De-identification of health records

Anurag Beniwal | SI 671 | December 17, 2015

[1]Background

The very popular Hippocratic oath states that "…All that may come to my knowledge in the exercise of my profession or in daily commerce with men, which ought not to be spread abroad, I will keep secret and will never reveal."

With the increased use and availability of healthcare data for research by clinicians and public health researchers, protecting patient confidentiality has become more and more important and unavoidable.

In the United States, the Health Insurance Portability and Accountability Act (HIPAA; codified as 45 CFR §160 and 164) protects the confidentiality of patient data, and the Common Rule [2] protects the confidentiality of research subjects. These laws typically require the informed consent of the patient and approval of the Internal Review Board (IRB) to use data for research purposes, but these requirements are waived if data is de-identified, or if patient consent is not possible (e.g., data mining of retrospective records). For clinical data to be considered de-identified, the HIPAA "Safe Harbor" technique requires 18 data elements.

The list of all 18 data elements is present in Annexure 1

Literature review:

The review included reading several papers on de-identification techniques in general and also particular to the healthcare data. "Automatic de-identification of textual documents in electronic health records: A review " by Meystre , Fhriedlin , South,Shen and Samore served as a retrospective guide to the past work done in this area .

The summary of results, techniques used and author names is present in Annexure 2 .

In the Machine learning front, the tutorial on Conditional random fields by Sutton was the primary reference.

[2]

[1] http://goo.gl/mrtPLW

[2] https://people.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf

Data collection:

The data used for this project is adopted from NLP research datasets from Informatics for Integrating Biology & the bedside (i2b2) (https://www.i2b2.org/NLP/DataSets/Download.php)

The data contains discharge summaries of the patients who got treated in the hospital.The data was obtained after signing on a data use agreement which can be found in the Annexure 3 .

The dataset is in XML format.

The following named entities are tagged in the training set and we are going to focus on only these entities:

- Doctor
- Hospital name
- Patient name
- Date
- Phone
- Patient ID

Corpus statistics:

Number of sentences:

Train data: 52600

Test data: 18500

Number of word tokens:

Train data: 386052

Test data: 162796

Data processing:

Data preprocessing is an important task in any data mining project , however it becomes even more important to be extra cautious while cleaning text data due to its unstructured nature .

The training data has the discharge summaries of patients from different hospitals. The HTML tags (labels) represent the named entity labels in the training dataset . <PHI TYPE="PATIENT">Blind</PHI> . The PHI here stands for Protected Health Information.

Data processing Steps:

1) **Parsing the XML file**

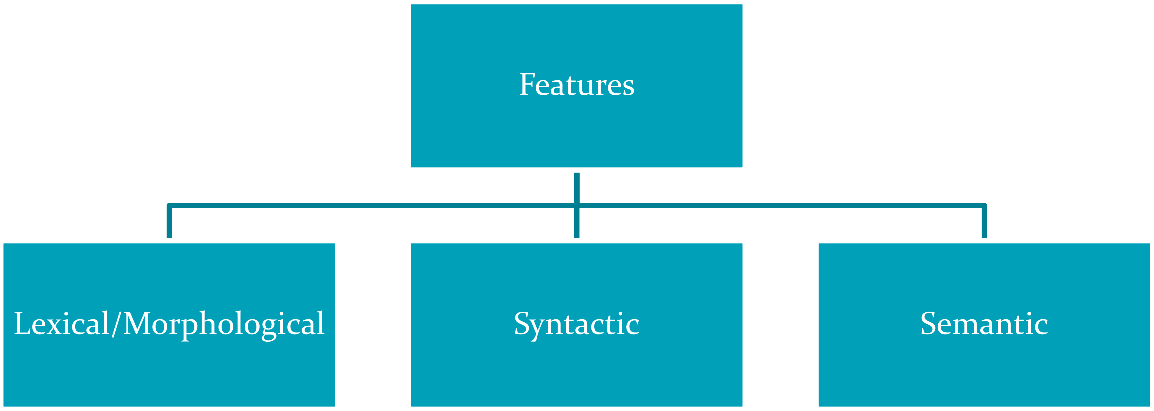
The XML file is parsed using beautiful soup library in Python . The library does a good job of poarsing improper XML files. The file is then converted into text format .

2) **Tokenization of the discharge summaries**

The paragraphs are tokenized into sentences. However , the tokenization of discharge summaries at different levels of abstractions does impact the accuracy of classification especially when using machine learning techniques that use the syntactic and semantic structure of the text to classify (for ex. HMM , Conditional random field etc.) . This is an important research question which is interesting to explore. The summary can be tokenization can be split at one level and the information of other level of abstractions can be included in the form of features.In this case, I have split the summary based on the sentences (Considering a line ending with a "." as a sentence.

