



NTNU – Trondheim
Norwegian University of
Science and Technology

De-identification of Norwegian Health Record Notes

An Experimental Approach

Roar Bjurstrøm
Jaspreet Singh

Master of Science in Computer Science

Submission date: June 2013

Supervisor: Øystein Nytrø, IDI

Co-supervisor: Thomas Brox Røst, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Problem Description

De-identification of Electronic Health Records

To facilitate medical research and sharing of medical health information from Electronic Health Records (EHRs), the content must be anonymized, de-identified or pseudonymized in order to preserve patient confidentiality. Information is considered sensitive if it can be used to identify an individual, thus such information is privacy protected and encompassed by a number of statutes. Accordingly, such information must either be removed or replaced. Sensitive information includes information about persons (patients, health care providers, relatives), places (city names, street names) and institutions/agencies (medical clinics, hospitals, workplaces). De-identification performed manually on large sets of EHRs is time-consuming, prohibitively expensive and error-prone [1]; consequently, methods for automatic or semi-automatic de-identification is of scientific, health and commercial interest.

The task is to implement a java based application that de-identifies free text clinical notes, with the purpose of recognizing sensitive information. This is an experimental approach in which different methods and techniques will be applied in order to measure mutual performance, as well as the performance achieved by different method-combinations. The application will be developed and tested on clinical notes from a realistic Norwegian EHR-corpus. Finally, the application will be experimentally evaluated by its ability to detect sensitive information, on a manually annotated reference standard.

Supervisor: Øystein Nytrø, IDI

Norsk Sammendrag

Helsevesenet har de senere årene gått over fra papirbaserte til elektroniske pasientjournaler. Dette skaper en god del utviklingsmuligheter innen medisinsk forskning, pasientbehandling og utdanning, ettersom det blir mulig å nyttiggjøre digitale journaler på en helt ny måte. For å drive etisk forsvarlig medisinsk forskning på pasientjournaler, må journalene aidentifiseres eller anonymiseres hvorved konfidensiell informasjon må henholdsvis erstattes eller fjernes. Manuell aidentifisering er en tidkrevende og kostbar prosess, som setter store begrensninger på mengden pasientjournaler som blir gjort tilgjengelig.

Målet med denne oppgaven er derfor å lage en automatisk aidentifiseringsapplikasjon for norske kliniske fritekstnotater. Ettersom det ikke finnes tidligere studier direkte relatert til automatisk aidentifisering av norske kliniske fritekstnotater, har vi brukt en eksperimentell metode der vi utviklet en applikasjon basert på et utvalg av flere forskjellige metoder og eksperimenter. Disse har så blitt evaluert i kombinasjon med hverandre for å oppnå best mulig resultat. Dette har ledet oss til en endelig versjon av applikasjonen.

Applikasjonen er basert på mønstergjenkjenningsteknikker i tillegg til en enkel statistisk metode. Systemet er realisert på en måte som gir mulighet for justeringer av den statistiske metoden, der forskjellige justeringer er testet for å undersøke ytelsesforandringer. Resultatene er evaluert mot 225 selvannoterte kliniske notater, hvorav disse er utvalgt fra et realistisk pasientnotat-korpus. Systemets beste konfigurasjon gjenkjenner 77% sensitive identifikatorer, med en presisjon på 68%, samtidig som den lar store deler av pasientnotatets ikke-sensitive innhold forbli intakt med en feilklassifisering (fallout) på 5%.

Abstract

The conversion of paper-based health records to electronic health records creates new opportunities within medical research, medical education and patient treatment. However, electronic health records have to be de-identified or anonymized before disclosure, in order to conduct ethically sound research. Manual de-identification is time-consuming and costly, and thus limits the amount of health records that can be disclosed for research purposes.

The aim of this project was to develop an application for automatic or semi-automatic de-identification for Norwegian free text clinical notes. As no directly related studies have been performed on Norwegian clinical notes, our approach was highly experimental. We have employed different methods and techniques. These have been evaluated in different combinations to find the best match. The method combination which obtained the best evaluation results constitutes our final de-identification application.

The application we have developed is based on pattern matching techniques and a simple statistical method. It produces de-identified output which is evaluated against a manually annotated reference standard consisting of 225 clinical notes. Our best system configurations recognized 77% of the total 3320 sensitive identifiers, with a precision of 68%. Most of the insensitive contents remained intact with a fallout of 5%.

Preface

This thesis is submitted to the Norwegian University of Science and Technology for partial fulfillment of the requirements for a master's degree.

This work has been performed at the Department of Computer and Information Science, NTNU, Trondheim, with Øystein Nytrø as supervisor and Thomas Brox Røst as co-supervisor.

Acknowledgment

We would like to thank our supervisor Øystein Nytrø and co-supervisor Thomas Brox for good guidance and encouraging feedback throughout the project. The weekly EVICARE-meetings have been a source of many great and fruitful discussions; hence we would like to thank the EVICARE-team for providing new and better ideas for this project. Finally, we want to thank Karl Johan V Heimark for technical support and solving all of our latex related troubles.

Contents

Problem Description	i
Norsk Sammendrag	iii
Abstract	v
Preface	vii
Acknowledgment	vii
Contents	xiii
List of Figures	xvi
List of Tables	xviii
List of Abbreviations	xix
I Introduction	1
1 Introduction	3
1.1 De-identification Overview	3
1.1.1 Evicare	4
1.2 Assignment	4
1.2.1 Problem Description	4
1.2.2 Research Questions	6
1.2.3 Experiment	7
1.2.4 Constraints	8
1.2.5 Realistic EHR-notes	9
1.2.6 Simplifications	9
1.3 Report Outline	10
1.3.1 Part II	10
1.3.2 Part III	11

1.3.3	Part IV	11
-------	---------	----

II Background 13

2 Legislation 15

2.1	Personal Health Data	15
2.1.1	Personal Privacy	16
2.2	Medical and Health Research	16
2.3	Anonymization	17
2.4	Directly and Indirectly Identifiable Information	18
2.4.1	Directly Identifiable Information	18
2.4.2	Indirectly Identifiable Information	18
2.5	Anonymous Health Data	18
2.6	Pseudonymous Health Data	19
2.7	De-identified Health Data	20
2.8	Comparison	21
2.9	Health Registers	21
2.10	Relevant Theory	22
2.10.1	Protected Health Information (PHI)	23
2.10.2	Record Linkage	25
2.10.3	κ -Anonymity and ℓ -diversity	25

3 Medical Records 29

3.1	Electronic Health Records	29
3.2	EMR-systems	30
3.3	Benefits	30
3.4	Issues	30
3.5	De-identification	31

4 State of the Art 33

4.1	Methods	33
4.2	Pattern Matching	34
4.2.1	Pattern Matching and De-identification	34
4.2.1.1	Reference Works	34
4.2.1.2	Regular Expressions	36
4.3	Machine Learning	37
4.3.1	Text Classification and Data Mining	38
4.3.2	Machine Learning and De-identification	38
4.3.3	Part of Speech Tags	39
4.4	Evaluation Methods	40
4.4.1	Recall	41
4.4.2	Precision	41
4.4.3	Fallout	41
4.4.4	F-measure	42
4.4.5	Summary Example	42
4.5	Pattern Matching Approaches	43
4.5.1	Gupta	43

4.5.2	Neamatullah	44
4.5.3	Pantazos	44
4.6	Machine Learning Approaches	45
4.6.1	Dalianis	45
4.6.2	Uzuner	45
4.7	Hybrid Approach	46
4.7.1	Ferrandez	46
 III Implementation		49
 5 Introduction		51
 6 Preprocessor		53
6.1	Cleaning	54
6.2	Sentence Splitting	54
6.3	Part of Speech Tagging	54
6.4	Tokenizing	55
6.5	Data Structure	56
6.5.1	Token	56
6.5.2	Sentence	57
 7 Pattern Matching		59
7.1	Categorytag	60
7.2	Token-level Tagging	60
7.2.1	Regular Expression Matching	61
7.2.2	Dictionary Lookup	62
7.2.3	Medical Codes Search	63
7.3	Sentence-level Tagging	64
7.3.1	Phrase Search	65
7.3.2	N-Gram Search	68
7.3.3	Dosage Matching	69
7.4	Handling Unmatched Tokens	69
7.4.1	Stemming	69
7.4.2	Innertokens	70
 8 Statistical		71
8.1	Weighting	71
8.2	Inverse Document Frequency	71
8.2.1	Formula	72
8.2.2	Supported Features	73
 9 Classification		75
9.1	Classification Algorithm	75
9.2	Classification Summary	77
 10 Postprocessor		79

IV	Results	81
11	Experiment Preparations	83
11.1	Reference Standard	83
11.1.1	Note Types	83
11.1.2	Annotation	84
11.1.3	Choices	84
11.2	Modified Discharge Summaries	85
11.3	Properties	85
11.4	Maximum Token IDF	85
12	Results	87
12.1	Experimental Approach	87
12.2	Regular Expressions and IDF	87
12.2.1	Results	88
12.3	Pattern Matching	91
12.4	Pattern Matching and IDF	93
12.5	Modified Discharge summaries	97
13	Discussion	99
13.1	Sources of Errors	99
13.1.1	Evaluation Method	99
13.1.2	Reference Standard	100
13.1.3	Classification	100
13.2	Performance	100
13.2.1	Statistical vs Pure Pattern Matching	100
13.2.2	All Methods	101
13.2.3	Note Types	101
13.3	Structured vs Unstructured	102
14	Conclusion	103
14.1	Conclusion	103
14.1.1	Question 1	103
14.1.2	Question 2	103
14.1.3	Question 3	104
14.2	Further Work	105
14.2.1	Rule Base	105
14.2.2	Machine Learning	105
14.2.3	Preprocessor	105
14.2.4	POS-tagger	106
14.2.5	Compound words	106
	Bibliography	112
V	Appendix	113
	Appendices	

Appendix A Regulation	115
A.1 Regulation on health records, §8	115
Appendix B Results	117
B.1 Regular Expressions and IDF	117
B.2 Pattern Matching and IDF	119
Appendix C Unicode Whitespace Table	123
Appendix D Input-Output Example	125

List of Figures

1.1	A fictitious clinical note	5
2.1	Pseudonym example	20
2.2	Illustration of the clinical data types	20
2.3	κ -Anonymity example	26
2.4	Example of 4-anonymous and 3-diversive information	26
4.1	POS-tag example	39
4.2	POS-tag ambiguities	40
5.1	Overview of the pipeline	51
6.1	The main elements in the data structure	53
6.2	Overview of the preprocessor pipeline	53
7.1	Overview of the pattern matching	59
7.2	Overview of the <i>token</i> -level tagging	60
7.3	Overview of the regular expression matching	62
7.4	Overview of the dictionary lookup component	62
7.5	Example medical codes search	64
7.6	Overview of the sentence-level tagging	65
7.7	Overview of the phrase search	68
7.8	Stemming example	70
8.1	Overview of the statistical implementation	72
10.1	Example output	79
12.1	De-identification results on the entire reference standard using IDF and regular expressions, with <i>sentence-idf</i> = 2	88
12.2	De-identification results on entire reference standard using IDF and regular expressions, with <i>sentence-idf</i> = 4	89
12.3	De-identification results on discharge summaries using IDF and regular expressions, with <i>sentence-idf</i> = 2	89

12.4	De-identification results on discharge summaries using IDF and regular expressions, with <i>sentence-idf</i> = 4	90
12.5	De-identification results on nursing notes using IDF and regular expressions, with <i>sentence-idf</i> = 2	90
12.6	De-identification results on nursing notes using IDF and regular expressions, with <i>sentence-idf</i> = 4	91
12.7	Results on all notes	95
12.8	Results on discharge summaries	95
12.9	Results on nursing notes	96
12.10	Results on record notes	96
D.1	The output produced by our de-identification application on the fictitious clinical note presented in the introduction 1.1	125

List of Tables

4.1	Regular expression for national identification numbers	37
4.2	Definitions used in binary classification tests	40
4.3	Example distribution of Protected Health Information (PHI) and non-PHI in an Electronic Health Record (EHR)	42
4.4	Output of an example de-identification process	42
4.5	Result of an example de-identification process	42
4.6	Evaluation results achieved by Neamatullah et al.'s de-identification application	44
4.7	Evaluation results achieved by Pantazos et al.'s de-identification application	45
4.8	Evaluation results achieved by Dalianis et al.'s de-identification application	45
4.9	Evaluation results achieved by Uzuner et al.'s de-identification application	46
4.10	Evaluation results achieved by the BoB-application on VHA clinical documents	47
4.11	Evaluation results achieved by BoB-application on the i2b2-challenge corpus	47
6.1	Example of problematic output	55
6.2	Example of problematic output	55
6.3	Example of innertokens	56
7.1	The two example vectors used in the cosine similarity example	66
11.1	The amount of sensitive and insensitive tokens in the reference standard .	85
12.1	The pattern matching results on all note types	92
12.2	The pattern matching results on discharge summaries	92
12.3	The pattern matching results on nursing notes	92
12.4	The pattern matching results on record notes	92
12.5	Tuning	94
12.6	Result on modified discharge summaries	97
14.1	Best results	103

B.1	The statistical component's results on the entire reference standard . . .	117
B.2	The statistical component's results on discharge summaries	117
B.3	The statistical component's results on nursing notes	118
B.4	The statistical component's results on record notes	118
B.5	Results on all notes	119
B.6	Results on discharge summaries	120
B.7	Results on nursing notes	120
B.8	Results on record notes	121
C.1	Unicode space characters	123

List of Abbreviations

ATC Anatomical Therapeutic Chemical Classification System.

BoB Best of Breed.

BoW Bag of Words.

CDS Clinical Decision Support.

CRF Conditional Random Fields.

CVK Central Venous Catheter.

EHR Electronic Health Record.

EMK the European Convention on Human Rights.

EMR Electronic Medical Records.

GIC Group Insurance Commission.

HIPAA Health Insurance Portability and Accountability Act.

i2b2 Informatics for Integrating Biology and the Bedside.

IAIMS Integrated Advances Information Management Systems.

ICD International Classification of Diseases.

ICPC International Classification of Primary Care.

ICT Information and Communications Technology.

ICU Intensive Care Unit.

IDF Inverse Document Frequency.

IDI Department of Computer and Information Science.

IG Information Gain.

IR Information Retrieval.

MeDLEE Medical Language Extraction and Encoding.

MeSH Medical Subject Headings.

MIMIC Multiparameter Intelligent Monitoring in Intensive Care.

NCMP Norwegian Classification of Medical Procedures.

NCRP Norwegian Classification of Radiological Procedures.

NCSP The NOMESCO Classification of Surgical Procedures.

NLP Natural Language Processing.

NTNU Norwegian University of Science and Technology.

PHI Protected Health Information.

POS Part of Speech.

REK Regional Committee for Medical and Health Research.

RIDF Residual Inverse Document Frequency.

SNOMED Systematized Nomenclature of Medicine Clinical Terms.

SOAP Subjective Objective Assessment Plan.

SSB Central Bureau of Statistics.

SVM Support Vector Machine.

SVO Subject Verb Object.

TPF Trusted Pseudonym Agency.

UMLS Unified Medical Language System.

VHA Veterans Health Administration.

Part I

Introduction

1.1 De-identification Overview

De-identification denotes the process of removing/replacing sensitive information. This thesis propose to implement a de-identification application with the purpose of removing sensitive information from free text clinical notes. Automatic or semi-automatic de-identification is getting more valuable as health records are stored digitally in Electronic Health Records. An Electronic Health Record (EHR) consists of several types of documents, including medical history, allergies, laboratory results, pathology reports, discharge summaries etc. Health data also contains a lot of sensitive information, including name, date of birth, relatives, address and several other sensitive elements that can be used to identify the person concerned. Sensitive information is legally privacy protected, making EHRs visible to a limited audience.

When EHRs are de-identified, i.e. when confidential information and person identifiable information are removed, these are legally outside the privacy protection which in turn makes them available to a broader range of audience[2]. Large parts of medical research are based on de-identified EHRs, but as practiced today, these EHRs are manually de-identified. Conducting medical research needs an adequate amount of de-identified data in order to reach statistically significant conclusions, whereby the validity of the conclusions increases with the amount of data. Manual de-identification is time and resource consuming, thus making automated or semi-automated de-identification tools attractive. The win-win compromise allows analysts to use this information, while preventing potential identity thieves from identifying the individuals.

By disclosing clinical documents and making them accessible to a wider range of audience, it might be easier to develop creative and innovative patient care tools, especially with a view to technological progress within other fields as data mining and information extraction (discussed in section 4.3.1). Tools can be developed for recognizing and tag standard medical terminologies with the corresponding codes within an EHR, which in turn can be used to enrich and adapt EHRs for other NLP-tools. Useful tools, like decision support systems, can be developed for health care workers in order to make proper decisions and somewhat provide relieve from the everyday stress.

1.1.1 EviCare

The de-identification task is given by the Healthcare Informatics-section at Department of Computer and Information Science (IDI) in connection with the Norwegian research project EviCare (Evidence-based care process). The aim of the EviCare-project is to develop valid tools, and produce high-quality research within the field of Clinical Decision Support (CDS). In short, CDS systems are designed to assist health care providers who have decision making tasks as a part of their daily activities, by contributing with choices and decisions at the point of care. CDS is a major topic of artificial intelligence in medicine[3].

De-identification systems are not directly related to CDS-systems in particular, but we are still a part of the EviCare-team; having access to the same realistic corpus of EHR-notes (described further in section 1.2.5) with the purpose of developing tools of health care interest. We also cooperate during our research and utilize efforts across projects. The EVICARE project-description can be found through Regional Committee for Medical and Health Research (REK)s homepage (see [4]), with the reference number 2010/3380.

1.2 Assignment

This section presents our interpretation of the assignment, and the objectives we want to accomplish through this project.

1.2.1 Problem Description

To facilitate medical research and sharing of medical health information from EHRs, the content must be anonymised or pseudonomized in order to preserve patient confidentiality. All sensitive information must either be removed or de-identified. Information which can be used to identify an individual is considered as sensitive information. Such information includes name, date of birth, sensitive information about relatives, address, hospital name, patient number, workplace etc.

De-identification performed manually on large sets of EHRs is time-consuming, prohibitively expensive and even error-prone[1]. Dorr et al.[5] evaluated the time cost to manually de-identify narrative text notes (average of 87.2 ± 61 seconds per note), and concluded that it was difficult to exclude all PHI required by Health Insurance Portability and Accountability Act (HIPAA)¹ (HIPAA is further described in section 2.10.1). M. Douglas et al.[6] demonstrated that the recall (discussed in section 4.4.1) of a human expert working alone ranged from 0.63 to 0.93, with an average of 0.81, whereas an algorithm based on pattern matching, lookup lists and common sense heuristics achieved a recall value of 0.85; consequently, methods for automatic de-identification are of both medical research and commercial interest.

¹ In US, the Health Insurance Portability and Accountability Act has defined 18 personal health identifiers, which are required to be removed from medical records before disclosure.

De-identification is an important step towards making clinical data more available and facilitate for useful applications. Having access to automatic de-identification software makes it more efficient and cost-effective to obtain a de-identified corpus of a desired subset of clinical notes, for example leukemia²-patients with a specific blood type and age, and further disclose the corpus for cancer-research or other objectives; In other words, easier and cost-effective access to tailored corpuses. On a more general basis, automatic or semi-automatic de-identification can serve good and valuable purposes within medical education.

There are also several technological benefits. De-identified clinical notes are important in the field of Natural Language Processing (NLP) in order to develop new tools and concepts for improving patient care, such as information extraction and information retrieval. De-identified EHRs may provide more secure sharing of clinical data across health institutions, which today is considered unsafe due to the degree of information visibility.

Dato: 18.06.1999
Godkjent av: Ola Nordmann
Dokumentnummer: 32313478

INNKOMSTJOURNAL

Innleggende lege: Amk-sentralen

Pasient: Nils Hansen

Alder: 53 år

Diagnose: Brystsmerter

Tidligere sykdommer:

Pasienten hadde hjerteinfarkt i 1995.

Aktuelt:

Smerter og nummenhet som startet i venstre hånd og spredte seg oppover og mot hjertet ca kl 14 i dag. Kvalm, ingen effekt av nitro. Ikke økt dyspnoe, ikke vært kaldsvett. Får nitro og morfin i ambulansen, og innkommer smertefri

Medikamenter:

Fragmin 2500 IE x 1

Panodil 500 mg x 4

Nexium 40 mg x 1

Figure 1.1: A fictitious clinical note where red lettering expresses sensitive information.

In particular, the main objective of our project is to implement an application that recognizes sensitive information in free text EHR-notes. The application will consist of

²A type of cancer of the blood or bone marrow

several modules performing de-identification on different levels (word, sentence, statistical), whereby each module will function independently and their respective outcomes will be further used to make decisions. The modules will be based on different techniques, and as this approach is experimental, one sub-goal is to investigate the performance of the modules (the research questions of this experimental approach is presented in the next section). The application will be realized as a java based application, which will be developed and tested on a realistic corpus. Finally, it will be evaluated on a self-annotated reference standard (discussed in section 1.2.3), module by module and in module-combinations experimentally, and measure its ability to detect sensitive information on a predefined set of patient notes. The performance will be measured in recall, precision and f-measure, which are presented in section 4.4.

1.2.2 Research Questions

The main purpose of this project is to implement an application that detects sensitive information in Norwegian EHR-notes. Further, the aim is to answer some research questions. The primary goal is to make a best-effort applications. However, a certain focus is also directed towards experimenting with different techniques and methodologies, rather than only focusing on the end-product. Our research questions reads as follows:

- How well, in terms of recall, precision, fallout and f-measure, can we implement a de-identification application in the course of a semester, on the basis of rule-based and simple statistical methods?
- How will different combinations of the implemented algorithms and techniques affect the performance?
- Can such a system be realized in the Norwegian health care system and replace/simplify manual annotation/de-identification?

The first and main question: How well, in terms of recall, precision and fallout, can we implement a de-identification application in the course of a semester, on the basis of rule-based and simple statistical methods? Considered the constraints of this project: lack of a Norwegian gold standard, lack of annotated training data (for machine learning purpose, see 4.3) and strict time limitation, the aim is to implement a best-effort de-identification application within a semester (5 months). This question will be answered by evaluating the performance through the performance measures recall, precision and fallout against a self-annotated reference standard. The performance measures and the interpretation of these will be elaborated later, see section 4.4.

The second question: How will different combinations of modules and methods affect the performance? Our approach will be to experimentally de-identify EHR-notes using different combinations of techniques; One-by-one or in various combinations. One example would be to compare a regular expression-module combined with statistical methods against dictionary lookup-module combined with statistical methods, and further observe how well these perform together. It can be interesting to observe in which cases wrong decisions are made due to disturbance between the modules, and to what

extent the final decision of one module should be prioritized above another. This part of the experiment will emphasize pros and cons of different techniques used in the application. As far as the question is concerned, the answer will partially be used to sharpen and finalize our application in order to achieve best possible performance and improve the end-result according to the first research question. Furthermore, the answer may serve as valuable information for further work on de-identification of Norwegian EHR free text notes.

The last question: Can such a system be realized in the Norwegian health care system, and replace/simplify manual annotation/de-identification? This question is more of a discussion topic rather than a question to be answered through an experiment. Since we already know that perfect de-identification applications for EHRs are very difficult, if not impossible, to develop, we already know the partial answer to this question; Experiments are broadly speaking needless to prove this. The purpose of asking such a question is to look at issues related to automatic de-identification from different viewpoints and open a more fundamental level of discussion of the challenges involved; what is perfect de-identification/anonymization according to the legislation, who holds the information (EHRs), constraints within the public health service' information systems, EHR-standards and their impacts, and to what extent our results can benefit today's practice.

1.2.3 Experiment

The experiment will consist of several parts, as mentioned in research question one and two. The application will mainly employ (thoroughly described in chapter 5) regular expressions (see 4.2.1.2) and reference works (see 4.2.1.1) together with simple statistical methods, whereby the decisions made by the statistical methods will be based on certain thresholds. Each of these will be implemented independently in such a way that they easily can be enabled and disabled. Hence, the experimental approach is to run the application using different combinations of the aforementioned methods, and use different thresholds whenever the statistical methods are enabled. The obtained results will form the basis for answering the second research question, and the method-combination achieving best results will obviously constitute our final de-identification application.

