie, and K. Ruccione. *Childhood Cancer Survivors: A Practical Guide to Your Future.* O'Reilly & Associates, 2000. http://www.patientcenters.com/survivors/news/jobs.html.

[25] J. Levine. *De-identification of ICU Patient Records.* MIT Press, 77 Mass. Av. Cambridge, MA, USA, 2003. MEng Thesis.

[26] D. A. Lindberg, Humphreys B. L., and McCray A. T. The unified medical language system. *Methods Inf Med*, 32(4):281–91, Aug 1993.

[27] D. F. Linowes and R. C. Spencer. How employers handle employees' personal information. *Employee Rights and Employment Policy Journal*, 1(1):154–172, 1997. http://www.kentlaw.edu/ilw/erepj/abstracts/v1n1/linowes.html.

[28] R. G. Mark. Bioengineering research partnership project proposal, 2001.

[29] L. J. Melton. The threat to medical-records research. *New England Journal of Medicine*, 337(20):1466–1471, Nov 1997.

[30] P. Ruch, R. Baud, A-M Rassinoux, P Bouillon, and G Robert. Medical document anonymization with a semantic lexicon. *Proc AMIA Symp*, pages 729–733, 2000.

[31] M. Saeed, C. Lieu, G. Raber, and R.G. Mark. Mimic ii: A massive temporal icu patient database to support research in intelligent patient monitoring. *Computers in Cardiology*, 29:641–644, 2002.

[32] L Sweeney. Replacing personally-identifying information in medical records, the scrub system. *Proc AMIA Symp*, pages 333–337, 1996.

[33] L Sweeney. Guarenteeing anonymity when sharing medical data, the datafly system. *Proc AMIA Symp*, pages 51–55, 1997.

[34] L. Sweeney. Information explosion. In L. Zayatz, P. Doyle, J. Theeuwes, and J. Lane, editors, *Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies.* Urban Institute, 2001.

[35] R. K. Taira, A. A. T. Bui, and H. Kangarloo. Identification of patient name references within medical documents using semantic selectional restrictions. *Proc AMIA Symp*, pages 757–761, 2002.

[36] S. M. Thomas, B. Mamlin, G. Schadow, and McDonald C. A successful technique for removing names in pathology reports using an augmented search and replace method. *Proc AMIA Symp*, pages 777–781, 2002.