3) After tokenization, sentences with tags are identified and the labels are enclosed in tuples with the word (word, label), removing all other text within the label that is not useful. The words that are not named entities are assigned the label "Other"
4) The spaces between words within the labels are replaced with special characters so that they don't split apart while tokenizing a sentence into a word.
5) The stop words are not removed as they may be helpful in predicting if a word is a named entity or not
6) However, blank spaces are removed from the sentences
7) Part of speech tagging is done using perceptron tagger in the NLTK library.
8) The part of speech tag of the word is merged in the tuple of the words having the word and the label. Now the tuple looks like (word, label , pos-tag)
9) Paragraph information is added to the tuple. For ex. (xyz , PATIENT , NN , HISTORY). However, this comes with some inaccuracies as the data is not explicitly divided into paragraphs. It is assumed that the text after a paragraph heading belongs to that heading until another heading is encountered.

Feature creation:



1) Lexical features :
   - Word
   - Surrounding words
   - Words Length
   - Regular expressions for labels
   - Sentence length
   - Sentence position
   - Capitalization
   - Number of words within a label
   - N grams
2) Syntactic features :
   - POS tags of the word
   - POS tag of the previous and the successive words
   - The sentence structure and hierarchy
3) Semantic features:

The semantic features include dictionary terms from various dictionaries, medical dictionaries and other medical databases. However, these features have not been included in the current model.

Technique:

Many of the studies in the past have used pattern matching/rule based approaches for de-identifying health documents. Use of medical dictionaries have also been common. The techniques work really well on the data they have been trained on.A detailed description of the pattern matching techniques used and the authors have been provided in the appendix 4.

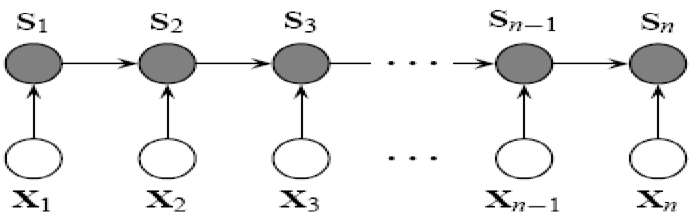
Due to increasing veracity in the healthcare data, it is impossible to build a custom de-identification tools for every source of such data. Need for more and more general tools are needed that can de-identify data from multiple sources with a reasonable accuracy. These approaches may not do as well as pattern matching approaches but are highly generalizable.

This was the motivation for keeping the scope of this work focused towards approaches that can be generalized across datasets and corpuses. A combination of patterns (not too specific to a dataset) and Machine learning techniques quite well .In the current work, I have used Conditional random fields as the machine learning technique as it leverages the lexical and syntactic structure of the data really well and is more generalizable.

What are conditional random fields?

CRFs are a type of discriminative undirected probabilistic graphical model. It is used to encode known relationships between observations and construct consistent interpretations. It is often used for labeling or parsing of sequential data, such as natural language text or biological sequences and in computer vision.

Conditional random fields (CRFs) are a class of statistical modelling method often applied in pattern recognition and machine learning, where they are used for structured prediction. Whereas an ordinary classifier predicts a label for a single sample without regard to "neighboring" samples, a CRF can take context into account; e.g., the linear chain CRF popular in natural language processing predicts sequences of labels for sequences of input samples.



CRF’s in Named Entity Recognition

Linear chain CRF’s have been widely used in Part of speech tagging and in NER recently due to its simplicity and superior performance.

The states of the CRF in our context are the named entities and the output are the words. This is a discriminative approach of modelling and the states are predicted using the outputs and the previous states.

Current work:

In this project, we train linear chain CRF using the python crf-suite library. In order to leverage the power of pattern matching techniques, few patterns which are observed ubiquitously in datasets are used as features.

INPUT PARAMETERS

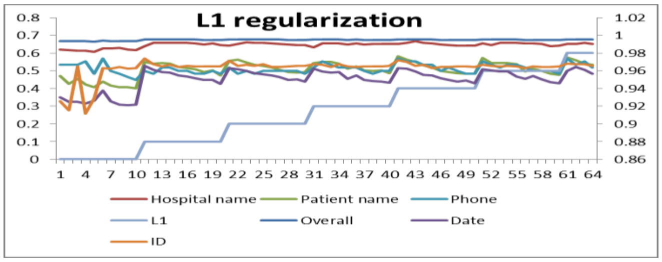
The input parameters used to train the model include:

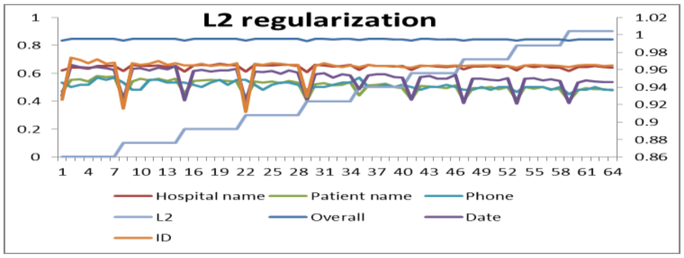
1) L1 and L2 regularization penalties :

Due to large number of features in the model, there are high chances that many of the features provide overlapping information, this makes the model over fitted and less general.

