A Method for Automatic De-identification of Medical Records

Arya Tafvizi
MIT CSAIL
Cambridge, MA 02139, USA
tafvizi@csail.mit.edu

Maciej Pacula
MIT CSAIL
Cambridge, MA 02139, USA
mpacula@csail.mit.edu

Abstract

Electronic medical records contain information valuable to researchers in various disciplines. However, the use of records in practice is limited due to privacy protection laws. Clinical documents such as discharge summaries contain many instances of protected health information (PHIs), which have to be removed before the data is made publicly available. Manual de-identification, where PHIs are removed by human experts, is both slow and in many cases inaccurate.

In this work, we describe the design and implementation of an automatic de-identifier, and evaluate its performance on a large corpus of discharge summaries. We demonstrate that our de-identifier can effectively locate PHIs in free-text medical documents, achieving 97.37% precision and 93.52% recall, and outperforms humans both in accuracy and processing speed. We further evaluate the robustness of our solution through feature selection, and by varying the size of the training corpus. We show that performance improves as a function of the training set size, implying that even better results can be obtained by using a larger training set. We also perform SVM kernel selection, demonstrating that various kernels are effective for de-identification.

1 Introduction

Electronic medical records contain information valuable to many research areas - decision support systems, patient monitoring and epidemiology, among others (Neamatullah et al., 2008). However, such records often contain protected health information (PHI), which limits their use outside of hospitals (Uzuner et al., 2008). Under the Health Insurance Portability and Accountability Act of 1996 (HIPAA), the amount of individually identifiable health information released for a specific purpose, except if explictly authorized by the patient, cannot be more that "minimum necessary" for the purpose. In practice, this requirement means that a patient's medical records can rarely be disclosed outsise the normal course of treatment without the patient's written consent (Annas, 2003), limiting the records' use in research.

In order for a record to be considered deindentified, HIPAA requires that it does not explicitly identify an individual, or provide other information which can be used to identify an individual. Such de-identification can be accomplished in one of two ways:

1. A person with "appropriate knowledge of and experience with generally accepted statistical and scientific principles and methods" must determine that the risk of patient identification is "very small" (HHS.gov, 2010).

2. Instances of seventeen classes of identifiers (PHIs) are removed from the record. These include names, geographic subdivisions smaller than state, telephone numbers and all elements of date except the year, among others (Uzuner et al., 2007).

The first option is generally infeasible and inefficient, since an approval by a human expert would be required for each new study. Prior de-identification performed by human experts (either by clinicians or persons familiar with medical terms), on the other hand, has been shown to be prohibitively time-consuming and expensive (Neamatullah et al., 2008). In addition, human performance on de-identification tasks is highly variable and some studies show that computer algorithms can perform at least as well (Uzuner et al., 2007; Neamatullah et al., 2008).

While automatic de-identification of medical records is appealing, it has challenges of its own. Documents such as discharge summaries, for example, are characterized by incomplete utterances and frequent use of medical language, which makes them unsuitable for many NLP tools that deal with more colloquial language. Furthermore, an expression's classification as PHI can be highly ambiguous depending on the context (Uzuner et al., 2008). A conservative approach which, when in doubt, always classifies an expression as a PHI is unaccaptable since it can remove potentially useful information from the document.

In this work, we describe the design and implementation of an automatic de-identifier for medical documents, and evaluate its performance on a large corpus of discharge summaries. We demonstrate that our de-identifier can effectively locate PHIs in free-text discharge summaries, achieving 97.37% precision and 93.52% recall, and outperforms humans both in accuracy and processing speed. We further evaluate the robustness of our solution through feature selection, and by varying the size of the training corpus. We show that performance improves as a function of the training set size, implying that even better results can be obtained by using a larger training set. We also perform SVM kernel selection, an aspect of de-identification not addressed in literature, demonstrating that kernels other than linear are effective for de-identification.

2 Problem specification

An automated de-identifier can be formalized as follows. Given a medical document $D = w_1, w_2, \ldots, w_n$, where w_i's are the document's words, and a classifier $f(w_j)$ st. $f(w_j) = +1$ if w_j is a PHI and $f(w_j) = -1$ otherwise, a de-identifier outputs $D_{de-id} = w'_1, w'_2 \ldots, w'_n$, where:

$$w'_j = \begin{cases} w_j & \text{if } f(w_j) = -1 \\ NULL & \text{otherwise} \end{cases} \tag{1}$$

$NULL$ is a special symbol meaning that a word has been removed. In other words, a de-identifier removes all PHI instances from the document on a per-word basis.

