MEDINFO 2019: Health and Wellbeing e-Networks for All
L. Ohno-Machado and B. Séroussi (Eds.)

doi:10.3233/SHTI190404

An Extensible De-Identification Framework for Privacy Protection of Unstructured Health Information: Creating Sustainable Privacy Infrastructures

Stefano Braghin[a], Joao H Bettencourt-Silva[b], Killian Levacher[a], Spiros Antonatos[a]

[a] Data Privacy Group, IBM Research, Dublin, Ireland
[b] Health and Person-Centred AI Group, IBM Research, Dublin, Ireland

Abstract

The volume of unstructured health records has increased exponentially across healthcare settings. Similarly, the number of healthcare providers that wish to exchange records has also increased and, as a result, de-identification and the preservation of privacy features have become increasingly important and necessary. Governance guidelines now require sensitive information to be masked or removed yet this remains a difficult and often ad-hoc task, particularly when dealing with unstructured text. Annotators are typically used to identify such sensitive information but they may only be effective in certain text fragments. There is at present no hybrid, sustainable framework that aggregates different annotators together. This paper proposes a novel framework that leverages a combination of state-of-the-art annotators in order to maximize the effectiveness of the de-identification of health information.

Keywords:

Privacy, De-Identification, Natural Language Processing

Introduction

Information exchange is a routine activity among practitioners in healthcare organizations. Much of the information that physicians require to deliver care is recorded and exchanged internally in patient records and, externally, via formal communication letters between referring and consulting physicians across organizations. Other forms of exchange include, for example, legal reasons or claims between providers and payers. As electronic medical records evolved and larger amounts of data were recorded, the value of the secondary uses of healthcare data became apparent [1; 2]. Both the new uses as well as improvements in exchange systems mean that information can more easily be misplaced or accessed by third parties without justifiable reasons.

Early efforts took place in the UK in the 1990s [4]. In the early 2000s, the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule was introduced in the US in order to regulate and safeguard the use and disclosure of protected health information (PHI), which is sensitive and identifiable [5]. The same rule sets out clear definitions of the types of information that are considered to be PHI, such as a patient's name or social security number, as well as guidelines on how to remove such information, such as the Safe Harbor method [5]. De-identification is the process in which PHI in a given patient record is either removed or masked such that it is not possible to link that particular record back to the identity of the original patient. Approaches for estimating the likelihood that de-identified information can be re-identified have been proposed [6] and blanket protection policies and methods such as the Safe Harbor, although effective, may still leave organizations susceptible to re-identification.

De-identification approaches may focus on structured data, for example, estimating the population size cut-offs for geographical areas so that no data suppression or further aggregation is necessary [8]. Other approaches, such as the anonymisation of clinical profiles in the form of International Classification of Diseases (ICD) codes, have also shown that it is possible to share information while safeguarding the privacy of the underlying individuals [11]. However, the rising need for sharing free-text records has boosted the need for Natural Language Processing (NLP) methods and techniques [9]. A systematic review of de-identification techniques revealed that the majority of systems focused on structured data and only a few addressed automatic text de-identification [10]. Furthermore, most systems performing annotation relied on pattern matching, rules, or dictionaries and the latter has shown better performance overall but lacks generalisation [10].

Free-text de-identification tools such as MIST [15] and the various systems in [10] have been more recently evaluated in [14]. The results of this evaluation showed the most successful system (MIST) had a precision of 97.8% and a recall of 95.1%. These evaluation metrics are defined in Figure 1. A different study [13] compared the performances of human annotators against a system using pattern matching based on a dictionary, regular expressions and heuristics. The results showed low precision (75%) but high recall (96%), meaning that the output is mostly de-identified, but also that there is a large amount of over de-identification. Similarly, in [12], the authors tested the performance of human annotators against MIST and an in-house identifier based on MALLET [18] that replicates MIST's design. The results were reassuring, reaching 95% precision in some entity types after adding pre and post processing to the workflow, in addition to model training. Similarly, [16] proposes a system called Medical De-identification System, designed to process HL7 messages in order to remove HIPAA and non-HIPAA specified identifiers. This system relies on scrubbing instead of providing a configurable masking system, yet the authors report results in line with the state-of-the-art systems.

$$P = \frac{detected\ relevant\ entities}{detected\ entities},\quad R = \frac{detected\ relevant\ entities}{relevant\ entities},\quad F_1 = 2\frac{P*R}{P+R}$$

Figure 1 Evaluation metrics: Precision (P), Recall (R) and the harmonic mean of P and R (F_1).

