An Evaluation of Feature Sets and Sampling Techniques for De-identification of Medical Records

James Gardner
Department of Mathematics
and Computer Science
Emory University
jgardn3@emory.edu

Li Xiong[*]
Department of Mathematics
and Computer Science
Emory University
lxiong@mathcs.emory.edu

Fusheng Wang
Center for Comprehensive
Informatics
Emory University
fusheng.wang@emory.edu

Andrew Post
Center for Comprehensive
Informatics
Emory University
arpost@emory.edu

Joel Saltz
Center for Comprehensive
Informatics
Emory University
jhsaltz@emory.edu

Tyrone Grandison
Core Healthcare Services
IBM Services Research
tyroneg@us.ibm.com

ABSTRACT

De-identification of textual medical records is of critical importance in any health informatics system in order to facilitate research and sharing of medical records. While statistical learning based techniques have shown promising results for de-identification purposes, few such systems are publicly available. It remains a challenge for practitioners to build an accurate and efficient system as it involves a significant amount of feature engineering, i.e. creation and examination of new features used in the system. A comprehensive evaluation is needed to thoroughly understand the effects of different feature sets and potential impacts of sampling and their trade-offs between the often conflicting goals of precision (or positive predictive value), recall (or sensitivity), and efficiency.

In this paper, we present the Health Information DE-identification (HIDE) framework and evaluate the open-source software. We present an evaluation of various types of features used in HIDE, and introduce a window sampling technique (only the terms within a specified distance from personal health information are used to train the classifier) and evaluate its effect on both quality and efficiency. Our results show that the context features (previous and next terms) are particularly important and the sampling technique can be used to increase recall with minimal impact on precision. We obtained token-level label precision of 0.967, recall of 0.986 and F-Score of 0.977 when not including true negatives. The overall HIDE system achieves token-level precision of .998, recall of .999, and f-score of .999 on the previous i2b2 challenge task.

[*]Corresponding author.

IHI'10, November 11–12, 2010, Arlington, Virginia, USA.
Copyright 2010 ACM 978-1-4503-0030-8/10/11 ...$10.00.

Categories and Subject Descriptors

H.2.0 [**Database Management**]: General—Security, integrity, and protection; J.3 [**Life and medical sciences**]: Medical information systems; K.4.1 [**Computers and Society**]: Public Policy Issues—Privacy

General Terms

Measurement, Performance, Experimentation

Keywords

Medical text, De-identification, Conditional random fields

1. INTRODUCTION

De-identification of medical records is of critical importance in any health informatics system in order to facilitate research and sharing of medical records. Under the HIPAA Privacy Rule[1], Protected health information (PHI) is defined as individually identifiable health information and is subject to restrictions on access, use, and disclosure. Health information is qualified as de-identified if it neither identifies nor provides a reasonable basis to identify an individual and is exempt from the above privacy restrictions. De-identification systems focus on detecting and removing (or replacing) PHI in the medical record resulting in a de-identified record. De-identified data can be used for a variety of purposes such as quality improvement, research, and teaching.

As a considerable amount of medical records resides in textual forms such as clinical notes, SOAP (subjective, objective, assessment, patient care plan) notes, radiology and pathology reports, an important task of de-identification is to detect PHI references within medical text which can be then replaced or removed.

CLINICAL HISTORY: 90 year old female with a history of B-cell lymphoma (Marginal zone, SH-02-22222, 6/22/01). Flow cytometry and molecular diagnostics drawn.

Figure 1: A Sample Pathology Report Section

Figure 1 shows a sample pathology report section with personally identifying information such as age and medical

[1] Health Insurance Portability and Accountability Act (HIPAA). http://www.hhs.gov/ocr/hipaa/

record number highlighted. It is necessary for the de-identification process to detect the personal identifiers from the text and replace or remove them.

The main approaches for PHI detection from text can be classified into rule-based or statistical (machine learning)-based methods. Rule-based systems can be quite powerful, but they lack the portability necessary for multiple institutions to quickly adopt a software package based on such techniques.

The statistical learning techniques use a list of feature attributes to train a classification model and classify the terms in new text as either identifier or non-identifier. While it requires manually annotated training data, it can be ported to other domains or genres of text much more rapidly.

While several works studied statistical learning techniques such as conditional random fields (CRF) for de-identification purposes and shown promising results, few such systems are publicly available. It remains a challenge for practitioners to build an accurate and efficient system as it involves a significant amount of feature engineering. A comprehensive evaluation is needed to thoroughly understand the effects of different feature sets and potential impacts of sampling and their tradeoffs between the often conflicting goals of precision (or positive predictive value), recall (or sensitivity), and efficiency. Any medical de-identification system requires high recall of PHI, but the precision must be acceptable. It is possible to detect PHI with high precision in many types of highly unstructured data, but the recall is sometimes low.

In this paper, we present a detailed study of various types of features used in our learning based de-identification system. We also present a window sampling technique to increase performance and tailor the system for a particular user's precision and recall requirements. The paper builds on top of the open-source Health Information DE-identification (HIDE) system we developed at Emory [4, 5, 3], which uses Conditional Random Fields (CRFs) as the underlying machine learning technique.