Reference Standard

A selection of clinical notes from the Norwegian EHR-corpus will be annotated manually in order to evaluate the performance. We will perform the annotation ourselves, which will be somewhat simplified, as manual annotation is challenging and often carried out by domain-experts. In order to make the best of the situation, our only choice is to perform a best-effort annotation and prepare a simplified reference standard.

A more detailed description of reference standard and the annotation is provided in chapter 11, under section 11.1.

1.2.4 Constraints

There are several constraints affecting the project, these will be presented and discussed in this section.

Time

The development of de-identification software is a time-consuming process and requires a lot of organization and preparations. First of all, similar de-identification projects from the state of the art review (chapter 4), shows that robust software based on pattern matching requires a rich and comprehensive rule-base. The time is limited to one semester (5 months), with a team consisting of two students. In order to develop a comprehensive rule-base and other relevant functionalities we first have to get a basic preprocessor going which will take some time, and then start off building the rule-base which will be enriched gradually. The main challenge is to make generalized rules in order to detect different types of sensitive information, on different types of clinical notes, which requires incremental method of work by trying - failing - improving. Since automated de-identification of free texts still is on a research stage internationally, there is not any concrete solution to the task that can be followed; neither any directly related work which can be built upon. Hence, the required preparations will place a noticeable time-limitation on the project.

Almost all work will be done from scratch as no previous work directly relates to de-identification of free text Norwegian clinical notes. This is the main reason for choosing and framing our first research question, presented in the previous section (1.2.2): How well, in terms of recall, precision and fallout, can we implement a de-identification application in the course of a semester, on the basis of rule-based and simple statistical methods? Knowing that other similar projects (see sec 4.5 and 4.6) have been conducted by large and professional teams over long periods in contrast to our team consisting of two students, we know it will be difficult, thus chose to not set any concrete and unattainable goals which cannot be reached at the end of this semester. Hence, the focus is to observe how well we are able to implement a de-identification application within a semester, and additionally initialize a valuable project for the Norwegian health care service.

Gold Standard

A significant part of an experiment is to perform a test with measurable outcome. The performance of a de-identification application is often considered with respect to a gold standard, which is a manually annotated corpus. A gold standard is worked out by human judges (professionals, i.e. linguists or domain experts), where the de-identification is based on a number of criteria. It does not exist any Norwegian gold standard for our purpose. Consequently, we will not get the opportunity to compare our results against a professional prepared solution, and are forced to define our own criteria. Some phrases can be difficult to decide whether they reveal sensitive information or not,

even for professional annotators. Hence, the lack of a gold standard is a major limitation to the validity of the results.

The lack of any annotated data also makes it difficult to use machine learning algorithms, because they need sufficient annotated training data. This is another apparent limitation as it prevents us from make use of off-the-shelf machine learning tools, which could have been of great value. Machine-learning based de-identification software has shown to achieve satisfactory and robust results. A few machine learning approaches will be presented later, in section 4.6.

Late Start

Since the EHR-notes contain extremely sensitive information, and several other students needed access to the notes for their respective projects, IDI decided to set up a separate lab. This lab has restricted admission, limited to students who have relevant projects. The EHR-lab was ready to use almost one and a half months after semester-start, thus it took some time before we got the chance to test our application on real EHR-notes and discover the corpus-related bugs.

1.2.5 Realistic EHR-notes

As part of the EviCare team we have access to EHR-notes. IDI received the data this semester, intended for students working on EviCare-projects. The corpus contains over 40000 various types of free text notes, based on 800 hospital stays. Another feature of the corpus is that among the 800 patients, there are 152 known patients with Central Venous Catheter (CVK). Furthermore, the dataset also include a partitioned sample in which the corpus has been divided into ten subsets, whereof each subset contains equal quantity of CVK-patients. However, these features are contemplated to students working on projects related to classification of medical contents in the notes, in case someone wants to perform a 10-fold cross validation. In terms of de-identification, these features are rather irrelevant.

The corpus is located on a net-disconnected server inside a separate lab which is a sensitive and secure zone. The files are handed over in text (UTF-8) – and html format. The naming of the text files also follows a pre-defined pattern, and reveals certain information about the notes, in order to follow particular patients or dates. This is also a less relevant property, thus not utilized by our de-identification application.

1.2.6 Simplifications

It is often a challenge to identify sensitive information, even manually[1]. The main challenge is to decide on what should be classified as so-called indirectly identifiable information, which will be explained in 2.4.2. In short, the annotator has to assume the knowledge of a potential intruder, and foresee that this knowledge together with the disclosed information do not link to an individual. The automation of this process requires

complex analysis in which κ -anonymity and ℓ -diversity (discussed in section 2.10.3), among others, are frequently used techniques. These are often used on structured data and require structuring of the free text notes, which in turn is more than we are able to do within a semester.

We have made some simplifications and narrowed down the range of information regarded as sensitive, whereby purely context based indirectly identifiable information will not be considered, such as: demographic information, background information and information revealing living conditions and circumstances. Below follows examples of phrases revealing such information:

- *The patient has a daughter and a son*
- *Drug addict, bruised face and missing incisors*
- *Served in the war against Iraq for two years, lost three of his closest buddies, clearly traumatized*

All of these sentences reveals indirectly identifiable information. The first example reveals that a patient has a daughter and a son. This piece of information is enough to significantly narrow the range of potential individuals, and further identify the person concerned by additional knowledge or information that might as well be present in the EHR.

The same applies for the two latter examples, besides, these also reveal recognizable characteristics which actually might be utilized to directly identify the concerned individuals.

1.3 Report Outline

The rest of the report is organized as follows:

1.3.1 Part II

Chapter 2 introduces relevant Norwegian legislation related to de-identification. The chapter provides a basic understanding of personal privacy and patient security, together with an explanation of de-identified, anonymized and pseudonymized patient data, and further the legislative limitations regarding the access and processing of clinical documents.

Chapter 3 gives an overview of EHRs, and further discusses the opportunities and challenges brought into light by the technological progress within the health sector, as well as the impact of an automatic de-identification application.

Chapter 4 presents relevant de-identification techniques, with main focus on machine learning and pattern matching methods, followed by a state of the art investigation.

1.3.2 Part III

Chapter 5 provides a short and basic overview of the implemented application.

Chapter 6 describes the implementation of the preprocessor, and how the clinical documents are prepared by our application, before the actual de-identification.

Chapter 7, 8, 9 describes the implementation of the pattern matching- and statistical component together with the classification algorithm, respectively, and provides detailed description of the implemented techniques, as well as the choices that have been made during the development.

Chapter 10 describes the implementation of the postprocessor.

1.3.3 Part IV

Chapter 11 explains the preparations that were made to facilitate the experiment.

Chapter 12 presents the results.

Chapter 13 provides a discussion about the reliability of the obtained results, and decisive sources of errors.

Chapter 14 summarizes the project and provides a conclusion by answering the initially asked research questions.

Part II

Background

Automatic or semi-automatic de-identification of free text clinical notes can be time-efficient, cost-effective and as good as manual de-identification[1][6]. The solution of the task mainly involves technical and scientific methods within text technology and NLP, however, it is equally important to analyze and get familiar with the actual problem before jumping to the conclusion. In our case, the main obstacle is the Norwegian legislation that places certain limitations on disclosure of health information. This chapter gives a brief overview of relevant rules and regulations according to Norwegian legislation, and provides legislative restrictions that forms the basis for the de-identification task.

2.1 Personal Health Data

Research on personal health data is regulated by the *Health Research Act*[7], where the purpose is to promote secure and ethically sound health research. As a matter of form, it is appropriate to provide the definition of personal health data, as a start-up, in regard to our actual motivation of de-identifying health data. According to §4 in the *Health Research Act*, personal health data is defined as:

“Confidential information pursuant to §21 of the Health Personnel Act and other information and assessments concerning health issues or that are significant for health issues that can be linked to an individual person.”

The clause mentions confidential information as a part of personal health data, which in turn is defined in the *Health Personnel Act*[8], cf. §21:

“Health personnel shall prevent others from gaining access to or knowledge of information relating to people’s health or medical condition or other personal information that they get to know in their capacity as health personnel.”

As per the above-mentioned paragraphs, EHRs are considered as personal health data, and hence have to comply with these.

Handling personal health data is comprised by rigid and strict regulations, which in turn ensures organized and responsible research. There are also several other acts which deals with personal health data, supplementing the Health Research Act (cf. §2 third paragraph), i.e. *Personal Data Act*[9] and *Personal Health Data Filing System Act*[10], whereby the latter will be further described in connection with *health registers* (discussed in 2.9). The essence of the *Health Research Act* is to maintain personal privacy and patient security.

2.1.1 Personal Privacy

The law of privacy is defined in the *Personal Data Act*[9]. Personal privacy mainly consists of two parts, personal integrity and personal data. Personal integrity deals with the social perspective of individual privacy as self-determination and self-expression, such as freedom of speech. Put differently, the right to live an autonomous and private life. Personal data privacy on the other hand, is more of our interest and defines norms and standards for processing personal data. The aim is to ensure that each individual controls the publicly available knowledge about his/her private situation (social status, age, address etc) . This is often interpreted as the technical part of personal privacy. It states that every individual has the ability to control when-, how- and how much personal information is disclosed to other parties. Personal privacy follows from "the European Convention on Human Rights (EMK) - article 8" , and provides a right to respect for one's private and family life. Privacy is thus a basic human right that must be respected in all circumstances, hence also highly relevant within the health sector.

The public health service is dependent on health information in order to give each patient proper and individualized health care. Information concerning illnesses, diseases, lifestyle, habits and other health related information is substantial to the health service in order to provide best possible treatment. In worst case, the lack of adequate personal health information may lead to wrong treatment; hence access to personal health information is necessary. Health care providers are obliged to keep secret all acquired information about patients in the course of their professional work. This is regulated by the *Health Personnel Act*[8], which has to be complied by doctors, nurses, physiotherapists and all kind of health personnel.

2.2 Medical and Health Research

There are two clauses in *the Health Research Act*[7] §13 & §20 which frames the terms for this very project. The first clause, cf. §13, states:

"Consent must be obtained from participants in medical and health research, unless otherwise laid down in law."

Personal health data is privacy protected, hence consents from respective participants are demanded. This has to be complied with when performing research on personal health data. Hence, when medical research projects depend on health data together

with the belonging person identifiable information (described in 2.4), consent is required from each and every participant, at least pursuant to the principal rule. There are some exceptions to this rule, but they are rather irrelevant in our case. Even if consents are acquired, medical research cannot be conducted on health data without further ado. In addition, an application has to be prepared and further evaluated by the *REK*, pursuant to the Health Research Act, §33:

”Research projects must have authority to process data. Prior approval from the regional committee for medical and health research ethics (REK) in accordance with Chapter 3 is necessary and adequate authority to process personal health data in medical and health research. REK is an administrative body, and has to follow The Public Administration Act when considering applications for medical research.”

Applications are thoroughly examined by REK in order to assure that projects preserve personal privacy, and promote ethically acceptable medical and health research. Our very EviCare-project (described in 1.1.1) was examined in a similar manner before disclosing the Norwegian EHR-notes corpus.

2.3 Anonymization

Medical research projects on health data can vary regarded purpose, size, budget, period of time etc., and the kind of health data required for the projects depends accordingly. Such variations may include document types, structured or unstructured data, specific diseases, geographic distribution and several other specifications. However, the common property of research corpora is often high quantity; Research projects often requires a sufficient amount of data in order to obtain statistically significant results, which in turn involves procurement of consents from each participant. Consequently, big datasets will demand a lot of consents, which in turn is time-consuming and costly. However, §20 of the Health Research Act[7] provide a means of solution to this impracticality, and is the other main clause defining the boundaries for our project:

”Consent is not required for research on anonymous human biological material and anonymous data.”

As addressed in the beginning of this chapter, the main purpose of the *Health Research Act* is to promote ethically sound medical and health research, whereby one of the many risks is that health data falls into the wrong hands. However, when sensitive information is removed from the persona health data, the health data is not considered personal anymore, as it no longer meets the requirements (2.1), cf. §4, *Health Research Act*, whereby the consents are needless. This alternative may be more suitable for projects conducting medical research purely on the medical contents of health data, independent of identifiable information.

Research on anonymous data still requires approval from REK, cf. §35, *the Health Research Act*. Furthermore, the legislation defines three levels of anonymization that can be performed before disclosure, whereby each level is anonymized differently depending on the purpose, and more importantly, by REK’s judgment.

Further, the distinction between directly and indirectly identifiable information will be explained, before moving onto the three anonymization-levels.

2.4 Directly and Indirectly Identifiable Information

Information is often divided in sensitive and insensitive information, whereby sensitive information is subdivided in directly and indirectly identifiable information.

2.4.1 Directly Identifiable Information

Directly identifying information is information about an individual X, wherewith this information exclusively, individual X can be identified. Evident examples of such information are name, address and phone number. This type of information usually easy to catch, and there is rarely any doubt whether a string of characters signifies directly identifying information. Besides, if there should be any doubt, the information is most probably indirectly identifying and hence has to be removed.

2.4.2 Indirectly Identifiable Information

Indirectly identifiable information does not reveal any identity in itself, but on the other hand exposes explicit identifying variables (gender, age, county, municipality etc.) which can be used to reveal the concerned individual's identity, by combining the disclosed variables with certain background information. Such background information is called quasi-identifiers[11]. Hence, the patient referred to in a clinical note wherein direct identifying information is removed, can still be re-identified by a potential intruder if he knows exactly the right quasi-identifiers.

Quasi-identifiers may come from various types of registers which are made publicly available or just common knowledge acquired through internet or acquaintance. For instance birth registers; it is quite common that parents announce the exact weight of their babies after the birth. Assuming a publicly available birth register with the age of mother, hospital of birth and the baby's weight, the age will make most births unique. Hence, the baby's exact weight can turn out to be a powerful quasi-identifier, providing potential intruders to obtain mothers age and hospital of birth.

2.5 Anonymous Health Data

The definition of anonymous health data is given by *the Health Data Filing System Act*[10], 2nd section, 4th paragraph:

"Data from which the name, national identity number and other characteristics serving to identify a person have been removed, so that the data can no longer be linked to an individual person."

Note that it should be impossible to link anonymous data to an individual, something which never can be guaranteed unless everything is removed from the document. In order to perform perfect anonymization, one must be familiar with all knowledge the receiving party holds, including background information (quasi-identifiers 2.4.2), which in turn is impossible. The risk of linking attacks (described in 2.10.2) makes the task of anonymization challenging and requires that all possible quasi-identifiers are carefully considered such that a sufficient amount of variables are removed, in order to declare the data as anonymous. Sometimes, this also involves that medical contents are removed from the data, for example due to a rare and recognizable disease, which unfortunately results in research material of poor quality.

As practiced today, a dataset is anonymous if the information can be associated to 4 or 5 individuals[2]. In theory, a dataset is anonymous as long as it can be linked to 2 individuals. However, as more information is digitized the risk of information leakage through unauthorized access increases, thus the limitation is adjusted accordingly.

2.6 Pseudonymous Health Data

Pseudo is Greek for false whereby a pseudonym is used to give patients false identities. Pseudonymous health data is health data, in which directly and indirectly identifying information is replaced with pseudonyms. For instance the name Peter could be replaced with the string AXB77, in which AXB77 is the pseudonym, with a direct mapping to the name Peter. Pseudonymizing health data provides the possibility to follow patients over a period of time by their pseudonymous identity, and most importantly without identifying the concerned individuals in real life. This is also possible across some health registers, which will be further described in section 2.9. This possibility adds a new dimension for researchers and might be very valuable.

The definition of pseudonymous health data is given by *the Health Data Filing System Act*[10]:

Personal health data where the identity has been encrypted or otherwise concealed, and nonetheless individualized serving the possibility to follow each patient through the health system without the identity being revealed.

It is required that a system for person-unique pseudonymisation is established by a trusted third party or a so-called *Trusted Pseudonym Agency (TPF)*. One example of a trusted pseudonym agency is *Central Bureau of Statistics (SSB)*. The main principle is that no authority should have access to both the health data and the mapping between the pseudonyms and the true identity. For instance, the *Prescription Register*[12] which is a central health register (see section 2.9), is a register containing pseudonym health data where *SSB* is the corresponding *TPF*. Figure 2.1 provides an simple illustration:

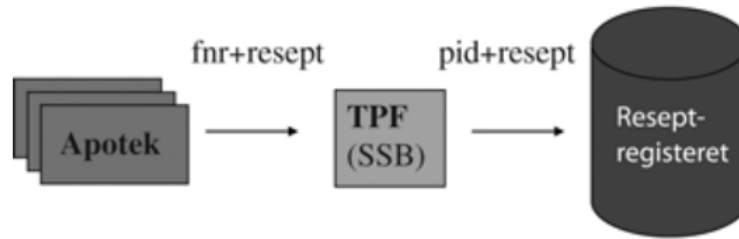


Figure 2.1: SSB encrypts national identification numbers into pseudonyms [13]

2.7 De-identified Health Data

A medical record is considered de-identified if the sensitive information is replaced by some value, which has a direct mapping to the original information. The difference between de-identified and pseudonymized health data is that the replacement value is independent of the original information in de-identification, whereas pseudonymized data does. Hence, the national identification number for patient X can be replaced with the string AAB in one note, and BBA in the other. The mapping is required to be stored apart from the holders of the de-identified data. The mapping is usually held by a trusted body, i.e SSB.

The definition of de-identified health data is given by *the Health Data Filing System Act*[10]:

”Personal health data from which the name, national identification number and other characteristics serving to identify a person must be removed, so that the data can no longer be linked to an individual person, and where the identity can only be traced through alignment with the same data that were previously removed.”

Since the recipient of de-identified data not has access to the mapping, re-identification should not be possible, hence the data will appear as anonymous to the recipient.

Dato:	Dato: <Date:12233>	Dato: <Date>
Godkjent av:	Godkjent av: <NAME:232> <NAME:488>	Godkjent av: <NAME> <NAME>
Dokumentnummer:	Dokumentnummer: <DOCNUMBER:422>	Dokumentnummer: <DOCNUMBER>
INNKOMSTJOURNAL	INNKOMSTJOURNAL	INNKOMSTJOURNAL
Innleggende lege: Amk-sentralen	Innleggende lege: Amk-sentralen	Innleggende lege: Amk-sentralen
Pasient:	Pasient: <NAME:233> <NAME:1>	Pasient: <NAME> <NAME>
Alder: år	Alder: <3F> år	Alder: <YEAR> år
Diagnose: Brystsmerter	Diagnose: Brystsmerter	Diagnose: Brystsmerter

Figure 2.2: Illustration of the clinical data types. left: Anonymous, center: Pseudonymous, right: De-identified

2.8 Comparison

As illustrated on figure 2.2, exactly the same information is removed from all three types, with the difference being that the removed information in anonymous health data isn't replaced by any value or key, just censored. This is what separates anonymous data from the other two types. On the other hand, the replaced value in de-identified health data won't be of any use for the recipient of the data, and thus will appear as anonymous. Further, the only difference between de-identified data and pseudonymous data is that the replacement key/value in pseudonymous health data depends on the actual value. In other words, every individual has a unique pseudonym which provides the recipient a link to a patient, and thereby follow this patient through the health system. Hence, the pseudonyms are useless without the mapping considered re-identification purposes, and thus are anonymous to the recipient.

2.9 Health Registers

Health Registers are filing systems where health information about patients is systematically stored, in which an individual can be able to track his belonging health data. The registers are either stored electronic or as hard copies. There are two types of registers: *Central Health Registers* (Norwegian: "Sentrale helseregistre") and *Medical Quality Registers* (Norwegian: "Medisinsk kvalitetsregistre"). Central Health Registers contains nationwide health information and mainly established to monitor the state of health on a national scale, maintain and improve the health service, and to conduct medical research. The health data is fetched from health care institutions, and is continuously updated in order to get reliable information about present state of health and to quickly detect new patterns. *The Cancer Register* (Norwegian: "Kreftregisteret") and *The Vaccination Register* (Norwegian: "Nasjonalt Vaksinasjonsregister") are examples of *central health registers*. The *medical quality registers* are used to ensure the quality of treatment, but also used for research purposes.

There are a total of 15 *central health registers* and about 200 *medical quality registers*. These are stored as either person-identifiable form, de-identified form or pseudonomized form. A noteworthy fact is that research projects on health data from these registers is controlled by the *Personal Health Data Filing System Act*[10]. This is explicitly clarified in the *Health Research Act*[7] cf. second paragraph of §2 (referred to earlier 2.1), explicitly states:

"The Act does not apply to establishment of health registers."

The intention is that the establishment of registers whereby subsequent processing of health data is governed by *Personal Health Data Filing System Act*. However, when it comes to medical research, we need to keep steady hands. The *Health Research Act* applies to all types of concrete medical and health research projects. Hence, if a register is established as a part of a concrete medical and health research project, the *Health Research Act* will be the prevailing regulation. However, if a register is established without any concrete aim and independently of a medical and health research project,

for instance as entry to future projects, the *Personal Health Data Filing System Act* will apply[2]. Hence, even if a health register is established under the provisions of *Personal Health Data Filing System Act*, the specific use of the health data from the registers will be under the jurisdiction of *Health Research Act*, if it is for medical and health research purpose.

As mentioned, the health registers are realized in either: person-identifiable, de-identified or pseudonymous form. This is pursuant to §7 and §8 in *Personal Health Data Filing System Act*, which governs the establishment of *medical quality registers* and *central health registers* respectively, stating:

”... *The name, national identity number or other characteristics that directly identify a natural person may only be processed with the consent of the data subject. The latter’s consent is not necessary if the regulations provide that the personal health data may only be processed in pseudonymized or de-identified form ...*”

This excerpt from the *Personal Health Data Filing System Act* is of significant importance for our project as it denotes that consents are needless when processing de-identified or pseudonymized health data. Hence, whether it comes to a concrete health and medical research project or just the establishment of a health register, automatic or semi-automatic de-identification will in both cases be time-saving, which is the actual reason for mentioning the health registers in first place.

Furthermore, there are several rules and regulations on the anonymization levels described above, regarding medical research. However, a detailed study of the legislative framework for medical research is outside the scope of this paper, whereas adequate legislation still has been covered for our purpose. The main reason for presenting these three types is to clarify why sensitive information, is replaced or removed with a view to the legislation.

The manual process of locating and replacing sensitive information before it is disclosed is extremely time-consuming and expensive[1]. The importance of medical research is quite obvious, however, the premises are clear; anonymity must be maintained. This is what forms the basis for our task, namely automatic de-identification

2.10 Relevant Theory

This chapter has presented the legislative definition of *directly* and *indirectly* identifiable information together with *anonymous*, *de-identified* and *pseudonymous* data. Although clinical documents are adapted to fit within the legislative boundaries, it is always discussable whether the personal privacy is sufficiently protected. Automatic de-identification studies conducted in other countries signifies just this challenge of privacy preservation, which will be presented in chapter 4. However, existing de-identification studies have been prepared and developed within their respective legislative boundaries, whereby a majority of these are conducted in the US. This section presents a brief overview of the juridical definitions in the US, which also are used outside the US

(e.g. Sweden[14], Canada[15]) due to clear guidelines, followed by the weaknesses of these guidelines.

2.10.1 Protected Health Information (PHI)

HIPAA[16] dictates allowed contents of disclosed patient information. The HIPAA Privacy Rule specifies; patient data accessibility, the circumstances in which patient data can be disclosed and between whom. Also in the US, clinical data must be de-identified in order to be disclosed for research purpose. However, the *HIPAA Safe Harbour* standard defines 18 specific categories of patient data elements that must be removed before the data can be shared. Such information is called *Protected Health Information (PHI)* and is sensitive health information, or any information contained inside patient documentation (including medical records or payment history) that can be used to identify the patient. Hence, in contrast to Norwegian legislation, the Safe Harbor standard provides somewhat clearer provisions for de-identification, by providing an explicit listing of health information that is regarded sensitive. The following identifiers of the individual or of relatives, employers, or household members of the individual, are removed [16]:

- Names
- All geographic subdivisions smaller than a State, including street address, city, county, precinct, zip code, and their equivalent geocodes, except for the initial three digits of a zip code if, according to the current publicly available data from the Bureau of the Census
- Dates (other than year) directly related to an individual
- Telephone numbers
- Fax numbers
- Electronic mail addresses
- Social security numbers
- Medical record numbers
- Health plan beneficiary numbers
- Account numbers
- Certificate/license numbers
- Vehicle identifiers and serial numbers, including license plate numbers
- Device identifiers and serial numbers
- Web Universal Resource Locators (URLs)
- Internet Protocol (IP) address numbers
- Biometric identifiers, including finger and voice prints
- Full face photographic images and any comparable images;

- Any other unique identifying number, characteristic, or code, except the unique code assigned by the investigator to code the data

Although clinical documents are de-identified pursuant to the HIPAA-specific identifiers, re-identification will always be possible [17]. HIPAA is too naive in today's data rich society because other data sources often exist that contain some or all of the same values, allowing redacted identity information to be restored by linking datasets [17]. *Record linking* is a common technique used to perform re-identification, and will be presented in next section.

