Paper DH04

Text as Data in the Context of Anonymising Clinical Study Reports

Lukasz Kniola, Biogen, Maidenhead, UK
Woo Song, Xogene, Englewood, NJ, USA
Tim Perin, Xogene, Englewood, NJ, USA
Nitin Chaudhary, Xogene, Englewood, NJ, USA

ABSTRACT

In the expanding data sharing landscape, there is an increasing expectation and demand to share clinical trial results and data. Techniques and standards are being developed to support anonymisation of patient-level and aggregate data stored in tabular form. One task that remains largely manual is anonymising/ redacting clinical study reports.

This problem can be tackled by turning human-readable documents into data form that can be processed by machines. This paper explains techniques helpful in finding redaction candidates within text. From simple search, through regular expressions (RegEx, or RE), to Natural Language Processing (NLP). It introduces the ideas of parameterising and contextualising text to increase the accuracy of the findings. It discusses the challenges of extracting text from documents and explores ways to automatically apply the necessary edits and redactions.

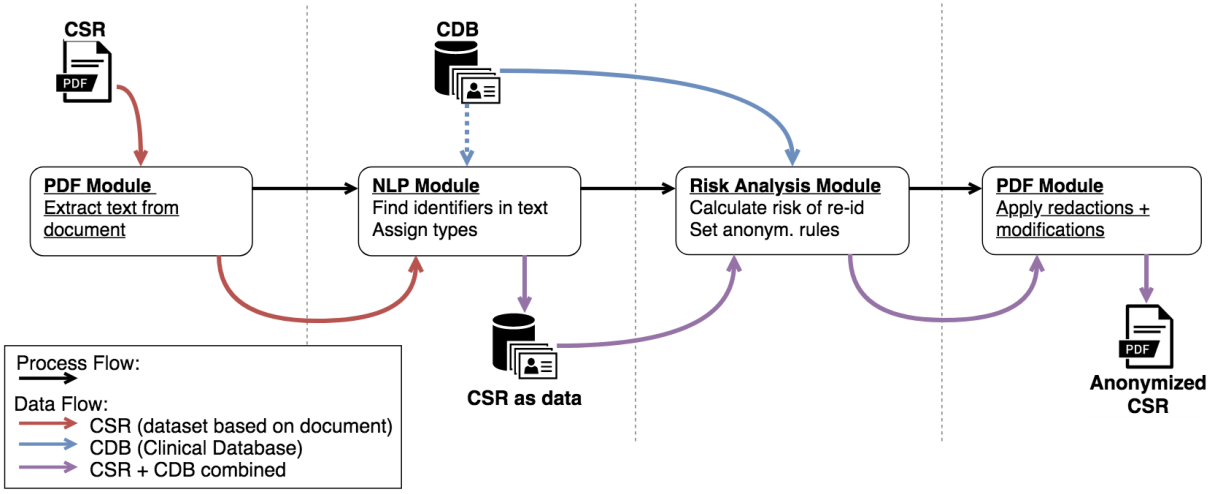
INTRODUCTION

Anonymisation of data has been a hot topic over the last few years. There is a growing expectation and demand to make the clinical trial results and data available for wider audiences, from patients themselves and patient groups, to researchers, to general public. The recent changes and new requirements in the regulatory landscape necessitate development of techniques, and standards which allow for appropriate, defendable, and secure anonymisation of data ahead of sharing and using it for secondary purposes.

The term “data” is typically associated with clinical database and datasets, results in tabular forms, however, data can also be provided as documents, such as Clinical Study Report and similar. While there is an increasing understanding of assumptions, rules and methods in the pharmaceutical industry, which leads to better efficiency and automation, anonymisation of clinical documents and reports presents a completely different set of challenges and problems.

The process can be split into four individual modules as shown on Figure 1. This paper describes the individual steps in detail, including the necessary assumptions, methodology and challenges.

Figure 1. Process flow.



EXTRACTING TEXT FROM DOCUMENTS

Clinical data can be rendered into document form using various formats such as Adobe Portable Document Format (PDF), Microsoft Word (DOCX), and Rich Text Format (RTF). These documents include narrative text, tables and charts, thus making it easier to communicate clinical information to a human reader. However, in the process of creating these easy-to-read documents, the underlying structure and order of the data is typically rearranged in monolithic blocks of text, making it difficult for a computer program to tag, extract and manipulate the data embedded in document form.

To further complicate the problem, Adobe PDF, the most prevalent file type that is used to render clinical data, is the most difficult to manipulate. The PDF was created in 1993 by Adobe Systems with the goal of creating an electronic document format that would appear the same on different devices independent of the environment on which they were created. While the PDF format achieved this goal, it did not contain any functionality for easy tagging or retrieval of text or tabular content.