It presents an extension of the features in the existing HIDE system and a more thorough evaluation of different feature sets and sampling techniques for CRF-based de-identification and the latest version of HIDE on gold standard datasets. Our results show that the context features are particularly important and the sampling technique can be used to increase recall with minimal impact on precision. The overall HIDE system achieves token-level precision of .998, recall of .999, and f-score of .999 on the previous i2b2 challenge task.

The remainder of this paper is organized as follows. Section 2 presents on overview of existing work on de-identification systems. Section 3 presents the HIDE framework and software. Section 4 describes the sequence labeling problem and the various types of features used in HIDE. The features can be classified into regular expression, affix (prefix and suffix), context, and dictionary features. Section 5 introduces a window sampling technique. Section 6 reports the detailed evaluation results of the different feature sets and sampling techniques. Further discussion and conclusions are in Section 7.

2. RELATED WORK

This section briefly describes other de-identification systems and approaches. The most common approaches to de-identification are based on rules and dictionaries or statistical learning techniques. HMS Scrubber [1] is an open-source system implemented in Java that utilizes the header information associated with a record, rules for detecting common PHI (e.g. dates), and a dictionary of common names (and names associated with the institution). Any information that matches is then removed from the record. An alternative open-source system implemented in Perl using similar techniques as the HMS Scrubber can be found in [9]. This system is associated with PhysioNet [6]. The PhysioNet webpage[2] also includes a gold standard dataset of nursing notes. We evaluate our system on this dataset in Section 6. Rules based on local context and semantic lexicons were studied in [17]. The system builds rules based on the surrounding terms and information gleaned from a sematic lexicon to detect PHI.

An alternative approach that uses a dictionary of safe (guaranteed non-PHI) terms and removes all terms that are not in the list can be found in [2]. The Concept-Match algorithm steps through the record replacing all standard medical terms with the corresponding code, leaves all high frequency (stop words) and removes all other terms leaving a de-identified record. This technique has high recall, but suffers from lower precision.

Other systems use machine learning techniques. The best performing systems use a variety of features and use either Support Vector Machine (SVM) [11], variations of decision tree [16] or CRF [13] classifiers as their underlying statistical learning frameworks.

Uzuner, *et. al.* [12] introduced the i2b2 datasets as a gold standard for evaluating medical record de-identification solutions. The best performing systems on this data (and most similar to HIDE) was the Carafe system of Wellner, *et. al.* [13]. Carafe also utilizes conditional random fields. See [14] for more discussions of privacy and de-identification techniques on electronic health records. We evaluate our system on the i2b2 dataset in Section 6.

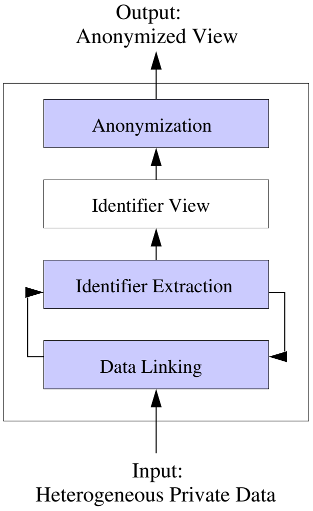


Figure 2: HIDE Conceptual Framework

3. HIDE

HIDE (Health Information DE-identification) [4, 3, 5] is a framework for de-identifying both structured and unstruc-

[2] http://www.physionet.org/physiotools/deid/

tured data. This section gives a brief description of the HIDE framework and the software implemented according to the framework.

3.1 Conceptual Framework

HIDE consists of three major components: *identifier extraction*, *data linking*, and *de-identification and anonymization.* Figure 2 gives a graphical description of the conceptual framework.

Identifier Extraction. The identifier extraction component extracts the identifying information including HIPAA identifiers as well as sensitive attributes from unstructured (text) data. Note that in order to apply advanced data anonymization techniques, HIDE can extract a much broader set of information than existing de-identification systems that typically focus on the set or a subset of HIPAA identifiers.

Data Linking. In relational data, we assume each tuple corresponds to an individual entity. This mapping is not present in heterogeneous medical data repositories. For example, one patient may have multiple pathology and lab reports prepared at different times. In order to preserve privacy for individuals and apply statical de-identification in this complex data space, the data linking component links relevant attributes (structured or extracted) to each individual entity and produces a person-centric representation of the data. HIDE provides an interative process between the identifier extraction and data linking components. Once an identifier is detected it is assigned to an existing patient in the system, and using the information for each patient can be used to help extract more PHI from the text.

De-identification and Anonymization. Once the identifying attributes are extracted and linked to individuals, they form a structured identifier view. This notion of identifier view allows application of advanced anonymization algorithms that are otherwise not applicable to unstructured data. Given an identifier view, the anonymization component anonymizes the data using different privacy models, including k-anonymization, l-diversity, or differential privacy. Finally, using the resulting values from the anonymized identifier view, we can remove or replace the identifiers in the original data.

This paper focuses on the underlying technologies of the identifier extraction component in the framework. Section 4 gives more details of the underlying machine learning component for identifier extraction. We refer the readers to [3] for a thorough description of the HIDE framework.

3.2 Software

HIDE is a web-based application that utilizes the latest web-technologies. HIDE is written in Python on top of the Django[3] web application framework. It uses Apache CouchDB[4] as the document storage engine. HIDE provides users (primarily honest brokers and de-identification researchers) with the ability to either manually or automatically label (annotate), de-identify, anonymize, and analyze the data. HIDE provides a web-based annotation interface (javascript) that allows iterative annotation of documents and training of the classifier for detecting PHI. This allows the user to quickly create training sets for the CRF classifier. HIDE uses the CRFSuite [10] package for the underlying CRF. This provides fast training and auto-labeling (Section 4) functionality in the system.