A corresponding list of identifiers in Norwegian is not defined by the legislation. Hence, an essential issue is to decide whether the de-identification is satisfactory. However, *Oslo university hospital* has defined a set of sensitive identifiers, but in addition also emphasizes the removal of these might be insufficient. Their internal rule states that the information is satisfactory de-identified if and only if it prevents a link to a group of 3-5 individuals [18] (mentioned in section 2.5). We will use Oslo University Hospital's list of sensitive identifiers as a basis for de-identification. The list is not explicitly realized as presented below, however the composition of the following list is based on their internal guidelines for de-identification which is available on the Internet. [19]:

- Name
- National identification number
- Address
- Zip code
- Phone number
- Fax number
- Birth date
- Hospitalization and conscription date
- Patient number
- Bank account number
- Driver's license number
- Car registration number
- Link to personal web pages
- Mail addresses
- Biometric characteristics
- Photos
- Information about ethnicity
- Political point of view
- Religion

- Sexual relations
- Memberships in labor unions

2.10.2 Record Linkage

The purpose of removing *HIPAA*-specified[16] PHI is to prevent/minimize the possibility of re-identification. However, such lists will never rule out the possibility of re-identification since we know that perfect anonymization is impossible, due to *quasi-identifiers*. One common re-identification technique is *Record linkage*.

The process of combining quasi-identifiers with a disclosed anonymous dataset and perform re-identification is called linking attacks. This is a common re-identification approach. The main idea of record linkage is to associate pairs between two lists of tuples, and use this to derive sensitive information about individuals.


One example is the re-identification carried out by Dr. Latanya Sweeney [11]. The Massachusetts Group Insurance Commission (GIC) decided to release "anonymized" (HIPAA-specific) data on state employees that showed every single hospital visit, with the purpose of helping researchers. Every obvious identifier was removed, such as name, address and national identification number, and William Weld (Governor at that time) assured that patient privacy was maintained in the dataset. However, to make a point and prove the limits of anonymization, Dr. Sweeney re-identified the Governor's health records, including prescriptions and diagnoses, and sent them to his office. Through the re-identification "attack", she surprisingly revealed that 87% of the US population could be identified by just ZIP codes, date of birth and gender. Sweeney carried out a linking attack by purchasing voter rolls from the city of Cambridge and combined this information with the GIC records.

Furthermore she states that about half of the US population (132 of 248 million) are likely to be uniquely identified by only *place, gender and date of birth*, where place is basically the city, town or municipality in which the person resides [20].

2.10.3 κ -Anonymity and ℓ -diversity

The abovementioned example illustrates the weaknesses of *HIPAA*-specific[16] PHI and other specified lists of identifiers. Some proposed methods for preventing attacks on structured data is κ -*anonymity*[11] and ℓ -*diversity*[21]. A κ -anonymised dataset has the property that that each record is similar to at least $\kappa-1$ other records with respect to "identifying" variables. Hence, worst case scenario for a κ -anonymous released dataset is that it narrows down an individual entry to a group of κ individuals. The following example shows a database that is 2-anonymized by suppressing a few variables.

First	Last	Age	Race
Harry	Stone	34	Afr-Am
John	Reyser	36	Cauc
Beatrice	Stone	34	Afr-Am
John	Delgado	22	Hisp




First	Last	Age	Race
*	Stone	34	Afr-Am
John	*	*	*
*	Stone	34	Afr-Am
John	*	*	*

Figure 2.3: Rows 1 and 3 are identical and rows 2 and 4 are identical (equivalence classes). [22]

However, Machanavajjhala et al.[23] addressed the limitation of κ -anonymity, and showed that κ -anonymity does not guarantee privacy against attackers, using background knowledge. κ -anonymity prevents identity disclosure, but not attribute disclosure. Assume that an intruder has access to a disclosed data set which is 5-anonymous. This data set alone can not be used to identify the individuals. But if the intruder is interested in a particular attribute, e.g. medical diagnosis, and all 5 are flagged with cancer, then the data set implicitly reveals that the person the intruder intends to identify, has cancer. This is called a *homogeneity attack*, after which Machanavajjhala et al. [23] introduced a new notion of privacy, called ℓ -diversity. ℓ -diversity requires that the distribution of a sensitive attribute in each equivalence class has at least ℓ *well-represented* values. Figure 2.4 illustrates how the level of anonymity of a 4-anonymous table (left) can be improved, by modifying it into a 3-diversive table (right).

Non-Sensitive				Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	*	Heart Disease
2	130**	< 30	*	Heart Disease
3	130**	< 30	*	Viral Infection
4	130**	< 30	*	Viral Infection
5	1485*	≥ 40	*	Cancer
6	1485*	≥ 40	*	Heart Disease
7	1485*	≥ 40	*	Viral Infection
8	1485*	≥ 40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer



Non-Sensitive				Sensitive
	Zip Code	Age	Nationality	Condition
1	1305*	≤ 40	*	Heart Disease
4	1305*	≤ 40	*	Viral Infection
9	1305*	≤ 40	*	Cancer
10	1305*	≤ 40	*	Cancer
5	1485*	> 40	*	Cancer
6	1485*	> 40	*	Heart Disease
7	1485*	> 40	*	Viral Infection
8	1485*	> 40	*	Viral Infection
2	1306*	≤ 40	*	Heart Disease
3	1306*	≤ 40	*	Viral Infection
11	1306*	≤ 40	*	Cancer
12	1306*	≤ 40	*	Cancer

Figure 2.4: Left: 4-anonymous table right: 3-diversive table. [23]

The sensitive identifier in the 3-diversive table has three *well represented* values: *heart disease*, *viral infection* and *cancer*. Using the 3-diversive, we no longer are able to tell if our "neighbor" Bob (a 31 year old American from zip code 13053) has cancer, which can be derived from the 4-anonymous table. We also cannot tell if our "colleague" Umeko (a 21 year old Japanese from zip code 13068) has viral infection or cancer. Machanavajjhala et al.[23] have shown that the ℓ -diversity algorithms provide a stronger level of privacy than κ -anonymity routines.

κ -anonymity and ℓ -diversity are techniques to generalize information and prevent re-identification. However, these techniques are limited to structured data and requires

the structuring of free text notes before they can be utilized. There are tools to process text-based data, and extract coded data from standardized nomenclatures such as *Systematized Nomenclature of Medicine Clinical Terms (SNOMED)*[24], which can be used to structure free text clinical notes and further anonymized. However, this is a different approach to the de-identification task than the one we have chosen (see implementation 5), thus these techniques will not be utilized.

Medical Records

As our task deals with EHRs, it is appropriate to know what a health record actually is and contain. This section provides an overview of EHRs in general and its main components, together with the opportunities and challenges that come into light from the technological development.

3.1 Electronic Health Records

A medical record is one of the most important tools health care providers holds. Medical records have been used in Norway since 18th century and, internationally before 17th century. A health record is systematic clinical documentation of the health care that has been received. The EHR is an individual journal, recorded by a specific person with a specific profession within health care. Hence, an individual can have several EHRs at various clinics or health units/institutions such as psychologists, physiotherapists, doctors etc. Digital health records are called electronic health records (EHRs), in which the journaling is carried out through EHR-systems (discussed in 3.2), pursuant to *the Health Personnel Act* §46[8].

There are certain demands for the content of health records, cf. §8 in *Regulation on health records*[25], in which there are defined 21 items that should be included if they are relevant and necessary. The reason for explicitly stating “*relevant and necessary*” is because the 21 items considers a number of various conditions and scenarios, i.e. enforcement(*r*) or (*drug addiction*), hence not always necessary and relevant. The items are presented in the appendix A, however the essence of the paragraph is simply to include all medically related information about the patient, closest relatives, dates, health care providers, and sufficient personal information in order to make identification possible.

Medical records consist of both, structured and unstructured data. Structured documentation contains a fixed structure, where the information is recorded point by point and follows certain criterions. Unstructured data, on the other hand, is free form notes dictated straight into the record by the clinicians, of their observations and commentary. Examples of this type are discharge summaries and physician notes. EHRs also

contains non-textual elements, i.e. X-rays, CT scans and MR.

3.2 EMR-systems

Since health care providers are obliged to make a note of the medical help provided to patients, this implies that patients have to disclose personal information in order to receive medical help, cf. *Health Personnel Act* §39[8]. It is required that EHR-systems are established on organizational level, meaning that institutions/clinics providing health care are to establish a health record system. The obligation of documentation incumbent on each and every health care providing individual, and the record-systems must be designed in order to comply with the law.

3.3 Benefits

In addition to personal health care, the health sector is under constant pressure of improving their services: Treat more patients, implement new - tools, methods and technologies, increase quality and security and ensure cost-effectiveness. At the same time, personal information and personal health information requires safety, and the health sector has to maintain the information during these changes, both legally and practically.

EHRs have improved a lot of processes in the health sector, and is more effective and practical to use than non-electronic health records. The rapid progress in Information and Communications Technology (ICT) facilitates quick and precise distribution of EHRs and makes the information available to health care providers. One of the main advantages is the simplicity of sharing health records across different health care institutions, assuming smart and fast sharing-systems between them. However, this assumption is far-fetched as it demands a thorough renovation of today's ICT-systems, which is unlikely to be realized in the near future (see section 3.5).

ICT tools can provide the opportunity to easily verify that the records are up to date by logging the modifications made in a health record. Faster access to precise and necessary patient information may lead to more efficient treatment.

Moreover, with the reality of encryption and access control, the information remains safe and better secured against unauthorized access. Another major advantage is that medical and health research can be performed more efficiently by exploiting statistical tools like data mining (see section 4.3.1).

3.4 Issues

There is a somewhat danger to the realization of the abovementioned opportunities which might go at the sacrifice of personal privacy. EHRs contain a lot of sensitive information, whereby easier access to the information makes it easier to misuse, which

is a challenge in terms of EHR-sharing across institutions. The legislation clarifies constraints regarding information sharing across institutions, cf. *Health Register Act* §13[26]:

“Only persons responsible for managing data or persons working on their command, may be granted access to health information to the extent that this is necessary for the work and in accordance with confidentiality provisions.”

The latter part of this act prevents direct information sharing between health institutions. Information sharing in accordance to personal confidentiality is difficult without a satisfactory national infrastructure with access control and logging opportunities. It is a necessity to know who the information seeker is, and to whom the information will be exposed to.

Development and improvements within the health sector brings changes internally and across different health institutions, and there is doubt whether the maintenance and protection of personal health information keeps up during the improvements of the system. This allegation was made by *The privacy commission*, who published a detailed study in 2009, “Individ og integritet – Personvern i det digitale samfunn”[27]. This is a wide study and includes a lot of subjects relevant to personal privacy, where for our purpose it also discuss issues and challenges regarding personal privacy in the health sector.

The study reveals that computer systems used in health sector do not provide satisfactory security of the patient information. One specific problem addressed is that the information flows across all user-levels within the institutions, and redundant information is being exposed. Even though there are several exceptions, the general rule states that only health care providers have legitimate access to health records. This rule is not followed properly within health care clinics/institutions. Too much data is accessible to employees without a legitimate need.[27]

Furthermore, *The Norwegian Data Inspectorate* has conducted a number of inspections on health institutions and revealed unsatisfactory practice[28][29]. *The Norwegian Data Inspectorate* is an independent administrative body, and their task is to monitor and control that the Personal Data Act is followed. The Data Protection Authority expects better practice and even sent a letter (2008) to Health – and Care Department in which they proclaim that illegitimate EHR access threatens personal confidentiality in hospitals[27].

3.5 De-identification

One way to improve the weaknesses addressed by *The Privacy commission* and *The Norwegian Inspectorate* is to arrange for customized access on all levels. However, there is still a great need for standardization and coordination of information - and communication systems, in order to ensure that the patient privacy is maintained and national guidelines are followed by every institution. To put it in a realistic perspective, Helse Sør-Øst, which is one among four Norwegian regional health authorities, has a total of 3500 digital systems with relevant health information. Hence, standardization will require a lot of time.

Today's practice is to either omit information about personal characteristics and all non-medical information before sharing the data with the requesting institution, or by directly requesting the patient concerned. Both of these alternatives comply with the legislation[27]. These processes, however, are time consuming and will not be as efficient as direct information access, indicating the need of automatic de-identification, which also was concluded in the investigation conducted by *The Privacy Committee*, which states:

“The technical possibilities’ EMR gives for de-identification and other methods to hide identity should be fully utilized“.

Automatic or semi-automatic de-identification of free text health records first of all simplifies the very de-identification process which today is performed manually, reducing the time-consumption and in turn the costs.

By assuming access to such an application, it provides the opportunity to de-identify a significantly higher amount of clinical documents and finalize these manually within the same amount of time it takes to perform the whole process manually. Consequently, more de-identified clinical data will be available for various applications as health research, data mining and education, and provide researchers and operations with greatly increased access to patient data, thereby combining sources of data previously unavailable.

The need for such an application increases accordingly with the technological progress within the health sector. The increasing use of EHR-systems simplifies illegitimate exposure to confidential information, and is thus easier to misuse. As per the inspections revealing unsatisfactory security of patient privacy by *The Norwegian Inspectorate* and the statement made by *The Privacy Committee*, automatic de-identification may serve as a partial solution to patient privacy issues, and at the same time improve patient treatment.

Automatic or semi-automatic de-identification of free text clinical notes is becoming more relevant as medical records are being converted to Electronic Medical Records (EMR). The opportunities and benefits have been considered for a while by the Norwegian healthcare (as per the Privacy Commission's statement 3.5), but rather few serious attempts have been made to develop such a tool. However, there has been conducted a series of attempts by various research communities in other parts of the world, experimenting with different methodologies.

As a preliminary study for this project, a state of the art investigation was performed last semester, where the main focus was to look at different tools and techniques used for de-identification purpose, on different languages. This chapter will not repeat the investigation in particular, but rather provide a summary and point out the key observations, directly relevant for our project.

4.1 Methods

Technically, de-identification of free text clinical notes is a binary classification problem, with the purpose of classifying a sequence of text as either sensitive or insensitive. The complexity of this task increases significantly when dealing with free texts as it requires semantic analysis in order to interpret the text. There isn't any precise solution to the task of de-identification, however, the state of the art investigation shows that broadly two types of techniques are used: pattern matching (see 4.5) and machine learning (see 4.6), or sometimes a hybrid (see 4.7) of these. Pattern matching and machine learning are essential techniques used in NLP, hence the two following sections will briefly explain these techniques with a view to de-identification. Further, a selection of the various de-identification approaches from the state of the art investigation will be presented together with their respective evaluation results.

4.2 Pattern Matching

Pattern matching is a term in computer science and denotes the procedure of recognizing predetermined text-sequences. The main cycles of a pattern matching approach is: chunk input text into self-defined textual units (often called tokens) - process the text-units - locate desired text patterns - and finally perform desired operations with the retrieved text. Tokens can be characters, words, sentences or paragraphs. Predetermined patterns can occur in a lot of unexpected ways, causing ambiguities, which requires in-depth analysis. Several techniques are used to carry out such analysis, which will be described in the further sections.

It is important to separate the two terms, pattern matching and pattern recognition. Pattern recognition is a machine learning technique, denoting the likelihood of a match, whereas in pattern matching, patterns usually have to match exactly. However, the pattern recognition topic resumes in connection with machine learning in section 4.3.

Pattern matching is a huge field with a widespread range of applications, i.e. NLP, image recognition, speech recognition and data mining. The following sections will somewhat restrict the range, and only cover methods most relevant to de-identification.

4.2.1 Pattern Matching and De-identification

As mentioned above, de-identification of free text medical notes is a task of determining a sequence of text as either sensitive or insensitive. In order to do so, one firstly has to define what sensitive information involves, and further get familiarized with how such information is shaped, when it typically occurs, and furthermore the context such information occurs in. Pattern matching methods have to be adapted accordingly, in order to make precise decisions.

Among the pattern matching approaches which were investigated last semester, there are three very common techniques which are utilized in almost every approach: Reference works, regular expressions and Part of Speech (POS)-tags. The two former techniques will be described in the following sections, whereas POS-tagging will be described in section 4.3.3.

4.2.1.1 Reference Works

De-identification applications based on pattern matching are quite dependent on reference works. Reference works can be described as the supporting beam of such applications as they constitute a substantial part of the systems. The reason is quite simple; smallest units in a text providing semantics (meaning) are words (or acutally morphemes to be linguistically correct). And to interpret a phrase/sentence, the words have to be analyzed in order to extract a meaning out of the phrase/sentence. However, it is tremendously challenging to make a computer interpret human language due to

the complexity¹. Hence, the task of semantic extraction has to be somewhat simplified.

One cheap way to weed out sensitive text is to locate sensitive words/terms by means of reference works, whereby the contents of the reference works are classified as sensitive. This method, however, presumes that each and every word/term in a dictionary is sensitive. Take for instance the sentence: “*Patient Name: Peter Smith*”, from the anonymization example in 1.1. By using a name dictionary on this sentence, the strings Peter and Smith will match, and since names are directly identifiable information, these can be annotated as sensitive. This method can be used to identify a number of sensitive information categories. The following dictionary-types can be utilized by a de-identification application, in order to perform a word/term-level sensitivity-classification.

- Person-names
- Geographic Locations
- Clinical Institutions Names
- Religions
- Zip codes

Various dictionaries and lexicons are typically fetched from publicly available resources, which in turn are inexpensive and easily accessible. Not only sensitive dictionaries, but also various other domain-relevant term-dictionaries are used to recognize the text contents in clinical notes. Medical terminologies and classifications systems are often built from various sources, at the cost of months or even years of work. Domain relevant reference works are very often used in order to recognize medical terms, which do not exist in general dictionaries. The following list shows frequently used reference works in de-identification software, within the clinical domain.

- Medical Language Extraction and Encoding (MeDLEE): A natural language parser used to recognize medical concept entities. It was originally designed for decision support applications in the domain of chest X-ray reports, but showed high accuracy in extracting specific clinical information from discharge summaries, hence the application of MeDLEE has expanded to different medical fields [31].
- Medical Subject Headings (MeSH):[32] is a *controlled vocabulary* and links medical concepts and terms, and arranges these hierarchical. It is a twelve-level hierarchy, consisting of 26853 *descriptors* or medical headings. MeSH is used for indexing/-tagging articles from 5400 of the world’s leading biomedical journals, and further used to perform search queries.
- SNOMED[24] is called a *reference terminology* and includes over 311000 unique concepts. The terms are systematically organized and adapted for computer processing, recognizing concepts and relationships from health data and further used for aggregating the contents. It is used to index, store and retrieve clinical data across specialties, and helps to organize the contents of medical records.

¹Computational Complexity and Natural Language is another huge research field, among linguists and cognitive scientists [30]

- Unified Medical Language System (UMLS) [33] is a huge compendium of many *controlled vocabularies* and provides a mapping between these. The UMLS integrates over 2 million names for some 900 000 concepts from more than 60 families of biomedical vocabularies. The vocabularies are integrated as a *thesaurus*², and can be viewed as a big *ontology*³ of biomedical concepts, and is a powerful tool in NLP and medical informatics. Among the many vocabularies it includes MeSH and SNOMED, where terms from various vocabularies are connected through *concepts*, which in turn are connected in *relationships*.

Dictionary lookups for sensitive words and medical terms is a common technique, but also simplified and inconvenient, as words may have several meanings which again requires contextual information from the belonging sentence.

Take for instance the sentence *Hope Cushing has a fracture in the right elbow*. Here, the patients name is *Hope Cushing*, which will match against a name-dictionary. However, the word *right* will also match in name-dictionary lookup, and *hope* is a verb which isn't sensitive and will also match against multiple dictionaries. Moreover, the word *Cushing* will match in a medical dictionary as *Cushing* is a medical condition. Evidently, word/term-level lookups can provide indications to the contents of a sentence, but will in most cases be insufficient and require contextual analysis.

4.2.1.2 Regular Expressions

Regular expression is a pattern matching technique, and used to identify common patterns which occur in texts. It is widely used in computer science areas like compilers and text editors. One of the most advantageous property of regular expressions is the ability to catch numerical patterns. Numerical patterns are frequently found in clinical notes, and can be very sensitive. As certain types of information have a fixed nature and predetermined structure, regular expressions can be used to recognize these. The following items are examples of information that occurs in clinical which can be recognized by a regular expression.

- Dates
- National Identity Numbers
- ZIP codes
- Phone Numbers
- Medical Record Numbers
- Patient Numbers
- Bank Account Number
- Fax number
- Electronic Mail Addresses

²Reference work that lists words together according to similarity of meaning.

³Explicit formal specifications of terms in a domain and relations among them.

- IP addresses

Every item on this list has a determined structure, in which neither the characters, nor their positioning, can be random. For instance, Norwegian national identification numbers consists of 11 numbers, whereof; the first 6 digits represents the date of birth, the next two are randomly fixed followed by a gender specific digit in which even numbers are assigned to women and vice versa, and the final two digits representing a checksum, validating the preceding numbers. However, regular expressions do not have the property of validating checksums, as they only look at the characters and their positioning, unable to perform further computation. Hence, in order to validate a Norwegian national identification number, a regular expression has to be used in combination with a special rule that has the quality to confirm a checksum. In order to clarify how a regular expressions works, table 4.1 provides an explicit example of a regular expression used on Norwegian national identification numbers.

DDMMYY IIICC: (Day)(Month)(Year) (IdNumber)
Day: 0[1-9] [12][0-9] 3[01]
Month: 0[1-9] 1[012]
Year: [0-9][0-9]
IdNumber: [0-9][0-9][0-9][0-9][0-9]

Table 4.1: Regular expression for national identification numbers

Regular expressions can also be used on phrase-level as well, but rarely used due to complexity and unpredictability. Besides, this technique is also utilized to detect non-numerical patterns. The de-identification application developed by Deléger et al.[34] used regular expressions to match medication dosages. Another example is Neamatullah et al.[1] who used regular expressions to recognize address patterns. Such regular expressions often rely on dictionaries and special rules in order to increase the applicability.

4.3 Machine Learning

Machine learning is a field within computer science and a branch of artificial intelligence. The field is extensive and deals with a great number of today's engineering challenges, including NLP challenges. The main principle of machine learning systems is to learn from data, without being explicitly programmed, but use experience to perform predictions. Machine learning algorithms take data as input, with the purpose of identifying underlying relationships and predict behavior.

One practical example is automated speech recognition; If the sound waves of 1000 people uttering the word *yes* are collected, and likewise for the word *no*, a machine learning algorithm will use the sound waves as training data and establish a classifier (binary classifier for this particular example). The classifier will be used to predict new utterances as either the word *yes* or *no* by their sound waves and a likelihood which

depends on the machine learning algorithm. A classifier's robustness is often decided by the level of variation in the training data[35].

4.3.1 Text Classification and Data Mining

Machine learning is a frequently used technique in the field of NLP. The range of use is broad, but commonly used for text classification and information extraction.

Data mining or text mining (*large scale data analysis*) denotes the process of discovering patterns in large data sets. Automated pattern discovery in texts is a hot topic at the present time due to digitalization, and the ability to discover complex patterns within large sets of data, beyond human reach. For example, data mining can help companies to find customers with common interests, through the information stored inside their customer databases. Data mining requires comprehensive quantity of data, often fetched from databases and focuses on the discovery of unknown properties in the data. Data mining approaches utilizes a variety of methods within different fields, i.e. artificial intelligence, statistics and machine learning. The range of use is very broad as it provides the ability to process raw data into useful information, and used in medicine, business, games, engineering and visualization.

