Patterns, Volume *2*

Supplemental information

Building a best-in-class automated de-identification tool for electronic health records through ensemble learning

Karthik Murugadoss, Ajit Rajasekharan, Bradley Malin, Vineet Agarwal, Sairam Bade, Jeff R. Anderson, Jason L. Ross, William A. Faubion Jr., John D. Halamka, Venky Soundararajan, and Sankar Ardhanari

Supplementary Methods

Ensemble Architecture Implementation details

We employed the *bert-base-cased* model (https://huggingface.co/bert-base-cased) through the HuggingFace/Transformers (https://github.com/huggingface/transformers) library. This is a case-sensitive English language pre-trained model based off of the BERT architecture trained using a masked language modelling (MLM) objective. The BERT model was pretrained on BookCorpus (https://huggingface.co/datasets/bookcorpus), a dataset comprising 11,038 unpublished books in addition to English Wikipedia.

Our ensemble involved employing at least one individual model for names, organizations, locations and ages. An additional *text normalized* model was also trained and utilized for names. Here, text normalization refers to the process of converting all uppercase words to title case (lowercase words are retained as is). A total of 61,800 tagged example sentences were used for fine-tuning the models. The final number of examples for each entity type is shown in **Supplementary Table 1**.

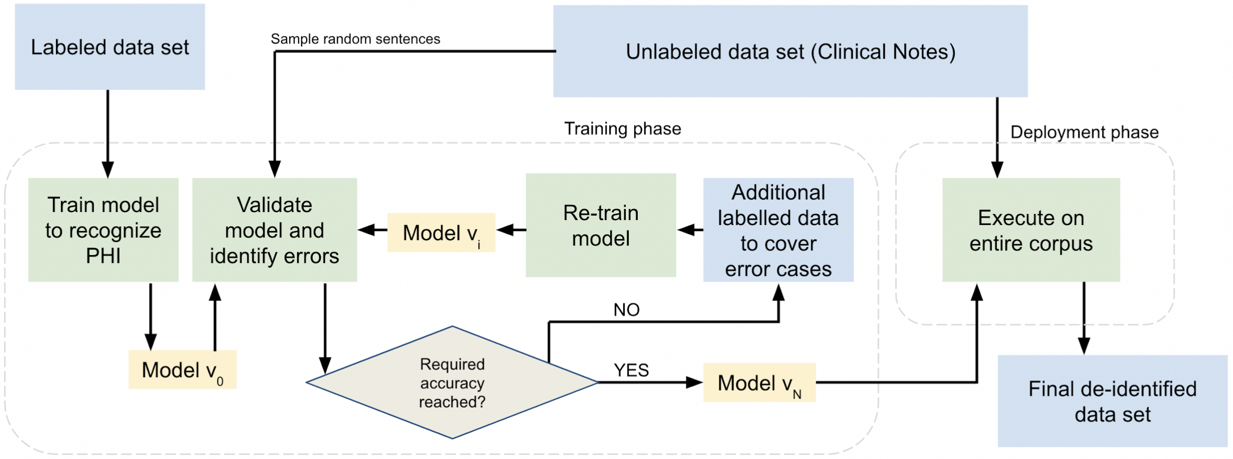
Model Priority #	Entity Type	Text Normalized?	Fine-tuning Examples
1	Name	No	44,929
2	Name	Yes	44,929
3	Location	No	11,461
4	Age	No	5,409
5	Organization	No	44,825

Supplementary Table 1: *BERT models employed in our ensemble and the corresponding entity type and number of fine-tuning examples. The Model Priority # denotes the order of precedence in the event that a word is tagged as PII by multiple models. For example, if a word is tagged as both a name and a location, it will be assigned the name entity (which has higher priority).*

Each transformer model is fine-tuned with a maximum sequence length of 256 (after tokenization) over 4 epochs. We use a training batch size of 32 and a learning rate of 5e-5 with a warmup proportion of 0.4. The Adam optimization algorithm was employed to update network weights. Loss was computed using cross entropy loss.

Each model is iteratively fine-tuned with training samples being continuously added to the initial set of training samples. The sentences chosen for fine-tuning the model are specifically selected from the space of errors that was seen in prior models. The iterative process of fine-tuning models therefore results in the generation of multiple individual neural networks (different versions) for each PII type each having a specific performance. To maximize the overall recall, we choose the two best performing models for each entity type and employ them in tandem.