In order to penalize the redundant information the impact of each feature is penalized by L1 or L2 penalty.

The L1 and L2 penalty values were determined by validation on the test dataset.





The parameters used in the final model are

L1 = 0.5 L2 = 1e-03

OPTIMIZATION PROCEDURE:

L-BFGS algorithm was used for optimization of the likelihood function.

Other optimization algorithms were not tried under the current scope, however it is a potential area of research. L-BFGS is used commonly in CRF's.

Overview:

In numerical optimization, the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm is an iterative method for solving unconstrained nonlinear optimization problems.The BFGS method approximates Newton's method, a class of hill-climbing optimization techniques that seeks a stationary point of a (preferably twice continuously differentiable) function. For such problems, a necessary condition for optimality is that the gradient be zero. Newton's method and the BFGS methods are not guaranteed to converge unless the function has a quadratic Taylor expansion near an optimum. These methods use both the first and second derivatives of the function. However, BFGS has proven to have good performance even for non-smooth optimizations (Source : Wikipedia)

NUMBER OF ITERATIONS:

The number of iterations was kept to 100 given the limitations of computational power.

Results

Named Entity	Accuracy (%)
Overall	99.66
Hospital	74.43%
Doctor	99.9%
Patient	72.2%
Date	95.78%
Phone	94.82%
ID	98%

Observations:

- The overall accuracy is good but is of little use as the number of labels is very less as compared to the total number of words in the document.
- The accuracy for hospital names is about 75% which may be due to huge heterogeneity in the names of hospitals w.r.t length, capitalizations etc.
- The accuracy for Doctors is high, may be because the words adjacent to the name of doctors are powerful in predicting the label for ex. Keywords like Dr. etc.
- The patient accuracy is as bad as hospital mainly due to heterogeneity in the names
- Dates/ Phone/ID have a good accuracy may be because of inclusion of few general regular expressions for them