Therefore, extracting text from a PDF document is one of the first challenges one faces, as PDFs are not structured for data and most PDF creation programs do not add sufficient meta-data or tags to allow for easy retrieval of embedded textual information. A PDF file usually consists of text, vector graphics, and raster graphics. Vector graphics are used to store illustrations and designs, while raster graphics are used for images and text stored as content streams. Specific libraries are required to programmatically extract text stored as content streams. Existing PDF specifications only deal with annotations, encryption etc. and not with extracting and reusing data from a PDF, except to allow for accessibility for use by people with disabilities.

There are quite a few open source and commercial libraries available to extract data from PDFs such as PyPDF, PDFMiner, PDFNet as well as the Acrobat SDK from Adobe. For this project, we used PDFNet SDK from PDFTron to extract text from CSRs in PDF format. PDFNet, while not great at extracting data from tables, does offer the best support for overall text parsing as well as redacting, annotating, highlighting of text and also has a WebViewer component which enables displaying PDF files in a web browser.

We used the Python programming language in conjunction with PDFNet to iterate through the pages in a PDF document and extract blocks of text using the TextExtractor class. The extracted text is then subjected to a variety of methods to identify candidates for anonymisation or redaction.

FINDING CANDIDATES FOR ANONYMISATION/REDACTION

Before focusing on how to find data points within the text, the first step is to establish what type of information should be looked for. At this stage, we are not deciding if and how to anonymise, we are simply extracting the information so it can be processed in subsequent steps.

The following details should be identified within text for later processing:

- Personal information like names, phone numbers, addresses, social security numbers, etc.,
- IDs, whether they are classified as direct identifiers (e.g. subject ID, test ID, etc.) or quasi-identifiers (e.g. site ID, lab ID, etc.),
- Demographic information: sex, age, race, ethnicity,
- Geographic information: country, region, continent,
- Body measurements: height, weight, BMI, etc.,
- Adverse events and medical history,
- Medications, therapies, and procedures,
- Mentions of family,
- Sensitive info, like pregnancy, abortions, mental health, substance abuse, etc.
- Dates.

Some details will be fairly straight-forward to distinguish in the text, since they will follow certain patterns (IDs, dates, etc.), or use a finite list of values (sex, country, race, etc.), while others will require more complex rules and algorithms to identify them in the text.

Another challenge is to attribute each piece of information to a specific subject. This may not be necessary if redaction is used to anonymise the document. However, when values are to be replaced with their anonymised versions, it may be required to understand which subject they belong to.

SIMPLE METHODS

A simple way to find elements of text is by simple search. Each programming language has its string search functions, which can be used to find specific terms in the text. This approach has obvious downsides. Creating static lists of terms to search for is very hard to maintain, prone to error and omission. Lists would need to be kept up to date, need to predict all possible values, regardless of their being present in the document or not.

Simple search can still be useful in very limited scenarios, but it should only be used to capture small numbers of static texts.

REGULAR EXPRESSIONS

Regular expressions (regex) are a powerful tool which can be used to find specific strings of text, and especially patterns (e.g. Subject IDs, dates, sentences, etc.). Regex principles are system agnostic, but different programming languages will have different functions to implement their functionality.

Without regular expressions, to extract the string 54 Years Old would simply involve searching for the exact string.

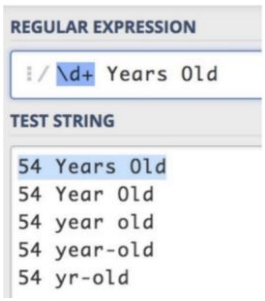
```
s = "Patient 10001 was 54 Years Old"
print s.replace("54","XX")
# Output: Patient 10001 was XX Years Old
```

This fails when the age changes, for example 53 Years Old. This is where regular expressions come in by allowing the programmer to specify a pattern.

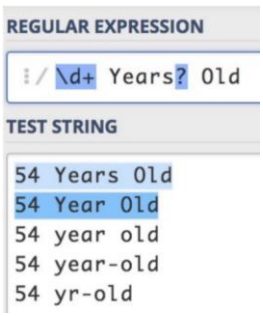
```
s = "Patient 10001 was 53 Years Old"
print re.sub(r"\d+","XX",s)
# Output: Patient XX was XX Years Old
```

Broken down, this expression replaces all occurrences of 1 or more digits with the string XX.