However, a detailed elaboration on data mining is outside the scope, but on the other hand there is a reason for mentioning this topic. One highly appropriate example is the use of data mining on medical records. The fact that medical records are converted to digital format brings along a lot of possibilities, whereby data mining is one among others. Data mining can have big impact on medical research, discovering complex patterns by the means of clinical data. Muneo Khushima et al.[36] did an interesting study, using data mining techniques to extract useful information from EMRs of chronic hepatitis patients, which successfully identified patterns in the vocabularies. However, the initial problem of privacy protection remains and prevents the fully exploitation of data mining. Data mining requires large amounts of de-identified data, which is too expensive to perform manually. Hence, this proves just another advantage of automatic or semi-automatic de-identification, making clinical information available for such powerful tools.

4.3.2 Machine Learning and De-identification

Binary classifiers are primarily used in machine learning approaches as de-identification is a binary classification problem. However, some approaches also choose to regard de-identification as a multi-classification problem with the goal of classifying each PHI(described in 2.10.1) category (Name, Date, Address, etc.). The state of the art investigation shows that Support Vector Machine (SVM), Conditional Random Fields (CRF) and Random Forests are commonly used machine learning methods.

The common property between such algorithms is to acquire knowledge and learn underlying relationship between *features*, and perform predictions. Machine learning features are measurable properties of the data, which is analyzed in order to acquire distinctive

characteristics. Some relevant machine learning features within NLP is: word length, sentence length, word count, caption, POS-tags (see 4.3.3) and word-frequencies. The learning is purely based on these features, which is why *feature-selection* is an actuating step and has huge impact on the end results. Robust machine learning models are developed by testing different sets of feature-combinations on training data, whereby the combination of features producing most optimal results are chosen.

One simplified example is a classifier trained and developed on newspapers. This classifier will probably achieve much better results when tested on newspapers than clinical notes, as clinical notes often contains a lot of abbreviations, grammatical incorrect sentences and a completely different vocabulary. This also illustrates the importance of using relevant training data adjacent to the test set. Thus, feature selection and domain relevance is crucial to the robustness of classifiers.

4.3.3 Part of Speech Tags

The process of marking a text in grammatical units is called POS-tagging, and plays an important role in NLP. This is used to find relationship between words in a phrase, sentence and paragraphs. A POS-tagger is a valuable part in text classification because the order of these syntactic units can be used to reduce the possibility of misinterpretations, and to resolve ambiguities.

One example is the sentence: *heat water in a large vessel*. A POS-tagger will assign a POS tag to each word as following

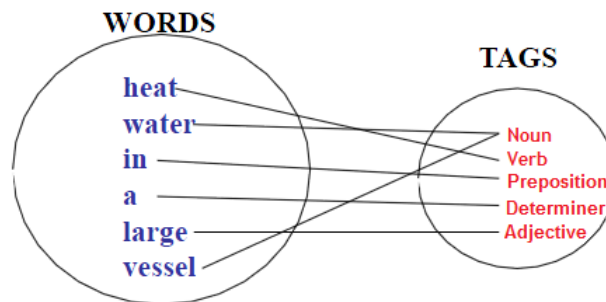


Figure 4.1: POS-tag example[37]

But even POS-taggers are prone to error, as words can have several valid POS-tags. These are either rule based or based on statistical techniques, including machine learning methods such as *SVM*, *Maximum entropy classifier* and *Nearest-neighbor*. One of the first and widely used English POS-taggers is rule based and called E. Brills tagger[38] which achieved 95% accuracy. It is challenging to obtain high accuracy rates as a word maps to several POS-tags, and a tagger that makes predictions based on likelihood may make wrong predictions. The next figure (4.2) illustrates the ambiguities which occurs during the tagging of the abovementioned (4.1) sentence.

<u>Word</u>	<u>Tag</u>
heat	verb noun
water	noun verb
in	prep noun adv
a	det noun
large	adj noun
vessel	noun

Figure 4.2: POS-tag ambiguities[37]

This example shows that a POS-tagger has to make certain choices based on experience, illustrating that relevant learning material is crucial.

A POS-tagger can be exploited by a de-identification application in several ways. By converting words to their respective POS-tags, the resulting sequences of POS-tags can be utilized as a feature by a machine learning classifier. POS-tags can be analyzed in order to solve ambiguities in pattern matching approaches by exploring the surrounding POS-tags. POS-tags are commonly used in machine learning-based systems. In Meystre et al.'s[39] review about recent de-identification studies, six of the total eight studies which uses machine learning used POS-tags as a features.

4.4 Evaluation Methods

De-identification software is commonly measured in terms of recall, precision, fall-out and f-measure. These terms stems from Information Retrieval (IR), which is the process of obtaining relevant information from a resource corpus, and the task of de-identification can partially be considered as clinical information retrieval. As explained in section 2.10.1, PHI refers to sensitive information, and the evaluation method classifies textual information as either PHI or non-PHI, without any intermediate grades.

The next sections will provide an introduction to each measure. The following table(4.2) defines central definitions used in binary classification. The following sections are fetched from the preliminary study.

True Positive	Correct classified PHI
True Negative	Correct classified non-PHI
False Positive	Incorrect classified PHI
False Negative	Incorrect classified non-PHI

Table 4.2: Definitions used in binary classification tests

4.4.1 Recall

Recall is the proportion of correct labeled PHI elements among the total amount of PHI elements in the text, and is given by the following equation:

$$\text{Recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}}$$

Recall is also called sensitivity in binary classification functions in the field of statistics. The recall value itself is a weak indicator because it only provides information about correct chosen PHI elements by the de-identifier and excludes the mistakes. In other words, a de-identifier can label every single word of a text as PHI, and gain a perfect recall rate of 100%, which is useless in practice as every word is removed from the text. It proves more useful combined with precision and fallout rate.

4.4.2 Precision

Precision is the proportion of correct labeled PHI elements among the total amount of labeled PHI elements. Precision is given by:

$$\frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}}$$

Precision is equivalent to positive predicted value in binary classification. The precision value of a de-identification tool gives a good indication of the performance. A good precision rate implies that the de-identifier has a good accuracy of labeling words as PHI, however it excludes the proportion of ignored PHI elements, which is vital in the context of sensitivity. Thus, neither precision rate nor recall rate provides optimal measures separately. The combination of these is (f-measure 4.4.4) much better and says a lot more about quality of the de-identification tool.

4.4.3 Fallout

Fallout is the proportion of incorrectly labeled PHI elements among the total amount of non-PHI elements. Fallout is therefore given by the equation:

$$\frac{\text{FalsePositives}}{\text{TrueNegatives} + \text{FalsePositives}}$$

This rate can be interpreted as the opposite of recall, as recall gives the rate of correct labeling out of total possible amount of correct labeling. Fallout gives an indication on the degree of incorrect labeling, ergo the amount of incorrect labeling out of the maximum possible amount of incorrect labels (non-PHI). The purpose is generally to achieve a low fallout rate as possible. However, in the context of de-identification of sensitive texts, high fallout rate is not as fatal as low recall rate. High fallout rate only indicates over-scrubbing ⁴. This will of course produce less specific output, but high recall is prioritized against low fallout.

⁴Removal of too much data, i.e. insensitive data

4.4.4 F-measure

Average of the precision and recall, where the best possible score is 1 and worst 0. F-measure is calculated as:

$$F_{\beta} = \frac{(1 + \beta^2) \times Recall \times Precision}{Recall + (\beta^2 \times Precision)}$$

Mathematically, this is the harmonic mean between recall and precision. This rate provides a holistic evaluation of the system and is commonly used in de-identification. Further in this report, we will use the traditional f-measure (F_1 -score) with $\beta = 1$.

4.4.5 Summary Example

To wrap up the terms described in previous sections and relate them to de-identification, this section provides a short example. The example was also been used in the preliminary study.

Let us assume that we have an EHR containing a total of 30 words. The 30 words consists of 13 PHI tokens, and 17 non-PHI tokens.

30 words
13 PHI
17 non-PHI

Table 4.3: EHR Example

Further, this EHR(4.3) has assumedly been de-identified by a de-identification application. The output of the de-identification process shows that 15 words have been classified as PHI, and 15 words as non-PHI. Comparing the application-output to the "gold standard", it shows that the 15 words classified as PHI consists of 5 non-PHI and 10 PHI words, meaning that 5 words have been over-scrubbed. By using this piece of information, we can derive the results shown in the tables below (4.4 & 4.5):

Output	Correct	Incorrect
PHI	10	5
non-PHI	12	3

Table 4.4: De-identification output

True positives	10
False positives	5
True negatives	12
False negatives	3

Table 4.5: De-identification results

Finally, the recall, precision, fallout and f-measure can be computed as follows:

$$\text{Recall} = \frac{10}{10+3} \implies \frac{10}{13} \approx 0.77$$

$$\text{Precision} = \frac{10}{10+5} \implies \frac{10}{15} \approx 0.67$$

$$\text{Fallout} = \frac{5}{5+12} \implies \frac{10}{17} \approx 0.29$$

$$F_1\text{Score} = \frac{(1+1^2) \times 0.77 \times 0.67}{0.77 + (1^2 \times 0.67)} \implies \frac{1.0318}{1.44} \approx 0.72$$

This is a very simple and specific example, at the same time as it describes how each rate is related to de-identification.

4.5 Pattern Matching Approaches

The following section provides a selection of existing de-identification approaches based on pattern matching. The pattern matching and machine learning approaches presented in the following sections were also a part of our preliminary study, however some of these have been slightly modified.

4.5.1 Gupta

One of the recent well-known de-identification systems is *de-id*, developed by Gupta et al.[40]. De-id was originally developed at Pittsburgh University, in connection with the Integrated Advances Information Management Systems (IAIMS) program intended to de-identify various kinds of clinical documents. However, the de-identification tool was further improved and adapted to surgical pathology reports by Gupta and his team. This system employs pattern matching, various dictionaries, and a set of complex rules. Among other elements, the special rules extracts information from the report headers, which often contains a lot of sensitive information, and use this to search similar text further in the reports.

Sensitive information is replaced by pseudonyms, in order to increase the readability. The team improved and optimized De-id in a continuously collaboration with domain experts, through a tripartite evaluation process, whereof each evaluation consisted of 967, 1000 and 300 reports, respectively. Overall evaluation results were never published, but they reported that the system was reliably and specifically removing safe-harbor identifiers. Only the amount of overmaking and undermaking errors (i.e. false positives and false negatives) were presented without any number of the total amount of PHIs, however the presented numbers shows consistent improvement throughout the three evaluations, and their application has later been used to de-identify more than 35000 pathology reports, where minor problems continuously are identified and fixed.

4.5.2 Neamatullah

Neamatullah et al.[1] described an approach, similar to Gupta et al.[40] which also extracts information from the report headers. They developed a perl⁵-based de-identification application, with a primary focus on discharge summaries and nursing notes. A certain level of modifiability was added gradually, in order to increase the applicability. This is also a pure pattern matching approach, using regular expressions, lexical lookup tables, dictionary lookups, and simple heuristics. Names and locations were recognized by dictionaries, whereas regular expressions were used to catch numerical PHI-categories (described in 2.10.1) and addresses. Furthermore, a string algorithm was used to detect potentially misspelled patient names. The software package, including source code and dictionaries, is freely available on the internet[41][42], so that its working can be studied, customized and improved.

To evaluate the de-identification application, a randomly selected subset of 2434 nursing notes were extracted from the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC) II database which is a database containing Intensive Care Unit (ICU) patient data[43]. The following results were obtained:

F-Measure	Recall	Precision	Fallout
0.845	0.967	0.750	0.002

Table 4.6: Evaluation results achieved by Neamatullah et al.'s de-identification application

4.5.3 Pantazos

Pantazos et al.[44] performed a de-identification study on Danish EMRs, which also was a first-time approach in this country. Like most others, their strategy was to extract sensitive information from the headers, and exploit the acquaint information on the rest of the body. However, their procedure was somewhat different. The team got access to an EMR database containing 437164 medical records, whereby 9 of 11 tables in the database contained structured data. Sensitive information was obtained from the structured part, and searched for in the remaining 3 tables of free text notes. Furthermore, sensitive information was replaced by pseudonyms; hence the output EMRs belonged to artificial persons, which was the goal in the first place.

A separate replacement algorithm was defined for each type of identifier, such as person names were replaced by other names with specific frequencies. Beyond that, the application was based on simple language analysis, special rules and dictionaries. The total amount of 323122 records were de-identified with an acceptable degree of anonymity, readability and correctness, achieving an F-measure of 0.984. The system was evaluated on a smaller test set, consisting of 369 random medical free text records, achieving following results:

⁵Practical Extraction and Reporting Language, a dynamic programming language

F-Measure	Recall	Precision	Fallout
0.957	0.995	0.923	x

Table 4.7: Evaluation results achieved by Pantazos et al.'s de-identification application

4.6 Machine Learning Approaches

The following section provides a selection of de-identification approaches based on machine learning.

4.6.1 Dalianis

There has been performed a de-identification attempt in Sweden by Dalianis et al.[45], on the Swedish gold standard called the Stockholm EPR PHI corpus. This de-identification approach was somewhat different as the purpose was to compare two state of the art machine learning algorithms, with and without the actual words of the corpus, implying that only word-features were used to localize sensitive information. 14 features were used to represent a token (words and sentences), i.e. POS-tag, token-length, whether initial letter is capital, token-lengths of two tokens before etc. The two machine learning algorithms were CRF and random forests. The experiment used an off-the-shelf CRF implementation, CRF++. CRF is a machine learning method for segmenting and labeling sequence data, and used for structured prediction. Random forests is a so-called ensemble classifier which involves a set of classifiers, and combines their prediction to make a decision. Random forest consists of several decision trees and outputs the most repetitive one. The method is a combination between Breimans "bagging"⁶ and "random selection of features". CRF performed best in terms of recall and f-measure, but random forest achieved better precision rate. Their initial goal was to compare the evaluation results with and without (only word-features) the actual words, whereby they arrived at the conclusion that the results indicated severe performance losses without the actual words, hence the chosen features were not sufficient for the suggested approach to be viable. The results of both methods are presented below:

Type	F-Measure	Recall	Precision	Fallout
CRF	0.76	0.71	0.85	x
Random forest	0.67	0.54	0.89	x

Table 4.8: Evaluation results achieved by Dalianis et al.'s de-identification application

4.6.2 Uzuner

Uzuner et al.[46] developed one of the better de-identification tools based on pure machine learning. SVM is used to map tokens to one or several PHI categories, also called

⁶A machine learning ensemble meta-algorithm to improve statistical classification and regression models, in terms classification accuracy.

multiclass classification. The system considers syntactical and orthographical properties of each concerned token and surrounding tokens. Orthography is a term in linguistics and concerns elements as word length, letters, punctuation marks and spelling. POS-tags were used as syntactic features. A distinctive characteristic in this approach is the usage of a link grammar parser. A Link Grammar Parser builds relations between sequences of words in sentences and adapts it to the respective language. For example, Norwegian and English are so called Subject Verb Object (SVO)-languages, thus a link grammar parser builds relations between the words and maps the relation to the respective pattern. The benefit of this approach is the ability to parse partially malformed sentences. The output of the link grammar parser is further used as a syntactical feature in the SVM. The test-corpus was picked from the Informatics for Integrating Biology and the Bedside (i2b2)-challenge⁷ which included 889 discharge summaries. The result from the study are presented below:

F-Measure	Recall	Precision	Fallout
0.98	0.98	0.99	x

Table 4.9: Evaluation results achieved by Uzuner et al.’s de-identification application

in order to increase the performance. The second sequence labeling toolkit is called Lingpipe, where the named entity tagging feature was used. This tagger is based on hidden Markov models which uses n-grams and text chunking. Their main focuses was to observe how well these tools worked "out of the box" and how much work was needed for additional performance gains. The results also showed that both Carafe and Lingpipe worked well out of the box, with minimal tailoring. In order to improve the recall, a number of regular expressions and a bias parameter were added. The best result was achieved by the Carafe-system with f-measure, precision and recall greater than 96%.

4.7 Hybrid Approach

Some de-identification approaches employs both machine learning and pattern matching techniques and combines this in various ways. The following section presents one so-called hybrid approach.

4.7.1 Ferrandez

Ferrandez et al.[48] developed a de-identifier called Best of Breed (BoB), adapted to Veterans Health Administration (VHA) clinical documents. Their main focus was to utilize existing rule and machine learning based methods, and alleviate the need of large annotated corpora. BoB mainly consists of two components, high sensitivity extraction component and false positive filtering component. The former component prioritizes

⁷A NLP-competition of developing the best de-identification software, arranged in 2006 by Dr. Ozlem Uzuner [47]

patient confidentiality and focus to achieve high recall, meaning that the threshold for classifying a token as PHI is very low. Hence, this component contemplates too many tokens as PHI, resulting in high recall - and low precision rate. They have used pattern matching i.e. dictionaries, regular expressions, fuzzy search techniques and heuristics. Additionally, machine learning was used in order to catch unusual PHI elements, where a CRF classifier was used.

The second component was employed to mitigate the large amount of false positives produced by the first component, hence the main focus is to find true positives among already classified false positives. It is completely based on machine learning, and consists of SVM classifiers. Firstly, they trained three SVM classifiers for names, numerals and eponyms. Secondly, a linear multiclass classifier was used to detect all other types of PHI. BoB was trained on 500 clinical documents and tested on 300. These included consult notes as: nursing notes, discharge summaries, emergency room notes, progress notes, preventive health notes, surgical pathology reports, psychiatry notes, history notes, physical and other less common note types. Additionally, in order to test the generalizability, they tested BoB on the i2b2 corpus from 2006. The following results were achieved:

F-Measure	Recall	Precision	Fallout
0.87	0.92	0.83	x

Table 4.10: Evaluation results achieved by the BoB-application on VHA clinical documents

F-Measure	Recall	Precision	Fallout
0.89	0.92	0.87	x

Table 4.11: Evaluation results achieved by BoB-application on the i2b2-challenge corpus

Part III

Implementation

There have been proposed several solutions to the problem of automatic de-identification of free text clinical notes. The state of the art chapter(4) addresses two principal methods, namely machine learning and pattern matching, which have shown promising results in earlier experiments (described in 4.5 & 4.6). Unfortunately, machine learning approaches are dependent on adequate training data (4.3.2), which was unavailable. However, we had access to various kinds of reference works which makes it possible to develop a pattern matching approach. In addition to reference works, some simple statistical methods were also included in order to assist the decision making and increase the performance.

Our implementation is java-based. The team is familiar with the java platform, which will facilitate time efficient development. Moreover, considered the wide range of methods and techniques used in de-identification, it might prove beneficial to organize the proposed solution in an object oriented manner in order to ensure modifiability. Besides, none clear reasons suggest that choosing another programming language would constitute a significant advantage. Even though the implementation is java based, some minor parts are implemented using python and c++.

The Application is implemented as a pipeline, and the design is inspired by Apache cTakes [49]. The pipeline consists of 4 components; pre-processing, pattern matching, classification and post-processing. Each of the components are implemented separately and the information flows as illustrated in figure 5.1. Further sections will provide a detailed description of each component.

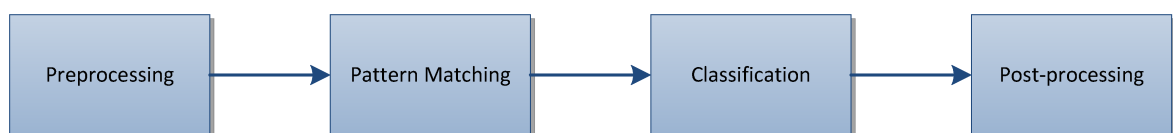


Figure 5.1: Overview of the pipeline

Preprocessor

The preprocessor transforms the raw text files and organizes them into a data structure in order to facilitate for further processing. Hence, the need for a preprocessor is quite apparent and is an important part of any NLP-application. The preprocessor reads each document and transforms these into the following three structures; *section*, *sentence* and *token* (illustrated in figure 6.1). Each *section* consists of multiple *sentences* and each *sentence* consists of multiple *tokens*. A more detailed description of each structure will be presented later. The performance of the system is greatly dependent on the performance of the preprocessor, as its output forms the basis for further investigations; an error at this stage will persist throughout the entire pipeline.

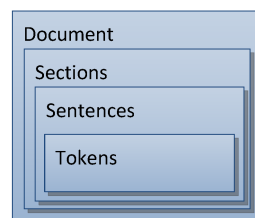


Figure 6.1: The main elements in the data structure

The preprocessor consists of several components which follows a pipeline structure. Figure 6.2 illustrates the preprocessor pipeline. This section will firstly describe each of the components and their functions, before presenting an overview of the finalized data structure of each document, and further a discussion of drawbacks that affects the system.

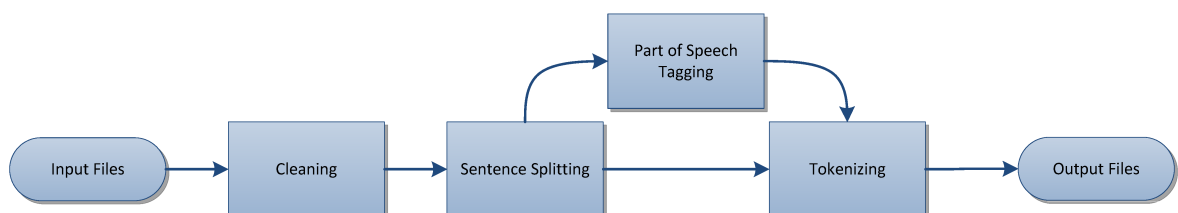


Figure 6.2: Overview of the preprocessor pipeline

6.1 Cleaning

The first step of the application is to clean the input files by replacing problematic characters that may cause problems later in the pipeline. As the raw files are UTF-8-formatted (UTF-8 is an encoding that represents every unicode¹ character), it is convenient to standardize characters like whitespaces and line separators. One practical example is the cleaning of whitespace characters; Unicode defines several types of whitespace characters, and since the whitespaces are used to split the text into *tokens* (explained in 6.4) and *sentences* (explained in 6.2), it is beneficial if the whitespaces remain consistent. It is for instance easier to create robust regular expressions(7.2.1) with standardized characters. The "file-cleaner" simplifies these tasks by substituting each type of whitespace character with the standard unicode space *U+0020*. A table of unicode space characters has been added to the appendix, see figure C.1.

6.2 Sentence Splitting

The cleaned data is divided into *sentences* by a sentence-splitter, which is a slightly modified off-the-shelf splitter provided by Apache Lucene [50]. The modifications were made by fellow students working on other EviCare-projects (see 1.1.1), and adapted to Norwegian texts.

The splitter identifies most of the sentences correctly, but struggles with malformed sentences that lack either punctuation marks or whitespaces. Initially, the intention was to improve the sentence splitter which may have lead improved performance , but the time-limitation did not allow. Although this factor may affect as a source of error, the contributions from the sentence analysis is less influential to final classification.

When the sentence-splitting is completed, the output is further used by the the POS-tagger(6.3) and Tokenizer(6.4).

6.3 Part of Speech Tagging

Output from the sentence splitter is passed into a POS-tagger. The POS-tagger is developed at Norwegian University of Science and Technology (NTNU) by Brox et al.[51]. It is based on the theory of Hidden Markov Models [52], a machine learning technique that assigns tags by likelihood. The tagger was evaluated on a Norwegian corpus with 100,000 words by a 10-fold cross validation², achieving an accuracy rate of 95.04%. As POS-tags are important features for the classification of a *token*, the performance of the POS-tagger is crucial in order to achieve good results. The output produced by the POS-tagger is further used by the tokenizer.

¹A digital standard representation of text, defining a mapping between characters and integer code points representing them

²Divide data into 10 equal parts, train the classifier on 9 parts and test on the last part.