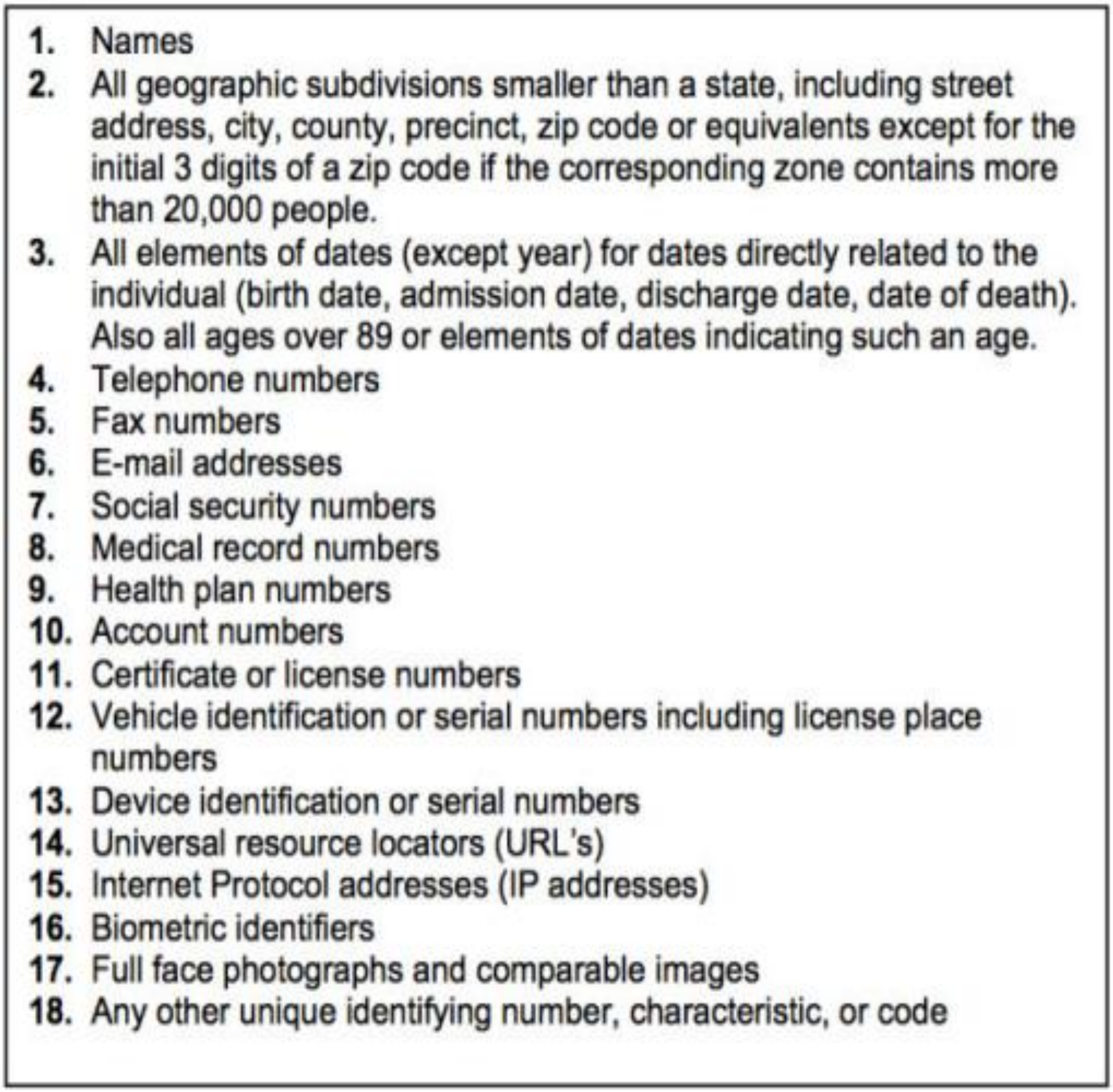
Future work

Current study had a limited scope due to time and knowledge constraints. This study has provided me with a great deal of idea about handling text data , feature engineering and behavior of different NER levels which is bound to serve as a strong foundation for future research in this area.

Next steps:

1) Leveraging medical dictionaries for improving accuracy
2) Understanding CRF in more detail and making changes to the existing learning algorithms to build custom CRF for NER applications
3) Create features at different abstractions of data to improve predictions
4) Fit the model on corpuses from multiple sources
5) Compare the results with existing toolkits like the Stanford NER tagger
6) Trying alternative machine learning techniques like SVM

Annexure 1 - The 18 elements necessary to be de-identified



1. Names
2. All geographic subdivisions smaller than a state, including street address, city, county, precinct, zip code or equivalents except for the initial 3 digits of a zip code if the corresponding zone contains more than 20,000 people.
3. All elements of dates (except year) for dates directly related to the individual (birth date, admission date, discharge date, date of death). Also all ages over 89 or elements of dates indicating such an age.
4. Telephone numbers
5. Fax numbers
6. E-mail addresses
7. Social security numbers
8. Medical record numbers
9. Health plan numbers
10. Account numbers
11. Certificate or license numbers
12. Vehicle identification or serial numbers including license place numbers
13. Device identification or serial numbers
14. Universal resource locators (URL's)
15. Internet Protocol addresses (IP addresses)
16. Biometric identifiers
17. Full face photographs and comparable images
18. Any other unique identifying number, characteristic, or code