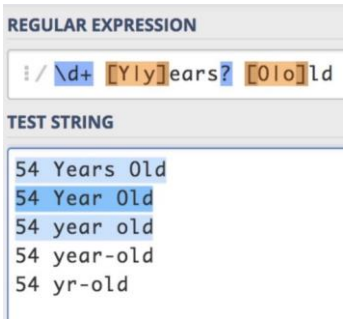
Applying this to a real-world example is much more complex. Patient narratives can differ within the same study and across other studies. Running our pattern against some common examples of age reveals its inaccuracy.



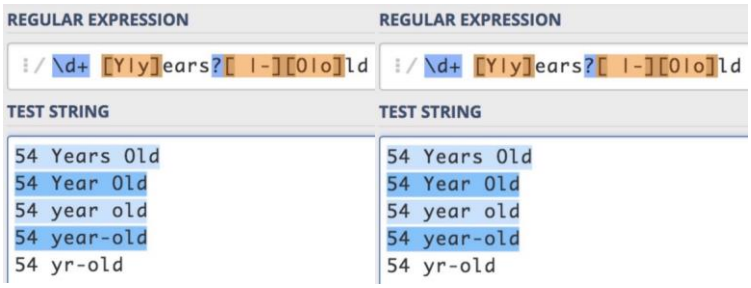
We can improve the pattern by making the “s” in years optional.



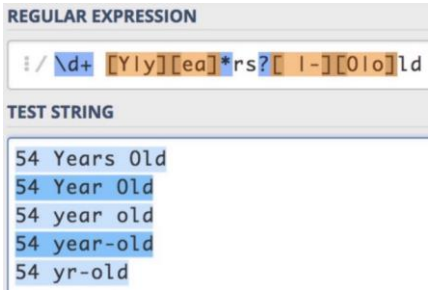
Next, we make the "Y" and "O" case insensitive.



Next, we add an optional dash instead of a space between "Year" and "Old."



Finally, we make the "ea" in year optional.



In order to have a suitable degree of accuracy, regular expressions require more specific and nonspecific descriptors, increasing the length and making them more difficult to work with.

VALUE LISTS BASED ON UNDERLYING CLINICAL DATABASE

Sponsors anonymising their clinical study reports will also have the clinical database, which underlines the results and conclusions of the report. Being able to use this data can be extremely valuable at consequent steps of the anonymisation process.

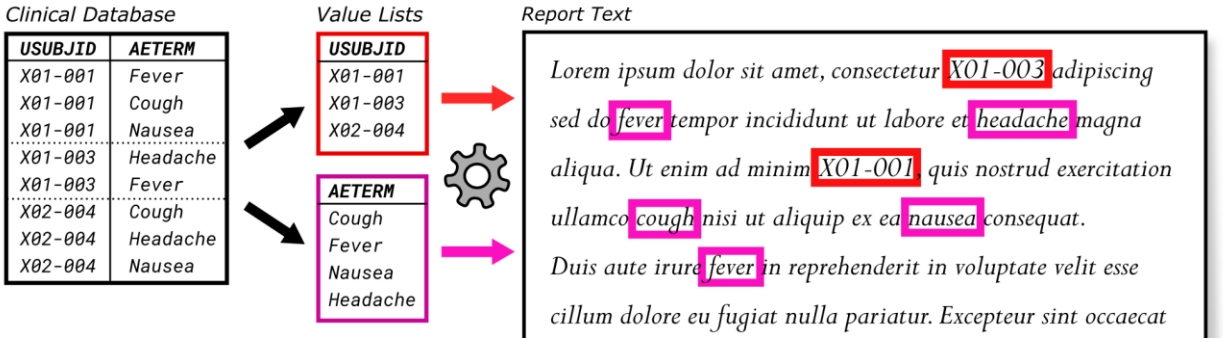
Clinical database can be used to build lists of strings to search for. Examples of such lists include:

- IDs of all subjects taking part in the trial,
- Distinct regions/countries where subjects were recruited,
- Adverse event, concomitant medications, medical history terms (both verbatim and coded) experienced by all subjects.

It is very easy to programmatically generate the lists based on the datasets and since the process is automatic, it is much more robust. Another advantage of this approach is that the lists will be limited only to terms reported during the trial, which makes the search more efficient.

Those generated lists can then be fed to macros or functions which will automatically identify all occurrences of any of the terms in the text.

Figure 2. Generating value lists based on clinical database, and using them for automated search.



NATURAL LANGUAGE PROCESSING / MACHINE LEARNING

Certain data points such as age, body measurements (height, weight, BMI, etc.) or sensitive information such as pregnancy, abortions are not easily identified by using a simple search or regular expressions. In such cases, when a text search candidate is not easy to define by using traditional methods, natural language processing (NLP) or Machine Learning (ML) can be used to round out the approach.