De-identification remains a task that cannot be perfectly handled by automated systems [14] yet, in practical scenarios, such systems have the potential to assist humans in performing de-identification if all the limitations are known and

acknowledged by the users. Contrary to previous work, the approach presented in this paper leverages an ensemble of annotators, taking advantage of the strength of each specific annotator. Moreover, this approach exposes a more flexible configuration to the user, thus allowing easier transition between domains (such as between clinical and financial/billing).

This paper presents the first hybrid framework for automatic text de-identification that combines different annotators together in order to maximise de-identification performance and increase generalisation. The methods section describes the framework in detail, how it was built and their individual components. The results section reveals the performance of each of the annotators, both individually and in combination, based on a representative corpus of clinical notes.

Methods

We have designed and implemented a de-identification pipeline that consists of five main steps, shown in Figure 2. The first step includes input handling and content extraction. The second step performs the pre-processing of the extracted text followed by an annotation step, where the individual annotators perform entity detection. The fourth step includes the post-processing of the results, false positives (i.e. incorrectly detected entity types) elimination and prioritisation. Finally, the de-identification step ensures the protection of the identified entities.

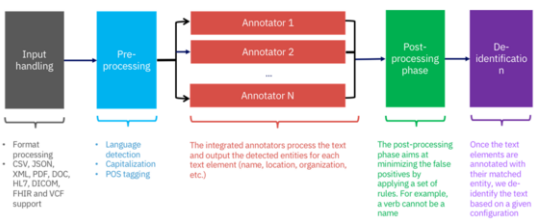

Figure 2 The high-level architecture of our de-identification pipeline

Input Handling

As we need to deal with a diverse set of input formats, our pipeline supports the following formats: CSV, JSON, XML, XLS/XLSX, PDF, DOC/DOCX, HL7, DICOM, VCF as well as plain text documents with no specific format. The user can specify which input elements contain free text information, for example, specific DICOM tags or HL7 segments. The input handling step is responsible for content extraction from the various input formats and preservation of the semantics that are necessary during the reconstruction phase.

Pre-Processing

The pre-processing phase includes three key steps: language detection, correct input capitalization and part-of-speech tagging.

Step 1 – Language Detection: the first step of the pipeline is to perform language detection. Based on the detected language, the system decides to proceed with the remaining steps depending on which annotators are available. For the vast majority of our use cases, we have to deal with input based on the English language but since we are integrating with international platforms, this step is essential. We rely on Apache OpenNLP[1] language detection components,[2] that support 103 languages.

Step 2 – Capitalization for Caseless Input: The input to the pipeline may be caseless; it can be either all lowercase or all uppercase. This has a significant impact on most of the annotators, hence this step is necessary for normalizing the input. We rely on Stanford CoreNLP's truecase annotator to handle such cases.[3]

Step 3 – Part-Of-Speech Tagging: Part-of-speech (POS) tagging is the last part of the pre-processing phase. Each token of the input is tagged with its part-of-speech information. This information is later used in the post-processing phase to alleviate false positives. Several annotators come with POS tagging functionality, such as Apache OpenNLP or Stanford CoreNLP. The POS tags used in this phase must be part of the Penn Treebank POS tagset.[4] For the POS taggers that do not follow this tagset, we manually created the mapping. POS tagging functionality can be extended in our pipeline.

Annotation Phase

During this phase, each annotator is called separately for the same input and then the individual results are gathered and merged. There is no inter-dependency among the annotators so this phase can be fully parallelized for performance.