6.4 Tokenizing

Clean sentences from the sentence splitter, together with POS-tagged sentences, are passed into the tokenizer where they are divided into *tokens*. The tokenizer employs the POS-tagged sentence files to assign a POS-tag attribute to each *token*, which is used during the final classification. *Tokens* are split by whitespace characters, and since the documents are cleaned, we only have to consider the standard unicode whitespace character. Most usually, these *tokens* represent words, dates and numbers. A few examples are presented below:

- 15/3-2011
- Word
- 68293050

The advantage of splitting *tokens* on whitespace characters is that patterns, such as dates and phone numbers, often are preserved within the *token*. However, this only includes *tokens* that are correctly spelled and follows a certain format, without any whitespace characters. A common problem related to tokenizing is that one particular and coherent piece of information is split into multiple pieces, because whitespace characters occur in between the information. Such cases are not handled in the preprocessor, which increases the complexity of the searches (thoroughly explained in section 7.2). The reason for choosing this approach is to maintain a consistent preprocessor by not making special rules during the splitting, and prevent unnecessary complications. Besides, this is also more effective since such errors are handled together with the search, which saves us from redundant checks. Table 6.1 illustrates some examples where the preprocessor will create “problematic” output, which later needs to be handled during searches:

Input	Tokenized
120678 44587	(120678),(44587)
November 2012	(November), (2012)
68 29 30 50	(68),(29),(30),(50)

Table 6.1: Example of problematic output

Another drawback with the simple tokenization approach is that the preprocessor becomes vulnerable to misspellings or wrongful punctuation markings. A big part of the approach is to look up *tokens* in reference works, such mistakes need to be handled before proceeding with further investigations. For example, if two words are divided by a comma without whitespace, the resulting *token* will consist of two words. Table 6.2 presents some examples:

Input	Tokenized
”received medication.He is”	(received),(medication.He),(is)
Day/evening/night	(Day/evening/night)

Table 6.2: Example of problematic output

Such behavior is handled inside the tokenizer, with a simple method that examines each *token* and looks for sequences of alphabetical characters separated by special characters. Further, the tokenizer creates a list of all the possible words existing within a *token*, which are denoted as *innertokens*. The list is created by splitting each *token* by every special character (illustrated by example in table 6.3).

Input	Tokenized	Innertoken
medication.He	medication.He	(medication),(He)
Day/evening/night	Day/evening/night	(Day),(evening),(night)
15/3-2012	15/3-2012	(15),(3),(2012)

Table 6.3: Example of innertokens

As the table illustrates, the *innertokens* can contain sensitive information. A detailed description how the *innertokens* are used, is described in section 6.5.1.

6.5 Data Structure

The data structure produced by the preprocessor forms the basis for rest of the pipeline. One important clarification is that further on, the words “token” and “sentence” will be written with italic³ font when we refer to object instances in our implementation.

The following section describe their most important attributed.

6.5.1 Token

Tokens are the smallest and most important units in our system. Tags are always assigned on *token*-level, which makes the classification fully dependent on this information. A *token*-instance has several attributes, whereby the most important are presented in the following list:

OriginalToken: The *token* as it occurs in the text.

StrippedToken: A string where special characters occurring at the beginning or ending of a *OriginalToken* is removed. The *token*, “-Date:”, will get *Date* as *StrippedToken*. *Strippedtoken* is used to facilitate for dictionary lookups.

Innertokens: A list of possible words within a *token*, split by special characters. This attribute is further explained in section 6.4.

POS-Tag: The part-of-speech tag assigned by the POS-tagger (6.3).

Stemmed Token The stem of the *token* (if it is a word), i.e. the word “*diseases*” has the stem form “*disease*” and “*increasing*” has *increase*.

FinalTag: The final *categorytag* (explained in 7.1) of a token.

³ *This is italic font*

6.5.2 Sentence

A *sentence* consists of several tokens, and has some attributes which enables sentence-level searching. The following list contains the most important attributes of a *sentence*-instance:

OrginalSentence: The sentence as it occurs in the text.

StrippedSentence: A string composed of each *tokens's strippedtoken*. This attribute is used during the *sentence*-level tagging.

Phrases: Holds information of the *phrases* within the sentence. A phrase is a list of tokens composed of *token*-level n-grams. Phrases can also be classified. For instance, if the sentence contains an International Classification of Diseases (ICD)-10 description, a *phrase*-object with the corresponding *tokens* and an ICD-10[53] tag is added to the sentence.

Pattern Matching

The pattern matching component is the main component of the application. The pre-processed data is sent into this component as a *document*-object (as illustrated in figure 6.1), in which several types of analysis is performed, on both word- and phrase-level.

Our main strategy is to consider every *token* as sensitive, and accept one-by-one as long as they meet certain criteria. The process is tag-based which means that *tokens* and *phrases* are tagged by categories during the investigation. For instance, when a *token* gets positive date-validation, it will be tagged as a date, and further be examined for other categories. Hence, *tokens* and *phrases* will be able to receive multiple tags, whereby each tag has a Boolean¹ value denoting either sensitive or insensitive. When every *token* and *phrase* have been investigated and tagged, the *document*-object passed into the postprocessor. But first, a detailed description of the pattern matching component will be presented.

The pattern matching component can be further divided into two subcomponents based on whether it uses *tokens* or *sentences* as input. Figure 7.1 illustrates the overall structure of the pattern matching.

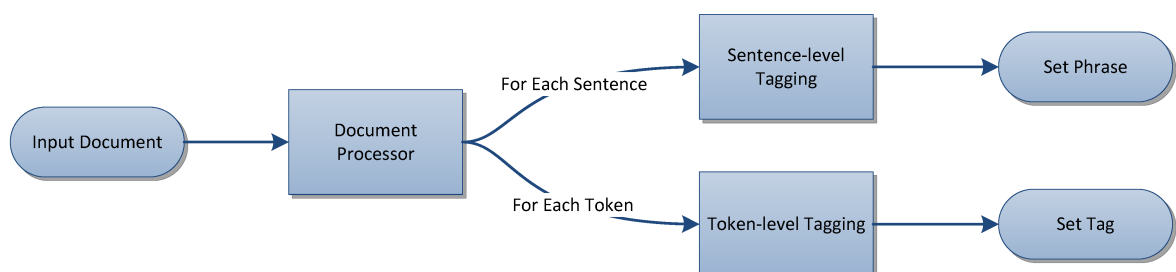


Figure 7.1: Overview of the pattern matching

¹binary value: true or false

7.1 Categorytag

As our approach is tag-based, the main technique is to assign classification tags to *tokens* and *phrases*. The classification tags are implemented as independent data structures called *categorytags*. A *categorytag* denotes a category, and is assigned to a *token* when a *token* is classified. The *categorytag* has three attributes; tag, code and description.

- The tag attribute indicates the type of the *categorytag*, i.e. ICD10. It includes an associated sensitivity-value, indicating the sensitivity of the tag; for instance if a *token* is tagged as a name, the corresponding sensitivity value is true.
- The Code attribute enable medical codes to be associated with a tag. Currently, only medical codes (described in section 7.2.3) use this feature.
- The Description attribute holds a textual description of the *categorytag*, mostly used for medical codes.

7.2 Token-level Tagging

Token-level tagging is performed by three components: regular expression, dictionary lookups and medical codes search. This is the performance-intensive part of the application as each *token* is looked up in an extensive library of reference works. However, a substantial drawback is that most of the *token*-level tagging is performed independently, without any contextual considerations, which causes unreliable tagging and in turn makes the system prone to wrong classification. When the *token*-level tagging is completed, a final *categorytag* is classified. The final classification is a simplified process, which is described in chapter 9. The sections below describe the three main components of the *token*-level tagging. Figure 7.2 provides an overview of the *token* matcher.

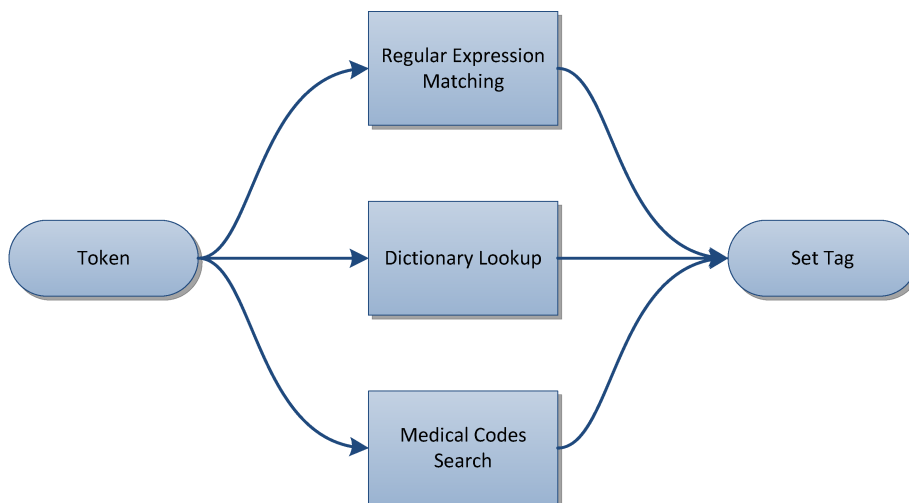


Figure 7.2: Overview of the *token*-level tagging

7.2.1 Regular Expression Matching

Regular expressions are used to recognize sequences of characters with determined structure. Inspired by Neamatullah et al.[1], among others, the application use regular expressions to recognize several patterns. The following list shows each category of sensitive information recognized by regular expressions, whereby each has a belonging *categorytag*.

- Date
- Mail Address
- Website Url
- National Identification Number
- Phone Number
- Bank Account Number
- Time (hour)
- Month
- Year
- Drivers License Number
- Car Registration Number
- Zip Code

Every category listed above has a true sensitivity-value, implying that the matching tokens reveals sensitive information, and thus will be removed. At the same time, the regular expression matcher is not limited to only accept sensitive categories, which provides the possibility to extend the application with insensitive patterns as well.

As figure 7.3 illustrates, some of the regular expressions are assisted by checks to increase the precision. These checks are necessary since regular expressions only recognize patterns without any further possibilities to perform calculations (i.e. in order to validate checksums). For instance, dates have a certain format, and the application use regular expressions to match the format of a potential date, and then use additional methods to ensure that the potential date is a valid date.

The regular expressions component is mainly dependent on *tokens*, but since the tokenizer (thoroughly explained in section 6.4) splits on whitespaces it is necessary to analyze multiple *tokens* in case familiar patterns are split. Our workaround for this problem is to use a pattern called *part*, which only match if a *token* may be a part of a pattern across multiple *tokens*, and then match the pattern on these.

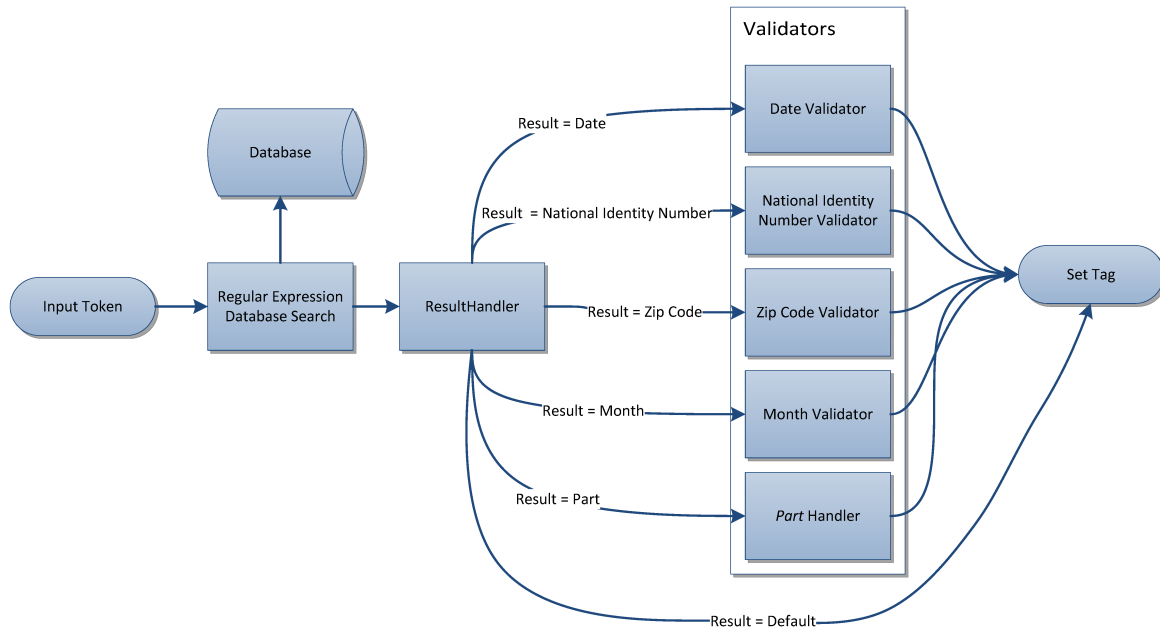


Figure 7.3: Overview of the regular expression matching

7.2.2 Dictionary Lookup

The dictionary lookup is a simple dictionary search, where each of the dictionaries are matched against each *token's* lowercase form *strippedtoken* (see 6.5.1). The dictionary database is indexed using Apache Lucene[50], which was chosen simply because it offers appropriate functionalities for our purpose, making it fast to implement. Each dictionary in the dictionary database has an associated *categorytag*, constraining a dictionary to consist of only one type of words. Hence, if a *token* match against a person name dictionary, the token will be tagged as a person name. An issue related to this approach is that it is dependent on reference works of distinctive categories, such as: person-names, street names, geographic locations, which simplifies the process of classification. When it comes to more general dictionaries, as *norkompleks* ([54]) and *ordnett* ([55]), it is difficult to decide a sensitivity value. Such dictionaries include most of the Norwegian words which makes it challenging to assign a *categorytag*.

It is straightforward to add new dictionaries, which is a big advantage, providing high modifiability. Figure 7.4 illustrates overall structure of the dictionary search.

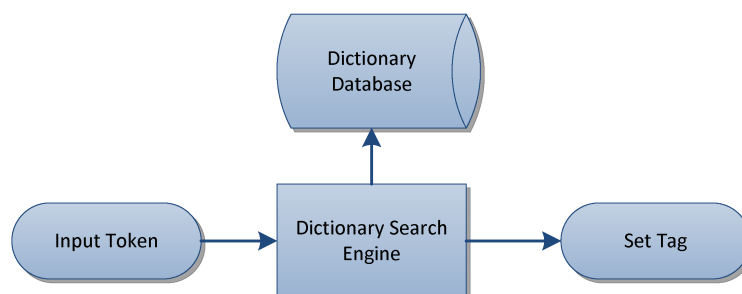


Figure 7.4: Overview of the dictionary lookup component

Dictionary lookup is an essential part in most pattern matching approaches (4.2.1). This is also the case in our implementation through which the majority of the *tokens* are assigned (several) *categorytags*. The following list shows all the *token* types supported in the dictionary database.

- Dosage unit
- Person Name
- Medicament
- Element
- Nationality
- Organization number
- Place name
- Weekday
- Numeral
- Physicians' Desk Reference (*Felleskatalogen*[56])

7.2.3 Medical Codes Search

Medical notes often contain standard medical terminologies from *medical vocabularies* and *medical classification systems*. *medical vocabularies* are system of disease names with explanation of their meaning, whereas *medical classification systems* is descriptions of medical diagnoses and procedures into universal medical code numbers. These terms are important to recognize as they constitute the medical contents of the notes, which in turn is alpha and omega to researchers. However, some diseases are "critical" to keep inside a de-identified clinical note as they may be rare and apparent, thus indirectly identifiable (described in 2.4.2). Our initial plan was to use a statistical overview of disease-occurrence, and use this to discard rare diseases. Unfortunately, this was never implemented because of late access to statistics, but the application is able to recognize several codes, making it ready for such an extension. Below follows a list of all the supported codes.

- **A**natomical **T**herapeutic **C**hemical Classification System (ATC) [57]
- **I**nternational **C**lassification of **D**iseases Version 2010 (ICD-10) [53]
- **I**nternational **C**lassification of **P**rimary **C**are Version 2 (ICPC-2) [58]
- **M**edical **S**ubject **H**eadings (MeSH)[32]
- **N**orwegian **C**lassification of **M**edical **P**rocedures (NCMP) [59]
- **N**orwegian **C**lassification of **R**adiological **P**rocedures (NCRP) [60]
- **T**he **N**OMESCO **C**lassification of **S**urgical **P**rocedures (NCSP) [61]
- **S**ystematized **N**omenclature of **M**edicine **C**linical **T**erms (SNOMED CT) [24]

In the same manner as dictionary lookup, each *token* is looked up inside the medical terminologies and classification systems, and gets the corresponding tag when it matches. The information about the medical terms is stored inside the code and description attribute of the tag. Figure 7.5 illustrates the searching procedure of a input *token* containing a Norwegian Classification of Medical Procedures (NCMP) code.

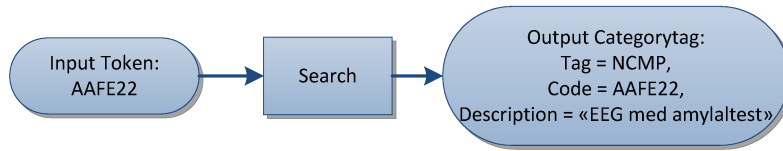


Figure 7.5: Example medical codes search

7.3 Sentence-level Tagging

When the *token*-level tagging is accomplished, the input text is further investigated and tagged on a *sentence*-level. This is partially performed through dictionary matching, and partially through the tags that already exists within each *token*. *Sentence*-level analysis is beneficial in order to fully exploit dictionaries that contains expressions and phrases consisting of several *tokens*. As referred to earlier, the *token*-level tagging is performed independently and makes the judgment unreliable, which places a huge limitation on the decision making phase.

Sensitive information is often composed of several *tokens* with several meanings which the *token*-level tagging may ignore:

- *Grensen skole (Grensen school)*
- *Ås kommune Oppvekst og Kultur (A company in Ås municipality)*
- *Legevakten i Klepp (The accident and emergency unit in Klepp)*

These examples show the importance of considering multiple *tokens* which also might provide indications about the context thereby a *sentence*-level tagging component is included in the implementation design.

There are a lot of ways to use the sentences from the input text to search in the dictionaries. A simple approach is to look for exact matches in the dictionaries and obtain highly reliable tags. However, this also requires that the dictionary descriptions must be equivalent to the search-phrases, which can be impeded by misspellings, the lack of words or different order of words. In order to regard such factors, the *sentence*-level tagger is composed by three components: Phrase-search, N-gram search and dose-matcher. When a phrase is matched by one of these components, the matched phrase and information about the respective dictionary is added to the *sentence*-object as a *phrase*-object. Mistakes made by sentence splitter(6.2) may have an impact in this process as the output is used to perform *sentence*-level tagging. Figure 7.6 illustrates the overall structure of the *sentence*-level tagging.

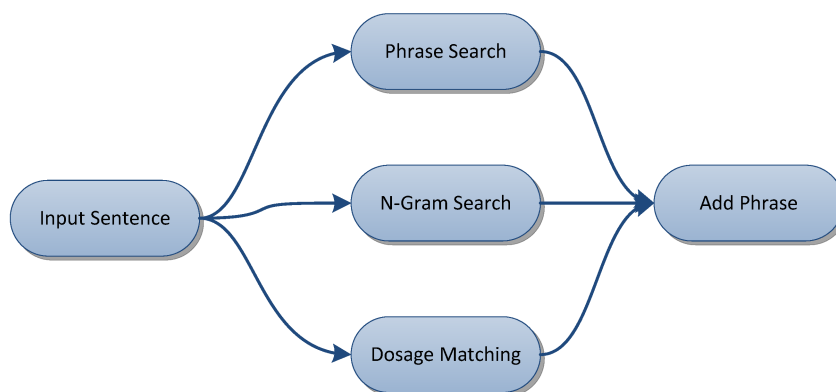


Figure 7.6: Overview of the sentence-level tagging

7.3.1 Phrase Search

The phrase-search component is used to recognize full sentences and phrases composed of more than 3 *tokens* (since n-gram is restricted to a maximum of three *tokens*, see 7.3.2). The search phrases are looked up in the dictionary database, which includes medical dictionaries and classification systems (ICD-10[53], MeSH[32], Anatomical Therapeutic Chemical Classification System (ATC)[57], etc.) together with sensitive dictionaries (Place names, clinical firm names, etc.) In the same manner as the *token*-level tagging, each dictionary has a belonging tag. The following list presents the types of phrases currently supported by this search:

- The NOMESCO Classification of Surgical Procedures (NCSP)
- ICD-10
- International Classification of Primary Care (ICPC)-2
- SNOMED
- NCMP
- Norwegian Classification of Radiological Procedures (NCRP)
- Organization Names

Phrase search in the abovementioned dictionaries enables the possibility to recognize medical diagnoses and procedures from classification systems without the corresponding code present. When a description is matched against a search-phrase, the corresponding code is looked up, after which a *phrase*-object is instantiated and assigned to the *sentence*. Even though the majority of the list-items are medical classification systems which includes a code and a description, the search works on plain dictionaries as well, such as organizational names.

Each sentence is compared to the phrases in the phrase dictionaries using SimString[62] with the cosine similarity as similarity measure. To determine the similarity between the search-phrase and the dictionary contents, the cosine similarity score is compared

with a self-tuned threshold, whereby a match is denoted when the score exceeds the threshold. Cosine similarity is widely used in the field of IR[63].

The cosine similarity uses word-frequencies to vectorize both the search-phrase and dictionary-phrases, and computes a score based on the cosine angle between these vectors. The score is negative correlated with the angle, hence the score increases when the angle decreases. Another property of this technique is that score is computed independently of the word ordering, which is exactly why we employ this method; in order to recognize medical contents and sensitive information in a wider range than exact match.

Cosine similarity is defined the following equation:

$$Similarity = \frac{\vec{S} \cdot \vec{P}}{\|\vec{S}\| \|\vec{P}\|}$$

Where S is the sentence vector and P is the phrase vector.

The vectors are based on the Bag of Words (BoW)-model. Here follows a simple example:

Phrase: Congenital heart block

Sentence: The patient has congenital heart block

Word	Sentence	Phrase
block	1	1
congenital	1	1
has	1	0
heart	1	1
patient	1	0
the	1	0

Table 7.1: The number of occurrences of each word

$$\vec{S} = [1, 1, 1, 1, 1, 1]$$

$$\vec{P} = [1, 1, 0, 1, 0, 0]$$

$$Similarity = \frac{\vec{S} \cdot \vec{P}}{\|\vec{S}\| \|\vec{P}\|} = \frac{3}{\sqrt{3}\sqrt{6}} \approx 0,71$$

The similarity is then compared to the decided threshold.

When the phrase achieves a score which exceeds the similarity thresholds, it is further investigated. This is necessary because the cosine similarity threshold needs to be low in order to find potential phrases, resulting in a lot of incorrect matches. Hence, the cosine similarity is only used as a filter to weed out the most similar phrases from the dictionaries. *Simstring* has proven to be fast and efficient [62], which is an advantage as the application makes use of several huge dictionaries. Even though we have disregarded high time efficiency as a requirement to the application, it is a convenient and time-saving method which serves the purpose of phrase searching in an effective manner.

Depending on the threshold, the search will find several potential phrases within a sentence, whereby lower thresholds give higher hit-rates. As initially mentioned, the threshold will be fine-tuned as a part of the experiment since the threshold may influence the overall performance.

The most similar phrases from the dictionaries filtered out by the cosine similarity component are further investigated, one by one, by means of three techniques:

Levenshtein distance: The phrase is compared to the input-sentence by *token*-level overlapping, in which a Levenshtein distance (edit distance) on each *token* is allowed. Levenshtein distance is used in a wide range of applications like text string matching, character recognition, spell checking, fuzzy search and record linkage (2.10.2). The Levenshtein distance between two sequences of texts denotes the number of *edits/primitive operations* required to change one textual sequence into the other. The primitive operations are insertion, deletion and substitution. The application use equal weighting for these operations whereof each cost-value is set to one. Here follows some example to illustrate the method, in which the function $\lambda(\alpha, \beta)$ denotes the Levenshtein distance between text sequences α and β , or the cost of transforming α into β :

- $\lambda(\text{"cool"}, \text{"fool"}) = 1$, (substitution of "c" for "f")
- $\lambda(\text{"De - identification"}, \text{"identification"}) = 3$, (deletion of "D", "e" and "-")
- $\lambda(\text{"prove"}, \text{"improvement"}) = 6$, (insertion of "i", "m", "m", "e", "n", "t")

This is a convenient way to account for potentially misspelled words. Clinical notes may contain several misspellings, Ruch et al.[64] reported error-rates up to 10%, whereby follow-up notes are most vulnerable. The application has an adjustable *edit*-distance per *token* in a phrase, and different *edit*-distances will be tested during the experiment. However, the distance is not normalized with respect to token-length which might be a drawback since longer *tokens* may involve a higher amount of misspellings. One proposed solution is to use a dynamic Levenshtein distance, which has shown to give better results [65].