For example, a subject's age can be written as "aged 43," "of age 43","43 years of age," or "43 years old." It is possible to create a RegEx search by enumerating through various permutations, but the approach will fail when it encounters a passage written as "The subject was 43 at the time of the event." A human reader will not have any problems interpreting the last example as an age, but traditional computing methods will fail.

In such cases, one can apply machine learning techniques, which can better deal with ambiguity and contextual meaning. Machine learning involves using documents which have already been tagged manually. The documents and tags are then fed to a neural network, which is trained from the data, also called a corpus. The network learns to recognize "age" as an entity that is not defined by a finite set of values, but as a concept that, with some statistical measure of confidence, is deemed "age."

As with human learning, this type of machine learning is highly dependent on both quality and quantity of the training data. In our experiment we used the spaCy NLP library, which uses residual CNN architecture (convolutional neural network layers with residual connections) for NER Tagging (named entity recognition). The library has various pipelines for tagging, dependency parsing, entity recognition etc. The library also has a tokenizer which automatically parses the incoming document/text and creates discrete units of text. A custom model was trained for custom entity recognition and tagging. As spaCy uses CNNs and various other pipelines, it is context aware and classifies or tags words based on their context rather than solely on their content.

Due to unavailability of an existing dataset of patient data, we created our own dataset. We used 4 PDF documents (CSRs) with a total of 520 pages to create it. Only the patient narrative sections of the CSRs were used for this purpose. These CSRs were manually tagged for patient ID, lab measurements, age, gender, race/ethnicity and date. spaCy's existing dataset was used for geographical information. The dataset consisted of tagged words, text on the page and entity types. The generated dataset consisted of approximately 7000 entities. The dataset was split 70:30 for training and testing sets respectively. The custom model was trained using the training set and tested on the test set. Despite the small amount of data available, we were able to achieve an accuracy of 83.4% on the test set.

TEXT PARAMETERISING

Adding some metadata to the extracted text can be very helpful in processing it. It can help distinguish between free-flow text and tables, titles and paragraphs, etc.

Regardless of the method (or combination of methods) employed to find identifiers in the text, it can be very useful to also collect some metadata which describes those strings. Depending on how the report is going to be processed and anonymised, these may include:

- Page number on which the text was found,
- Position on the page,
- Type of entity (subject ID, gender, country, etc.),
- Text is part of a title, a paragraph or within a table,
- Font and style.

TEXT CONTEXTUALISING

It may not be needed to redact all instances of a specific string or pattern. Understanding if data is given in the context of an individual subject, a group or study population can help flag false positives and avoid unnecessary redactions.

Information classified as identifiers (based on the text) does not always refer to individual subjects. In summary sections of documents, dates, lists of adverse events and others can be discussed not in relation to any particular subject but to the trial overall. In that case, anonymisation is not necessary and would destroy valuable data utility. By being able to distinguish between instances when information is related to an individual and when it is discussed at a trial level, we can make sure that only information which may be helpful in re-identification of individuals is anonymised, while all general discussions and conclusions in the report remain unredacted.

SELECTING ANONYMISATION/REDACTION RULES BASED ON RE-IDENTIFICATION RISK ASSESSMENT

PASSING INFORMATION BETWEEN MODULES

The text of the report has been read in and the identifiers, both direct and indirect, have been found and tagged within the text. This information is then passed onto the risk assessment module. The below table shows an example of such metadata describing all findings.

Table 1. Example of redaction metadata passed from NLP module to Risk Analysis module.

Start position	Page number	Text	Length	Entity type
194	93	101-002	7	SUBJECTID
429	93	101-002	7	SUBJECTID
440	93	30-year-old	11	AGE
452	93	male	4	GENDER
520	93	21 August 2017	14	DATE
722	93	Injection Site Erythema	23	AEDECOD
896	93	03 July 2017	12	DATE
…	…	…	…	…

The next step is to decide which of the identifiers can remain unchanged, and which need to be modified or redacted. Any direct identifiers found in the text should always be anonymised. This can be achieved by redacting or replacing the values with pseudonyms, consistently across the document. Quasi-identifiers are then used to calculate the residual risk of re-identification. Depending on the results of the risk assessment, some or all of them are likely to also require some level of anonymisation. Since they contain different types of information, multiple rules and techniques are available to modify the values and, consequently, reduce the re-identification risk. More detailed methodology of measuring the risk of re-identification using the quantitative approach is described in (Kniola, 2016). The assumptions and methods described in this section refer specifically to anonymizing documents in preparation for public release.