HIDE has been currently integrated into the caTIES[5] de-identification pipeline. The software package can be configured to use HIDE as a de-identification option for pathology reports in the caTIES database. HIDE can import data from a variety of sources. The system is currently being implemented and tested in real-world settings by multiple institutions. More details can be found at the HIDE project[6] and code[7] web pages.

4. SEQUENCE LABELING AND FEATURES

De-identifying medical text can be viewed as the often encountered task of named entity recognition (NER) in natural language processing (NLP). One of the most successful methods for NER is to cast it into a sequence labeling problem.

4.1 Sequence Labeling

Sequence labeling is the process of labeling each token in a sequence with a label corresponding to features of the token in the sequence. One of the most common examples of sequence labeling is part-of-speech (POS) tagging, where each token in the sequence is labeled with its corresponding part-of-speech. Detecting PHI in medical text is very similar, except that the labels correspond to whether or not the term is (or is part of) a name, date, medical record number (MRN), *etc.* If the term is not PHI, it is labeled with an "O."

CLINICAL HISTORY: <age>90</age> year old female with a history of B-cell lymphoma (Marginal zone, <id>SH-02-22222</id>, <date>6/22</date>/01). Flow cytometry and molecular diagnostics drawn.

Figure 3: A Sample Marked Pathology Report Section

Figure 3 shows an example pathology report with the PHI surrounded by tags. Our task is to train the computer to label the sequence of tokens in the pathology report with the correct PHI labels corresponding to the tags. In order to predict the correct label for a token it is necessary to build features for each token that can be used to calculate the probability of a label given the set of features. This set of features (corresponding to and including the token) are referred to as a feature vector. This sequence of feature vectors is then used in the machine learning framework for predicting PHI and for training the underlying classifier.

PHI extraction in HIDE consists of *training* and *labeling* phases. In order for HIDE to automatically label the PHI in the document it must first be trained on how to predict the correct labels. The training phase consists of (1) tokenizing the records in the gold-standard training set, (2) building the feature vector for each token, and (3) constructing a statistical model of the feature vectors corresponding to the known labels. The labeling phase consists of (1) tokenizing the record, (2) building the feature vector for each token, and (3) predicting the correct label sequence given the feature vector sequence.

[3] `http://www.djangoproject.com/`

[4] `http://couchdb.apache.org/`

[5] `http://caties.cabig.upmc.edu/`

[6] `http://mathcs.emory.edu/hide/`

[7] `http://code.google.com/p/hide-emory`

Label	Token	ALPHA?	NUMBER?	PREV_WORD	NEXT_WORD	PRE1	SUF1
O	HISTORY	1	0	CLINICAL	90	H	Y
age	89	0	1	HISTORY	year	7	7
O	year	1	0	age	old	y	r
O	old	1	0	year	female	o	d

Table 1: Example subset of features in feature vectors generated from marked report section.

<table>
<tr><th>Regular Expression</th><th>Name</th></tr>
<tr><td>^[A-Za-z]$</td><td>ALPHA</td></tr>
<tr><td>^[A-Z].*$</td><td>INITCAPS</td></tr>
<tr><td>^[A-Z][a-z].*$</td><td>UPPER-LOWER</td></tr>
<tr><td>^[A-Z]+$</td><td>ALLCAPS</td></tr>
<tr><td>^[A-Z][a-z]+[A-Z][A-Za-z]*$</td><td>MIXEDCAPS</td></tr>
<tr><td>^[A-Za-z]$</td><td>SINGLECHAR</td></tr>
<tr><td>^[0-9]$</td><td>SINGLEDIGIT</td></tr>
<tr><td>^[0-9][0-9]$</td><td>DOUBLEDIGIT</td></tr>
<tr><td>^[0-9][0-9][0-9]$</td><td>TRIPLEDIGIT</td></tr>
<tr><td>^[0-9][0-9][0-9][0-9]$</td><td>QUADDIGIT</td></tr>
<tr><td>^[0-9,]+$</td><td>NUMBER</td></tr>
<tr><td>[0-9]</td><td>HASDIGIT</td></tr>
<tr><td>^.*[0-9].*[A-Za-z].*$</td><td>ALPHANUMERIC</td></tr>
<tr><td>^.*[A-Za-z].*[0-9].*$</td><td>ALPHANUMERIC</td></tr>
<tr><td>^[0-9]+[A-Za-z]$</td><td>NUMBERS_LETTERS</td></tr>
<tr><td>^[A-Za-z]+[0-9]+$</td><td>LETTERS_NUMBERS</td></tr>
<tr><td>-</td><td>HASDASH</td></tr>
<tr><td>’</td><td>HASQUOTE</td></tr>
<tr><td>/</td><td>HASSLASH</td></tr>
<tr><td>‘~!@#$%\^&*()\-=_+\[\]{}|;’:\",./<>?]+$</td><td>ISPUNCT</td></tr>
<tr><td>(-|\+)?[0-9,]+(\.[0-9]*)?%?$</td><td>REALNUMBER</td></tr>
<tr><td>^-.*</td><td>STARTMINUS</td></tr>
<tr><td>^\+.*$</td><td>STARTPLUS</td></tr>
<tr><td>^.*%$</td><td>ENDPERCENT</td></tr>
<tr><td>^[IVXDLCM]+$</td><td>ROMAN</td></tr>
<tr><td>^\s+$</td><td>ISSPACE</td></tr>
</table>

