



University of Bonn  
Bonn-Aachen International Center for Information Technology (b-it)

## **Anonymization of Electronic Health Care Records: The EHR Anonymizer**

Thesis submitted as partial fulfillment of the requirements  
towards the passing of the module Programming Lab 2

By  
**Alex Hoch, Bryce Fransen, Thomas Lordick**

**February 2020**

# Abstract

Electronic health care records (EHRs) are the main source of information between medical professionals. Therefore, EHRs would be a good source for advanced data analysis. However, the current data privacy laws complicate the utilization of patient data; due to the concern of evaluating personal information without consent. Since personal information is only partially useful for analysis, removing it from the documents is a reasonable solution that respects the patients privacy and makes patient data accessible for third parties. The electronic health care records anonymizer (EHRA) is a BRAT-based application that can be used to tag personal information in EHRs and replace them with previously defined placeholders. In a first annotation step, machine learning (Stanford NER) and rule based approaches generate tags for potential personal information and prior user annotations are added to the document. The user can now add/edit/remove annotations until the annotations are adjusted. The performance of the automated tagging was evaluated on the basis of 10 example EHR input documents, and resulted in a F1 score of 0.72 and an accuracy of 0.96. Since the goal was to keep false negative (FN) tags at a low rate, the false negative rate (FNR) of 0.1 motivated the implementation of a 'feedback loop' that saves user annotations for future uses and therefore reduces FNs. Further, the moderate number of false positive (FP) tags results a low precision score of 0.6, which could be counteracted with the implementation of a 'black list', that saves tags that were deleted by the user. In combination the 'feedback loop' and 'black list' would boost the automatic annotation performance, yet in the end, the user has control.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Methods</b>	<b>7</b>
<b>3</b>	<b>Results</b>	<b>15</b>
<b>4</b>	<b>Discussion</b>	<b>17</b>

# List of Figures

1.1	<i>The beginning of a German electronic health care record in form of a letter. Written by a clinical physician to the patients family physician to describe the medical status. . . . .</i>	3
2.1	<i>Diagram that displays the workflow of the electronic health care anonymizer (EHRA). INP = Document File, PAR = Parsing, NER = Name ENTity Recognition, FLE = Feedback Loop Part I, GUI = User Interface, ANO = Annotation, OPT = Options, FLL = Feedback Loop Part II, EXP = Confirm Export, OUT = Anonymized Document . . . . .</i>	7
2.2	<i>BRAT (Brat rapid annotation tool) annotations in different languages. Tags/labels and relationships are annotated. . . . .</i>	9
2.3	<i>The main graphical user interface window is divided into the button row and the two web views that display the original and the annotated text. . . . .</i>	11
2.4	<i>Annotation File Example . . . . .</i>	12
4.1	<i>The performance of the electronic health care records anonymizer was measured based on 10 example letters. The pie diagram displays the total ratio between true positive (TP), false positive (FP), true negative (TN) and false negative (FN) results. . . . .</i>	18

# List of Tables

3.1 *Left Table: Displays all sample documents names (Doc) and their corresponding True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN) counts. Right Table: Displays the statistical metrics (FPR = False Positive Rate, FNR = False Negative Rate) calculated using the left tables TP, FP, TN and FN totals. . . . .* 16

# 1 Introduction

## Electronic Health Care Data

Electronic health care records (EHR) are known to be a major part of data science now days. Before digging into EHRs and its relevance in particular, it's important to revise the term „data science“. The term “data science” describes expertise associated with taking (usually large) data sets and annotating, cleaning, organizing, storing, and analyzing them for the purposes of extracting knowledge. It merges the disciplines of statistics, computer science, and computational engineering [5]. Finding patterns within the data and using it for the prediction of new properties (regression) or classify new data points (classification), e.g whether a tumor is malignant or benign, is a widely application of researchers dealing with medical data. One has to consider, determining the appropriate algorithm, deeply depends on the type(s) of data we are dealing with.

Data can be a table of samples or data points (e.g patients) with corresponding properties describing each sample, which is probably the most famous type of data. With respect to medical data, it could involve several measurements like blood pressure or heart rate, but also categorical features that indicate the absence or presence of certain disease. Apart from that, data in the form of pictures (X-Ray screenings, etc.) and natural language exist in the medical field to a large extent.