CONTROLLED DATA SHARING AND PUBLIC RELEASE

The context of sharing the data is key in selecting the right measures and methodology for the risk assessment. The most common scenario for sharing data sets from clinical databases is a controlled release of data, to a specific requestor, under a contractual agreement, within pre-set rules and constraints. These conditions reduce the risk of re-identification and allow for a more permissive approach to anonymisation. Conversely, documents like CSRs are most likely to be anonymised with the goal of public release. Since the recipient, their motives and means are unknown, and there is little stopping them from misusing the document and the information it contains strict and conservative approach to anonymisation is required to ensure that identity and privacy of the trial subjects is protected.

Overall risk of re-identification is a product of two measures – risk of re-identification attempt and the risk of the attempt being successful. Under public release, the former value is set to 100%. Since an attack cannot be explicitly

ruled out, then it possible. Therefore, the overall risk, in the context of public release is equal to the risk of successful re-identification.

REFERENCE POPULATION – PROSECUTOR AND JOURNALIST METHODS

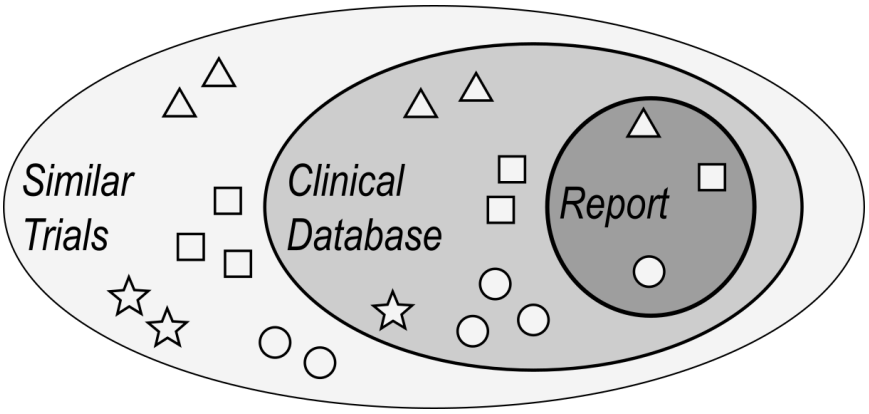
To calculate the risk of re-identification of a dataset it is first necessary to find individual risks for every record in that dataset. First, each record is assigned to an equivalence class, that is, a group of records which have the same values of the quasi-identifiers. Individual risks are then a function of the size of an equivalence class that the record belongs to. The bigger the class, the lower the risk of successful re-identification.

$$Individual\ Risk = \frac{1}{Equivalence\ Class\ Size}$$

When finding equivalence classes, it isn't necessary to only look within the list of subjects found in the text of the document. Patient narratives and summaries are likely to only describe subjects with significant events or findings. If the risk analysis was based only on that subset, this would result in smaller equivalence classes, and therefore higher individual and overall risks. This, in turn, would require excessive modification or redaction of the information in the report to minimize the risk.

It is possible to build the equivalence classes based on a larger reference population. The easiest option is to utilize the clinical database of the reported trial, which includes all study participants. Alternatively, a "similar trials" approach is a viable scenario, in which the reference population is built using trials which are similar to the one whose report is being anonymised. Although a standard definition of "similar trials" is not agreed at the time of writing this paper, it usually means trials within the same indication or drug, run around the same time, with comparable geographical coverage. If a clinical study report is part of a bigger submission then all subjects across all the trials in that group can also form a reference population.

Figure 3. Finding equivalence classes using subjects in the report, clinical database, and similar trials.



The figure illustrates distribution of subjects across different reference populations. Each symbol represents an individual subject belonging to one of four equivalence classes, distinguished by shapes. Depending on the reference population, subjects within classes will be characterized by individual risk values shown in the table below.

Table 2. Individual risks calculated on different reference populations.

	△	□	○	☆
Report	1/1	1/1	1/1	-
Clinical Database	1/3	1/3	1/4	1/1
Similar Trials	1/5	1/6	1/6	1/3

AVERAGE AND MAXIMUM RISK

Once all individual risks are known, the next step is to calculate the overall risk of re-identification for the data set. This can be expressed as the average or the maximum of all individual risk values across the set. Average risk can be used when every record has a similar probability of being targeted (e.g. acquaintance attack). However, while the final value may be sufficiently low, individual records may still carry a significant risk of being successfully re-identified. In the context of public release, a potential attacker is likely to select the most distinct record or records to demonstrate that re-identification is possible. As a result, the maximum risk across the dataset should be calculated to ensure that all records are protected adequately.