3 Related work

There exist a number of algorithms for de-identifying medical text, most of which can be classified as belonging to one of two categories: (a) rule-based and (b) statistical (Uzuner et al., 2007). An example of the former, Neamatullah et al developed a system which uses dictionary lookups, regular expressions and heuristics for PHI classification. The algorithm scans medical notes word-by-word, and relies on hard-coded rules to combine regular expression, context and dicitonary lookup information to decide whether a word is a PHI. The system exposes limited flexibility through user-swappable dictionaries (Neamatullah et al., 2008).

Gupta et al developed *De-Id*, an automatic de-identifier which uses a complex set of rules, dicitonaries, pattern-matching algorithms, and the Unified Medical Language System to identify and replace identifying text in clinical reports, while preserving medical information for use in research. Gupta et al evaluated De-Id on a set of 967 surgical pathology reports, where it failed to de-identify 103 accession numbers, 7 dates, 9 names, 25 initials and 46 hospital or laboratory names. It also erroneously removed non-PHI information from 15 reports (Gupta et al., 2004).

One of the most accurate statistical systems is *Stat De-id* developed by Uzuner et al. (Uzuner et al., 2008). Uzuner uses a multi-class, high-dimensional Support Vector Machine (SVM) trained on a large corpus of discharge summaries to identify PHIs. The Support Vector Machine operates on a vector representation of words in documents, where each vector dimension corresponds to a particular feature of a word. Supported features include punctuation, word length, bigram associations and part of speech (POS) tags, among others. Stat De-id achieves a precision and recall of

98.40% and 93.75%, respectively, on a corpus of authentic discharge summaries (Uzuner et al., 2008). The accuracy, while high, is not enough for all de-identification purposes. Stat De-id's performance as a function of corpus size was not evaluated, which makes it impossible to predict whether higher accuracy could be obtained through a larger training corpus. Furthermoe, Uzuner et al. only trained Stat De-id using a linear kernel, which may not be the optimal choice for the PHI/non-PHI classification problem.

Our de-identifier is an open-source re-implementation of Stat De-Id and achieves comparable accuracy on the same test set. We evaluate the system's performance as a function of the training set size, features used, as well as the choice of kernel in the underlying SVM implementation.

4 Approach

We define a *feature extractor*, denoted $\phi(w_j)$, which maps all words in a medical document onto a high dimensional vector representation $\in \mathbb{R}^d$. Each dimension in the resulting vector space corresponds to a specific feature of the word, such as part of speech, length, membership in medical dictionaries or syntactic role, among others (see Section 4.1 for a detailed description of all feature classes we use). The exact number of features d depends on training data but is usually above 100,000.

We use a two-class Support Vector Machine to classify words into the PHI $(+1)$ and non-PHI (-1) categories. Given the feature vector representation $\phi(w_j)$, the SVM uses the following classification rule:

$$f(w_j) = sign\left(\sum_{i=1}^{T} \alpha_i^* y_i K(\phi(w_j), \phi(v_i))\right) \quad (2)$$

where $v_1, v_2, \ldots, v_T$ is the training set, y_i is the $+1/-1$ label of v_i, and α_i^* for $i = 1, \ldots, T$ are the parameters of the model. $K(\phi(w_j), \phi v_i)$ is the *kernel*, which indirectly maps $\phi(w_j)$ and $\phi(v_i)$ onto a kernel-specific vector space by computing the inner product in that vector space.

We find the model parameters α_i^* by solving the following maximization problem:

$$\arg\max_{a_1,\ldots,a_T} \sum_{i=1}^{T} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{T} \alpha_i \alpha_j y_i y_j K(\phi(v_i), \phi(v_j))$$
$$\text{subject to } 0 \le \alpha_i \le C, i = 1, \ldots, T$$

where C is a parameter penalizing misclassifications.