To complement the above improvements on model architecture and algorithms for de-identification, an iterative learning framework is deployed in tandem that allows rapid validation and performance evaluation for trained models (**Supplementary Figure 1***)*. This allows each component of the ensemble framework to be re-trained and fine-tuned to learn from previous mistakes independently of other models.


Labeled data set
Sample random sentences
Unlabeled data set (Clinical Notes)
Training phase
Deployment phase
Train model to recognize PHI
Validate model and identify errors
Model v_i
Re-train model
Additional labelled data to cover error cases
Execute on entire corpus
NO
Model v_0
Required accuracy reached?
YES
Model v_N
Final de-identified data set

Supplementary Figure 1: *Iterative model generation process and learning from errors. Model performance improves during its evolution from v0 to vN.*

All of our experiments were performed on an Ubuntu 16.04 machine (12 CPU cores and 220GB RAM) with two NVIDIA Tesla V100 GPUs (16GB of RAM each). We used Python v3.6.9 with PyTorch v1.3.1 and pytorch-pretrained-bert v0.6.1 (now HuggingFace/Transformers). We first perform sentence tokenization to convert documents into sentences. On two GPUs, our system achieved an inference speed of 53 sentences per second (inference batch size was set to 128 and maximum sequence length was 256). Additionally, fine-tuning an individual model of our ensemble took 45 minutes for ~44k sentences with both GPUs being utilized.

In order to maximize recall of our ensemble, we employ a voting ensemble scheme across models of different entity types with a voting threshold of 1. That is, a word is determined to be PII if it is detected by at least one model. If a word is detected as PII by more than one model, it is assigned an entity type based on its priority (as described in **Supplementary Table 1**).

Creating an inclusion list of sentences

In a repository of 103 million physician notes (from 477,000 patients) from the Mayo Clinic, a total of approximately 3.1 billion sentences corresponded to approximately 700 million unique sentences, which highlights the redundancy in a corpus of this size and provides optimization opportunities in the de-identification processing pipeline. In particular, sentences with high prevalence were found to typically not contain PII (since they occur across a large number of patients, the chances that they contain information specific to any one patient is low). We computed the prevalence of all sentences and found that the top 1,600 most common sentences correspond to 1.01 billion sentences overall (one-third of the entire corpus).

These 1,600 sentences represented the initial inclusion list. Additionally, we filtered out the top 25,000 most prevalent sentences that contain a disease or a drug entity. This ensures that medically relevant sentences that are also highly prevalent are preserved. All of the sentences that are part of the inclusion list are manually verified.

Obfuscation methods

For each category of PII, obfuscation is performed through the replacement methods described in **Supplementary Table 2**.

Category	Sub-category	Replacement Method	Example
Name	First Name	Replace with sampled surrogate after gender and ethnicity matching	**Mohammad** visited the clinical today. → **Imran** visited the clinic today.
Name	Last Name	Replace with sampled surrogate after ethnicity matching	Ms. **Lopez** agreed with the procedure → Ms. **Hernandez** agreed with the procedure.
Name	Initial	Replace letters randomly	John **W.B.** Smith → Jack **G.S.** Parker
Name	IDs	Replace letters and numbers randomly	Signed **DF14** → Signed **AB76**
Location	N/A	Replace with sampled surrogate	She is from **Springfield, Illinois** → She is from **Ithaca, New Yor**k
Organization	N/A	Replace with sampled surrogate	Welcome to **Veterans Memorial Center** → Welcome to **Butler County Health Care Center**
Age	N/A	If age is greater than 89 years, replace with "89+"	Mr. Johnson is **92** years old → Mr. Michaels is **89+** years old
Date	N/A	Shift date by a randomly selected number of days. Maintain format of the date string.	Appt date: **04/12/2020** → Appt date: **03/29/2020**
Time	N/A	Do nothing	N/A
Website	N/A	Replace with sampled surrogate	For more info check **mayoclinic.org** → For more info check **healthcareforyou.org**
Email Address	N/A	Replace with sampled surrogate	Reach out to **john.smith@care.com** → Reach out to **primaryprovider@care.co**

			m
Vehicle Plate	N/A	Replace letters and numbers randomly	Vehicle plate: **6TR-435** → Vehicle plate: **7TH-129**
Phone Number	N/A	Replace numbers randomly	**546-123-0543** → **574-784-1122**
Numeric Identifier	N/A	Replace numbers randomly	Patient Clinic **#4433245** → Patient Clinic **#1382135**
Zip Code	N/A	Replace numbers randomly	Cambridge MA, **02139** → Tucson, AZ, **45241**
Pager	N/A	Replace numbers randomly	Dr. Jones **1-12435** → Dr. Smith **4-63259**
IP Address	N/A	Replace numbers randomly	**127.0.0.1** → **176.3.5.7**