Although the equivalence classes used to calculate individual risks should ideally use bigger reference classes, the overall maximum (or average) risk should be calculated only across the subjects found in the text of the report. This will exclude records, whose quasi-identifier values are highly unique, but are never explicitly mentioned in the report. Looking at Figure 1 and Table 1, the maximum risk of re-identification for all records in the clinical database is 1. However, no “star” record is ever mentioned in the text, therefore the maximum risk across the records found in the report is 1/3.

The point of finding bigger reference populations, which better reflect real life scenarios, as well as only using subjects at risk (present in the report) is to prevent over-anonymisation, which in turn means better retention of data utility in the anonymised document.

THRESHOLD

To establish if the calculated risk of re-identification is sufficiently low, it is set against a threshold. The widely agreed threshold in the clinical trial community is 0.09 (or 9%). If the risk value is below that threshold then the document is considered to be sufficiently anonymised. Otherwise, the report needs to undergo further modifications and/or redactions.

APPLYING ANONYMISATION RULES TO REDUCE THE RISK OF RE-IDENTIFICATION

As previously discussed, direct identifiers always need to be redacted or pseudonymised. However, they are not used for the risk calculation. Conversely, quasi-identifiers will be used in the risk quantification, but they don’t always need to be modified.

The following techniques are available to generalize quasi-identifiers, which results in less and bigger equivalence classes, and therefore reduced overall risk of re-identification:

- Sex – can only be suppressed or kept as reported,
- Age – can be categorized to age bands (e.g. 25-30, >89, etc.),
- Race, ethnicity – most common values are retained, others are pooled (e.g. White, Non-White, etc.),
- Geographic information – countries can be aggregated to bigger areas (e.g. US, EU, ROW, etc.),
- Body measurements: height, weight, BMI, etc. – replaced with value bands,
- Adverse events, medical history, medications, therapies, and procedures – verbatim terms removed, preferred terms replaced with higher classes,

It has to be stressed, that all quasi-identifiers don’t always need to be anonymised to the full extent. They can be retained in their reported form if the residual risk remains below the threshold. Risks should be computed for different combinations of rules and techniques, with each quasi-identifier being in turn: kept, anonymised or suppressed. The optimal scenario will then be the one for which the residual risk is below the threshold but as close to the threshold as this will ensure that the data has been anonymised sufficiently and not excessively, and as much data utility has been retained as possible.

When the only tool available to modify the document is redaction, then the rules are limited to keep/supress only.

If values can be edited within the text, then for the quasi-identifiers which use banding, pooling or aggregation, there are two options available. The original values can be replaced with:

- the higher-level category (e.g. “37-year-old” to “35-39-year-old”, “Germany” to “Western Europe”, etc.),
- a value of the same format as original from within the band (e.g. “37-year-old” to “X-year-old” when X is a random number between 35 and 39, “Germany” to “XYZ” where XYZ is one of the countries from the “Western Europe” group, etc.).

DATES

Dates are a special type of identifiers. They can be useful to attackers and therefore should be edited, however, they are not used for the risk calculation. Shifting the dates significantly reduces their usefulness for a potential attacker. When done consistently within each subject's data (so that for each subject, all dates are shifted by the same number of days) the relationship between events for each subject is retained. Another option is replacing absolute dates with study days, which has similar effect.

SENSITIVE INFORMATION

Sensitive information, like references to family, pregnancy, abortions, mental health, substance abuse, etc., should be carefully considered for unconditional redaction. Once tagged, these details should be suppressed. In such case they no longer need to be included in the risk calculations.

BUILDING THE SET OF ANONYMISATION RULES

Some identifiers may use a general rule, like generalization. Those are not subject-specific and can be applied without individual values being attributed to specific subjects. For example, age value may need to always be replaced with age band. In such case, an age of "37" will be replaced with "35-39" regardless of which subject it refers to.

Other identifiers will use rules which are individually tailored to each subject. For example, dates will be shifted by a different number of days for each subject. When a string "21 August 2017" is found, the number of days by which to shift the date will differ from one subject to another.

When deciding on anonymisation rules and creating instructions for entities found in the text, the pivotal condition is whether each entity can be attributed on an individual subject. If this is the case, then subject-specific values can be modified on a subject by subject, entity by entity basis.

When attribution is not possible, only general subject-independent rules (like age banding) can be used.

This problem is simplified when the text is redacted. Since redaction only uses a binary KEEP/REDACT ruleset, it is not necessary to attribute each entity to a subject. Based on the risk analysis, a variable like age, sex, or country will either always be kept or removed. Therefore, this can be done without linking the entity to a particular subject.

PASSING INFORMATION BETWEEN MODULES

Final rules need to be passed onto the editing module which will apply the anonymisation rules to all tagged text. The table received from the NLP module is updated with specific instructions for each entity, as shown in the example below.