Annexure 2 – Characteristics of the existing de-identification systems

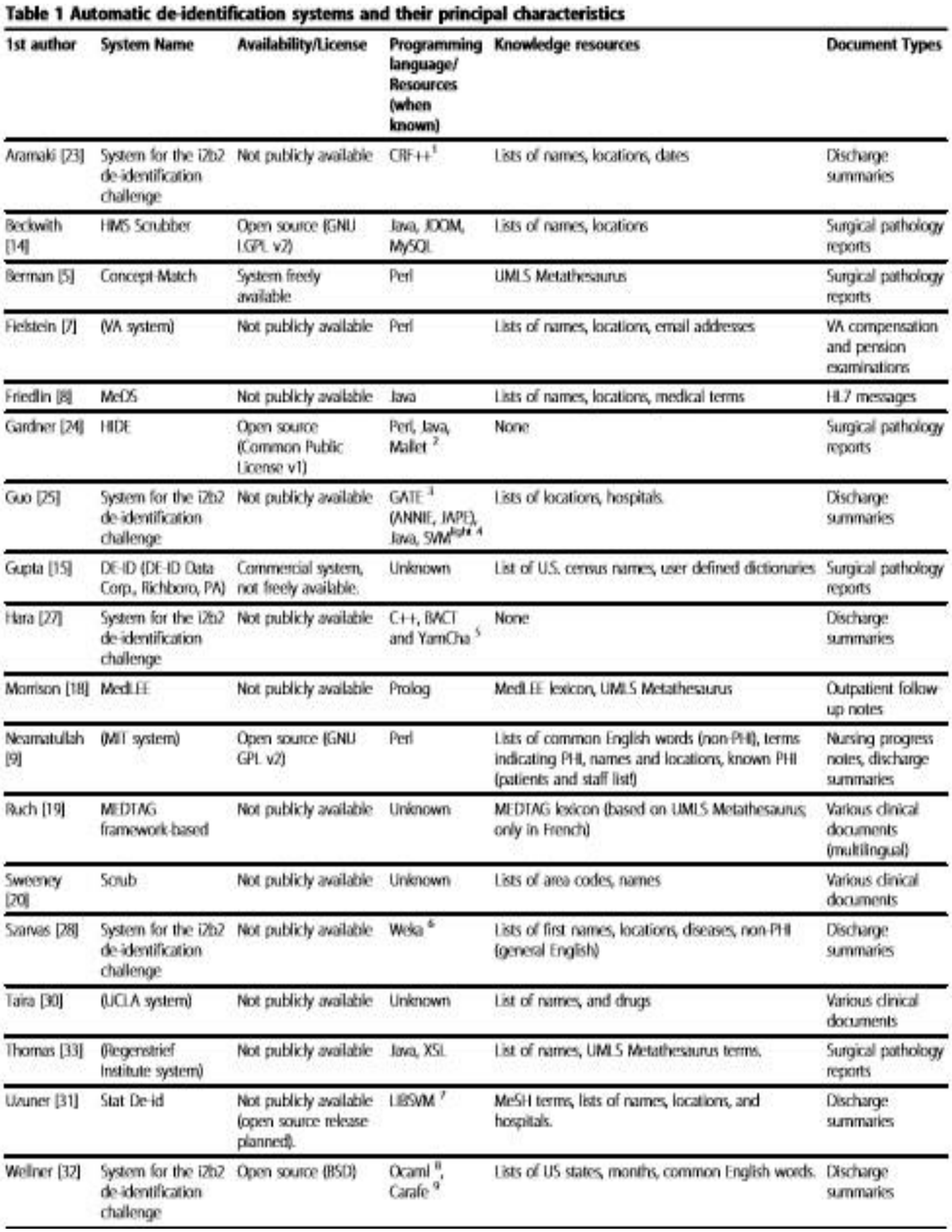


Table 1 Automatic de-identification systems and their principal characteristics

1st author	System Name	Availability/License	Programming language/ Resources (when known)	Knowledge resources	Document Types
Aramaki [23]	System for the i2b2 de-identification challenge	Not publicly available	CRF++^{1}	Lists of names, locations, dates	Discharge summaries
Beckwith [14]	HMS Scrubber	Open source (GNU LGPL v2)	Java, JDOM, MySQL	Lists of names, locations	Surgical pathology reports
Berman [5]	Concept-Match	System freely available	Perl	UMLS Metathesaurus	Surgical pathology reports
Fielstein [7]	(VA system)	Not publicly available	Perl	Lists of names, locations, email addresses	VA compensation and pension examinations
Friedlin [8]	MeDS	Not publicly available	Java	Lists of names, locations, medical terms	HL7 messages
Gardner [24]	HIDE	Open source (Common Public License v1)	Perl, Java, Mallet ^{2}	None	Surgical pathology reports
Guo [25]	System for the i2b2 de-identification challenge	Not publicly available	GATE ^{3} (ANNIE, JAPE), Java, SVM^{light} ^{4}	Lists of locations, hospitals.	Discharge summaries
Gupta [15]	DE-ID (DE-ID Data Corp., Richboro, PA)	Commercial system, not freely available.	Unknown	List of U.S. census names, user defined dictionaries	Surgical pathology reports
Hara [27]	System for the i2b2 de-identification challenge	Not publicly available	C++, BACT and YamCha ^{5}	None	Discharge summaries
Morrison [18]	MedLEE	Not publicly available	Prolog	MedLEE lexicon, UMLS Metathesaurus	Outpatient follow up notes
Neamatullah [9]	(MIT system)	Open source (GNU GPL v2)	Perl	Lists of common English words (non-PHI), terms indicating PHI, names and locations, known PHI (patients and staff list)	Nursing progress notes, discharge summaries
Ruch [19]	MEDTAG framework-based	Not publicly available	Unknown	MEDTAG lexicon (based on UMLS Metathesaurus; only in French)	Various clinical documents (multilingual)
Sweeney [20]	Scrub	Not publicly available	Unknown	Lists of area codes, names	Various clinical documents
Szarvas [28]	System for the i2b2 de-identification challenge	Not publicly available	Weka ^{6}	Lists of first names, locations, diseases, non-PHI (general English)	Discharge summaries
Taira [30]	(UCLA system)	Not publicly available	Unknown	List of names, and drugs	Various clinical documents
Thomas [33]	(Regenstrief Institute system)	Not publicly available	Java, XSL	List of names, UMLS Metathesaurus terms.	Surgical pathology reports
Uzuner [31]	Stat De-id	Not publicly available (open source release planned).	LIBSVM ^{7}	MeSH terms, lists of names, locations, and hospitals.	Discharge summaries
Wellner [32]	System for the i2b2 de-identification challenge	Open source (BSD)	Ocaml ^{8}, Carafe ^{9}	Lists of US states, months, common English words.	Discharge summaries