Table 2: List of regular expression features used in HIDE

The Conditional Random Field (CRF) framework [7] was developed for the sequence labeling task. CRFs are one of the best machine learning techniques for sequence labeling and hence are very good at detecting PHI in text. HIDE uses the CRF as its underlying statistical learning framework. A CRF takes as input a sequence of feature vectors, calculates the probabilities of the various possible labelings (the type of PHI for each token in the sequence) and chooses the one with maximum probability. The probability of a labeling is a function of the feature vectors associated with the tokens. More specifically, a CRF is an undirected graphical model that defines a single log-linear distribution function over label sequences given the observation sequence (feature vector sequence). The CRF is trained by maximizing the log-likelihood of the training data.

4.2 Features

We now describe the set of features used to construct the feature vectors in the HIDE system. Table 1 shows example feature vectors based on the sample marked report. The features can be categorized into regular expression, affix, dictionary, and context features. We empirically study the effects of the various features in Section 6.

Regular Expression Features. Regular expression features are those features that are generated by matching regular expressions to the tokens in the text. The value for a given regular expression is active (specifically the value for the feature is set to 1 in the CRF framework) if the token matches the regular expression. These featuers are useful for detecting medical record numbers and phone numbers. The regular expression features are fairly standard and similar to those in [13]. Table 2 contains the list of all regular expression features used in HIDE.

Affix Features. The prefix and suffix of a token are affix features. HIDE uses the prefixes and suffixes of length one, two and three for each token. E.g., if the token is “diagnosis” the affix features of `PRE1_d`, `PRE2_di`, `PRE3_dia`, `SUF1_s`, `SUF2_is`, and `SUF3_sis` would be active. These features can be useful for detecting certain classes of terms that have common prefixes or suffixes.

Dictionary Features. HIDE can use any number of dictionaries. If a phrase (or token) is encountered that matches any of the entries in the dictionary a feature indicating that each token is contained in the dictionary is added to the feature vector. Suppose that “John” is in a dictionary file called `male_names_unambig`. If “John” occurs in the text, then the feature `IN_male_names_unambig` would be active in the feature vector associated with the token “John.” HIDE currently uses all of the dictionaries from the PhysioNet de-identification webpage.

Context Features. Previous words, next words, and occurrence counts are examples of context features. Sibanda and Uzuner [11] demonstrate that context features are important features for de-identification. HIDE includes the previous and next four tokens and the number of occurrences of the term scaled by the length of the sequence in each feature vector

5. SAMPLING

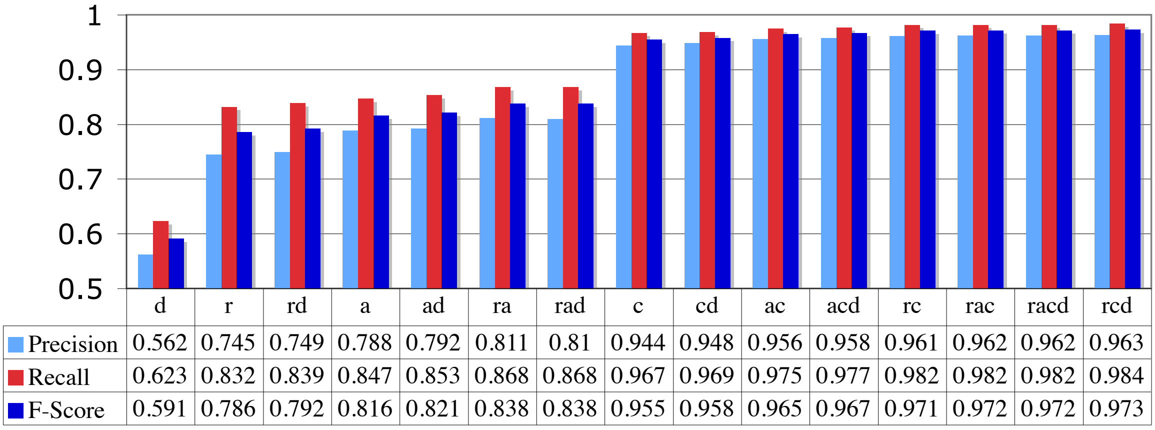
The overwhelming number of “O” tags biases the classifier into predicting “O” as the label. A simple technique for removing some of this bias is to remove the number of “O” in the training set. This will increase the recall of most labels at the cost of decreasing precision (positive predictive value). Gardner and Xiong [3] investigated the use of cost-proportionate rejection sampling to increase the accuracy of de-identification using Naive Bayes classifiers. Many of the “O” labels between real labels can be removed while still retaining the information necessary for the classifier to decide when an entity is going to appear in the text. Window sampling is based on this idea.

5.1 Random O-Sampling

We use random O-sampling as a baseline to compare to our window sampling method. Random O-samping keeps every non-“O” label and selects every “O” label with probability *p*. The intuition behind this method is a version of cost-proportionate rejection sampling [15], except that the order of the training data is always preserved and the non-“O” labels are always selected. This method decreases the number of “O” labels the classifier sees and thus, the classifier will choose the “O” label less often with the overall effect of increasing recall. Section 6 shows the effects of random O-sampling.