Post-processing Phase

The post-processing phase includes several steps intended to increase the accuracy and quality of the end results.

Eliminating False Positives Based on POS Information

Since the entities to be de-identified are comprising of either nouns or adjectives, we can eliminate false positives based on the part-of-speech information. For example, if our input is "Will increase Lisinopril to 40 mg daily." and one of the annotators detected that "Will" is a name (this can happen due to blind dictionary matching) then this result can be filtered out since "Will" is neither a noun nor an adjective in the context of this sentence.

POS information need to be ignored for certain entities otherwise true positives would also be removed. Numeric entities, like phone numbers or credit card numbers, e-mail addresses, IP addresses, URLs and datetime entities can be classified as neither nouns nor adjectives, but they must not be discarded. Therefore, we provide configuration options to make sure that the pipeline handles these special cases accordingly.

Priority System

Since we invoke multiple annotators, there may be conflicting results for the same input token(s). In order to handle the potential conflicts, we introduced a priority system per annotator and per entity type. For each annotator, we assign a weight (a value in range [0, 100], with 1 as default) to each of the entity types supported. The initial weight assignment was performed based on the evaluation results of each individual annotator. The priority system also allows us to disable entities for a specific annotator. A weight value of 0 for an entity will force the pipeline to ignore the results for that entity for that annotator. The priority system allows us to disambiguate entries like “Alzheimer” which can be identified both as a name and a medical term, given that medical term classifier has higher priority over the name classifier for the conflicting terms.

[1] https://opennlp.apache.org/

[2] https://opennlp.apache.org/news/model-langdetect-183.html

[3] https://stanfordnlp.github.io/CoreNLP/truecase.html

[4] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Merging of Connected Entities

There are cases where an entity is split into separate results by an annotator. For example, let us consider that the input is "Health Center of Washington". An annotator can detect two entities: "Health Center" as an organisation and "Washington" as a location. However, we want to detect only one entity and that is the entire sequence as an organization. The pipeline enables the merging of these two results into a single entity. Users can specify, through a configuration, the left and right-hand side types of the connected entities along with the connecting particles and/or pronouns, such as “of”, or “on”.

Blacklisting

The final step of the post-processing phase requires checking the results, per entity type, against a configurable blacklist. These contain tokens that are not to be considered matches. For example, for date and time entity type, we assigned that the token “currently” is a blacklist term. It is also possible to blacklist entity types from the final report. As an example, UMLS[5] dictionary entries if they are not going to be de-identified. This feature is enabled in our system in order to reduce over-classification, hence reducing the loss in utility.

Identifying Missing Repetitions of Identified Entities

There are cases where an instance of an entity is recognized once but then subsequent instances are not recognized. The pipeline includes a mechanism to detect unrecognized occurrences for entities previously identified by the annotators. The mechanism assigns types to the missed entities based on previous occurrences. Newly annotated entity types will be processed using the previously described steps (earlier in Post-processing) in order to remove false positives.

De-identification Core

The final step of the pipeline is the de-identification of the detected entities. This step ensures that any PI/PHI is sufficiently protected and that the usability of the end result is at an acceptable level. This can be performed, for example, by an expert accessor as defined in the HIPAA Expert Determination Method.

Data masking is applied on the detected entities and it replaces the original values with fictionalized ones. Our pipeline offers masking capabilities for all supported data types, as listed in Table 1, as well as some generic masking providers, such as redaction, nullification, hashing, randomization, truncation and numeric value shifting. Replacing values with their entity types (for example, e-mail address "test@domain.com" can be masked to "EMAIL-0") is also an option. Two key properties are supported**. Extensive format and semantic preserving masking**: format and semantic preservation is a key aspect of masking, since in many cases the data recipient needs to operate on the masked data. Approaches such as hashing, encryption or redaction, would limit (or even eliminate) the value of the masked data and allow users only to perform basic operations (e.g. counting). Our approach provides utility-preserving masking for each identified type, such as replacing Social Security Numbers with fictional numbers following the same standard, or replacing gender-specific names with other names of the same gender. **Consistent masking**: this feature ensures that the original values are always masked to the same fictionalized value. Consistency is an important requirement for data analytic purposes in healthcare, since it is the only accurate way to measure uniqueness and to do identity matching.