EHRs represent data in the form of natural language. To define the term briefly, we would like to quote the Centers for Medicare Medicaid Services (CMS) [2], which provides a very useful explanation:

„An Electronic Health Record (EHR) is an electronic version of a patients medical history, that is maintained by the provider over time, and may include all of the key administrative clinical data relevant to that persons care [...] “

The provider in that context would normally be a clinic or hospital, but could also be the patients private physician. The so-called „medical history“ isn't formally defined and could involve any kind of observation the „provider“ has made on the patient - for example, the patients problems, medications, vital signs, immunizations, laboratory data or radiology reports. With addition to, old information from observations from the past perhaps made by different providers. One key point is, that data in form of natural language has no well-defined structure as it is the case in simple tables which adds to the challenge of extracting computer-readable knowledge from this kind of data. Without going into detail at this point, text mining techniques are usually used for this purpose.

To illustrate this short introduction to EHRs, the following screenshot shows a typical sample we worked with.

Sehr geehrter Herr Dr. Müller,

ich berichte Ihnen über Ihre Patientin, Frau **Ina Müller**, \***21.08.1946**, die sich zuletzt am 18.06.2008 in unserem Diagnostik- und Behandlungszentrum für Gedächtniserkrankungen im Alter (DBGa) vorstellte.

**Diagnose:** Demenz vom Alzheimer Typ mit spätem Beginn  
Hypertonus

**Anamnese:**  
Bezüglich der Vorgeschichte möchte ich auf vorhergehende Arztbriefe aus dem Jahr 2007 verweisen. Zusammenfassend stellte sich Frau Müller erstmalig im Mai 2007 in unserer Ambulanz vor. Es wurde damals bei seit ca. drei Jahren bestehenden Gedächtnisstörungen die o.g. Diagnose gestellt. Es wurde eine Behandlung mit **Aricept** eingeleitet. Aktuell erscheint Frau Müller in Begleitung ihrer Tochter zur Verlaufsuntersuchung. Die Pat. selbst berichtet, dass ihr Gedächtnis allenfalls **leichtgradig** schlechter geworden sei. Die Tochter berichtet von einer deutlichen Verschlechterung des Gedächtnisses. Ihre Mutter würde häufig die gleichen Dinge erzählen. Sie müsse häufig Sachen wiederholen. **Gelegentlich** würde sie auch die Einnahme von Medikamenten vergessen. Ihrem Beruf als Politikerin könne sie immer schlechter nachgehen.

**Orientierende psychometrische Testung:**  
Mini-Mental-Status: 19 von 30 Punkten. (Mini-Mental-Status v. 12.11.2007: 20 von 30 Punkten).

Figure 1.1: *The beginning of a German electronic health care record in form of a letter. Written by a clinical physician to the patients family physician to describe the medical status.*

## EHRs and the problem of data privacy for research

The sharing of data in general is fundamental for science. Making it accessible for further analysis, for example x-ray screenings of a patient to research labs or specialist clinics, can support the patients recovery as well as research labs to use the data to develop new image analysis techniques, derive knowledge from it or, broadly said, drive the process of science forward. This applies not only to image data. It applies to any kind of data, especially the data we are dealing with: EHRs.

But, as indicated in the headline, reality is not as straight forward as it might appear in the last few lines. A fact about EHRs (and data in general) is that it contains private information. Private information in the context of



EHRs would be the patients name, his/hers date of birth or even his/hers profession - depending on the data that is included in the record. The problem that goes with this is, that according to data privacy laws (§ 1 I BDSG) it is not-permitted to share private data for secondary purposes unless the patients give their consent or authorization. In the worst case, without permission, the process of science is plainly said "abandoned before takeoff".

### **Anonymization**

So how do we bypass the problem of data privacy? In other words, how can we still provide EHRs for secondary purposes? The solution is Anonymization, sometimes also called de-identification. Anonymization is considered to replace all private terms by placeholders, so that the data can't be assigned to its original person the data was drawn from any more. An obvious example would be the persons name, date of birth or even his/her telephone number. So in the process of anonymization, those terms would be automatically tagged and replaced by placeholder in order to keep the structure of the data, but "de-identify" it. The question is, how to do so in a way that protects individual privacy, but still ensures that the data is sufficient in quality so that the analytics are useful and meaningful. Imagine the date of birth. Since this is a very private term and should be replaced by the system, one could raise the question: What if the date of birth is of importance for further research? The exact age of person could be tremendously relevant for a clinic or a physician that gets the data from the source and has to draw conclusions from it regarding the patients disease. Would it be sufficient to keep the year of birth and cut out the rest in terms of data quality? If so, do we still protect the individuals privacy as well? So in order to provide sufficient data, anonymization has to be defined dynamically depending on the specific terms we want to replace or alter. Furthermore, since we are dealing with text data of moderate size, the process has to be carry out automatically by a tool that prevents people from tagging private terms by hand. All those aspects have to be consid-

ered when building an anonymization tool. How we managed to tackle those problems and aspects will be outlined in the next chapter.