Annexure 3 – PHI targeted by each de-identification system

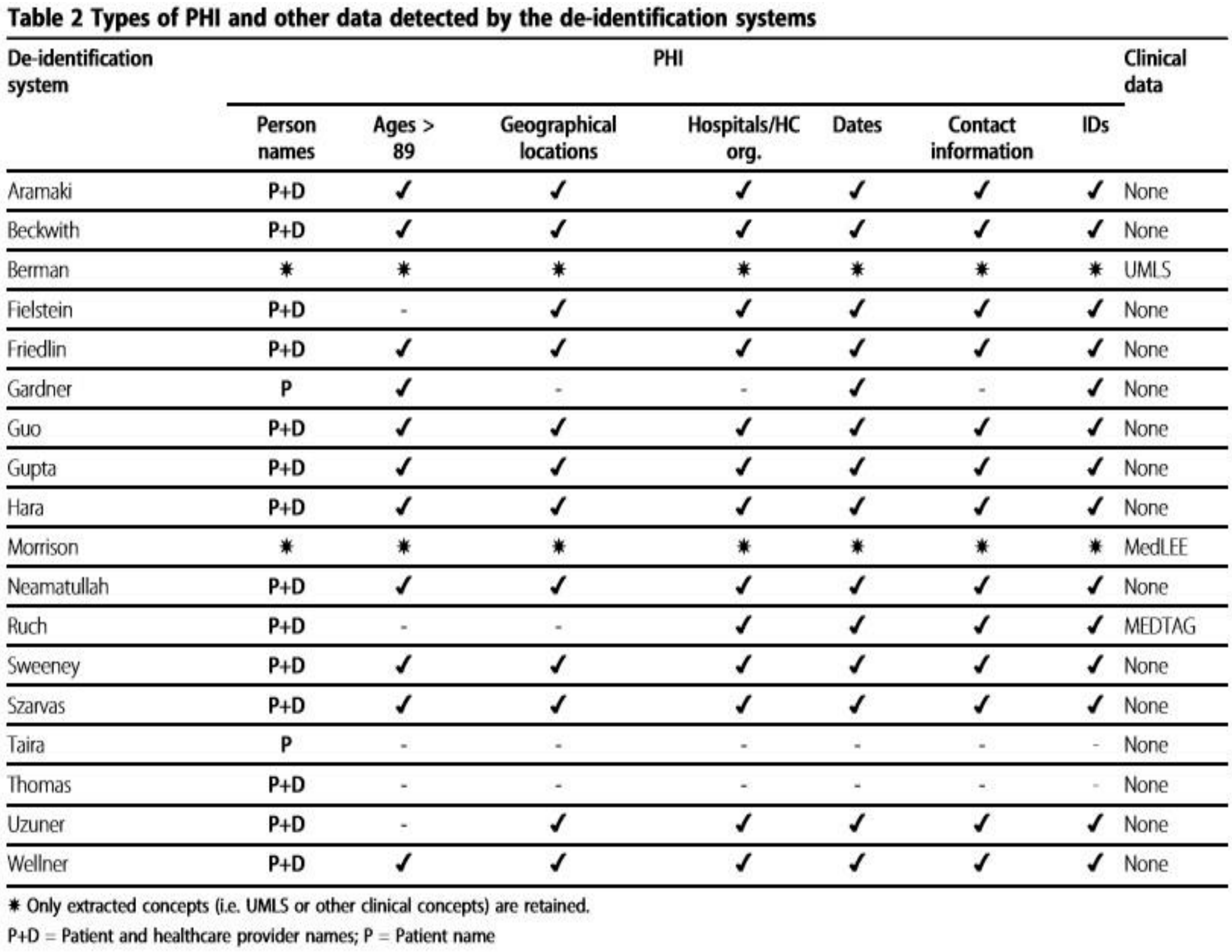


Table 2 Types of PHI and other data detected by the de-identification systems

De-identification system	PHI							Clinical data
	Person names	Ages > 89	Geographical locations	Hospitals/HC org.	Dates	Contact information	IDs	
Aramaki	P+D	✓	✓	✓	✓	✓	✓	None
Beckwith	P+D	✓	✓	✓	✓	✓	✓	None
Berman	✱	✱	✱	✱	✱	✱	✱	UMLS
Fielstein	P+D	-	✓	✓	✓	✓	✓	None
Friedlin	P+D	✓	✓	✓	✓	✓	✓	None
Gardner	P	✓	-	-	✓	-	✓	None
Guo	P+D	✓	✓	✓	✓	✓	✓	None
Gupta	P+D	✓	✓	✓	✓	✓	✓	None
Hara	P+D	✓	✓	✓	✓	✓	✓	None
Morrison	✱	✱	✱	✱	✱	✱	✱	MedLEE
Neamatullah	P+D	✓	✓	✓	✓	✓	✓	None
Ruch	P+D	-	-	✓	✓	✓	✓	MEDTAG
Sweeney	P+D	✓	✓	✓	✓	✓	✓	None
Szarvas	P+D	✓	✓	✓	✓	✓	✓	None
Taira	P	-	-	-	-	-	-	None
Thomas	P+D	-	-	-	-	-	-	None
Uzuner	P+D	-	✓	✓	✓	✓	✓	None
Wellner	P+D	✓	✓	✓	✓	✓	✓	None

✱ Only extracted concepts (i.e. UMLS or other clinical concepts) are retained.