True /Pred	B-age	B-date	B-doctor	B-hospital	B-id	B-location	B-patient	B-phone	I-date	I-doctor	I-hospital	I-id	I-location	I-patient	I-phone	O	Total	Prec	Rec	F1
B-age	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	1.0	0.667	0.8
B-date	0	1919	0	1	5	0	0	0	0	0	0	0	0	0	0	6	1931	0.996	0.994	0.995
B-doctor	0	2	1043	2	0	4	0	0	0	1	0	0	0	0	0	18	1070	0.985	0.975	0.980
B-hospital	0	1	2	663	1	2	0	0	0	0	0	0	0	0	0	7	676	0.987	0.981	0.984
B-id	0	1	1	1	1136	0	0	0	0	0	0	0	0	0	0	4	1143	0.987	0.994	0.990
B-location	0	3	3	5	0	104	1	0	0	0	0	0	0	0	0	3	119	0.920	0.874	0.897
B-patient	0	0	7	0	0	3	230	0	0	0	0	0	0	0	0	5	245	0.996	0.939	0.966
B-phone	0	0	0	0	2	0	0	55	0	0	0	0	0	0	0	1	58	1.0	0.948	0.973
I-date	0	0	0	0	0	0	0	0	3462	0	0	2	0	0	0	42	3506	0.997	0.987	0.992
I-doctor	0	0	0	0	0	0	0	0	0	2500	2	0	2	0	0	78	2582	0.974	0.968	0.971
I-hospital	0	0	0	0	0	0	0	0	2	4	1889	0	6	0	0	28	1929	0.991	0.979	0.985
I-id	0	0	0	0	1	0	0	0	2	0	2	602	0	0	0	21	628	0.711	0.959	0.816
I-location	0	0	0	0	0	0	0	0	0	14	4	0	196	2	0	28	244	0.951	0.803	0.871
I-patient	0	0	0	0	0	0	0	0	0	17	0	0	0	513	0	8	538	0.996	0.954	0.974
I-phone	0	0	0	0	0	0	0	0	0	0	0	0	0	0	220	24	244	1.0	0.902	0.948
O	0	1	3	0	6	0	0	0	8	31	10	243	2	0	0	0	304	NA	NA	NA

Table 3: Confusion matrix showing token label accuracy using all features



	d	r	rd	a	ad	ra	rad	c	cd	ac	acd	rc	rac	racd	rcd
Precision	0.562	0.745	0.749	0.788	0.792	0.811	0.81	0.944	0.948	0.956	0.958	0.961	0.962	0.962	0.963
Recall	0.623	0.832	0.839	0.847	0.853	0.868	0.868	0.967	0.969	0.975	0.977	0.982	0.982	0.982	0.984
F-Score	0.591	0.786	0.792	0.816	0.821	0.838	0.838	0.955	0.958	0.965	0.967	0.971	0.972	0.972	0.973

Figure 4: Figure showing dictionary, affix, regular expression, and context features in order of increasing importance (from all but one result).

5.2 Window Sampling

In window sampling we keep every non-"O" label and a window of size k around that label. The intuition behind this method is similar to the random O-sampling except that it treats all "O" labeled terms not "near" PHI as noise to the classifier as we are more interested in detecting PHI than non-PHI. Section 6 shows that the window sampling technique can be quite useful for tweaking the precision and recall for the HIDE system.

6. EVALUATION AND EXPERIMENTS

We performed all experiments on a machine with 8 cores at 2.2ghz and 16 gb of ram.

Datasets. Our datasets consist of the PhysioNet and the i2b2 datasets. These are some of the only publicly available (with some licensing restrictions) datasets for evaluating medical de-identification solutions. The i2b2 dataset consists of example pathology reports that have been re-synthesized with fake PHI. The reports are somewhat structured and have sentence structure. The PhysioNet data consists of re-synthesized nursing notes that are very sporadic and contains almost no sentence structure.

For the figures in the experimental section we constructed 10-fold cross-validation sets for the i2b2 and PhysioNet datasets. The cross-validation set for i2b2 is composed of the 220 reports in the i2b2 testing dataset[8] The cross-validation set for Physionet is composed of 163 records that are a subset of the full PhysioNet dataset.

Metrics. All numbers are reported for token-level accuracy and exclude all true negatives (those tokens that are correctly labeled as "O"). We report the standard precision (positive predictive value), recall, and f-score. True positives (TP) are those PHI which are correctly labeled as PHI, false positives (FP) are those tokens that are labeled as PHI when they should be labeled as "O," true negatives (TN) are those tokens correctly labled as "O" and false negatives (FN) are those tokens that should be labeled as PHI but are marked as "O." Precision (P) or the positive predictive value is defined as $P = TP/(TP + FP)$. Recall (R) or sensitivity is defined as $R = TP/(TP + FN)$ and f-score (F) is defined as $F = 2(P \cdot R)/(P + R)$. It is worth noting that *specificity* is defined as $TN/(TN + FP)$. We do not report specificity because the non-identifying attributes are dominating compared to the identifying attributes so specificity will be always close to 100% which would not be very informative.