Annotators

Our de-identification pipeline includes a list of existing annotators. This section describes the available annotators. However, as mentioned before, the real power of the toolkit is in its configurability and extensibility. The annotators listed below were selected based on availability and quality criteria.

PRIMA Annotator - The PRIMA annotator is a set of 38 classifiers that cover basic PI/PHI entities based on our work for structured data [17]. PRIMA embeds three types of classifiers: regular expression-based, dictionary-based and custom-logic-based. Entity types like e-mails, addresses and URLs can be detected via regular expressions. Other entity types, like names, cities, countries and ICD codes, are detected using dictionaries. For the rest of the entity types, we use custom code for type detection. For example, credit cards are firstly detected by a specified prefix but the correctness of the check digit is also verified. A list of the currently supported entity types is shown at Table 1. PRIMA exposes the necessary interfaces to easily extend and localize the available functionality, as described in [17]. PRIMA is able to parse HL7 and FHIR messages as it has been designed to be compatible with healthcare systems.

Table 1: Supported entity types for PRIMA annotator.

Demographics	Location	Healthcare	Technology	Banking	Telco
Date and time	City	ICD v9	E-mail	SWIFT	IMSI
Name	Country	ICD v10	URL	IBAN	IMEI
surname	Continent	NDC	MAC address	CC number	Phone number
Religion	Address	ATC	IP Address	CC type	
Marital status	Latitude/Longitude	CPT			
Ethnicity	US states	Hospital name			
National ID	US counties	DICOM metadata			
SSN US	US/UK ZIP Codes	Patient ID			
NINO		MR number			
Occupation					

The application of regex-based or dictionary-based classifiers over free-text can miss entities spanning across multiple tokens. For example, consider the input “Health Center of NY is currently expanding its ICU”, along with the objective to de-identify the organization term “Health Center of NY”. Applying classifiers only for every distinct token will not return any results, even if our dictionaries include terms like “Health Center of NY”. To tackle this issue, we implemented a shingle filter mechanism. For each token, we examine the token itself and the next 1 to N tokens. For example, with N=4 the filter will produce the shingles “Health”, “Health Center”, “Health Center of”, “Health Center of NY”, ..., "currently expanding its ICU". For each shingle, all available PRIMA classifiers are invoked. This mechanism enables detecting entities that span across multiple tokens. Shingles, however, may still return overlapping results yet the subsequent steps in this pipeline are used to merge them with results from the other annotators.

Apache OpenNLP - Apache OpenNLP supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity recognition, chunking, parsing, language detection and coreference resolution. Apache OpenNLP comes as an offline Java library. Trained models are also available for several languages in addition to English.

Stanford CoreNLP - Stanford CoreNLP [19] provides a set of human language technology tools. It can give the base forms of

[5] https://www.nlm.nih.gov/research/umls/

words, parts-of-speech tagging, named entity recognition, normalize dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases and syntactic dependencies, indicate which noun phrases refer to the same entities, sentiment analysis, extract particular or open-class relations between entity mentions, and more. Stanford CoreNLP comes with models for the most commonly spoken languages.

SystemT - SystemT [20] is a declarative information extraction system that has been designed and developed to perform the task of extracting structured information from unstructured or semi-structured data. It is based on the basic principle underlying relational database technology: complete separation of specification from execution. SystemT uses a declarative rule-based language and an optimizer that generates high-performance algebraic execution plans.

Advanced Care Insights - The Advanced Care Insights[6] (ACI) Service uses healthcare annotators that accelerate natural language processing capabilities to identify medical and social information in physician notes, discharge summaries, and pathology reports. ACI identifies, normalizes, and codes medical and social facts, including symptom, disease, procedures, allergies, medications, smoking status, lab results, ejection fraction, and various daily living assistance terms. Note that, like the other annotators, the ACI service can be used in a HIPAA compliant manner.