P+D = Patient and healthcare provider names; P = Patient name

Annexure 4 : Pattern based and Machine learning based systems

Pattern Matching based systems

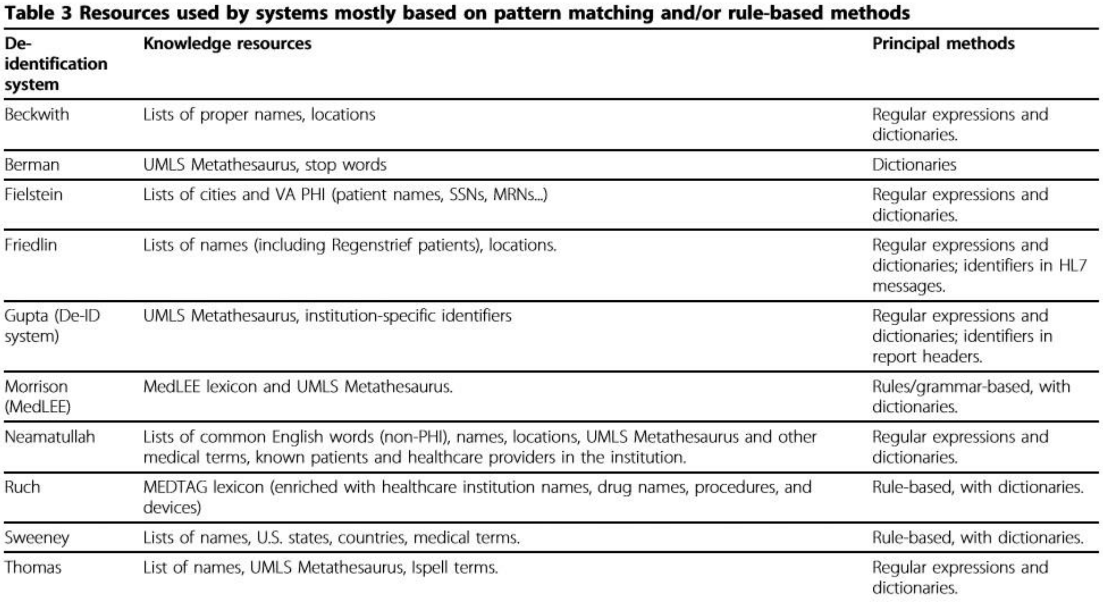


Table 3 Resources used by systems mostly based on pattern matching and/or rule-based methods

De-identification system	Knowledge resources	Principal methods
Beckwith	Lists of proper names, locations	Regular expressions and dictionaries.
Berman	UMLS Metathesaurus, stop words	Dictionaries
Fielstein	Lists of cities and VA PHI (patient names, SSNs, MRNs...)	Regular expressions and dictionaries.
Friedlin	Lists of names (including Regenstrief patients), locations.	Regular expressions and dictionaries; identifiers in HL7 messages.
Gupta (De-ID system)	UMLS Metathesaurus, institution-specific identifiers	Regular expressions and dictionaries; identifiers in report headers.
Morrison (MedLEE)	MedLEE lexicon and UMLS Metathesaurus.	Rules/grammar-based, with dictionaries.
Neamatullah	Lists of common English words (non-PHI), names, locations, UMLS Metathesaurus and other medical terms, known patients and healthcare providers in the institution.	Regular expressions and dictionaries.
Ruch	MEDTAG lexicon (enriched with healthcare institution names, drug names, procedures, and devices)	Rule-based, with dictionaries.
Sweeney	Lists of names, U.S. states, countries, medical terms.	Rule-based, with dictionaries.
Thomas	List of names, UMLS Metathesaurus, Ispell terms.	Regular expressions and dictionaries.