Unless stated otherwise, in this work we use the linear kernel defined as:

$$K(\phi(w_j), \phi(v_i)) = \phi(w_j)^T \phi(v_i) \quad (3)$$

In Section 5.5, we additionally evaluate our model on the quadratic and Gaussian kernels. Respectively:

$$K(\phi(w_j), \phi(v_i)) = \left(\gamma \cdot \phi(w_j)^T \phi(v_i) + r\right)^2 \quad (4)$$
$$K(\phi(w_j), \phi(v_i)) = \exp(-\gamma \|\phi(w_j) - \phi(v_i)\|^2) \quad (5)$$

γ and r are adjustable kernel paremeters, whose optimal values depend on problem characteristics.

4.1 Feature Representation

The features used by the feature extractor $\phi(w_j)$ can be divided into three broad classes: lexical/orthographic, syntactic and semantic. This section describes each feature class and its contituent features in detail.

4.2 Lexical/Orthographic Features

4.2.1 The Word Itself

There are words which consistently appear as non-PHIs such as the conjunction "and" and the pronoun "they" (Uzuner et al., 2008). Similarly, some words such as common first names are almost always PHIs. We incorporate this knowledge into our feature representation by creating a binary indicator feature for each unique word we encounter in the training corpus.

4.2.2 Capitalization

Some PHI types such as names and locations are usually capitalized (Uzuner et al., 2008). We incorporate this knowledge by using a single binary indicator feature specifying whether a word is capitalized or not.

4.2.3 Punctuation

Numerical PHIs such as telephone numbers and dates can sometimes be identified with the help of punctuation (Uzuner et al., 2008). For example, the “-” in “617-123-4567” can help identify the string as a phone number. Similarly to capitalization, punctuation is represented as a binary indicator feature.

4.2.4 Numbers

PHIs such as IDs, dates and phone numbers usually contain numbers (Uzuner et al., 2008). For each word, we have an indicator feature specifying whether the word is a number.

4.2.5 Word Length

Phone numbers, social security numbers and other numerical entities usually have a fixed length, which can aid their identification as PHIs. To represent word length, we use L indicator features where L is the length of the longest word in the training corpus. We set the l-th length feature to 1 if the given word has length l.

4.2.6 Lexical Bigrams

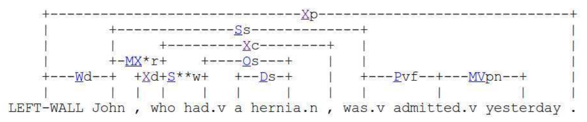
Sometimes PHIs can be identified using the immediate lexical context. For example, the phrase “was admitted” is usually preceded by the name of the patient. We incorporate lexical context into our feature representation by creating two indicator features per each bigram we encounter in the training corpus. For a given word, we set the corresponding bigram feature to 1 if the word is preceded/followed by that bigram.

4.3 Syntactic Features

4.3.1 Syntactic Bigrams

While lexical bigrams capture some of the context, they cannot deal with long-range dependencies. Consider for example the phrase “was admitted”, which is usually preceded by a patient’s name. In case of an immediate precedence, lexical bigrams can capture the dependency perfectly. However, they fail when there is another phrase in between the target PHI and the bigram of interest, as in “John, who had a hernia, was admitted yesterday” (Uzuner et al., 2008).

To deal with long-range dependencies, we use the *Link Grammar Parser* (Link Grammar, 2010).



```
    +------------------------------Xp------------------------------+
    |       +--------------Ss-------------+                        |
    |       |     +---------Xc--------+   |                        |
    |       +-MX*r+    +----Os---+    |   |                        |
    +---Wd--+  +Xd+S**w+   +--Ds-+    |   +---Pvf--+---MVpn--+     |
    |       |  |  |    |   |     |    |   |        |         |     |
LEFT-WALL John , who had.v a hernia.n , was.v admitted.v yesterday .
```

Figure 1: Output of the Link Grammar Parser for the sentence “John, who had a hernia, was admitted yesterday”. The parse tree correctly captures the long-range dependency between “John” and “was admitted”.

The parser extracts syntactic dependencies between parts of a sentence, enabling us to capture long-range links between words (Figure 1). We store such dependencies, for each word, as *syntactic bigrams* (Uzuner et al., 2008). That is, we store the two most immediate left and right syntactic dependencies, as indicated by the Link Grammar Parser.

4.3.2 Part of Speech

PHIs such as names and locations are generally more likely to be nouns than verbs (Uzuner et al., 2008). We incorporate this knowledge by having an indicator feature for each possible part of speech. The POS tagging is done by the OpenNLP Part-of-Speech tagger (openNLP, 2010).