Training Details. The CRF is trained using the CRFSuite application with the L-BFGS [8] algorithm. The L-BFGS algorithm stops when the log-likelihood on the training data improves by no more than 10^{-5} from the previous iteration.

6.1 Effects of Features

We performed the feature experiments on the i2b2 cross-validation sets. The feature experiments show all subsets of regular expression, affix, dictionary, and context features.

[8] We note that this provides a smaller training set than in the i2b2 challenge.

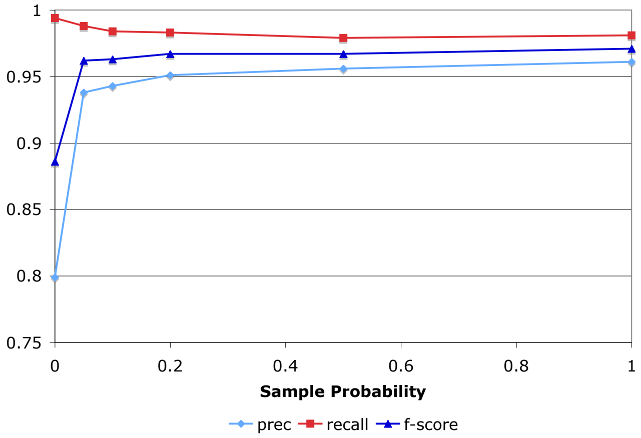


Figure 5: Effect of random O-sampling selection probability and a fixed history size of 4 on the i2b2 cross-validation data.

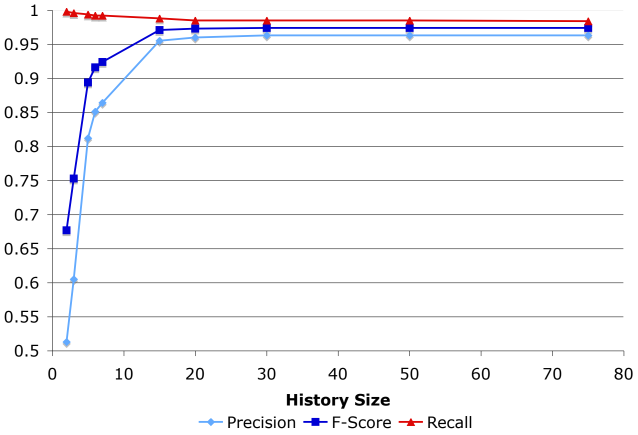


Figure 6: Effect of window history size for window filtering on i2b2 cross-validation data. History size of 10 gives a window of 20 tokens.

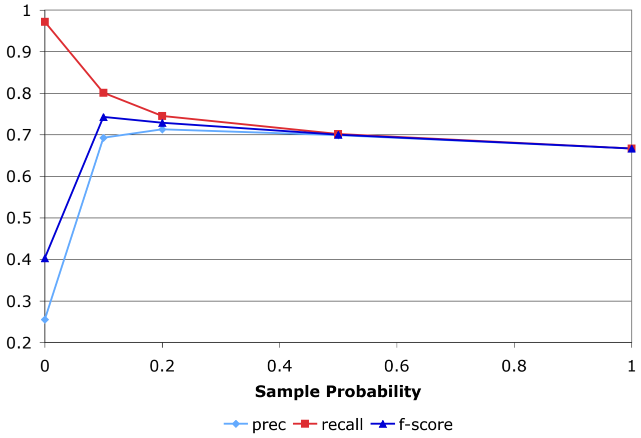


Figure 7: Effect of random O-sampling selection probability and a fixed history size of 4 on the PhysioNet cross-validation data

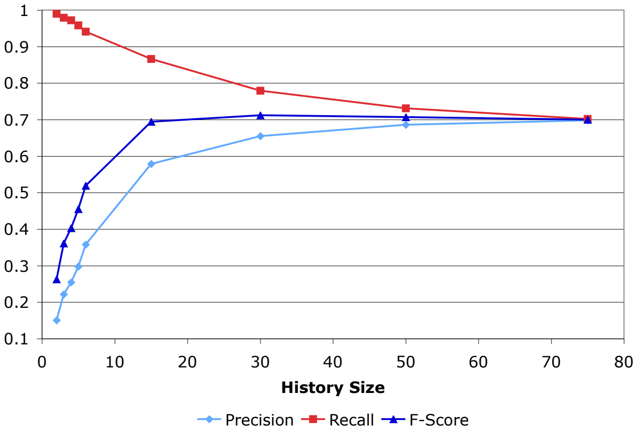


Figure 8: Effect of window history size for window filtering on PhysioNet cross-validation data.

Table 3 shows the token-level confusion matrix for labeling the PHI in the i2b2 dataset using all features. We note that the confusion matrix includes those tokens that are spaces. This only has an effect on the I-label entries and does not have significant effect on the precision and recall numbers. The results are quite good for the majority of PHI, as can be seen by the large values along the diagonal of the confusion matrix. The most commonly missed PHI are the I-id, which correspond to missing the continuation of a medical record number, e.g. detecting <id>1234</id>-123 instead of <id>1234-123</id>.

Figure 4 shows the overall term-level results for all subsets of the features. Our experiments indicate that the most important features for this task in increasing order are: dictionary, affix, regular expression, and context features. Using only the context features the classifier achieves f-score of 0.955. This verifies our intuition and the results in [11]. The regular expression features are the second most effective. The affix features are third. Readers may notice that the rcd slightly outperforms the racd feature set, but we believe this to not be significant. The least important features were the dictionary features. This is likely due to the fact that many of the terms in the text that are in the dictionaries are not PHI.