Machine learning based systems

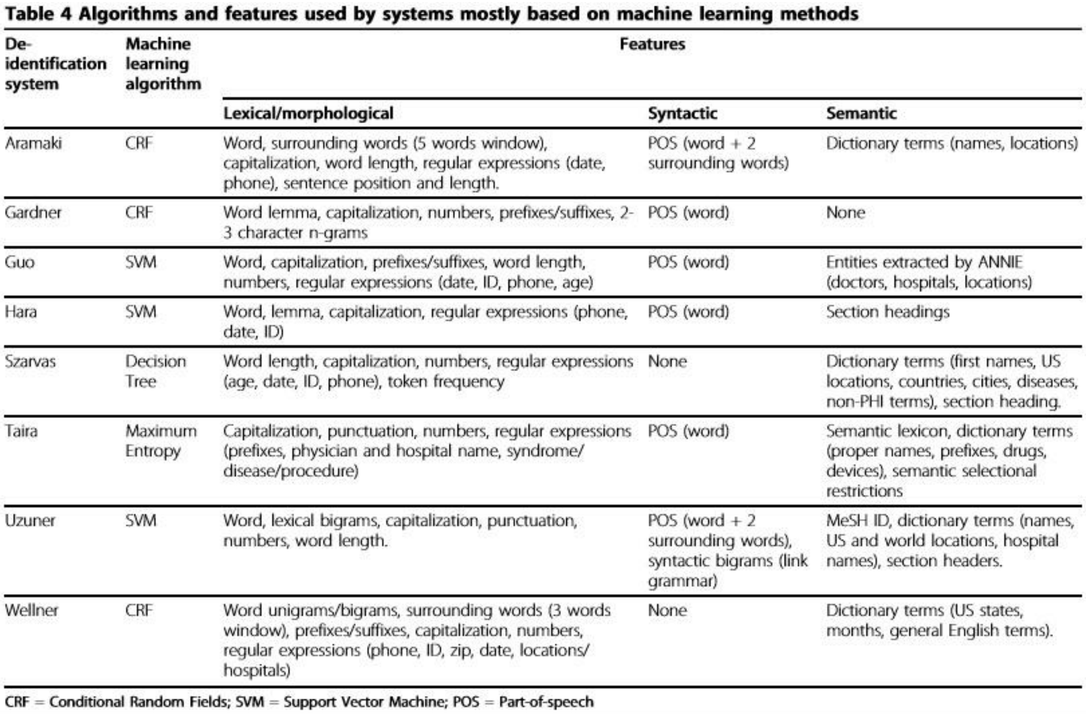


Table 4 Algorithms and features used by systems mostly based on machine learning methods

De-identification system	Machine learning algorithm	Features		
		Lexical/morphological	Syntactic	Semantic
Aramaki	CRF	Word, surrounding words (5 words window), capitalization, word length, regular expressions (date, phone), sentence position and length.	POS (word + 2 surrounding words)	Dictionary terms (names, locations)
Gardner	CRF	Word lemma, capitalization, numbers, prefixes/suffixes, 2-3 character n-grams	POS (word)	None
Guo	SVM	Word, capitalization, prefixes/suffixes, word length, numbers, regular expressions (date, ID, phone, age)	POS (word)	Entities extracted by ANNIE (doctors, hospitals, locations)
Hara	SVM	Word, lemma, capitalization, regular expressions (phone, date, ID)	POS (word)	Section headings
Szarvas	Decision Tree	Word length, capitalization, numbers, regular expressions (age, date, ID, phone), token frequency	None	Dictionary terms (first names, US locations, countries, cities, diseases, non-PHI terms), section heading.
Taira	Maximum Entropy	Capitalization, punctuation, numbers, regular expressions (prefixes, physician and hospital name, syndrome/disease/procedure)	POS (word)	Semantic lexicon, dictionary terms (proper names, prefixes, drugs, devices), semantic selectional restrictions
Uzuner	SVM	Word, lexical bigrams, capitalization, punctuation, numbers, word length.	POS (word + 2 surrounding words), syntactic bigrams (link grammar)	MeSH ID, dictionary terms (names, US and world locations, hospital names), section headers.
Wellner	CRF	Word unigrams/bigrams, surrounding words (3 words window), prefixes/suffixes, capitalization, numbers, regular expressions (phone, ID, zip, date, locations/hospitals)	None	Dictionary terms (US states, months, general English terms).

CRF = Conditional Random Fields; SVM = Support Vector Machine; POS = Part-of-speech

References

- http://bmcmedresmethodol.biomedcentral.com/articles/10.1186/1471-2288-10-70
- https://people.cs.umass.edu/~mccallum/papers/crf-tutorial.pdf
- http://www.chokkan.org/software/crfsuite/
- https://www.i2b2.org/NLP/DataSets/DataSheets/2006.php
- http://mistdeid.sourceforge.net/
- http://cogcomp.cs.illinois.edu/page/software_view/MedNER
- http://vis.stanford.edu/projects/adept/
- http://www.ucdmc.ucdavis.edu/compliance/guidance/privacy/deident.html
- http://www.hhs.gov/ocr/privacy/hipaa/understanding/coveredentities/De-identification/guidance.html