4.4 Semantic Features

4.4.1 Section Headings

Some PHIs are more likely to appear in specific sections or under specific headings. For example, the “Discharge Date” heading is usually followed by a date (Uzuner et al., 2008). We incorporate heading information by having indicator features for each section we encounter in the training set.

4.4.2 MeSH ID

MeSH ID maps nouns to specialized medical categories such as diseases, treatments and tests, among others. There are 15 top-level categories, and subcategories up to a depth of 11 (Uzuner et al., 2008). Nouns which can be successfully mapped to a MeSH ID category are less likely to be PHIs. We incorporate MeSH ID information by having an indicator feature for each MeSH ID type.

4.4.3 Dictionary Information

In addition to MeSH ID, we use specialized dictionaries such as a dictionary of common names and locations. We incorporate dictionary information through indicator features specifying whether the target word is present in the given dicitionary.

4.5 Tokenization

Before we can use the feature extractor, we use the *tokenizer* to transform free-text documents into lists of words. The tokenizer is designed to work with the format of the Challenge Corpus (Section 5.1), using the provided annotations to detect word boundaries. The tokenizer also extracts basic information from the annotations, such as section names and PHI types, and passes them forward to the feature extractor.

4.6 De-identification

The *de-identifier* combines the feature extractor, the SVM classifier and the tokenizer into a single functional entity which can be used to remove PHI instances from free-text medical documents (Listing 1). The de-identifier loops through words in the document, and uses the classifier $f(w_i)$ to decide whether a given word is a PHI. If it is, it removes the word from the document and inserts a generic PHI marker ($NULL$) in its place. Such markers are then used to evaluate the accuracy of de-identification.

Listing 1: Pseudocode of the De-identifier. The de-identifier uses a word classifier $f(w_i)$ to replace PHI instances with the $NULL$ symbol

```
DEIDENTIFY(document):
    $\alpha_1^*, \ldots, \alpha_T^* =$ TRAIN($v_1, \ldots, v_T$)
    $w_1, \ldots, w_n =$ TOKENIZE(document)
    deid = []

    for $w_i$ in $w_1, \ldots, w_n$:
        if $f(w_i)|_{\alpha^*}$ == $-1$:
            deid.append($w_i$)
        else:
            deid.append($NULL$)

    return deid
```

5 Experiments

5.1 Corpus

We obtained a large corpus of 889 discharge summaries called the *Challenge Corpus* (Table 1). The summaries in the corpus have been manually de-identified and all PHIs replaced with realistic surrogates. The PHIs have also been tagged and assigned to one of the seventeen categories by a human expert (Uzuner et al., 2008).

Table 1: Statistics of the Chalenge Corpus, which we used for training and testing our system. The corpus contained 889 manually annotated hospital discharge summaries.

Category	Instances	Fraction of Total
Non-PHI	444,127	94.03%
Doctor	7,697	1.63%
Date	7,651	1.62%
Hospital	5,204	1.10%
ID	5,110	1.08%
Patient	1,737	0.37%
Location	518	0.11%
Phone	271	0.06%

We randomly split the corpus into a training and test set, with 669 (75%) and 220 (25%) documents, respectively. All results reported in this paper, unless noted otherwise, are based on these two sets.

5.2 Evaluation

To evaluate the performance of our system on the Challenge Corpus, we use precision, recall and f-measure - three accuracy metrics popular in literature (Uzuner et al., 2008; Uzuner et al., 2007). Precision is defined as the fraction of true PHIs in all the PHIs identified by the system. Recall, on the other hand, is the fraction of true PHIs in the PHIs identified by the system, out of all PHIs present in the gold standard. The f-measure combines precision and recall into a single number that can be used to compare different systems (Uzuner et al., 2008):

$$\text{f-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \qquad (6)$$

Precision, recall and f-measure for non-PHIs are defined analogously. For completeness, we also report the confusion matrix between PHIs and non-PHIs.

5.3 Accuracy

We evaluated the performance of our de-identifier on a corpus of 220 hospital discharge summaries. With all features enabled, our system achieved a PHI precision, recall and f-measure of 97.37%, 93.52%

Table 2: Confusion matrix for our de-identifier trained on 669 and evaluated on 220 discharge summaries from the Challenge Corpus.