Comparator: The phrase is compared to the input-sentence by a *token*-level intersection, in which it is performed a count of common words independent of word ordering. This method provides the ability to recognize sentences with similar meanings and dissimilar structure. Take for instance this input-sentence from a clinical note: *Abnormalities on hair shaft and hair color after chemo therapy*. This sentence consists of every word present in the ICD-code *L67: Hair color and hair shaft abnormalities*. Hence, when a phrase from a medical classification system or sensitive dictionaries exists within the input-sentence, the matching *tokens* in the sentence are tagged with corresponding tag. In this example, the *tokens* *hair*, *color*, *and*, *hair*, *shaft*, *abnormalities* are tagged ICD. The drawback is that every word has to exist within the dictionary-phrases, which is a rather poor implementation, since redundancies such as stop words prevents the fully exploitation of the comparator. It would be sensible to remove stop words from both parts in order to obtain better effect.

Exact matching: Compares the phrase to the sentence by comparing if the phrase exactly overlaps some parts of the sentence. Misspellings are not accepted by using this technique, which is an obvious drawback when recognizing phrases in record notes.

The following figure illustrates how the potentially matching phrases are processed:

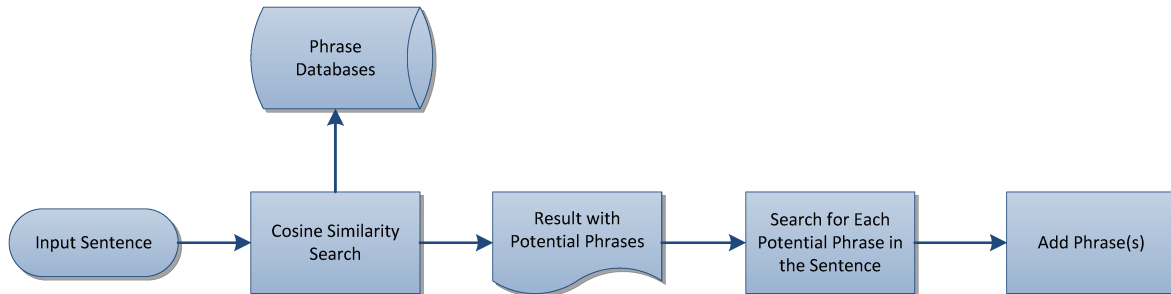


Figure 7.7: Overview of the phrase search

7.3.2 N-Gram Search

The second component of the sentence-level tagging is the n-gram search. A *token*-level n-gram denotes the processes of composing a phrase from n consecutive *tokens*. Hence, n consecutive *tokens* are fetched from the *sentences*, and their compositions are further used to search in reference works. Our application uses bigram (n=2) and trigrams (n=3). Here follows a short example of a bigram and trigram:

Sentence: This is a short sentence

Tri-gram:

This is a
is a short
a short sentence

Bi-grams:

This is
is a
a short
short sentence

N is limited to a maximum of three *tokens* because the phrase-search handles longer sequences, and is likely to match if a consecutive sequence of four or more *tokens* exists in both the search phrase and reference work. N-grams is a frequently used technique in the field of computational linguistics, and is also utilized in several de-identification applications on either character-level or word-level ([66],[67],[46])

The component is implemented in a similar manner as the dictionary lookup (described 7.2.2) with Apache Lucene[50] indexed dictionaries, whereby exact match is required for tagging. When a n-gram phrase finds a match in a reference work, the *n-gram* is assigned to the sentence as a phrase-object for use during the classification.

7.3.3 Dosage Matching

Medicament dosages are common parts of clinical notes and thus has to be recognized by the application. Dosage descriptions are recognized by regular expressions and dictionary lookups, an approach inspired by Deléger et al.[34]. Dosage descriptions includes whitespaces and has to be matched on a sentence-level, due to the problems discussed in section 6.4).

Unfortunately, the lack of robustness resulted in an unsatisfactory dose matcher which tags too many *tokens*. One recurring problem was that dates were mistaken for doses by the regular expression, which we never managed to improve. As a quick fix, the dosage matcher was assigned least priority and applied on *tokens* which were unrecognized by other components. However, the performance will not be crucially affected in terms of recall, as dosages are considered as insensitive information.

Dosage descriptions often include dosage units and medicaments which are tagged by the dictionary lookup component. Medicaments are looked up in the *pharmaceutical coding system* ATC[57], which is an international drug classification. In order to combine regular expressions with the dictionary lookup, we have made another simplification; If two *tokens* within a sentence is tagged as either dose, dose unit or medicament, a *phrase-object* is instantiated consisting of the *tokens* in between. This *phrase-object* is tagged as a *medicament_phrase*. By experience we observed that *tokens* which occurs between dosages and medicaments usually involve information related to the medicaments, hence through this technique the application will potentially be able to recognize more dose/medicament patterns. Below follows an example of a *medicament_phrase* added to a *sentence*. Green *tokens* denotes matched medicaments, doses or dose units, whereas blue *tokens* becomes part of the *medicament_phrases*.

Medicaments: Sodium chloride 1x1 in each eye, glycerol 1x1 in left eye, Fragmin 5000 lE

7.4 Handling Unmatched Tokens

So far, we have presented the main pattern matching components and the appurtenant search techniques. However, even though the initial search is accomplished, there might still be several untagged tokens. The *classification-component* will classify each untagged word as sensitive which will not harm the recall, but on the other hand negatively affect the precision and fallout. Hence, in order to somewhat reduce this negative effect, the untagged *tokens* are re-processed by means of two techniques, *stemming* and *innertokens*.

7.4.1 Stemming

Stemming is the process of reducing *inflected* words to their *stem*. For instance the word *chairs* has the stem *chair*, and the words *connected*, *connective* and *connection*

have the stem *connect*. Stemming algorithms are used in a lot of IR applications, also within the medical domain[68], and have shown to increase the retrieval accuracy[69].

Our application makes use of a simplified and self-defined stemming method. It is implemented by using a *computational lexicon of Norwegian words* called *Norkompleks*. *Norkompleks* contains stem words, their POS, inflected words and word-senses. Each untagged *token* and its' belonging POS-tag is searched looked up among the stemmed words of *Norkompleks*. The POS-tag is used to disambiguate between different *word-senses*, which in turn makes the stemming heavily dependent on the accuracy of the POS-tagger, which is an obvious drawback. A potentially better approach could have been to only use POS-tags on words with different *word-senses*. Unfortunately, this approach only reflects our hindsight and thus it was never tried. Figure 7.8 presents a small example of the stemming procedure. If the stemming is successful, the *token* is processed once more using the *token* stem as input, in case the stemmed *token* exist in some of the initial dictionaries.

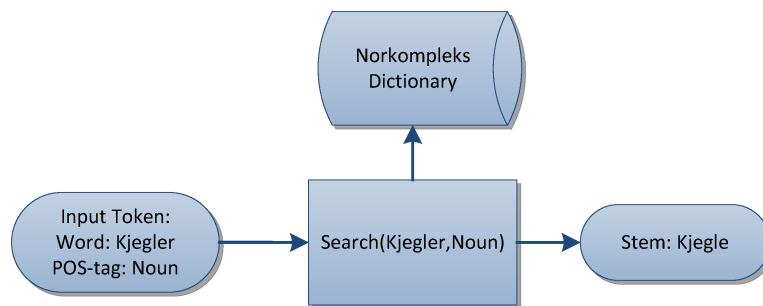


Figure 7.8: Stemming of the word "Kjegler" (English: Cones)

7.4.2 Innertokens

Unmatched tokens which have *innertokens* (discussed in 6.5.1) are processed once more. Each *innertoken* is processed as a normal token, but their resulting tags are added to the original *token*. Hence, if the original token is *Wednesday/Thursday*, the *innertokens* "Wednesday" and "Thursday" will be looked up in a dictionary and classified.

Wrongful punctuation will cause problems for this method, for instance for *tokens* as "Olsen.The". This *token* is split into two *innertokens*, "Olsen" and "The". Further, when "Olsen" successfully matches against a name dictionary, the original token "Olsen.The" will be tagged as sensitive. This is a partially wrong decision as "The" not is sensitive, however, it is better to be on the safe side. Hence, when a *token* has at least one sensitive *innertoken*, the remaining *innertokens* will be wrongfully tagged as sensitive. This is again a choice of preferring recall above precision.

This section will briefly describe the technique of weighting text units in automated text analysis, and further explain how the application makes use of Inverse Document Frequency (IDF) to weight *tokens* and *sentences*.

8.1 Weighting

Term weighting is a common technique used in computational linguistics, such as IR and data mining (described in 4.3.1). A *weight* is assigned to each term in a document or corpus, which often depends on the number of occurrences of the term in the document. Hence, we would like to compute a score (weight) $\omega(\tau, \delta)$ of term τ in corpus δ .

There exist several different term weighting methods, whereby IDF is one of the most widely used for estimating the term sensitivity in a corpus. Other commonly used techniques are Residual Inverse Document Frequency (RIDF), Information Gain (IG) and chi square [70]. Term frequency weighting has proven to be effective for filtering stop words (frequent words as “*the*”, “*is*” and “*a*”). For our purpose, term frequency is a cheap way to allow frequently occurring terms in the health record notes, as these presumably are insensitive. Nevertheless, the readability will be maintained by allowing frequent and common words. Infrequent terms in clinical notes indicates unique information, which might denote rare medical conditions or other kinds of peculiarities that can be used to identify the concerned patient. Hence, we chose to implement a statistical component based on term frequency.

8.2 Inverse Document Frequency

We have implemented a frequency module based on the term weighting technique IDF, with following assumption; terms present in a small subset of the corpus have higher probability of being directly identifying, in contrast to terms which are present in almost every document. Hence, the application makes the use of a lower and an upper threshold which is compared to each term’s IDF-score, whereby high IDF-score denotes

infrequent and rare terms, and vice versa. Our assumption is that insensitive information occurs more often than sensitive information, implying higher IDF-scores for sensitive information. However, this assumption might be on unsound basis since a corpus can contain repetitive sensitive information. Such information can be hospital or ward names, and also other kinds of repetitive identifiers occurring in the headers. The followings sections will present all the implemented statistical features in the application, which also is illustrated in figure 8.1.

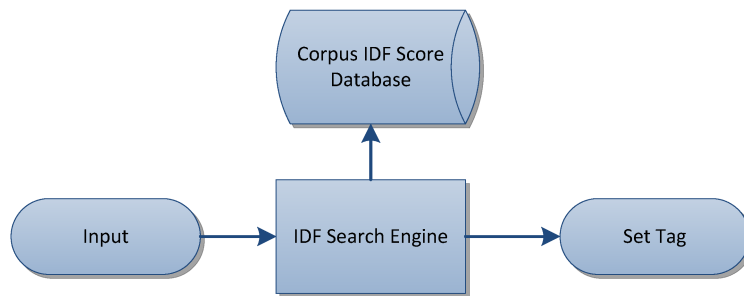


Figure 8.1: Overview of the statistical implementation

8.2.1 Formula

The inverse document frequency is expressed by the formula [71]:

$$idf_t = \log_{10} \frac{N}{df_t}$$

Where:

N denotes the number of documents in the corpus.

df_t denotes document frequency (amount of documents containing the term).

The formula has the following implications: The IDF-score of a rare term is high and the IDF-score of a common term is low. Hence, a text corpus consisting of 100,000 notes whereof 75,000 contains the word "and" and 6 contains the date "19.01.1967", will give the following IDF-scores:

$$idf_{and} = \log_{10} \frac{100,000}{75,000} = 0.125$$

$$idf_{19.01.1967} = \log_{10} \frac{100,000}{6} = 4.22$$

An important note regarding the thresholds we use is that these are corpus-specific and tuned with respect to the IDF-score of each term, which in turn is measured by the corpus size. Hence, if a corpus containing 50,000 notes is extended and doubled into 100,000 notes, the thresholds have to be re-tuned accordingly. This will probably have bigger influence on terms with high IDF-scores (rare terms), since the extension most likely will include instances of the common words.

For instance, if the above-mentioned text corpus is extended to 200,000 notes, the date "19.01.1967" will get following IDF-score if it still occurs 6 times:

$$idf_{19.01.1967} = \log_{10} \frac{200,000}{6} = 4.52$$

The IDF-score increases from 4.22 to 4.52, hence the idf-thresholds should not be fixed to an application should be re-tuned according to the corpus-changes.

8.2.2 Supported Features

The application has several features based on the IDF function, this section will describe the implemented features.

The IDF-function is used on both, *sentences*-level and *token*-level. Hence, when the application is executed, the IDF-score is computed for each *sentence* and *token*, and further stored in two databases. These are embedded in order utilize the IDF-scores of the entire corpus.

We use three different thresholds:

Threshold on *token*-level, where all tokens below the threshold α is considered to be insensitive. This is intended to recognize frequent words and terms, like stop words, in order to increase readability. As discussed, there is a potential risk that repetitive sensitive information achieves IDF-score below.

Threshold on *token*-level, where all tokens above the threshold β is considered to be sensitive. As assumed, sensitive information occurs less common and hence achieves high IDF-score.

Threshold on *sentence*-level where all sentence below the threshold γ is considered to be insensitive. This is to allow common sentences, which can prove beneficial by including *sentences* with untagged *tokens*. However, it is unlikely that sentences are equivalent except for document specific headings.

The third part of the pipeline is the classification part, in which the *tokens* and *sentences* are assigned one of two final tags, namely sensitive or insensitive.

9.1 Classification Algorithm

The classification procedure is carried out through a simple classification algorithm, in which the steps are prioritized by the assumed robustness of the searching techniques. The algorithm is described below:

1. ***Tokens* with certain POS-tags are marked as insensitive.**

The first step of the algorithm is to filter tokens by their corresponding POS-tags, in which following POS-tags are considered as insensitive:

- Preposition
- Conjunction
- Pronoun
- Subordinating conjunction
- Determinative
- Interjection
- Infinitive marker

The *tokens* which belongs to these POS-tags are assumed to be insensitive. It is still a risky simplification since these *tokens* may reveal indirectly identifiable information, such as gender (pronouns). However, we have chosen to classify gender as insensitive, despite the fact that gender might be used to identify a person (discussed in 2.10.2) .

2. **Every *token* within a tagged *phrase* in each *sentence* are assigned the *categorytag* and sensitivity value belonging to the *phrase*.**

During the *sentence*-level tagging, the sentences may have been tagged with several *phrase-objects*. Our assumption is that a match in the *sentence*-level tagging is more robust since several *tokens* have to match against the dictionaries, thus the *sentence*-level tagging is dedicated high priority. Hence, every *token* involved in the matched *phrase-objects* is assigned the respective *categorytag* and *sensitivity-tag*.

3. **Each *sentence*'s IDF-score is compared against the pre-defined *sentence*-IDF-threshold, whereby each *token* in a *sentence* falling below the threshold is classified as insensitive.**

Frequently occurring *sentences* are assumedly insensitive, thus the belonging *tokens* are classified as insensitive.

4. **For each unclassified *token*, the potential tags are examined using simple rules and heuristics, before a final classification is made.**

Each token is investigated through different checks, and a final classification is made if there is a match. The investigation proceeds as follows:

- (a) If the token has one tag, choose this tag.
- (b) If the token only contains the tags **YEAR** and **ZIPCODE**, the preceding token is investigated. If the preceding token is “i” (“in”), the token is set to **YEAR**.
- (c) If the *token* only contains the tags **NAME** and **GEOGRAPHICLOCATION**, one of these tags are assigned as *finaltag* (randomly).
- (d) If the token contains **NAME** or **GEOGRAPHICLOCATION**, the POS-tag of the token is investigated. If the POS-tag is **NOUN_PROP**, choose the **NAME** or **GEOGRAPHICLOCATION** as final tag (if only one exists, assign this), else remove **NAME** and **GEOGRAPHICLOCATION** from the tags.
- (e) If no other rule has been matched; choose a random tag.

This step is hugely simplified and a great deal of checks could have been added. It is not to conceal that the classification part could have been carried out better, which did not happen due to time constraints. The intention was to enrich this “rule base” continuously along the development.

Rule (b) is implemented because zip codes and years have similar patterns (Norwegian zip codes consist of 4 digits), which may cause some tokens to be tagged with both. This happens in spite of the regular expression validation which happens on both categories; a token is only tagged as **YEAR** if it is between 1900-2140, and **ZIPCODE** if it exists in the dictionary containing Norwegian zip codes. Hence, every number between 1900-2140 which also has a zip code will be tagged as both **YEAR** and **ZIPCODE**. However, both categories are sensitive; hence wrong classification won't have any impact on the performance.

Rule (c) is implemented simply because there is no need for distinguishing between **NAMES** and **GEOGRAPHICLOCATION**, as both is directly identifying.

Rule (d) is needed because the name- and geographic location dictionaries are huge and extensive, hence common insensitive Norwegian words often matches against these, causing ambiguities. The workaround is to only consider the *tokens* with **NOUN_PROP** as POS-tag ,in order to limit the amount of ambiguous *tokens*. Ideally, this assumption is always correct, however only with a POS-tagger with perfect precision.

Rule (e) is far from the optimal solution, but as mentioned, the time constraint did not allow a richer rule base; hence random classifications are made if no other rule is applied. The performance is not influenced when all the potential *categorytags* have equal sensitivity, but when this not is the case a certain a good dose of luck is needed. Hence, a potential for performance improvement is to enrich the rule base and prevent random classifications.

5. **For each *token* without any tag, the *token's idf*-score is compared to lower - and upper IDF-threshold, where each token below the lower threshold is set to insensitive and each token above the upper threshold is set to sensitive.**

Frequently occurring *tokens* are assumedly insensitive and thus classified as insensitive. Rare tokens are assumedly sensitive and thus classified as sensitive. This step has less impact on the performance as such tokens without any potential tags regardless are classified as sensitive in step 7.

6. **For each *token* without any tag, the token is stemmed, whereby a successful stemming is followed by a jump to step 4 with the stem *token* as input.**
7. **Each *token* without any tag is classified as sensitive. To be in the safe side, the only choice is to remove such unknown *tokens*.**
8. **The proportion of sensitive *tokens* are calculated in each *sentence*. The percentage is compared to a threshold; if the percentage of sensitive tokens is above this threshold, each token in the sentence is classified as sensitive. The threshold is further denoted as SSP (sensitive sentence proportion)**

9.2 Classification Summary

The classification algorithm is hugely simplified. Especially step 4 needs an improvement, since the rules don't fully exploit the available information. Besides, using a random classifier is a huge drawback as it causes unreliable results. Information classification is an important phase of a de-identification application and the simplified algorithm will undoubtedly weaken the performance. Unfortunately, we were unable to test different classification methods due to time constraints. The classification process is, however, independently implemented; hence, the rule base can easily be extended.

The postprocessor regenerates the clinical documents, in which the sensitive information is replaced by respective *finaltags*, as illustrated in figure 10.1. A somewhat drawback by replacing the sensitive information with specific tags is that some of these tags might be wrong, which results in misleading output. However, this can be regarded as a trifle as it won't affect the performance.

The original files are read and compared to each *token* in our system, in order to regenerate identical documents, whereby the sentences, paragraphs and overall structure is maintained.

The postprocessor is modifiable, which makes it easy to add support for pseudonyms. Unfortunately, pseudonymous information relies on precise tagging; hence, a better classification algorithm needs to be implemented before the application can produce pseudonymous output.

```
Dato: <DATE>
Godkjent av: <NAME>
Dokumentnummer: <DOCNR>

INNKOMSTJOURNAL
Innleggende lege: Amk-sentralen
Pasient: <NAME> <NAME>
Alder: <YEAR> år
Diagnose: Brystsmerter

Tidligere sykdommer:
Pasienten hadde hjerteinfarkt i <YEAR>.

Aktuelt:
Smertor og nummenhet som startet i venstre hånd og spredte seg
oppover og mot hjertet ca kl <TIME> i dag. Kvalm, ingen effekt av
nitro. Ikke økt dyspnøe, ikke vært kaldsvett. Får nitro og morfin i
ambulansen, og innkommer smertefri

Medikamenter:
Fragmin 2500 IE x 1
Panodil 500 mg x 4
Nexium 40 mg x 1
```

Figure 10.1: Example output

Part IV

Results

Experiment Preparations

The experiment was conducted on a net-disconnected server inside a separate lab described in section 1.2.5. The performance evaluation was carried out by comparing the de-identified output against a manually annotated reference standard.

11.1 Reference Standard

This section describes the manual annotation process.

11.1.1 Note Types

The reference standard used to evaluate the de-identification application was annotated manually in the laboratory. A total selection of 225 clinical notes from the Norwegian EHR-corpus was manually de-identified, whereby the notes consisted of 75 discharge summaries, 75 nursing notes and 75 record notes. The intention was to annotate a higher amount of documents (at least 150 of each type) in order to achieve reliable results, however the time constraint did not allow.

Discharge summaries are clinical reports written by different kinds of health professionals, for each patient who is discharged from a hospital. Discharge summaries contain discharge diagnosis, operations, laboratory, x-rays, hospital course and various other hospital related documentation. The reason for including discharge summaries in the reference standard is due to text richness. Another factor is that these often involve a lot of background information about the patient, which increases the de-identification challenge.

Nursing notes are documents created by nurses or other health care professionals. In almost all hospitals, nurses write notes about their patients on a daily basis. These include summaries of accomplished daily routines like medication, issues, diagnosis and further notes about the medical plan. Nursing notes are often short, in contrast to discharge summaries. The Subjective Objective Assessment Plan (SOAP)-procedure is often used to record the patient interaction [72]:

Subjective: What the patient has to say.

Objective: What the health care professional observes.

Assessment: Quick summary of symptoms, diagnosis and overall progress.

Plan: The plan for the treatment.

Such notes often contains non-standard medical language, which makes them different from discharge summaries[1]. Hence, it can be interesting to observe how the performance is influenced by the different note types.

The last type is called record-notes (*Norwegian: "Journalnotater"*). These are recorded by different health professionals, and contain quick status checks, or supplementary documentation to ongoing treatments. Nursing notes is one example of record notes, but our set does not contain nursing notes. The set of record notes consists of physician notes, physiotherapist notes and chief physician notes. These are usually very short compared to discharge summaries. The amount of sensitive identifiers varies a lot and depends on the profession, and were included due to their shortness and the mutual variation as they are recorded by different health care professionals .

11.1.2 Annotation

The original files were copied into a separate folder in which the annotation was performed. The annotation method was straight forward; a special tag `<::sens::>` was placed subsequently after each sensitive word or sensitive textual sequence without whitespace, as shown below:

Original text: "Patient name: Hans Gunnar"

Annotated text: "Patient name: Hans<::sens::> Gunnar<::sens::>"

We annotated sensitive information on a *token*-level. Hence, the example above contains two sensitive instances, even though the these are a part of one sensitive identifier, i.e. patient name. The annotation was performed by both of us, whereby each note was annotated by one person, then quality assured by the another. It is common to quality assure two - or three times as human annotation is prone to error[1].

11.1.3 Choices

It is often a challenge to recognize sensitive information. The main challenge is to decide on what should be classified as so-called indirectly identifiable information. The annotator has to assume the knowledge of a potential intruder, and foresee that this knowledge together with the disclosed information not links to an individual, as explained in section 2.10.1. We chose to ignore indirectly identifying information in most of the cases, except when the information revealed extreme incidents, for instance incidents which could have been mentioned in the newspapers. For such cases, the other person was consulted, before making a final decision.

We didn't follow any guideline, however we used the compiled list from Oslo University's (described in 2.10.1) consisting of sensitive identifiers. However, as some textual sequences were challenging to annotate, we made some simplifications along the way to ensure consistent annotation. These are presented in the list below.

- Gender was not considered to be sensitive
- Number of children, and gender of children, is not sensitive with the exception of cases where the patient has more than 6 children.
- Ward names are sensitive
- Week days are sensitive

11.2 Modified Discharge Summaries

In addition to the 225 notes, we created a special test set with the same discharge summaries, but without the structured information. The files were produced by creating a copy of the annotated discharge summaries, and removing the structured information placed at the top and bottom of the document. We created this set in order to evaluate the application's performance on pure narrative text, which is the actual aim of the application.