Extensibility and Integration -There are cases where we need to detect entities based on either dictionaries, regular expressions or custom logic. In order to address such cases, the pipeline supports the loading of dictionary-based and regex-based identifiers via configuration and external files. There is also provisioning for the user to load their own annotators and POS taggers as well as localize the resources used for the detection and de-identification components. This can be done by implementing a Java interface and instructing the pipeline on how to use and configure the annotators via a configuration file.

The pipeline also includes NLP Tools from the Cognitive Computation Group of the University of Illinois[7], Watson Natural Language Understanding[8] and out-of-vocabulary classifier. For brevity, we will not elaborate on each of these annotators.

Results

Entities and Coverage

The pipeline supports the identification and masking of 38 entities. However, this number can be increased by plugging in custom functionality. The out-of-the-box supported entity list is based on the HIPAA Safe Harbor compliance standard but was augmented based on use cases we encountered during development and deployment.

We have classified each entity either as “direct”, which means that it directly identifies an individual, or “indirect”. Some indirect identifiers can be treated as direct in specific contexts, for example, if occupation is “chief of medicine” then it is a direct identifier. However, we do not cover such cases as disambiguating these cases requires additional, contextual, information that might not be available.

Figure 3 describes the supported entities along with their classification and supported annotators. With respect to Stanford, SystemT and Apache OpenNLP annotators, the non-supported entities can be covered after creating custom models to the respective annotation core. The same applies for PRIMA, with the difference that PRIMA also accepts simpler regex-based and dictionary-based identifiers.

As shown in Figure 3, no single annotator covers all entities effectively, at least not without considerable customisation and custom model training. However, our pipeline allows the concurrent execution of all available annotators, so all entity types can be covered. Note that certain entities are marked as "*" in Figure 3. The reason behind this is that such entity types do not have a universally accepted format, contrary to other entity types such as IP or e-mail addresses. These entity types, e.g. medical device identifiers and health beneficiary numbers, can vary between manufacturers and countries and it is therefore non-trivial to provide general models.

Entity	Classification	Stanford	ACI	SystemT	OpenNLP	PRIMA	Overall
Name	Direct	S	S	S	S	S	S
Location	Direct	S	NS	S	S	S	S
Dates	Indirect	S	NS	S	S	S	S
Phone numbers	Direct	NS	S	S	NS	S	S
Email	Direct	NS	S	S	NS	S	S
Social Security Numbers	Direct	NS	NS	NS	NS	S	S
Medical Record Numbers	Direct	NS	NS	C*	NS	C*	C*
Health Beneficiary Numbers	Direct	NS	NS	C*	NS	C*	C*
Organizations	Indirect	S	NS	S	S	S	S
Account numbers	Direct	NS	NS	C*	NS	C*	C*
Certificate / License numbers	Direct	NS	NS	C*	NS	C*	C*
Medical Device Identifiers	Direct	NS	NS	C*	NS	C*	C*
URLs	Indirect	NS	NS	S	NS	S	S
IP Addresses	Direct	NS	NS	C	NS	S	S
Occupation	Indirect	NS	NS	C	NS	S	S
Symptoms (optional)	Indirect	NS	S	C	NS	NS	S
Procedures (optional)	Indirect	NS	S	C	NS	NS	S

Figure 3. List entities supported by each of the annotators. S – Supported; NS – Not Supported; C – Customisable, not prebuilt.

Evaluation and Outcomes

We evaluated the performance of our pipeline in terms of precision and recall based on a random sample of 400 manually annotated medical notes, each referring to an individual patient. The sample was drawn from a large dataset from a US-based healthcare provider and ethical approval was obtained. The full dataset includes medical notes for more than 100,000 patients each of which has at least a medical note. The sampling processes removed machine generated notes and notes not containing any PHI. The notes contained entities for age, date, location, medical record numbers (MRN), names, phones and URLs. We compared our pipeline against NLM Scrubber.[9]

NLM Scrubber is a freely available, HIPAA compliant, clinical text de-identification tool designed and developed by the National Library of Medicine. It is the de-facto standard for de-identification of clinical notes in practical use cases and this allowed us to directly compare similar entity types. For the purpose of evaluation.