		Actual	
		PHI	non-PHI
Predicted	PHI	9,013	625
	non-PHI	243	158,838

and 95.41%, respectively, and 99.61%, 99.85% and 99.73% for non-PHI tokens. Out of 159,081 detected non-PHIs, only 243 (0.15%) were false negatives, while only 625 out of 9,638 (6%) detected PHIs were false positives (Table 2). Our results compare favorably with Uzuner's Stat De-id, which achieved a PHI precision of 98.46%, recall of 95.24% and f-measure of 96.82% on the same dataset (Uzuner et al., 2007).

Our system outperforms an average human de-identifier, who achieves a recall of 81% on similar datasets (Neamatullah et al., 2008). We also outperform human de-identifiers in speed, with our system being able to process 1 document per second.

5.4 Feature selection

5.4.1 One Feature Class Enabled

We performed feature selection by training the system on the full training set with only one feature class enabled, and evaluating the resulting classifier on the test set (Figures 2 and 3). We learned that lexical bigram information contibuted the most to precision and recall, achieving a PHI precision and recall of 89.76% and 67.91%, respectively, and non-PHI precision and recall of 98.08% and 99.53%. Syntactic bigrams also exhibited good precision and a slightly worse recall. Other feature classes, such as section heading information, achieved a PHI precision and recall of 0%, which implies that they are not by themselves useful in PHI classification.

5.4.2 One Feature Class Disabled

In addition to the "one feature class enabled" selection described in Section 5.4.1, we trained the system on the full training set with all feature classes *except* one enabled (Figures 4 and 5). We discovered that disabling the "word itself" feature had the most detrimental result on precision, bringing it

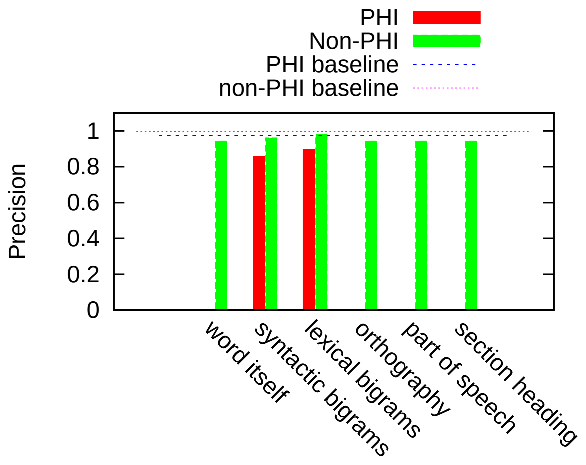


Figure 2: Precision of the de-identifier on the test corpus with only one feature class enabled. The baselines represent the system's precision with all features enabled.

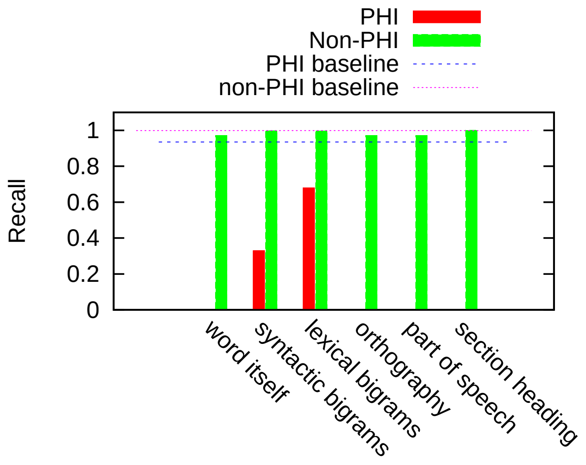


Figure 3: Recall of the de-identifier on the test corpus with only one feature class enabled. The baselines represent the system's recall with all features enabled.