11.3 Properties

As described above, sensitive information was annotated on *token*-level. This means that every sensitive identifier split by one or several whitespace characters were tagged. The table below provides a simple overview of the total amount of sensitive and insensitive tokens:

Type	Sensitive	Insensitive	Total	Sensitivity Percentage
Discharge summaries	2,152	16,786	18,938	≈ 11.4%
Nursing notes	662	6,320	6,982	≈ 9.5%
Record notes	506	3,032	3,538	≈ 14.3%
Entire reference standard	3,320	26,138	29,458	≈ 11.3%
Modified discharge summaries	668	14,933	15,601	≈ 4.3%

Table 11.1: The amount of sensitive and insensitive tokens in the reference standard

11.4 Maximum Token IDF

The inverse document frequency was explained in section 8.2.1, and is expressed by this fraction [71]:

$$idf_t = \log_{10} \frac{N}{df_t}$$

Where:

N denotes the number of documents in the corpus.

df_t denotes document frequency (amount of documents containing the term)

Our realistic EHR-corpus consists of 45614 clinical notes (N). The maximum IDF-score a *token* can achieve is when it only occurs once in the corpus. Since every token at least has one occurrence, the df_t can be set to 1. Hence the maximum *token* IDF-score can be calculated as follows:

$$idf = \log_{10} \frac{45,614}{1} = 4,66$$

The following chapter presents conducted experiments together with the achieved results.

12.1 Experimental Approach

Pursuant to research question 2 (1.2.2), one of the main objectives in this project is to investigate the performance of different methods and techniques, and how different combinations of these affects the performance. The various experiments were conducted on the manually annotated reference standard (described in the previous chapter), consisting of discharge summaries, nursing notes and record notes. The experiment is divided into three parts in which each part tests de-identification approaches based on different method combinations:

Part 1: A de-identification approach based on regular expressions and statistical methods.

Part 2: A pure pattern matching approach, without statistical methods.

Part 3: A de-identification approach, combining all components.

The results are presented in terms of recall, precision, fallout and f-measure. These fractions are normally used for de-identification purpose and are explained under section 4.4. Some results are presented in diagrams without the precise digits, however the precise results have been added to the appendix B.

12.2 Regular Expressions and IDF

The first experiment is to measure the performance of a de-identification approach based on regular expressions and IDF-scores. The aim is to observe how well a de-identification application performs without dictionary lookups. High performance is not expected as the application is tag-based, whereas the reference works can be considered as the

supporting beam of our application. On the other hand, it is interesting to observe how well the *tokens* and *sentences* are tagged on a statistical basis, by the support of regular expressions. Regular expressions are included in order to recognize certain types of directly identifying information, i.e. dates and national identification numbers.

As explained under section 8.2, the statistical component makes use of thresholds which are compared against each *token's* and *sentence's* IDF-score. The textual sequences that fails to be recognized by the regular expressions will be classified by the statistical component, which most likely involves the majority the *tokens* and *sentences*.

As the majority of the textual sequences will be classified by the statistical component, the binary classification is performed by looking at the *token/sentence idf*-score, in which a score above the IDF-thresholds denotes sensitive information and vice versa. The application is tested on three different test sets: Discharge summaries, nursing notes and the entire reference standard. The *token idf*-threshold is varied (x-axis) in order to observe how the four performance measures behave. We in only use two distinct values for *sentence idf*-threshold. This decision is made upon the assumption that almost every *sentence* is unique, hence *sentence*-level IDF-scores will in most cases be quite similar in contrast to *token*-level IDF-scores, whereby the few repetitive *sentences* will have lower scores and thus easy to catch.

12.2.1 Results

The following graphs illustrate the performance of a de-identification application based on regular expressions and statistical methods.

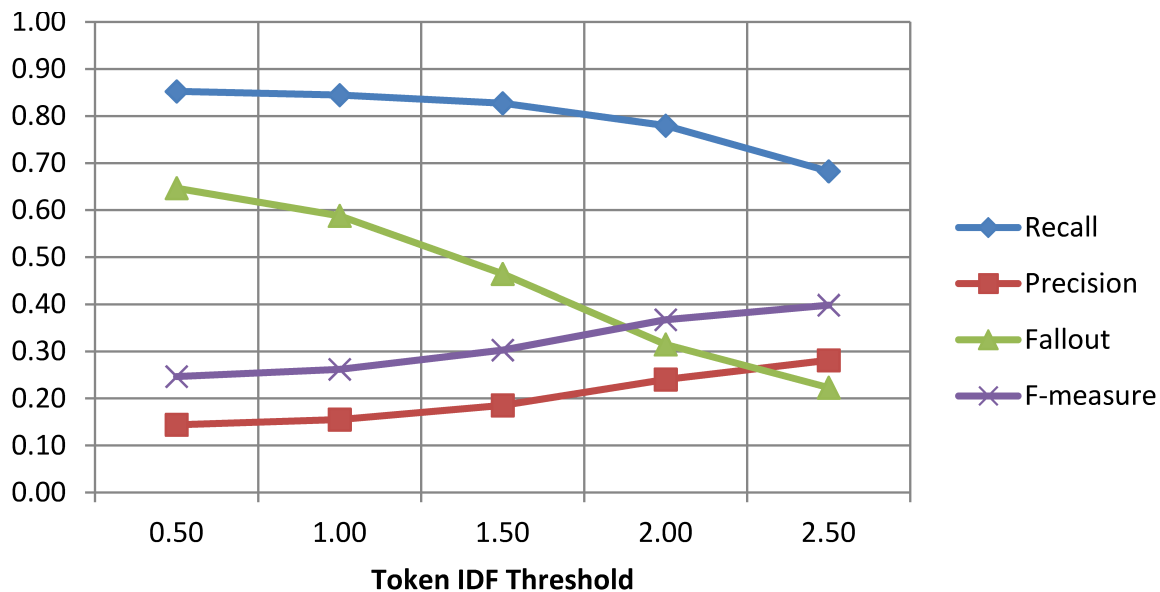


Figure 12.1: De-identification results on the entire reference standard using IDF and regular expressions, with *sentence-idf* = 2

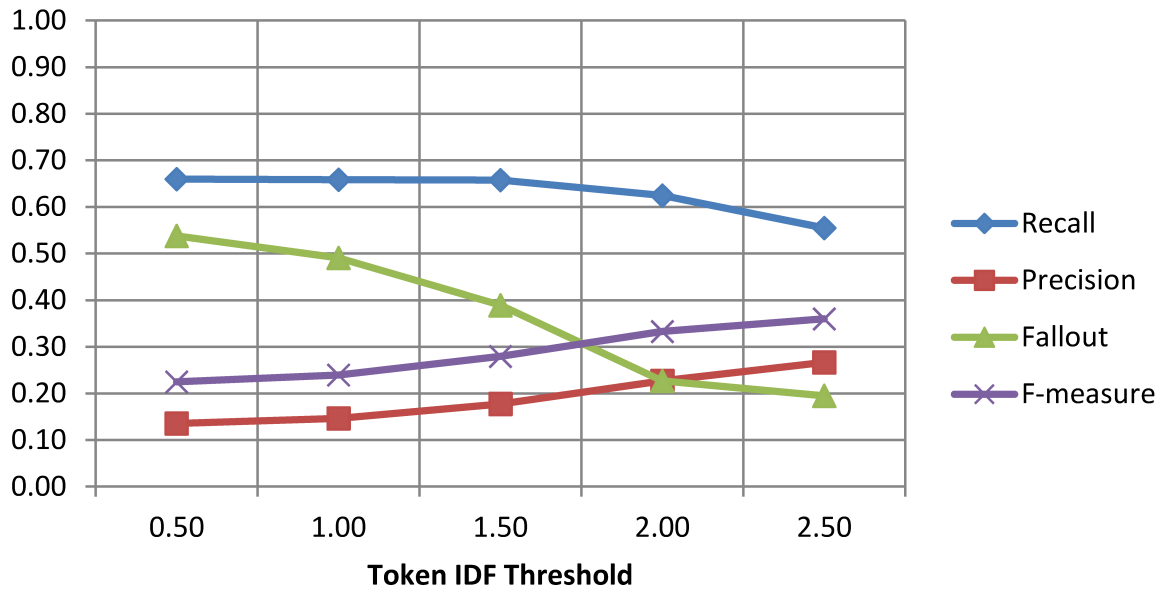


Figure 12.2: De-identification results on entire reference standard using IDF and regular expressions, with $sentence-idf = 4$

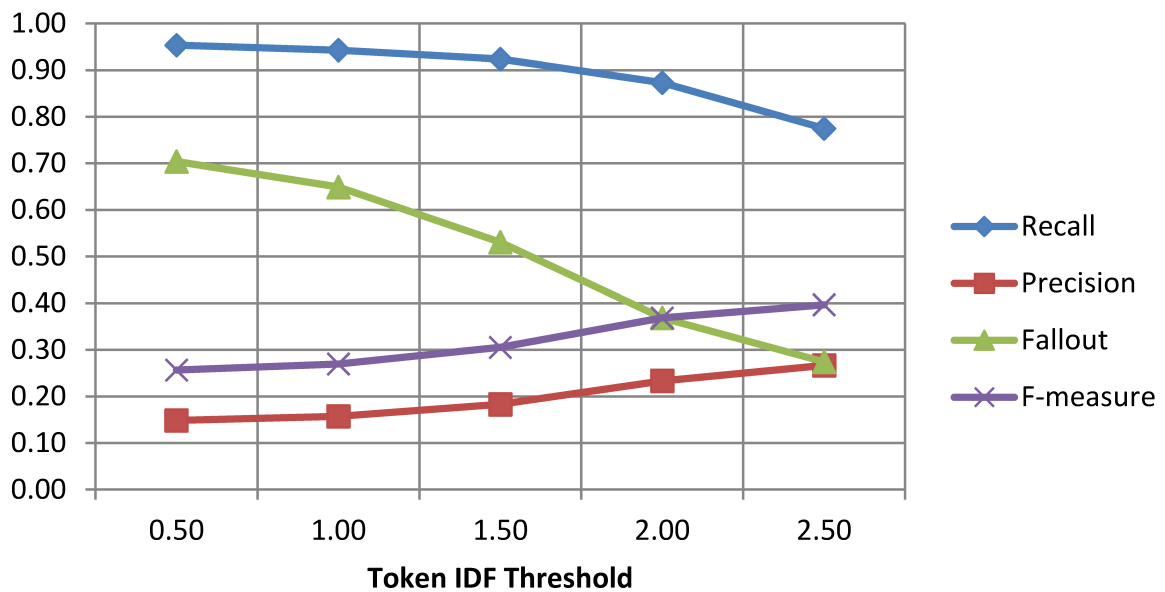


Figure 12.3: De-identification results on discharge summaries using IDF and regular expressions, with $sentence-idf = 2$

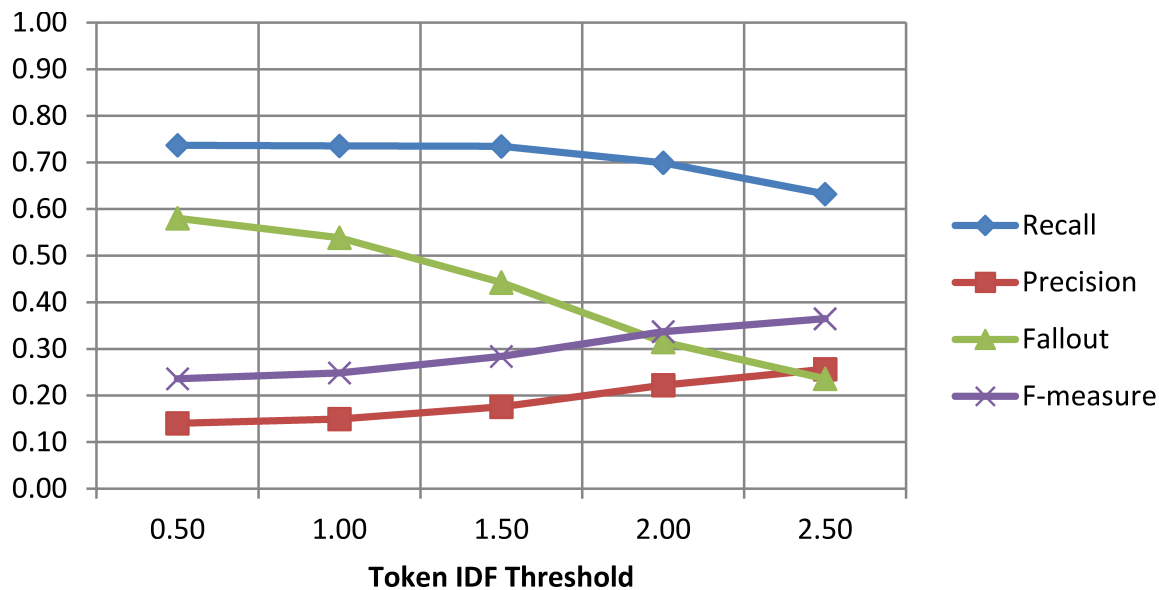


Figure 12.4: De-identification results on discharge summaries using IDF and regular expressions, with $sentence-idf = 4$

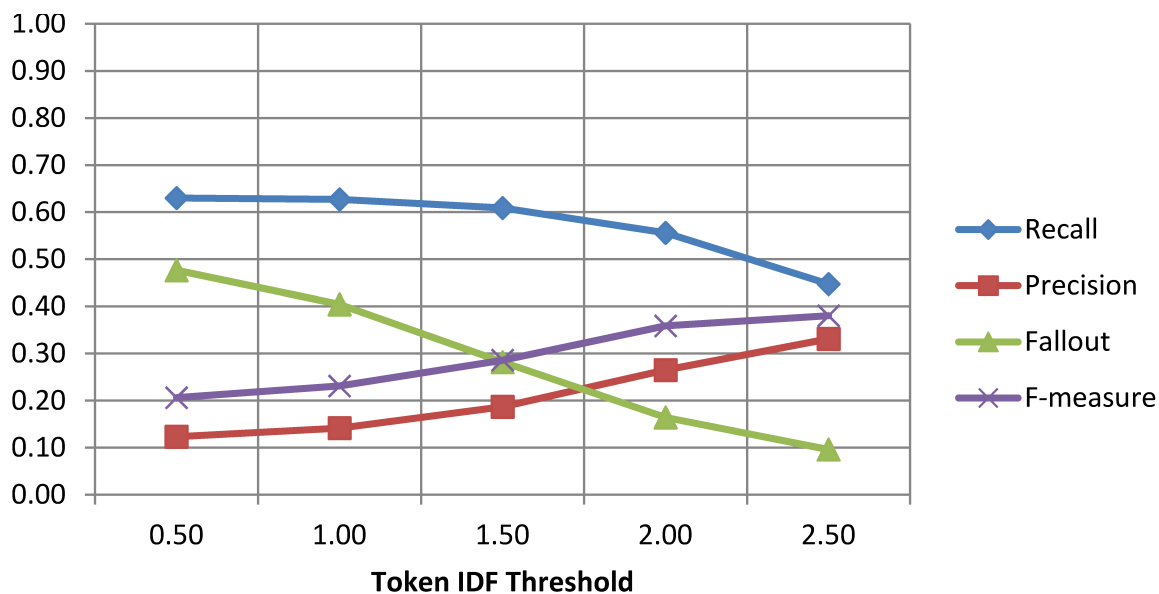


Figure 12.5: De-identification results on nursing notes using IDF and regular expressions, with $sentence-idf = 2$

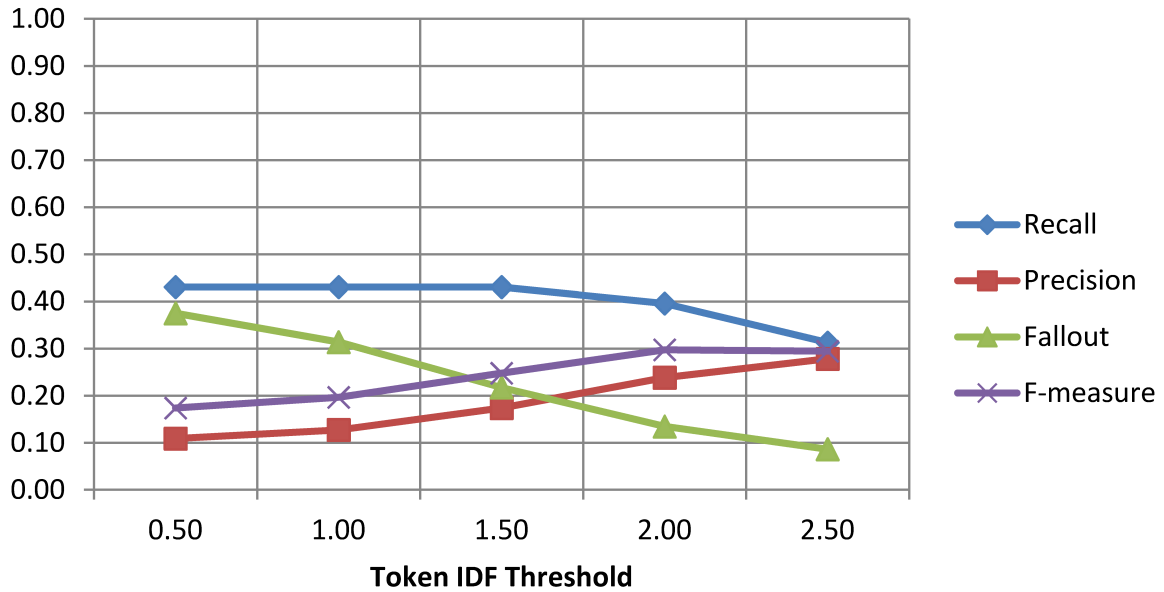


Figure 12.6: De-identification results on nursing notes using IDF and regular expressions, with $sentence-idf = 4$

The graphs above indicates weak performance when the *sentence* IDF-threshold is locked to 4. Changing the *sentence* IDF-threshold from 2 to 4 affects the recall significantly, the precision, however, remains unchanged. By increasing the *sentence* IDF-threshold, the criteria for considering sentences as sensitive is implicitly increased, which in turn has an evident impact on the recall as a lot of sensitive information “slips through” as insensitive. Although the recall has a clear leap, the overall binary classification isn’t affected to any significant extent in any of the test sets, which can be argued by looking at the precision and f-measure. Hence, more sensitive information is recognized, while the amount of miss-classifications also increases. All the diagrams indicate that the overall performance (considered f-measure) increases when the *sentence* IDF-threshold is increased.

The overall tendency is that an increase in the *token* IDF-threshold weeds out a certain amount of miss-classifications, which is denoted by a decreasing fallout-graph, however it also causes a significant decrease in the recall-graph. Put differently, high thresholds for classifying *tokens* as sensitive, decreases the recall, hence a lower amount of relevant information is recognized which in turn indicates an negative correlation between the *token* IDF-thresholds and recall graphs.

12.3 Pattern Matching

The next experiment measure the performance of a pure pattern matching approach. The statistical component is disabled, and the application totally relies on reference works and regular expressions.

The *phrase search* (7.3.1) is the only adjustable method. The Levenshtein-distance value was varied in the range $[0,3]$, in which a distance of 0 implies exact match. A

theoretical explanation of the Levenshtein-distance (or *edit-distance*) is provided in section 7.3.1. Additionally, this de-identification approach has been tested disregarding the word order which means that the words within a *phrase* has to overlap with a description in the reference works, without any spelling difference, in order to match.

LD = Levenshtein Distance

Phrase Search Setup	Recall	Precision	Fallout	F-measure
LD = 1	0,88	0,39	0,18	0,54
LD = 2	0,89	0,39	0,18	0,54
LD = 3	0,89	0,39	0,18	0,54
Unordered Words	0,88	0,39	0,18	0,54
Exact Match	0,88	0,38	0,18	0,53

Table 12.1: The pattern matching results on all note types

Phrase Search Setup	Recall	Precision	Fallout	F-measure
LD = 1	0,87	0,38	0,18	0,53
LD = 2	0,88	0,38	0,18	0,53
LD = 3	0,88	0,38	0,18	0,53
Unordered Words	0,87	0,38	0,18	0,53
Exact Match	0,87	0,38	0,18	0,53

Table 12.2: The pattern matching results on discharge summaries

Phrase Search Setup	Recall	Precision	Fallout	F-measure
LD = 1	0,85	0,32	0,19	0,47
LD = 2	0,86	0,32	0,19	0,46
LD = 3	0,87	0,32	0,19	0,47
Unordered Words	0,83	0,31	0,19	0,46
Exact Match	0,83	0,31	0,19	0,46

Table 12.3: The pattern matching results on nursing notes

Phrase Search Setup	Recall	Precision	Fallout	F-measure
LD = 1	0,98	0,55	0,13	0,70
LD = 2	0,98	0,54	0,14	0,70
LD = 3	0,98	0,54	0,14	0,69
Unordered Words	0,97	0,55	0,13	0,70
Exact Match	0,97	0,54	0,14	0,70

Table 12.4: The pattern matching results on record notes

It can be observed from these results that the mutual differences are quite small within each test set and that the overall performance is minimally influenced by the

Levenshtein-distance variations. This might be explained by the cosine similarity which is used in the phrase search; potentially matching phrases are filtered out by cosine similarity (explained in section 7.3.1), before investigating these with various Levenshtein-distances. Hence, the potential phrases filtered out by cosine similarity might be unvaried and monotonous, causing minimal impact on further analysis.

The results produced with a Levenshtein-distance value of 3 are slightly better than other results in terms of recall. In spite of minimal differences we chose to lock the Levenshtein-distance to 3 as an “optimized” (among the 4 results) value for further testing.

The overall performance achieved in the record notes increases impressively, compared to the first experiment. With a nearly top hitting recall and much better precision, this represents our best obtained results till now. The performance-increase on the record notes is also the main contribution for raising the overall performance on the entire corpus. Record notes are short and concise and rarely involves sensitive information other than names, dates and national identification numbers. These identifiers are easily recognized by regular expressions and the name-dictionary.

12.4 Pattern Matching and IDF

The last part of the experiment is to test a de-identification approach which is based on a combination of all the implemented methods and techniques during the project. Hence, the de-identification application includes regular expressions, statistical methods and reference works.

In the first part of the experiment, we used one IDF-threshold for *tokens* and one for *sentences*, in which we simply compared the idf-scores to the thresholds and made classifications accordingly. However, this part of the experiment does not fully depend on the idf-scores and has the advantage of utilizing reference works. We will use upper- and lower thresholds for *tokens*. When a *token's* idf-score exceeds the upper *token* idf-threshold it is classified as sensitive, and vice versa. In contrast to the first experiment, reference works and regular expressions will handle the *tokens* with *idf*-scores falling in between the upper and lower IDF-threshold. Tags from the reference works are assigned higher priority during the classification.

We will stick to only one *sentence* idf-threshold as most sentences are unique, hence the minority of similar sentences can most likely be separated by their significantly different IDF-score.

In order to maximize the performance for this approach, a “best-effort” setup was found by testing different values for the each variable, before proceeding with the tests. This involved the tuning of the upper *token* idf-threshold (I:U), *sentence* idf-threshold (I:S) and the *sensitive sentence proportion* (SSP). The lower *token* idf-threshold was the only variable during the third experiment.

The tuning was performed on the entire reference standard. The following table(12.5) only shows a selection of the obtained results by different setups:

I:L	I:U	I:S	P	SPP	Recall	Precision	Fallout	F-measure
1,50	4,00	3,00	LD = 3	0,80	0,75	0,44	0,12	0,55
1,50	4,00	2,00	LD = 3	0,80	0,78	0,43	0,13	0,56
1,50	4,00	1,70	LD = 3	0,80	0,83	0,43	0,14	0,57
1,50	5,00	1,70	LD = 3	0,80	0,83	0,44	0,14	0,57
1,50	5,00	1,70	LD = 3	0,60	0,84	0,41	0,16	0,55
1,50	5,00	1,70	LD = 3	1,00	0,83	0,44	0,14	0,58
1,50	5,00	1,60	LD = 3	1,00	0,85	0,43	0,14	0,57

Table 12.5: Tuning

Firstly, the *sentence* IDF-threshold was tuned. The values 3, 2, 1.7 and 1.6 were tried, where 1.7 gave the highest F-measure. The difference between 1.6 and 1.7 were minimal and chose 1.7 as the a suitable threshold.