Supplementary Table 2*: Obfuscation methods for each PII category*

Evaluation metrics

To evaluate model performance on the de-identification task, we computed the precision, recall and F1 scores. These were computed as follows:

$$\text{Precision} = \text{TP} / (\text{TP}+\text{FP})$$

$$\text{Recall} = \text{TP} / (\text{TP}+\text{FN})$$

$$\text{F1} = 2*\text{Precision}*\text{Recall} / (\text{Precision}+\text{Recall})$$

where TP is the true positive count, FP is the false positive count and FN is the false negative count.

De-identification on 2014 i2b2 test dataset

The 2014 i2b2 dataset consisted of 515 notes each in an individual XML file (present in the folder: ./2014 De-identification and Heart Disease Risk Factors Challenge Downloads/test_data/PHI Gold Set - Fixed).

Evaluation of existing methods: We report the performance of Scrubber, Physionet and Philter systems on the 2014 i2b2 data in their standard modes of operation (without additional dictionaries or gazetteers). To run MIST on the 2014 i2b2 data, we converted the dataset into the 2006 i2b2 data format since the stable software release of MIST directly supported the 2006 format (and not the 2014 format). Additionally, MIST assigns PII categories that are different from the 2014 i2b2 entity set. To address this issue, we constructed a mapping between the two sets of PII categories as described in **Supplementary Table 3**. In our implementation of MIST, we did not use gazetteers. As a result the scores we report for MIST are lower than those of the Dernoncourt et al. implementation which was configured to use the same gazetteers as their CRF model. We installed and implemented NeuroNER with instructions as outlined in the GitHub repository (https://github.com/Franck-Dernoncourt/NeuroNER/). In particular, we downloaded and ran the *i2b2_2014_glove_spacy_bioes* pre-trained model on the i2b2 validation set.

MIST PII Category	i2b2 PII Categories
NAME	PATIENT, DOCTOR, USERNAME
LOCATION	ORGANIZATION, STREET, CITY, STATE, COUNTRY, ZIP, LOCATION-OTHER
AGE	AGE
DATE	DATE
CONTACT	PHONE, FAX, EMAIL
ID	IDNUM, MEDICALRECORD, DEVICE
PROFESSION	PROFESSION

Supplementary Table 3: *Mapping between MIST and i2b2 PII categories*

Handling document IDs: The nference system was designed to identify document IDs in unstructured text (e.g. “*3-1272852*” in the sentence “*eScription document: 3-1272852 BFFocus*”). These entities were however not marked as PII in the ground truth of the i2b2 dataset and hence contributed to the false positive rate of our system. If we exclude such cases (we found 87 instances of document ID) our precision improves from 0.979 to 0.986.

PII entity-wise precision and recall comparison: For each entity class and i2b2 entity type we computed the precision and recall for both versions of the nference system (fine-tuned only on Mayo data and fine-tuned on Mayo as well as i2b2 data) as shown in **Supplementary Table 4**. Since the tagset used by nference is different from i2b2 entities, the recall could be calculated for each i2b2 entity and for each entity class. However, the precision could only be determined at the level of the entity class. Rule-based components on the nference ensemble performed identically across both versions of our system since they are not impacted by fine-tuning. Support was computed at the word level (i.e. “John Smith" corresponds to a support of 2).