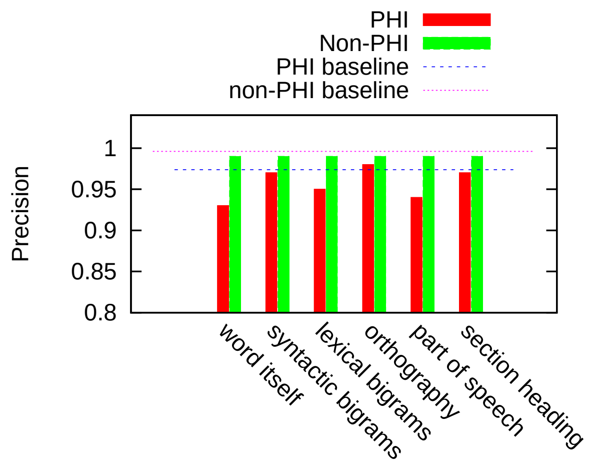


Figure 4: Precision of the de-identifier on the test corpus with all but one feature class enabled. The baselines represent the system's precision with all features enabled.

down from 97.37% to 93.02% for PHIs. Disabling "word itself" and orthographic features was the most detrimental for recall (degradation from 93.52% to 90.14% for PHIs). non-PHI precision and recall were not affected by disabling a single feature class.

5.5 Kernel Selection

We evaluated the performance of our de-identifier for different Support Vector Machine kernels - an aspect of SVM-based de-identification not adressed in literature. We performed kernel selection by training our system on a subset of the training set (50 randomly selected documents) with different kernels, and evaluating the system's performance on the full test set. Our results are shown in Table 3.

We discovered that in addition to the linear kernel, a quadratic and gaussian kernels work reasonably well. In particular, the gaussian kernel outperforms the linear kernel in terms of PHI precision, while sacrificing very little recall. This suggests that the fears of overfitting associated with kernels other than linear (Uzuner et al., 2008) are unwarranted.

5.6 Training Set Sensitivity

Knowing how the performance of the system improves with the size of the training set can help answer the question of whether even better accuracy could be attained with a larger training set. We trained our de-identifier on subsets of the full train-

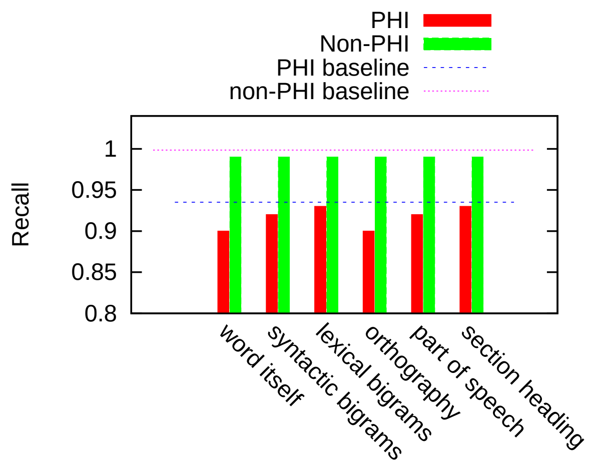


Figure 5: Recall of the de-identifier on the test corpus with all but one feature class enabled. The baselines represent the system's recall with all features enabled.

Table 3: Precision and recall of various kernels on the Challenge Corpus. The system was trained on a 50-document subset of the training set, and evaluated on the full test set.

kernel	PHI		non-PHI	
	Precision	Recall	Precision	Recall
linear	89.1%	79.7%	98.8%	99.4%
quadratic	98.7%	59.3%	97.6%	99.9%
gaussian	95.0%	74.2%	98.5%	99.8%

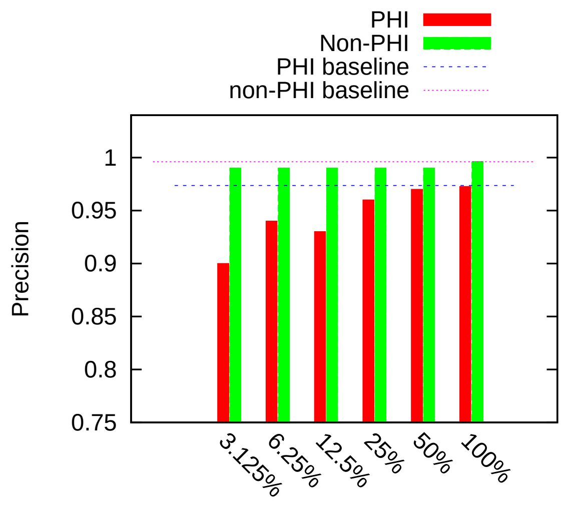


Figure 6: Precision of the de-identifier trained on sets of different sizes, where 100% corresponds to the full training set of the Challenge Corpus. Precision improves as the size of the training set increases. Baselines show the system's performance on the full training set.