### **The EHR Anonymizer**

Summarized in one sentence: we built a java tool that guides the user through the process of anonymization for EHRs in an semi-automatic fashion.

The form of the data the software uses as input meets the structure of formal letters as shown in Figure 1.1. Those letters are formatted in word (DOC/DOCX). Since workers in the health care sector are potential users of the tool and considered to be „non-IT“ individuals, we provide a graphical user interface of our software to support the users in the handling of the tool. After the input file has been read and edited, text mining techniques will be applied in order to annotate potential terms that could affect the patients privacy. Categories of private terms were defined beforehand. We were making extensive use of Stanfords Named Entity Recognizer (NER) software which provides a useful java API including several options to customize the pipeline for the specific core documents (EHRs) used [4]. Stanfords NER library is built on a combination of deep linguistic modeling and data analysis with innovative probabilistic, machine learning, and deep learning approaches and was trained with a vast amount of core documents. Moreover, we built an additional named entity recognizer only based on RegEx to support the NER workflow and tag important terms that Stanfords NER methods missed.

In the next step, having the input file text and the corresponding annotations (marked by the span), both raw text and annotated text will be displayed side by side. For that purpose, the software integrates BRAT (<https://brat.nlplab.org>), a web-based tool for text annotation for adding notes to existing text documents. As mentioned above, to provide a certain flexibility regarding the level of de-identification, we provide a way to let the user decide which terms should be anonymized and to which

extent. That allows the user to disregard certain annotation made by the software or even add more annotations that may be important for the final anonymized output. In the end, the record will be written to a new word file and can be used for sharing.

## 2 Methods

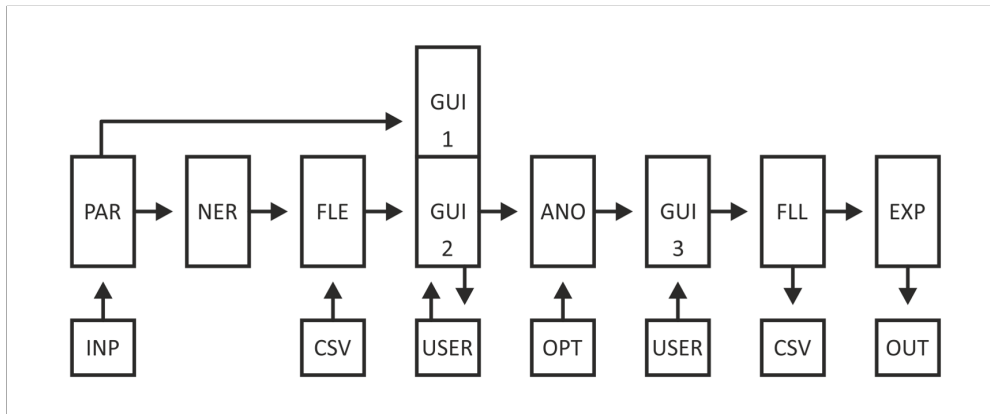


Figure 2.1: Diagram that displays the workflow of the electronic health care anonymizer (EHRA). INP = Document File, PAR = Parsing, NER = Name ENtity Recognition, FLE = Feedback Loop Part I, GUI = User Interface, ANO = Annotation, OPT = Options, FLL = Feedback Loop Part II, EXP = Confirm Export, OUT = Anonymized Document

## File handling

The first essential part of program is the handling of files. Since input files are formatted in DOC/DOCX, we made use of the Apache POI [1], which is a java API for handling Microsoft documents.

In the reading step, it was stored as a string without losing the original formatting. Since the EHR's, we were dealing with, contain formal headers mostly including information about the sender, this part was cut out of the string in an additional step. The final edited string serves as input for NER annotation part in the next step. Skipping a few steps of the pipeline,

when the entire anonymization procedure is done, a final output file in docx format is written to the disk.

### **EHR and Annotations**

Once the input file has been read in and converted to a simple string, the annotation part begins. Before annotating the private terms, categories of potential words that could reveal the patients identity were defined. Those categories are PERSON, ADDRESS, ORGANIZATION and GENDER. Under the tag PERSON, we consider names, for example the patients name or of the physician. ORGANIZATION and ADDRESS are tags that typically annotate company names and their locations mentioned in the text. In context of EHR's those can hospital or clinic names and their corresponding locations or even the address of the patients itself. The GENDER term describes words that may indicate the gender of the patient. While those are obviously simple pronounce like „Er“ or „Sie“, so called „Possessivpronomen“ like „ihr“ or „sein“ in the German language indicate the patients gender as well and will be tagged.