6.2 Effects of Sampling

We performed experiments with varying probability for random-O sampling[9] and various history sizes for window sampling.

Figures 5 and 6 show the effects of the random O-sampling with various selection probabilities and the effects of the window sampling on the i2b2 cross-validation dataset. When the selection probabillity is small the system is biased toward recall and when p is large the precision and recall begin to converge.

Figures 7 and 8 show the effects of the random O-sampling with various selection probabilities p and the effects of the window sampling on the PhysioNet cross-validation dataset. The curves follow similar paths as in the i2b2 data, but it

[9] In order to keep some of the context information we always keep a history size of four and randomly sample the remaining “O” labeled feature vectors.

True \ Pred	B-age	B-date	B-doctor	B-hospital	B-id	B-location	B-patient	B-phone	I-date	I-doctor	I-hospital	I-id	I-location	I-patient	I-phone	O	Total	Prec	Rec	F1
B-age	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	1.0	0.667	0.8
B-date	0	1930	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1931	0.996	0.999	0.998
B-doctor	0	1	1061	0	1	3	0	0	0	0	0	0	0	0	0	4	1070	0.985	0.992	0.988
B-hospital	0	2	3	663	1	4	0	0	0	0	0	0	0	0	0	3	676	0.982	0.981	0.981
B-id	0	0	1	1	1140	0	0	0	0	0	0	0	0	0	0	1	1143	0.990	0.997	0.994
B-location	0	3	7	11	0	96	0	0	0	0	0	0	0	0	0	2	119	0.906	0.807	0.853
B-patient	0	0	4	0	0	3	235	0	0	0	0	0	0	0	0	3	245	1.0	0.959	0.979
B-phone	0	0	0	0	2	0	0	55	0	0	0	0	0	0	0	1	58	1.0	0.948	0.973
I-date	0	0	0	0	0	0	0	0	3500	0	0	0	0	0	0	6	3506	0.998	0.998	0.998
I-doctor	0	0	0	0	0	0	0	0	0	2543	0	0	2	0	0	37	2582	0.986	0.985	0.985
I-hospital	0	0	0	0	0	0	0	0	2	6	1831	0	0	0	0	90	1929	0.984	0.949	0.966
I-id	0	0	0	0	1	0	0	0	0	0	2	616	0	0	0	9	628	0.720	0.981	0.830
I-location	0	0	0	0	0	0	0	0	0	14	12	0	192	0	0	26	244	0.980	0.787	0.873
I-patient	0	0	0	0	0	0	0	0	0	8	0	0	0	523	0	7	538	1.0	0.972	0.986
I-phone	0	0	0	0	0	0	0	0	0	0	0	0	0	0	220	24	244	1.0	0.902	0.948
O	0	1	1	0	6	0	0	0	6	8	16	240	2	0	0	0	280	NA	NA	NA

Table 4: Results on the i2b2 training and testing challenge data.

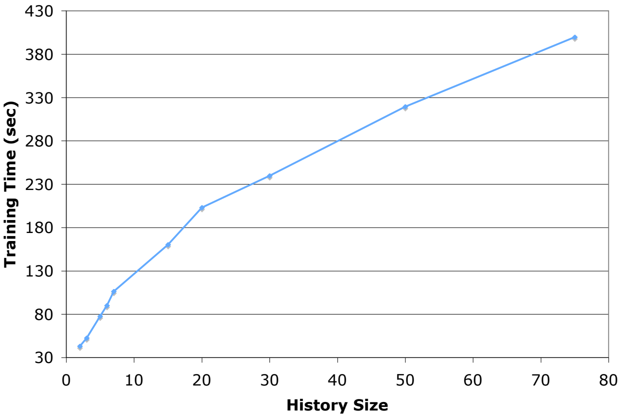


Figure 9: Training Time (seconds) vs. History Size on i2b2 dataset

should be noted that because of the highly sporadic nature of the nursing notes the recall is poor without a very small selection probability or window size. The convergence of precision and recall can can be seen in Figure 7, which indicates that sampling be stopped around 0.2. These results show that by decreasing the window size the classifier can detect all PHI. Neamatullah, *et. al* [9] report precision of 0.967 and recall of 0.749 on the full PhysioNet dataset of 1836 notes. We were only able to import a fraction of these from the site, but we believe our system would have similar results to those we have reported here on the full corpus. At a similar level of recall .972 we obtain precision of .255 with a history size of 4. This shows that the window sampling allows users to tweak the system to perform as well as hand tailored rule-based systems for recall.

6.3 Comparison to i2b2 Challenge

When training on the 669 document training set and testing on the 220 document testing set from the i2b2 challenge dataset we obtained token-level label (excluding “O”) precision of 0.967, recall of 0.986 and F-Score of 0.977.

Table 4 presents the token-level confusion matrix. This result is slightly better than the Carafe system [13] which reported a f-score of 0.975 when counting only true positives. If the Carafe system uses the feature sets described here, then theoretically it should acheive very similar or equivalent results. When counting true positives and negatives (without including spaces as tokens) as reported in the i2b2 challenge we obtain precision of 0.998, recall of 0.999, and f-score of 0.999.