Table 3. Example of redaction metadata passed from Risk Analysis module to PDF module.

Start position	Page number	Text	Length	Entity type	Anonymisation rule	Replacement/ overlay
194	93	101-002	7	SUBJECTID	REPLACE	X0014
429	93	101-002	7	SUBJECTID	REPLACE	X0014
440	93	30-year-old	11	AGE	REDACT	-
452	93	male	4	GENDER	KEEP	
520	93	21 August 2017	14	DATE	REPLACE	30 August 2017
722	93	Injection Site Erythema	23	AEDECOD	KEEP	
896	93	03 July 2017	12	DATE	REPLACE	12 July 2017
…	…	…	…	…	…	…

EDITING AND REDACTING THE SOURCE DOCUMENT PROGRAMMATICALLY

We made use of both regular expressions and NLP to identify items for redaction, as each method has its own advantages and disadvantages. For example, RegEx is better at identifying entities that conform to a specific pattern, but is less accurate for entities which rely on context. NLP takes a more semantic approach and uses the structure of the sentence and relationships between words to identify entities that could not be easily identified using RegEx. Certain entities are best identified using a combination approach. Using both techniques, text is parsed and processed twice. First, using the NLP technique, the text of each page is passed to a named entity recognition model which pulls out all relevant phrases. In the second phase, any text matching the RegEx patterns are also identified. Both of these phases return a collection of tagged words, entity types and the location of each phrase in the page's text. This data is stored in the database, which can later be used to refine the model after any incorrect matches are removed.

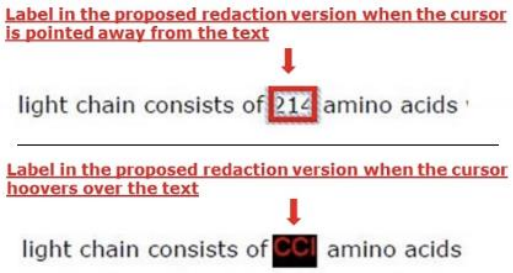
This data is then fed to the risk assessment module, which calculates and performs risk calculations. These calculated values are then compared against a threshold to determine whether the document is sufficiently anonymized or not. The risk assessment module also provides information on how the tagged text needs to be redacted, edited, or anonymized. PDFNet provides tools to redact and edit PDF documents. Using PDFNet, the tagged text is searched for on each page. Redaction marks are applied and tagged text is edited, depending on the rules and instructions provided by the assessment module. The redaction and anonymization are performed in accordance with EMA Policy 0070. After this last phase is complete, a PDF document with final redaction marks is generated. This document can then be opened in Adobe Acrobat for final review and the redaction marks can be applied.

TRANSFORMING DOCUMENT

Redaction is applied by overlaying text – as required – by black boxes. The length of the string does not affect the process. Modifying text poses challenges due to the fact that the replacement text is unlikely to match the original exactly (e.g. “France” >> “Western Europe”, “Asian” >> “Non-white”, etc.). Special consideration is required to decide how modifications will be applied. If the anonymised text is shorter then it leaves gaps. Longer replacement text may need to be “squeezed” to fit in the original page layout. This may affect readability. An alternative is to reflow the document. However, this means that the overall length of the new text may differ from the original. Pagination may be broken, additional rules to cater for extra pages are required.

EMA POLICY 0070 CONTEXT

There are specific technical requirements for redacting CSRs under EMA Policy 0070. The minimum requirements are that the file format is PDF and that "text proposed for redaction should be clearly identified as such (i.e. marked) and the text itself should be legible (read-through)" (http://www.ema.europa.eu/docs/en_GB/document_library/Regulatory_and_procedural_guideline/2017/09/WC500235371.pdf). An example is provided by the EMA.



Herein lies the difficulty with programmatically redacted CSRs for Policy 0070. In its simplest form, a programmatic redaction would involve converting the page to an image file and covering the redacted text with a black box. Adobe Acrobat has a redaction tool for highlighting and marking text for redaction. The redaction annotation example provided by the EMA is a specific PDF annotation mark used by Acrobat. While there are many open source tools that have the ability to manipulate PDF files, few of them support the redaction annotation mark. There are some licensed tools, such as PDFTron's PDFNet, but the licensing fee makes this a less appealing solution.