For that purpose, we made use of Stanfords Natural Language Processing (NLP) package and a self-written RegEx class, that was only trained for tagging GENDER. From the NLP package the specific Named Entity Recognition modules were used and fine-tuned. The so called „fine-grained NER“ options was turned off while rule-based models were further included. The goal was to keep the model as simple as possible while accounting for a faster running time and more accurate tags. Since fine-grained NER includes complex model trained for tagging terms indicating the ideology or criminal charge would add useless complexity and therefore larger running times, this option was disregarded. Details can be taken from the java documentation. Having the list of annotations, their entities and corresponding span marked by indices, that information will be written into the annotation-file which is used in the next part. Detail on the Annotation file format will be explained later on.

### BRAT rapid annotation tool

BRAT (Brat rapid annotation tool) is a web-based application for language independent text annotation, introducing an interface that enables a direct connection of text annotation and most recent natural language processing tools (NLP) [6]. BRAT is designed to support manual curation by giving users the opportunity to utilize advanced machine learning and rule-based approaches for e.g. relation extraction (RE) and/or named-entity recognition (NER), resulting in a supportive setting for curators, yet ensuring a high curation standard utilizing final human judgements. Furthermore, BRAT offers an intuitive interface controllable with mouse ‘dragging’ and ‘double clicking’ and supports different types of annotations e.g. relationships, tags/labels and/or free form annotations.

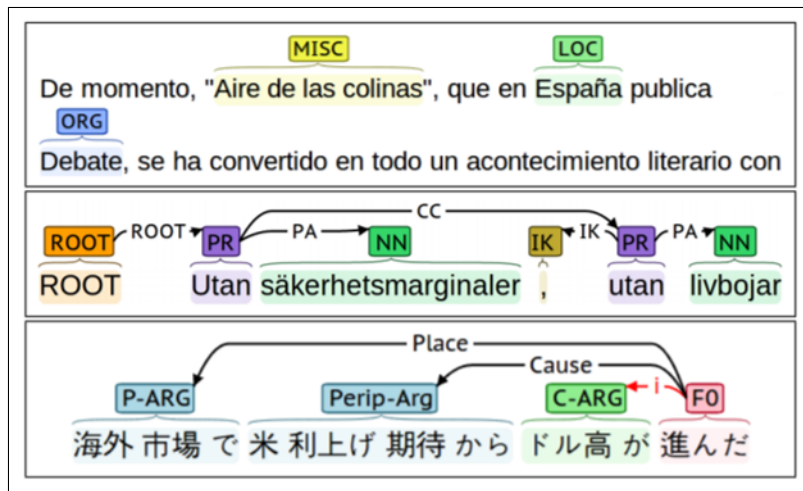


Figure 2.2: BRAT (Brat rapid annotation tool) annotations in different languages. Tags/labels and relationships are annotated.

### Graphical user interface

After the automatic annotation by the Stanford NLP module and the addition of previously annotated terms (Feedback loop), the graphical user

interface (GUI) is the main interaction point between the application and the user (Figure 2.1, middle part). In addition to previous annotations, the user can add annotations to the point the annotations are adjusted correctly. This concept is referred to as a “human in the loop (HITL) scenario”, as human judgements and machine learning techniques are combined and result in the most promising findings. Since the application targets a problem in the medical field and is mainly used by non-IT users, a user-friendly interface is required that is easy to use and that displays meaningful errors when wrong inputs are made. This is accomplished by constructing the GUI with JavaFX, one of the standard GUI libraries provided by Oracle Corporation. JavaFX is part of the open source JDK since the release 11 and therefore supported by a variety of desktop environments including Microsoft Windows, Linux and MacOS. The main GUI consists of three main parts: The first part is a button menu, that is used to open files, close files, refresh, change the path to the BRAT data directory and show credits (Figure 2.3). This enables the user to do basic interactions with the application. Second, the main window contains a view of the original text (embedded in a JavaFX webview). Besides the original text, the BRAT tool is integrated into the GUI (embedded in a JavaFX webview aswell) that displays the same text with added annotations. To run an anonymization up to this point, the user needs to download and install BRAT, run it on a server or locally, open a .doc/.docx document, the document will be annotated by the stanford NER tool and the feedback loop and finally annotations will be visualized in the BRAT window.