The results are summarized in Table 2. Using our pipeline, precision, recall and F_1 are consistently better across all entities. NLM Scrubber does not include models for ages and URLs. The recall is lower than NLM Scrubber only in the case of location and name entities but it is important to note that NLM Scrubber has very low precision on these entities, which means that NLM Scrubber will incorrectly identify as names or location “words” that are not actual names or location, respectively.

[6] http://www-03.ibm.com/software/products/en/advanced-care-insights

[7] Cognitive Computation Group @ Illinois, http://bilbo.cs.illinois.edu/

[8] https://www.ibm.com/watson/services/natural-language-understanding/

[9] https://scrubber.nlm.nih.gov/

Table 2 Performance of pipeline versus NLM Scrubber for all entities that need to be de-identified. Highlighted cells show best values.

	NLM Scrubber			Our Pipeline		
Entity Type	**Precision (%)**	**Recall (%)**	**F_1 (%)**	**Precision (%)**	**Recall (%)**	**F_1 (%)**
Age	0	0	0	93.18	78.85	85.42
Date	97.98	96.52	97.24	99.87	98	98.93
Location	31.25	100	47.62	88.24	88.24	88.24
MRN	53.85	95.45	68.85	100	100	100
Name	68.45	97.92	80.57	94.77	93.96	94.36
Phone	100	93.88	96.84	100	97.50	98.73
URL	0	0	0	100	100	100

Discussion

A multitude of highly specialized annotators are available and significant research has been undertaken to improve the accuracy of specific entities, such as names or medical terms. However, there is at present no existing framework to combine annotators so as to construct a robust pipeline that is able to satisfy multiple governance, regulatory and business purposes. This approach aims to create a flexible system that adapts to a plethora of scenarios. Similarly, a system able to support real-world heterogeneous data poses several challenges not captured by standard benchmarks. Clinical notes typically include typos, mixed cases (different across notes), and partially structured machine generated data (e.g., lab notes with alerts).

The approach proposed in this paper focuses on the identification of entity types, since regulatory requirements are not global or constant and on the premise that ensemble approaches outperform individual annotators. The pipeline proposed in this paper outperformed NLM Scrubber (see Table 2) and provided similar or better results than other approaches [12]. The latter proposes a framework and model that can be refined and where identifiers are customised. The generated models in [12] are trained and updated, however, this would typically mean a loss in precision when new entities are added to the classification model. The main difference between our approach and the one presented in [12] is that our models can return multiple types with weights in addition to the combination of information during post-processing with any other information available (e.g. using POS information to reduce false positives).

Combining positive with negative models helped increasing accuracy, for example, disambiguating between names (sensitive) and medical conditions (not sensitive according to HIPAA). Future development would include automating the priority system, for example, using semi-supervised learning techniques.

Conclusions

Automatic text de-identification is becoming increasingly important as larger volumes of narrative data containing PHI are being used. This paper presents a new hybrid system that combines different annotators and demonstrates how this new approach has a substantial effect on performance and generalisability. This is a useful tool for any data privacy expert following the HIPAA Expert Determination Method. To the best of our knowledge, this is the first framework that combines annotators together to provide precision and recall values better than any of the single annotators alone. Our results are also in line with, when not better than, other free text de-identification systems described in the literature. Further work is underway to evaluate this approach on larger corpora and to cover new data types.

References

[1] D. Blumenthal, Stimulating the adoption of health information technology, *N Engl J Med* **360** (2009), 1477-1479.

[2] C. Safran, M. Bloomrosen, W.E. Hammond, S. Labkoff, S. Markel-Fox, P.C. Tang, D.E. Detmer, and P. Expert, Toward a national framework for the secondary use of health data: an American Medical Informatics Association White Paper, *J Am Med Inform Assoc* **14** (2007), 1-9.