6.4 Performance

The HIDE system has integrated the CRFSuite [10], which is one of the fastest CRF implementations. The training time for the full 669 report training set with all features was 51 minutes, 39 seconds. For this experiment we simultaneously trained all ten CRF models using the cross-validation training sets for each history size and averaged the values. The training time to build all ten models for the PhysioNet data was 24 seconds. The training time to build all ten models for the i2b2 cross-validation dataset with no sampling was 12 minutes, 24 seconds (744 seconds).

Figure 9 shows the training time vs. window history size training time on the i2b2 dataset. The training time increases with the history size. Setting the correct sampling rate can allow users to optimize HIDE for their different speed, precision, and recall requirements.

7. DISCUSSION AND CONCLUSION

The HIDE system has proven to perform quite well on the i2b2 dataset and can achieve high recall on the highly unstructured PhysioNet dataset when using the window sampling technique. Encoding more specialized features could prove useful for de-identifying extremely unstructured data. Modifying the dictionary for institution specific tasks may also be of utility. The HIDE system is one of the fastest CRF-based de-identification systems due to the integration of the CRFSuite package[10].

Many de-identification systems could make use of the header information that is usually stored with each individual report. HIDE currently has preliminary support for HL7 Version 2, which contains the header information (usually much of the PHI) in a machine readable format. We will continue to enhance HIDE by using the “dictionary like” information in the header and constructing more meaningful non-local features.

We have described the HIDE framework and real-world software. We have demonstrated that context features are the most important for de-identification as well as shown the effect of a variety of features. We described the window sampling technique for tweaking the time, precision, and recall performance of the system.

8. ACKNOWLEDGEMENTS

The project described was supported by Grant Number R01LM009239 from the National Library of Medicine and a Career Enhancement Fellowship from the Woodrow Wilson

[10] `http://www.chokkan.org/software/crfsuite/benchmark.html`

Foundation. The content is solely the responsibility of the authors and does not necessarily represent the official views of the research sponsors. We thank the anonymous reviewers for their comments which helped improve the paper.

The i2b2 de-identified clinical records used in this research were provided by the i2b2 National Center for Biomedical Computing funded by U54LM008748 and were originally prepared for the Shared Tasks for Challenges in NLP for Clinical Data organized by Dr. Ozlem Uzuner, i2b2 and SUNY.

9. REFERENCES

[1] B. A. Beckwith, R. Mahaadevan, , U. J. Balis, and F. Kuo. Development and evaluation of an open source software tool for deidentification of pathology reports. *BMC Medical Informatics and Decision Making*, 6(12), 2006.

[2] J. Berman. Concept-match medical data scrubbing. how pathology text can be used in research. *Arch Pathol Lab Med*, 127(6):680–6, 2003.

[3] J. Gardner and L. Xiong. An integrated framework for de-identifying unstructured medical data. *Data Knowl. Eng.*, 68(12):1441–1451, 2009.

[4] J. Gardner and L. Xiong. Hide: An integrated system for health information de-identification. In *CBMS*, pages 254–259. IEEE Computer Society, 2008.

[5] J. Gardner, L. Xiong, K. Li, and J. J. Lu. Hide: heterogeneous information de-identification. In M. L. Kersten, B. Novikov, J. Teubner, V. Polutin, and S. Manegold, editors, *EDBT*, volume 360 of *ACM International Conference Proceeding Series*, pages 1116–1119. ACM, 2009.

[6] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. 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. *Circulation*, 101(23):e215–e220, 2000 (June 13). Circulation Electronic Pages: http://circ.ahajournals.org/cgi/content/full/101/23/e215.

[7] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning*, pages 282–289, 2001.

[8] D. C. Liu, J. Nocedal, D. C. Liu, and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

[9] I. Neamatullah, M. Douglass, L. Lehman, A. Reisner, M. Villarroel, W. Long, P. Szolovits, G. Moody, R. Mark, and G. Clifford. Automated De-Identification of Free-Text medical records. *BMC Medical Informatics and Decision Making*, 8(32), 2008.

[10] N. Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs), 2007.

[11] T. Sibanda and O. Uzuner. Role of local context in automatic deidentification of ungrammatical, fragmented text. In *In Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL*, page 73, 2006.

[12] O. Uzuner, Y. Luo, and P. Szolovits. Evaluating the State-of-the-Art in automatic de-identification. *JAMIA*, 14(5):550–563, 2007.

[13] B. Wellner, M. Huyck, S. Mardis, J. Aberdeen, A. Morgan, L. Peshkin, A. Yeh, J. Hitzeman, and L. Hirschman. Rapidly retargetable approaches to de-identification in medical records. *JAMIA*, 14(5):564—573, 2007.

[14] L. Xiong, J. Gardner, P. Jurczyk, E. Agichtein, and J. J. Lu. Privacy preserving information discovery on ehrs. In V. Hristidis, editor, *Information Discovery on Electronic Health Records*, pages 197–225. Chapman and Hall/CRC, 2009.

[15] B. Zadrozny, J. Langford, and N. Abe. Cost-Sensitive learning by Cost-Proportionate example weighting. In *IEEE International Conference on Data Mining*, 2003.

[16] G. Szarvas, R. Farkas, and R. Busa-Fekete. State-of-the-art anonymization of medical records using an interative machine learning framework. *JAMIA*, 14(5):574–580, 2007.

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