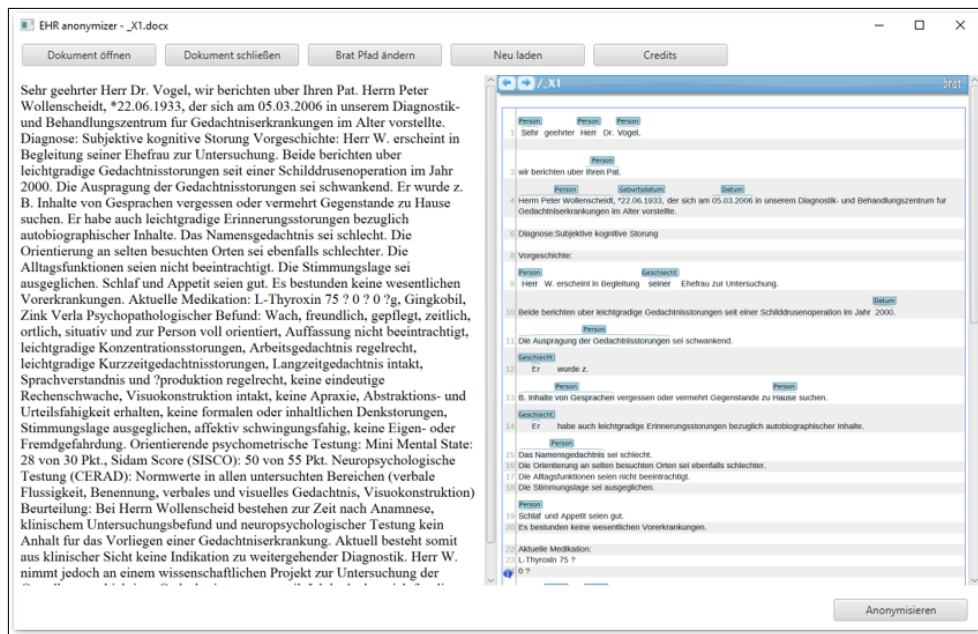
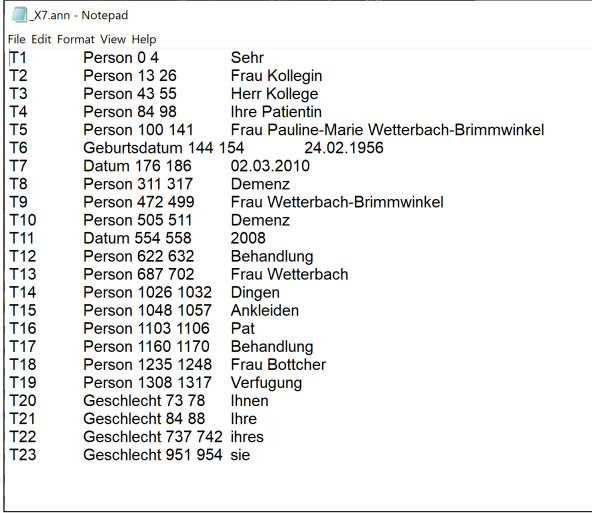


Figure 2.3: The main graphical user interface window is divided into the button row and the two web views that display the original and the annotated text.

The user can now add new annotations by logging into the BRAT tool via the BRAT menu and highlight words. New annotations can be set to the previously defined classes. Furthermore, annotations can be shifted to different positions or removed. When the user is ready and correctly annotated the text, the ‘Anonymisieren’ button will lead the user to the anonymization options menu, which displays all available options for anonymizing the classes (Figure 2.1, right part). When the user selects the options, the replacement mechanism will replace the tagged words with placeholders. After this, a last GUI window displays the whole anonymized text and requests the user to confirm that the text is completely anonymized and does not contain any personal information anymore.



## Annotation File



Term ID	Category	Start Index	End Index	Text
T1	Person	0	4	Sehr
T2	Person	13	26	Frau Kollegin
T3	Person	43	55	Herr Kollege
T4	Person	84	98	Ihre Patientin
T5	Person	100	141	Frau Pauline-Marie Wetterbach-Brimmwinkel
T6	Geburtsdatum	144	154	24.02.1956
T7	Datum	176	186	02.03.2010
T8	Person	311	317	Demenz
T9	Person	472	499	Frau Wetterbach-Brimmwinkel
T10	Person	505	511	Demenz
T11	Datum	554	558	2008
T12	Person	622	632	Behandlung
T13	Person	687	702	Frau Wetterbach
T14	Person	1026	1032	Dingen
T15	Person	1048	1057	Ankleiden
T16	Person	1103	1106	Pat
T17	Person	1160	1170	Behandlung
T18	Person	1235	1248	Frau Bottcher
T19	Person	1308	1317	Verfugung
T20	Geschlecht	73	78	Ihnen
T21	Geschlecht	84	88	Ihre
T22	Geschlecht	737	742	ihres
T23	Geschlecht	951	954	sie