			nference (fine-tuned on Mayo)		nference (fine-tuned on Mayo+i2b2)	
Entity Class	**i2b2 Entity**	**Support**	**Precision (False Positive Count)**	**Recall (False Negative Count)**	**Precision (False Positive Count)**	**Recall (False Negative Count)**
All	All	10861	0.961 (436)	0.988 (135)	0.979 (239)	0.992 (92)
Date		4951	0.975 (126)	0.994 (27)	0.975 (126)	0.994 (27)
	DATE	4951	N/A	0.994 (27)	N/A	0.994 (27)
Names		4131	0.974 (109)	0.991 (36)	0.996 (17)	0.994 (23)
	PATIENT	1353	N/A	0.992 (11)	N/A	0.998 (2)
	DOCTOR	2691	N/A	0.992 (21)	N/A	0.993 (17)
	USERNAME	87	N/A	0.954 (4)	N/A	0.954 (4)
Location		1177	0.911 (113)	0.980 (24)	0.968 (38)	0.987 (15)
	STREET	415	N/A	0.978 (9)	N/A	0.992 (3)
	CITY	327	N/A	0.982 (6)	N/A	1.0 (0)
	STATE*	188	N/A	1.0 (0)	N/A	1.0 (0)
	COUNTRY*	94	N/A	1.0 (0)	N/A	1.0 (0)
	ZIP	133	N/A	1.0 (0)	N/A	1.0 (0)
	LOCATION-OTHER	20	N/A	0.55 (9)	N/A	0.6 (12)
			nference (fine-tuned on Mayo)		**nference (fine-tuned on Mayo+i2b2)**	
Entity Class	**i2b2 Entity**	**Support**	**Precision (False Positive Count)**	**Recall (False Negative Count)**	**Precision (False Positive Count)**	**Recall (False Negative Count)**
Organization		1639	0.969 (43)	0.815 (302)	0.991 (13)	0.914 (140)
	HOSPITAL*	1502	N/A	0.821 (269)	N/A	0.922 (128)
	ORGANIZATION	137	N/A	0.759 (33)	N/A	0.912 (12)
Numeric Identifiers		576	0.926 (45)	0.977 (13)	0.926 (45)	0.977 (13)
	IDNUM	201	N/A	0.968 (7)	N/A	0.968 (7)
	DEVICE	10	N/A	0.9 (1)	N/A	0.9 (1)
	MEDICALR	365	N/A	0.986 (5)	N/A	0.986 (5)

	ECORD					
Contact		171	1.0 (0)	0.988 (2)	1.0 (0)	0.988 (2)
	PHONE	167	N/A	0.994 (1)	N/A	0.994 (1)
	FAX	3	N/A	0.666 (1)	N/A	0.666 (1)
	EMAIL	1	N/A	1.0 (0)	N/A	1.0 (0)

Supplementary Table 4: *PII entity-wise precision and recall for both versions of the nference system: (a) Fine-tuned on Mayo and (b) Fine-tuned on Mayo+i2b2. The first column corresponds to the entity class and the second column corresponds to the specific i2b2 entity type. Dates, numeric identifiers and contacts are implemented through rule-based methods and therefore have the same precision and recall across both system versions. For this analysis, only ages over 89 in the test dataset were considered (totally 8 instances of such an age were found) and our method detected all of those entities successfully. We therefore omit ages from this table. The tagset used by nference groups is different from i2b2 entities. Therefore, recall is calculated for each i2b2 entity and for each entity class but the precision is determined only at the level of the entity class. (*) While precision and recall have been computed for COUNTRY, STATE and HOSPITAL entities, we do not include for computing the final recall (in accordance with the group B entity set defined in Table 1.)*

Mayo test set annotation

Inter-rater reliability

Cohen's Kappa is used to compute the inter-rater reliability for categorical terms. We calculate Cohen's Kappa for the Mayo test dataset annotated by Mayo Clinic nurses in the following manner.

Step 1: In the ground truth tagged sentences for each nurse, we convert each PII entity (e.g., names, dates, and locations) to a universal " PII entity" type. Non PII entities are left as is.
Step 2: Since the full set of sentences to review is split into three groups and within each group every sentence is reviewed by two nurses, we consider two nurse extractor groups. Group 1 is comprised of nurses 1, 3, and 5 and group 2 is comprised of nurses #2, #4, and #6.
Step 3: We then construct an agreement/disagreement matrix. The numbers in the **Supplementary Table 5** denote the number of words for each category. For example, there are 4,919 words that were marked as PII by group 1 but were not marked as PII by group 2.

		Nurse Extractors Group 2	
		PII entity	Non PII entity
Nurse Extractors Group 1	PII entity	185455 (a)	4919 (b)
	Non PII entity	5411 (c)	1483221 (d)

Supplementary Table 5: *Agreement matrix for measuring inter-rater reliability*

Step 4: The observed proportionate agreement p_0 = (a+d)/(a+b+c+d) = **0.9938**
Step 5: The expected probability (i.e. probability of random agreement between the two groups) is the probability that both groups agreed on either yes or no. The probability that both groups agreed on yes (p_{yes}) is given below
P_{yes} = (a+b)/(a+b+c+d) . (a+c)/(a+b+c+d) = **0.0128**
P_{no} = (c+d)/(a+b+c+d) . (b+d)/(a+b+c+d) = **0.7858**
Therefore,
$p_e = p_{yes} + p_{no}$ = **0.7987**
Step 6: Compute Cohen's Kappa
$\kappa = (p_o - p_e)/(1 - p_e)$ = **0.9694**