The Policy 0070 redaction mark has the following properties: a red border, red font colour, and black overlay. In order to accomplish this using PDFNet, the SDF Object needs to be manipulated. By reverse engineering Acrobat's redaction annotations, the following properties can be set using PDFNet.

```
# Assuming the following variables are set:
#  annot - the RedactionAnnot

# To set the font color to red, we must manipulate the font
property, labeled as "DA". The red, blue, and green components
are the first three values in this string, which we will set to
(255,0,0) for red.
annot.GetSDFObj().PutString('DA','%d %d %d RG 0 g 0 Tc 0 Tw 100
Tz 0 TL 0 Ts 0 Tr /Helv 10 Tf' % (255,0,0))

# To set the overlay background color, we will set the interior
color to (0,0,0) for black
# Note: The ColorPt class uses a range of 0.0-1.0 to represent
color values. So, for example, rgb(0,255,0) would be
ColorPt(0.0,1.0,0.0)
annot.SetInteriorColor(ColorPt(0.0,0.0,0.0), 3)

# Finally, to set the border to red we simply set the annotation
color to (1,0,0) for red
annot.SetColor(ColorPt( 1.0, 0.0, 0.0 ),3)
```

CONCLUSIONS

Anonymizing Clinical Study Reports is a challenging task. There are four distinct steps in the process, each posing its own difficulties.

As discussed, working with PDF documents requires special tools to retrieve the contents. Due to many ways of creating PDF documents, there are no bulletproof ways to achieve this. However, tools are becoming more capable and even extracting information from figures within documents or fully scanned pages, can be achieved.

Processing text to find identifiers and redaction candidates can be achieved in a number of ways. Utilising the underlying database can simplify the process and make it more accurate. Tapping into the capabilities of machine learning removes the necessity of anticipating and often hardcoding terms which should be found in the text. The field of machine learning is growing at an exponential speed. New models are built, and existing ones are vastly improved. As these are perfected, the accuracy and efficiency of the natural language processing will also increase over time.

Selecting what to redact or modify and what to leave in the text relies on the quantification of re-identification risk. Although a big proportion of this process can be automated, this step still relies on human input as different context can lead to different decisions. The choices and decisions should, however, be informed by the computed results, to make sure the level of anonymisation is sufficient but not excessive.

The choice between redacting and modifying the text of the document depends less and less on the technical capabilities, and more on the context and environment of the release. There are still issues which need clarification and direction.

Thanks to the modularity of the process, each part can be developed separately, so long as the data and metadata passed between the modules is clearly defined. This means that as new tools, techniques and standards become available, individual steps can be updated or even replaced without affecting the remaining modules.

With the improvement of tools and standards and the increasing pressure from regulators, industry and public to share results of clinical trials, this is a very exciting area of focus and development.

REFERENCES

Canadian Institute for Health Information. (2010). *'Best Practice' Guidelines for Managing the Disclosure of De-Identified Health Information.* Ottawa, ON: www.cihi.ca.

El Emam, K. (2013). *Guide to the De-Identification of Personal Health Information.* Boca Raton, FL: CRC Press.

Institute of Medicine. (2015). *Sharing Clinical Trial Data Maximizing Benefits, Minimizing Risk.* Washington, DC: The National Academies Press.

Kniola, L. (2016). *Calculating the Risk of Re-Identification of Patient-Level Data Using Quantitative Approach.* PhUSE Annual Conference, 2016.

McKinney, W. (2018). *Python for Data Analysis*. Sebastopol, CA: O'Reilly Media.

Bird S., Klein E., Loper E. (2009). *Natural Language Processing with Python*. Sebastopol, CA: O'Reilly Media.

Bengfort, B., Bilbro, R., Ojeda, T. (2018) Applied Text Analysis with Python. Sebastopol, CA: O'Reilly Media.

El Emam, K., Arbuckle, L. (2013). Anonymizing Health Data Case Studies and Methods to Get You Started. Sebastopol, CA: O'Reilly Media.

Pharmaceutical Users Software Exchange (PhUSE). De-identification standards for CDISC SDTM 3.2.

European Medicines Agency. (2014). *EMA/240810/2013 - Publication of clinical data for medicinal products for human use.* http://www.ema.europa.eu

European Medicines Agency. (2017). *EMA/90915/2016 - External guidance on the implementation of the European Medicines Agency policy on the publication of clinical data for medicinal products for human use.* http://www.ema.europa.eu

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Lukasz Kniola
Biogen Idec Ltd
70 Norden Road, Maidenhead, Berkshire SL6 4AY, UK
Email: lukasz.kniola@biogen.com

Woo Song
Xogene
10 Sterling Boulevard, Suite 301, Englewood, NJ 07631
Email: woo@xogene.com

Tim Perin
Xogene
10 Sterling Boulevard, Suite 301, Englewood, NJ 07631
Email: tim@xogene.com

Nitin Chaudhary
Xogene
10 Sterling Boulevard, Suite 301, Englewood, NJ 07631
Email: nitin@xogene.com

Brand and product names are trademarks of their respective companies.