Figure 2.4: Annotation File Example

The annotation file (".ann" file) is created and read by BRAT. It consists of the term identification number (T), annotation category (i.e. "Person"), index numbers (start and end index) and the term. Each section is divided by either a space or tab; this allows for BRAT to parse and label each term within the browser window.

## Feedback Loop: Introduction

The reason for creating a feedback loop was for our program to reference past annotations the NLP library failed to recognize. This allows each user to have a customized local history of annotations that will automatically be annotated in future anonymization. Only 4 out of 5 annotation categories can be stored for future reference, Names (Person), Organizations (Organisation), Addresses (Adresse), and Gender (Geschlecht). Date information is strictly tagged by the NLP library because the identification of dates has proved to be very accurate. Each category (excluding Date) has a corresponding CSV file to store newly annotated words (Persons.csv,

Adresse.csv, Organisation.csv, and Geschlecht.csv). The CSV file format was chosen for ease of access. A user may want to edit the annotation history and delete any annoying/repetitive annotations from the file. Therefore, a CSV file can be easily opened, searched, and then delete a term using Excel or any other spreadsheet software readily available on most computers. The feedback loop consists of 2 parts; Part I.) identifying additional annotations from past anonymizations (Figure 2.1, "FLE"), Part II.) determine if any new annotations need to be stored for later reference (Figure 2.1, "FLL").

### **Feedback Loop: Part I**

Initialization of the feedback loop starts as soon as the user uploads a document to be anonymized. Once the user clicks upload, the document is converted into a raw text string and then tokenized (all words are split and appended into a new list; "Token List"). The Token List is then used to compare with all the words stored within each CSV file. This is accomplished by translating each CSV file into a list and running a list comparison. If any words from the CSV lists are contained within the Token List, the common words are indexed and labeled with its corresponding annotation category and appended to the annotation file (".ann"). After all the common historical words from every CSV file is added, a list of all the annotations from the ".ann" file is stored to be used in part 2 (original annotation list). Once these procedures have concluded, the document is then loaded on the GUI and ready for further annotating by the user.

### **Feedback Loop: Part II**

After the user has finished annotating and confirm the document has been anonymized, part 2 is initialized. For our program to determine if any new annotations need to be stored for later, a final list (final annotation list) from the ".ann" file needs to be compared with the original annotation list. This comparison will find the differences between the original list (from part 1) and final annotation list. These differing annotations are

then added to the corresponding CSV file for later reference. Once this has been completed the final anonymized document is exported.

**Feedback Loop: Conclusion**

The feedback loop does allow for a customized recall of annotations from a user's past experiences. However, this method of storing and identifying additional annotations is far from perfect and further improvements can be implemented in the future. Some potential modifications will be discussed later.

## 3 Results

### Annotation Performance

To determine the performance of the program, statistical analysis was performed using all 10 health care record samples. For each document we determined the number of annotations that were correctly labeled (True Positives), the number annotations falsely labeled (False Positives), number of correctly non-labeled annotations (True Negatives) and the number of annotations not labeled but should have been tagged for anonymization (False Negatives). A confusion matrix was created and the precision, accuracy, recall, specificity, F1 score, false positive rate (FPR) and false negative rate (FNR) were calculated over all documents.

### Statistical Metrics

#### Precision

$$\text{Formula} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

#### Accuracy

$$\text{Formula} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{False Negative} + \text{True Negative} + \text{False Positive}}$$

#### Recall

$$\text{Formula} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Specificity

$$\text{Formula} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}$$

F1 Score

$$\text{Formula} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Doc	TP	FP	TN	FN
0X	14	11	200	0
1X	13	21	299	1
2X	14	7	134	3
3X	22	8	273	6
4X	30	20	507	2
5X	20	14	373	3
6X	17	18	604	1
7X	15	8	172	1
8X	21	9	373	2
9X	19	5	289	2
<b>Total</b>	<b>185</b>	<b>121</b>	<b>3224</b>	<b>21</b>