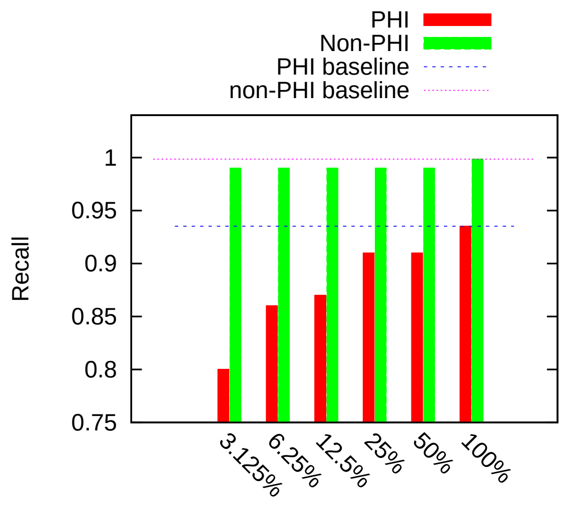


Figure 7: Recall of the de-identifier trained on sets of different sizes, where 100% corresponds to the full training set of the Challenge Corpus. Recall improves as the size of the training set increases. Baselines show the system's performance on the full training set.

ing set ranging from 3.125% to 100% of total. The results are reported in Figures 6 and 7.

Both precision and recall of our de-identifier improved with the size of the training set. While the effect is stronger for recall than for precision, the data implies that we could attain a higher precision and recall with a larger training set. Unfortunately, to our knowledge, no larger annotated corpora of discharge summaries are available.

6 Conclusions

Automatic de-identification of medical records is possible, and it works well. We implemented a de-identifier for discharge summaries based on Support Vector Machines. We evaluated our system on a large corpus of hospital discharge summaries, achieving a PHI precision and recall of 97.37% and 93.52%, respectively.

We further evaluated the system through feature and kernel selection. We found that there are feature classes such as lexical and syntactic bigrams which have the greatest impact on the performance of de-identification. We also demonstrated that different kernel types can be used successfully, achieving an overall performance comparable to the linear kernel and outperforming it on individual measures such as precision and/or recall.

We also measured the sensitivity of our de-identifier to the size of the training set. We demonstrated that precision and recall improve as the size of the training set increases, implying that even better accuracy could be achieved with a larger corpus.

A Implementation Details

We have open-sourced our implementation and made it available at `http://people.csail.mit.edu/tafvizi/6.864/deidentifier.tar.gz`.

References

[Neamatullah et al. 2008] Ishna Neamatullah, Margaret M. Douglass, Li-wei H Lehman, Andrew Reisner, Mauricio Villarroel. 2008. *Automatic de-identification of free-text medical records*. BMC Medical Informatics and Decision Making 8.1: 32

[Annas 2003] George J. Annas, J.D., M.P.H. 2003. *HIPAA Regulations - A New Era of Medical-Record Privacy?* The New England Journal of Medicine; 348:1486-1490

[Uzuner et al. 2008] Ozlem Uzuner, Tawanda C. Sibanda, Yuan Luo, Peter Szolovits. 2008. *A de-identifier for medical discharge summaries*. Artificial Intelligence in Medicine 42, p. 13-35

[Uzuner et al. 2007] Ozlem Uzuner, Yuan Luo, Peter Szolovits. 2006. *Evaluating the State-of-the-Art in Automatic De-identification*. Journal of the American Medical Informatics Association

[Gupta et al. 2004] Dilip Gupta, MD, Melissa Saul, John Gilbertson, MD 2004. *Evaluation of a De-identification (De-Id) Software Engine to Share Pathology Reports and Clinical Documents for Research* . American Journal of Clinical Pathology, 121, 176-186

[HHS.gov 2010] U.S. Department of Health and Human Services. 2010. *The Health Insurance Portability and Accountability Act of 1996 (HIPAA) Privacy and Security Rules*. `http://www.hhs.gov/ocr/privacy`, retrieved November 29, 2010

[openNLP 2010] The openNLP Toolkit. 2010. `http://opennlp.sourceforge.net`, retrieved December 10, 2010

[Link Grammar 2010] The Link Grammar Parser. 2010. `http://www.link.cs.cmu.edu/link/`, retrieved December 10, 2010