[3] A. Wright, C. Soran, C.A. Jenter, L.A. Volk, D.W. Bates, and S.R. Simon, Physician attitudes toward health information exchange: results of a statewide survey, *J Am Med Inform Assoc* **17** (2010), 66-70.

[4] T.H. Payne, D.E. Detmer, J.C. Wyatt, and I.E. Buchan, National-scale clinical information exchange in the United Kingdom: lessons for the United States, *J Am Med Inform Assoc* **18** (2011), 91-98.

[5] HHS, Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule, in, 2012.

[6] K. Benitez and B. Malin, Evaluating re-identification risks with respect to the HIPAA privacy rule, *J Am Med Inform Assoc* **17** (2010), 169-177.

[7] M. Kayaalp, Modes of De-identification, *AMIA Annu Symp Proc* **2017** (2017), 1044-1050.

[8] K. El Emam, A. Brown, and P. AbdelMalik, Evaluating predictors of geographic area population size cut-offs to manage re-identification risk, *J Am Med Inform Assoc* **16** (2009), 256-266.

[9] P. Zweigenbaum, D. Demner-Fushman, H. Yu, and K.B. Cohen, Frontiers of biomedical text mining: current progress, *Brief Bioinform* **8** (2007), 358-375.

[10] S.M. Meystre, F.J. Friedlin, B.R. South, S. Shen, and M.H. Samore, Automatic de-identification of textual documents in the electronic health record: a review of recent research, *BMC Med Res Methodol* **10** (2010), 70.

[11] R. Heatherly, L.V. Rasmussen, P.L. Peissig, J.A. Pacheco, P. Harris, J.C. Denny, and B.A. Malin, A multi-institution evaluation of clinical profile anonymization, *J Am Med Inform Assoc* **23** (2016), e131-137.

[12] L. Deleger, K. Molnar, G. Savova, F. Xia, T. Lingren, Q. Li, K. Marsolo, A. Jegga, M. Kaiser, L. Stoutenborough, and I. Solti, Large-scale evaluation of automated clinical note de-identification and its impact on information extraction, *J Am Med Inform Assoc* **20** (2013), 84-94.

[13] I. Neamatullah, M.M. Douglass, L.W. Lehman, A. Reisner, M. Villarroel, W.J. Long, P. Szolovits, G.B. Moody, R.G. Mark, and G.D. Clifford, Automated de-identification of free-text medical records, *BMC Med Inform Decis Mak* **8** (2008), 32.

[14] A. Stubbs, C. Kotfila, and O. Uzuner, Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/UTHealth shared task Track 1, *J Biomed Inform* **58 Suppl** (2015), S11-19.

[15] J. Aberdeen, S. Bayer, R. Yeniterzi, B. Wellner, C. Clark, D. Hanauer, B. Malin, and L. Hirschman, The MITRE Identification Scrubber Toolkit: design, training, and assessment, *Int J Med Inform* **79** (2010), 849-859.

[16] F.J. Friedlin and C.J. McDonald, A software tool for removing patient identifying information from clinical documents, *J Am Med Inform Assoc* **15** (2008), 601-610.

[17] S. Antonatos, S. Braghin, N. Holohan, Y. Gkoufas, and P. Aonghusa, *PRIMA: An End-to-End Framework for Privacy at Scale*, 2018.

[18] A.K. McCallum, MALLET: A Machine Learning for Language Toolkit, in, 2002.

[19] C.D. Manning, M. Surdeanu, J. Bauer, J.R. Finkel, S. Bethard, and D. McClosky, The Stanford CoreNLP Natural Language Processing Toolkit, in: *ACL*, ACL, 2014.

[20] L. Chiticariu, M. Danilevsky, Y. Li, F. Reiss, and H. Zhu, SystemT: Declarative Text Understanding for Enterprise, in: *Proceedings of the 2018 Conference of the North American Chapter,* ACL, New Orleans, Louisiana, 2018, pp. 76-83.

Address for Correspondence

Joao H Bettencourt-Silva. e-mail at jbettencourt@ie.ibm.com.