Score	Value
Precision	0.605
Accuracy	0.960
Recall	0.898
Specificity	0.964
F1 Score	0.722
FPR	0.037
FNR	0.102

Table 3.1: *Left Table: Displays all sample documents names (Doc) and their corresponding True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN) counts. Right Table: Displays the statistical metrics (FPR = False Positive Rate, FNR = False Negative Rate) calculated using the left tables TP, FP, TN and FN totals. .*

## 4 Discussion

### **Aim of the project**

The intention of this project was to build a BRAT-based tool that enables annotation of medical letters to anonymize privacy law protected patient data, including a automatic annotation of potential information. The automatic annotation is achieved with a combination of the Stanford natural language processing (NLP) library (Stanford named-entity recognition), the use of Regular expressions and the addition of prior user annotations ('Feedback loop'). The graphical user interface (GUI) is JavaFX based and displays the original text as well as the annotated text simultaneously. After human judgement, the final text is exported and saved. Because personal information needs to be completely eradicated from the input files, the main aim of the automatic annotation is to have a low number of false negative (FN) tags, which means that a part of the letter classifies as personal information, but is not recognized by the automatic annotation. The second priority is to prevent false positive (FP) tags, which are tags that are wrongly classified as personal information but do not contain any personal information, to occur.

### **Results interpretation**

The performance of the application was measured using 10 sample medical letters. Figure 4.1 displays the ratio of true positive tags (correctly labeled tokens), false positive tags (wrongly labeled tokens), true negative

tags (correctly unlabeled tokens) and false negative (wrongly unlabeled tokens) tags. The number of true negative tags dominates the chart, but since the letters contain only a small portion of personal information this is expected. The second largest portion of the diagram are true positive tags, which in combination with the previously explained metric leads to a high accuracy of 0.962 (Table 3.1) and shows that the automatic tagging is recognizing a very large portion of the personal information.

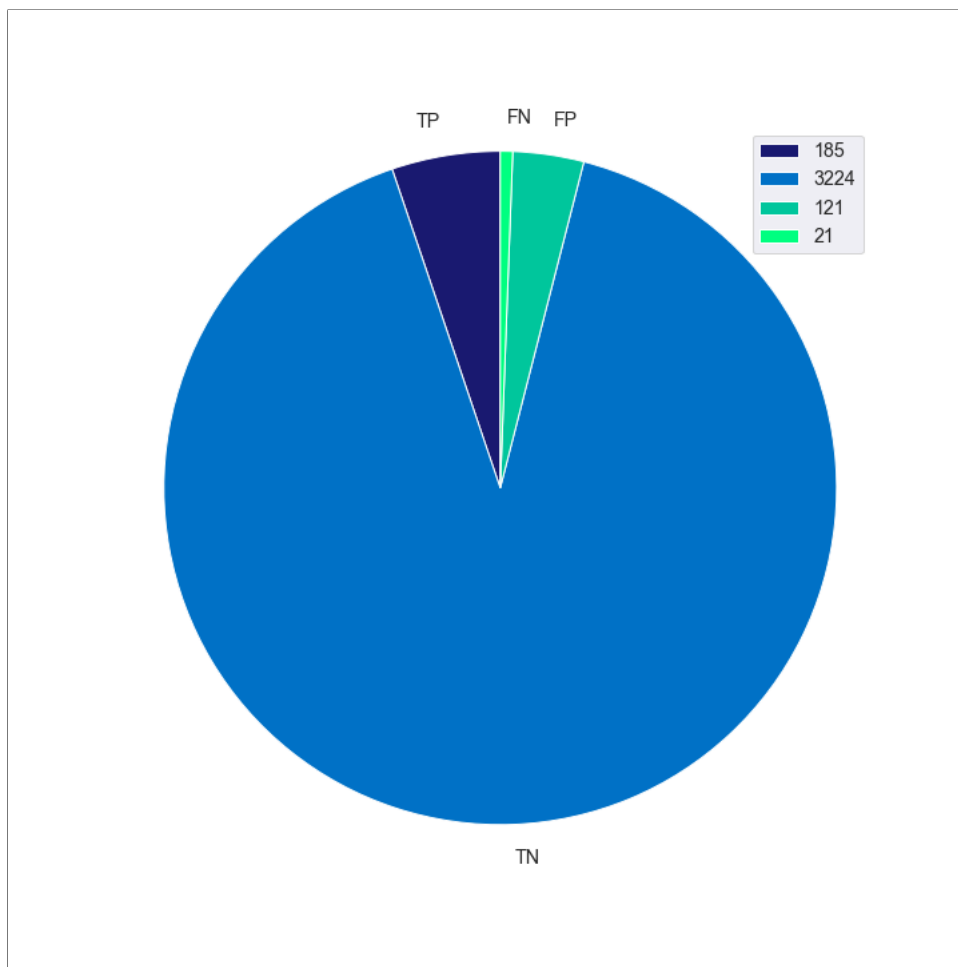


Figure 4.1: The performance of the electronic health care records anonymizer was measured based on 10 example letters. The pie diagram displays the total ratio between true positive (TP), false positive (FP), true negative (TN) and false negative (FN) results.

The relatively high number of false positive tags leads to a precision score of 0.605 (possible improvements for this are discussed in a later part), which shows that the automatic annotation falsely tags tokens as personal information. The biggest portion of these false positive tags rise due to the German language. Lots of surnames are nouns as well like 'Müller' or 'Bauer', and the automatic annotations tags these tokens in the text because they could be German surnames. In the end a higher false positive rate is more acceptable than a higher false negative rate. The number of false negative tags is illustrated by the false negative rate (FNR), which was 0.102 for the example letters. To reduce the number of FN tags without increasing the number of false positive tags, the application uses the 'Feedback loop', which saves human added and therefore non machine recognized tags to class separated comma-separated value (CSV) files. After the automatic annotations, the text is compared with these save files and similar words are tagged with the class the tag was saved in. This way, the number of FNs is reduced for future uses of the application. Further, the files can also be manually edited if the user wants to remove terms that are classified wrong due to very specific word uses etc. This gives the user more power and enables to specify the program for more precise field uses as frequently used terms will be saved and added to the automatic annotation. The F1 score is frequently used in natural language processing (NLP) classification and current NER tools trained on specific text libraries reach a F1 score of up to 100% [3], but since our tool uses German input files, common English NER classification metrics can not be compared.

### **Future work**

Since the results of a classification program mainly depend on the data and the corresponding trained model, first approaches to improve the program should be tackled at this point. EHR's differ from usual text cores in the frequency of technical terms of the medical field. For instance, diagnosis titles, e.g. "Alzheimer" and medication terms like "Bromazepam" really dominate the text corpora of EHR's. Those terms reveal to be quite prone



to be tagged as false positives. Since those terms are also tremendously important for the content of the EHR itself and for secondary purposes, the possibility of removing those terms a-priori does not really exist. One alternative would be a a-priori masking of these terms before annotating. However, this leads to an increased running time and with further model complexity. Alternatively, the training of NLP models with an increased focus on medical health care would be a serious way to improve it. Generally speaking, text mining models that use German text cores are sadly underrepresented in the NLP field and perform relatively badly in contrast to English trained models. A reinforced focus of researchers and developers in the text mining field on German text corpora can lead to an overall improvement regarding all application that NLP models as well.

Furthermore, to reduce the number of FPs, a 'black list' could be implemented, that saves words that are deleted by the user after the annotation process. The user could have a message coming up that asks if the words has a completely wrong tag or if the tag just does not make sense in this case. This way the chance to generate FN tags when using the 'black list' is lowered but still can not be ignored. The implementation of a 'black list' therefore needs to be well thought out.

Summarizing, the electronic health care anonymizer (EHRA) is a working tool for anonymization of medical letters. It is difficult to say how well our program will perform in a clinical setting due to the lack of training data. The automatic annotation could be improved in various ways but the main problem remains the nature of German language. Despite that, the utility and performance of the program nevertheless heavily depends on human judgement. With patience and good knowledge of German language, false negative and false positive scores are easily avoidable. Moreover, we are not medical professionals and comparing our ability to navigate through the application is still in question.

In conclusion, further testing and improvements are needed before distributing our software within any clinical setting.

# Bibliography

- [1] APACHE. Apache poi, 2019.
- [2] FOR MEDICARE MEDICAID SERVICES, C. Centers for medicare medicaid services (cms), 1965.
- [3] GUPTA, M. A review of named entity recognition (ner) using automatic summarization of resumes, 2018.
- [4] MANNING, CHRISTOPHER D, S. M. B. J. F. J. B. S. J. M. D. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60. (2014).
- [5] RUSS B. ALTMAN, M. L. What is biomedical data science and do we need an annual review of it? In *Vol. 1: i-iii* (July 2018).
- [6] STENETORP, P., PYYSALO, S., TOPIĆ, G., OHTA, T., ANANIADOU, S., AND TSUJII, J. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations Session at EACL 2012* (Avignon, France, April 2012), Association for Computational Linguistics.