Secondly, different values for SPP were tried, i.e.: 0.6, 0.8, 1.0. The results showed that 1.0 proved to be slightly better than the two others. However, the threshold value 1.0 theoretically denotes that every *token* within the *sentence* has to be sensitive in order to classify the fully *sentence* as sensitive. This is an extremely rare case, which is the actual reason for achieving better results. In other words, the application performs best without the SPP feature.

Thirdly, the upper *token* IDF-threshold was varied between the values 4.0 and 5.0. Since a *token* idf-score has a maximum value of 4.66 (see (11.4)), a *token* idf-score of 5.0 is unattainable which implicitly means that the feature is turned off. The difference was slight, but 5.0 gave better precision. This is expected since the upper token IDF-threshold is examined at a very late stage of the classification, having little or no impact on the overall performance.

The tuning resulted in a setup with the following values:

Token idf-threshold: 5.0 (Turned off)

Sensitive proportion: 1,0 (Turned off)

Sentence idf-threshold: 1.7

Phrase Search Setup: Levenshtein Distance 3

Further, the experiment was performed on each document type separately, and the entire reference standard. The lower *token idf*-threshold was varied in order to observe how the performance is affected when increasing the threshold.

The following results were achieved:

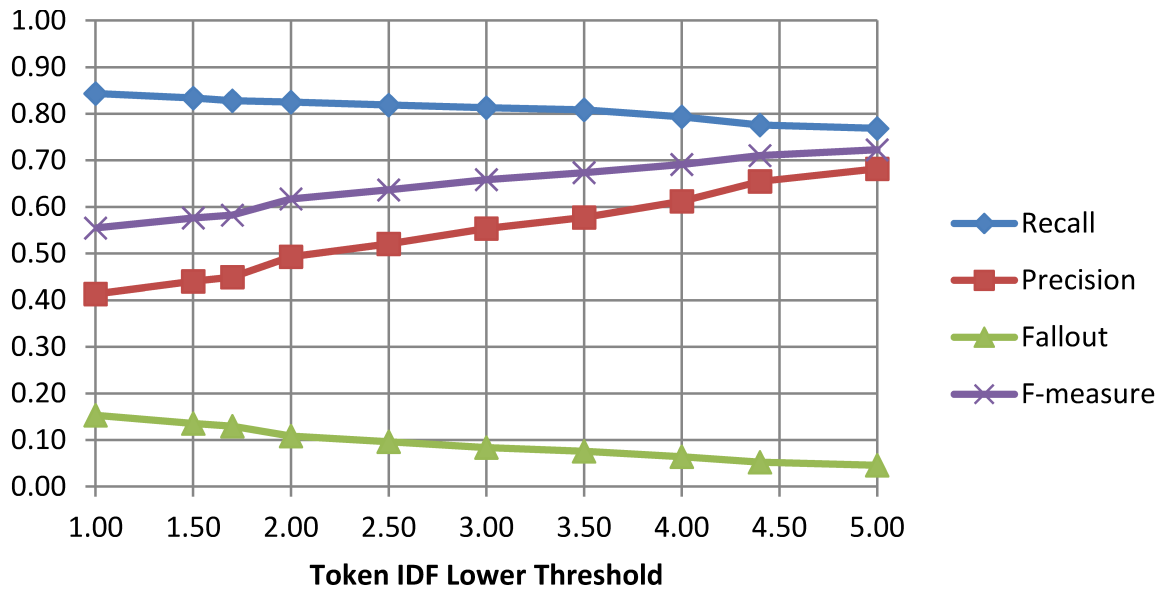


Figure 12.7: Results on all notes

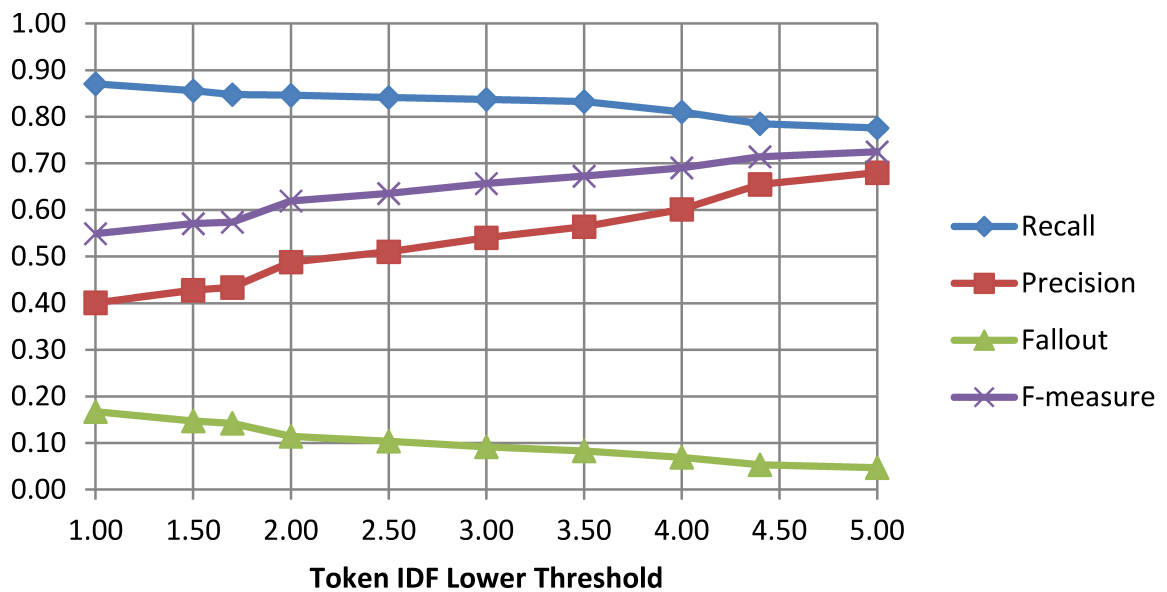


Figure 12.8: Results on discharge summaries

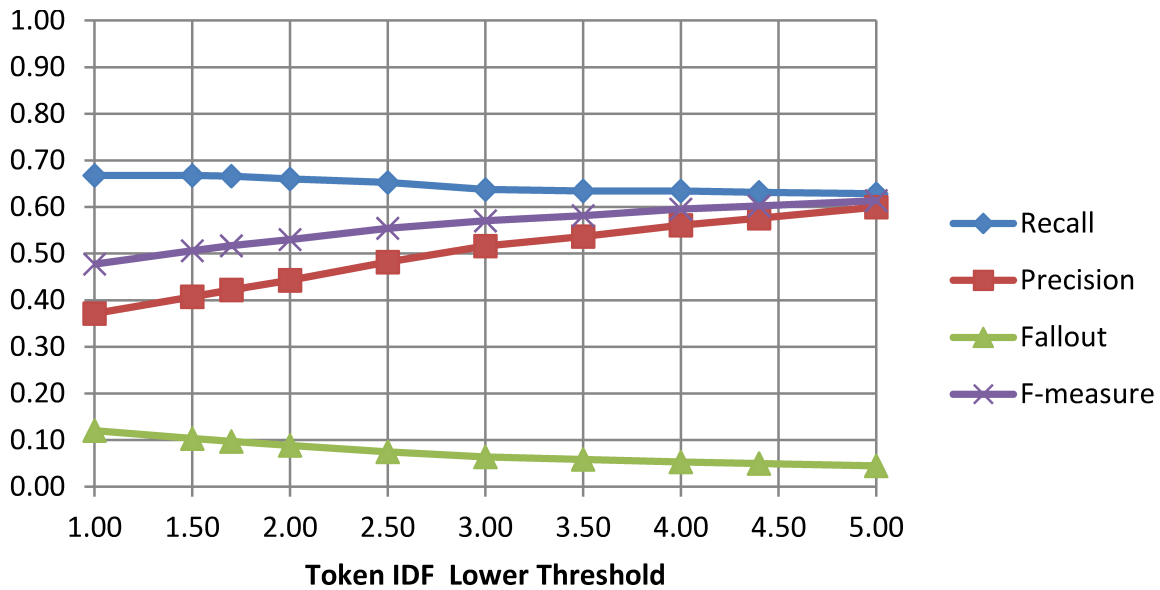


Figure 12.9: Results on nursing notes

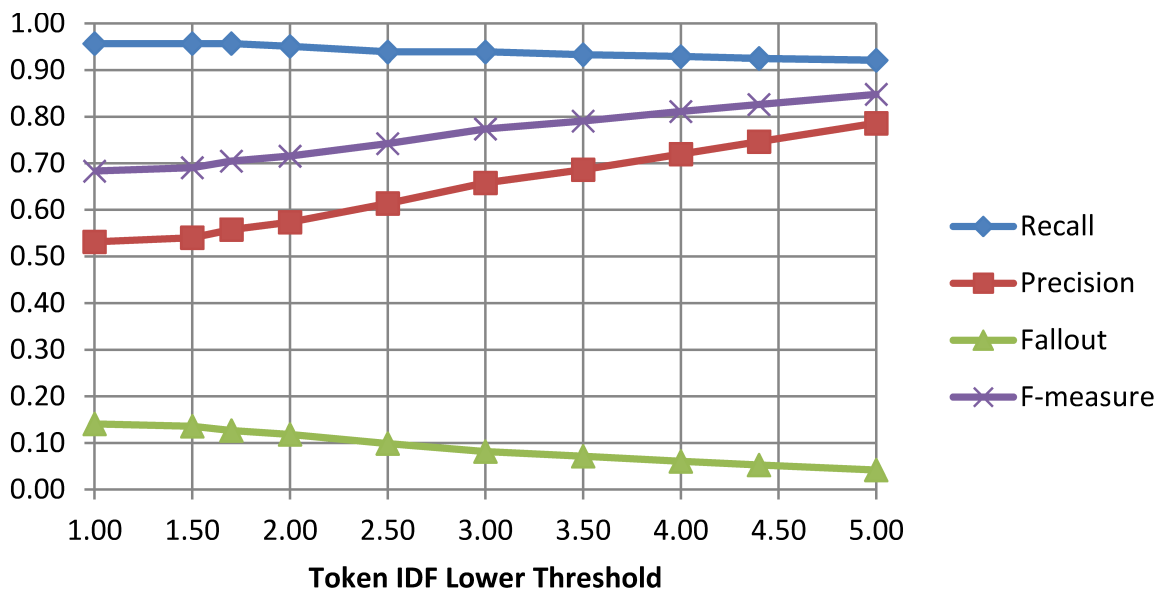


Figure 12.10: Results on record notes

The overall performance within each test set is well reflected in the first diagram 12.7 which presents the performance on the entire reference standard. The same tendency is repeated for each of the 4 graphs; recall and fallout decreases proportionally with increased *token IDF*-threshold, whereas the f-measure and precision have a corresponding increase.

A noteworthy fact is that the f-measure graph is squeezed by the recall and precision graphs, whereby this equation is true for every result: $\text{precision}(\text{idf}) \leq \text{f-measure}(\text{idf}) \leq \text{recall}(\text{idf})$, $\text{idf} \in [1, 5]$, which on a side note reminds about the *sandwich theorem*.

All the diagrams illustrates that the recall and fallout graphs starts at the top and decreases almost in parallel when the *token IDF*-threshold is increased, whereas the precision increases, which is the tendency for all results. Further, the f-measure graphs indicates that an increase in the token idf-threshold increases the overall performance; Less sensitive information is recognized (recall) but the classifications proves to be more precise.

Unrecognized textual sequences (without tags) are classified as sensitive. Disclosure of unknown textual sequences can be risky and thus have to be removed. However, this principle is turned contrary when the lower *token IDF*-threshold is locked to 5.0; the biggest idf-score a token can achieve is 4.66 this also implies that all information is regarded as insensitive on *token*-level. When every token is regarded as insensitive on *token*-level, the classification totally depends on *sentence idf*, dictionary lookups and regular expressions. If none of these components recognize the *token*, it is tagged as insensitive. The application performs best with maximum lower-*token idf*-threshold, in which our assumption is turned around and regards every token as insensitive. The number of *false positives* decreases, which results in falling fallout and increasing precision. Hence, the overall performance improves significantly when assuming that every textual sequence is insensitive, rather than sensitive; which is logical as the insensitive information constitutes the majority of clinical documents.

12.5 Modified Discharge summaries

The actual intention with this project is to develop an application with the purpose of de-identifying free text clinical documents. The clinical documents in our reference standard contain a lot of structured information; in fact the majority of the sensitive identifiers occur in the document headers. Hence, as a supplementary experiment, we also tested our best approach on pure narrative documents. As mentioned earlier 11.1.2, we modified the discharge summaries in the reference standard by removing the structured information at the top and bottom. The same setup were used as in the previous experiment 12.4.

Following results were achieved:

Recall	Precision	Fallout	F-measure
0.62	0.42	0.04	0.50

Table 12.6: Result on modified discharge summaries

The results indicates that the application performs significantly poorer on pure narrative text, by comparing these results with the results obtained in the third experiment 12.8. Even though the fallout has minimal differences, the overall performance in terms of f-measure has diminished from 0.75 to 0.5, due to significant decrease in both recall and precision.

13.1 Sources of Errors

The obtained results are influenced and weakened by several sources of errors. This section describes the most important ones.

13.1.1 Evaluation Method

The evaluation method used in the experiment has some weaknesses. Firstly, the evaluation disregards the loss of quality. Recognized information is only regarded as sensitive or insensitive and removed/retained accordingly, however the usability of the outcome is disregarded. The purpose of de-identification is to remove sensitive text in order to use the medical contents within legislative boundaries, however the performance measures don't consider the usefulness of the de-identified documents since each token is "equally weighted"; the performance is equally affected whether an *ICD10*-code[53] or a *connective (conjunction)* is removed, in which the former is code is much more crucial to retain. Hence, the overall binary classification performance measured by recall, precision, fallout and f-measure, does not necessarily denote the quality the de-identification.

Miss-classification of sensitive information units that are separated by several tokens, has more impact on the performance than they actually should. Take for example the phone number "74 12 93 10"; this phone number is split by four tokens, and if it isn't recognized it will be regarded as four missed identifiers rather than one. This is a drawback and creates a misleading picture of the performance.

Yet another weakness is related to the f-measure. The f-measure (f1-score) used in the experiment has equal weighting between recall and precision. As recall actually has higher priority than precision in a de-identification application, this should be regarded in the f-measure formula. The reason we chose to use this f-measure formula was to make our results comparable to the results presented in the state of the art chapter (4).

13.1.2 Reference Standard

Since we only annotated the most obvious indirectly identifying information, the reference standard might not be completely de-identified. For instance classifying gender as insensitive is a rather questionable choice. As previously mentioned, it is very difficult to ensure that the records are completely de-identified, since it is hard to determine what information a potential intruder holds (discussed in 2.10.2).

We made several simplifications during the annotation process which causes the results to be imprecise. An affective simplification is classifying ward name as sensitive; ward names are repetitive and a part of all headings. When these are unrecognized and regarded as sensitive the performance is noticeably affected. This is a partial reason for why the nursing notes and discharge summaries received significantly worse results than the record notes during the pattern matching, since almost every ward name was recognized in the latter type.

Manual annotation conducted by two students is an obvious drawback. The annotation should have been quality assured by an independent third party. Moreover, it is important to emphasize that the application also might be vulnerable to document variations since our reference standard is small, which also makes our obtained results highly unreliable.

13.1.3 Classification

Random classifications may produce variable results. However, our results indicates contrary behavior, having minimal impact by random classifications. For instance, the results produced by the pure pattern matching approach in part 2 (12.3), different Levenshtein-distances do not affect the results to any significant extent, which also supports the assertion that the random classification has minimal impact on the results.

The classification algorithms are too simplistic (as discussed in chapter 9). Multiple tags cause ambiguities, especially when the sensitivity labels are different, and resolving these in a robust manner requires an enriched rule base[40]. Too many *tokens* are looked up and matched in various sensitive dictionaries and tagged as sensitive, whereas the actual word is insensitive. Besides, the attempt of resolving ambiguities through POS-tags suffers the weakness of imprecise POS-tagging.

13.2 Performance

13.2.1 Statistical vs Pure Pattern Matching

The only difference between the first and second part is that the first used a statistical component (IDF), whereas the second used reference works. By swapping the statistical component with reference works, the overall performance was improved, in which the f-measure increased from 0.4 to 0.54. The primary reason for this increase is the recall boost in the second approach. The recall value increased from 0.7 to 0.89,

which indicates that the use of reference works recognizes more sensitive information. The statistical component failed to recognize repetitive sensitive information due to low idf-scores, whereas the pattern matching approach effectively looks up every token and recognizes almost every sensitive identifier, and has significantly better precision (0.29 vs 0.39). All in all, the statistical approach failed to recognize as much sensitive information as the pattern matching approach, which may indicate that de-identification based on pure *token*- and *sentence*-level *idf*-scores are insufficient, and needs supplementary statistical methods.

13.2.2 All Methods

There are clear differences between the results achieved throughout the three partial experiments. The last experiment, in which every component was included, achieved best the result with a f-measure of 0.75, distinct from 0.40 (*part 1*) and 0.54 (*part 2*). This was somewhat expected as most of the textual units (*tokens and sentences*) are investigated by several components, thus better grounds for decision-making.

By tuning various variables (*token* IDF-thresholds, *sentence* idf-threshold, Levenshtein Distance etc.) the performance of each component was maximized, and produced significantly better results. Even though the recall was better in the second part and decreased from 0.89 to 0.77, the precision boosted from 0.39 to 0.68, resulting in a solid increase in f-measure. The main strategy was to prioritize the tags assigned by regular expressions and references works, and further employ the thresholds provided by the statistical component.

Tokens without any tags are impossible to classify, thus classified as sensitive to be on the “safe” side. However, this choice significantly reduces the overall performance. Results from part 3 (12.4) emphasizes this impact when the lower IDF-threshold for *tokens* are increased. Every token is regarded as insensitive due to maximum lower IDF-threshold and produces the best results. The overall performance is increased while the recall is slightly decreased. This can be explained by the fact that the largest proportion of *tokens* in a note is insensitive, hence, most likely that an unknown *token* is insensitive.

13.2.3 Note Types

The de-identification performance on the nursing notes is quite poor compared to discharge summaries and record notes. One way to explain this is that nursing notes often contains inconsistent, error-prone, informal and oral language. Neamatullah et. al [1] states that clinical staffs frequently use technical terminology, non-standard abbreviations, ungrammatical statements, misspellings, and incorrect punctuation and capitalization in nursing progress notes. They also states that nursing notes appear to be significantly more challenging to de-identify than other forms of medical notes, such as discharge summaries. This statement is in accordance with our results, in which the discharge summaries obtained more satisfactory de-identification than the nursing notes.

13.3 Structured vs Unstructured

By comparing the results from the discharge summaries with the results from the modified discharge summaries, it is evident that the application performs better on discharge summaries with retained structured information. The structured information in the discharge summaries is easier recognized by regular expression, especially dates and national identification numbers, which proves the robustness of the regular expressions. The fallout value is close to equal in both types of discharge summaries, which indicates most of the miss-classification occurs in the free text part since most of the structured information is sensitive. This proves a solid de-identification performance in the pure structured part of the documents.

14.1 Conclusion

14.1.1 Question 1

With the obtained results, we are able to answer some of the questions presented in the introduction 1.2.2. The first question: *How well, in terms of recall, precision, fallout and f-measure, can we implement a de-identification application in the course of a semester, on the basis of rule-based and simple statistical methods?*

We have managed to develop a simple de-identification application, with following performance measures:

Recall	Precision	Fallout	F-measure
0.77	0.68	0.05	0.72

Table 14.1: Best results

It is important to emphasize that the results must be considered with regard to the fact that there has been made a number of simplifications.

14.1.2 Question 2

The second research question: *How will different combinations of the implemented algorithms and techniques affect the performance?*

This is mainly reflected through the results achieved in each part of the experiment.

As a summary, the results shows that a combination of regular expressions, reference works (sentence-level and word-level search) and statistical methods achieves better overall de-identification performance, considered f-measure, in contrast to component-wise performance.

Using idf-scores independently proves to be insufficient, and might need the support of more sophisticated statistical methods. However, the performance significantly increased when the idf-component classified every unknown textual sequence as insensitive. This indicates that the idf-component does not contribute to recognize any sensitive information in our "best-effort" approach, but only provides the advantage of accepting unknown textual sequences as insensitive.

Extending the de-identification application with additional methods gives better grounds for classification, at least in tag based approaches, but at the same time requires robust algorithms to resolve ambiguities and make final classifications.

The idf-component does not provide better precision without aggravating the recall, whereas reference works and regular expressions provides robust performance, and contributes to recognize most of the sensitive identifiers.

14.1.3 Question 3

The third research question reads as follows:

Can such system be realized in the Norwegian health care system, and replace/simplify manual annotation/de-identification?

There is no doubt that a semi-automatic de-identification application can be realized and used by the Norwegian health sector. The ground for this assertion is that two students have managed to implement an application that roughly recognizes 77 percent of the sensitive information in a corpus of 225 clinical documents. By the course of a semester (5 months), a de-identification application has been developed from scratch by the means of simple pattern matching and statistical methods.

There is need for manual adjustments to the de-identified output since we have disregarded some types of sensitive information, however, such adjustments can be quickly fixed; when a big corpus of clinical notes are de-identified by this application, manual adjustments can be performed much faster in contrast to manually de-identify the entire corpus. Even though our application only recognizes a limited range of sensitive identifiers, it is highly modifiable and can be extended by more reference works, regular expressions and even entire components can be added to the pipeline, for instance a machine learning component.

A fully automatic de-identification application, on the other hand, is obviously most practical. However, even though an application achieves perfect f-measure pursuant to a gold standard, it may have certain limitations when applied on new documents. After working on real Norwegian EHR-notes, our experience suggests that quality assurance of the de-identified output always will be needed, irrespective of the recall, precision, fallout and f-measure values. Certain clinical documents reveal a lot of sensitive information (indirectly) which cannot be recognized unless a computer obtains *cognitive skills*, and interprets human language. Besides, even if the documents can be interpreted, it can still be hard to decide whether or not a textual sequence constitutes indirectly identifying information. Since patient data is comprised by strict and rigid

legislation, manual quality assurance and adjustments seem to be needed, at least on simplified applications as the one we have developed.

Pursuant to the initial question, automatic de-identification applications are fully realizable, but we do not believe that these can replace manual de-identification. We also believe that a robust semi-automatic de-identification application can simplify the de-identification process, and prove to be significantly time- and cost effective within the health sector.

14.2 Further Work

The de-identification application adapted to Norwegian free text notes, developed through this project, can be considered as a preliminary study for future attempts. Since no previous studies deals with de-identification of Norwegian clinical notes, we do not have any directly related work to build our application upon; hence our study only focuses on experimenting with different techniques used for other languages. There are several elements that can contribute to enhance this application.

14.2.1 Rule Base

First of all, the rule base needs a considerable enrichment in order to improve the classifications and to resolve ambiguities. Enriched and adapted rules can contribute greatly to increase the performance, which has been shown by Gupta et. al [40]. Our pattern matching employs a minimalistic rule base and an even poorer classification algorithm. The application has to be extended by more reference works, linguistic rules, medical classification systems and a robust classifier.

14.2.2 Machine Learning

Machine learning methods could have been a part of this experimental approach. As we have described in the state of the art chapter 4, supervised machine learning methods have proved to be very effective for de-identification purpose. A large corpus of annotated text is required to train the machine learning algorithms, which does not exist in Norwegian. This is a huge drawback as several machine learning algorithms can be employed off the shelf, and could serve as a cheap classification feature in our approach. An annotated corpus should be prepared in order to provide new and valuable opportunities for NLP-researchers. However, it is important to emphasize that this requires significant work by domain experts.

14.2.3 Preprocessor

The file-cleaning process was not intended as a part of the preprocessor, but we early realized the need of this component during the first preprocessor runs on the realistic

clinical notes. Unfortunately, the files are not satisfactory cleaned as there are still unhandled character-types causing somewhat incorrect splitting. Apache Lucene provides word-splitters which could have been a better alternative; in fact the entire Document-class offered by Lucene[50] could have been used, as it offers a lot of tools which could prove beneficial at later stages of the system. Hence, a replacement of the document class could prove beneficial.

14.2.4 POS-tagger

The POS-tagger does not perform nearly as good as expected. The tagger was trained on a newspaper corpus. There can be several reasons for this behavior whereby one can be explained by domain-difference. Clinical texts contains sentences that are grammatical incomplete in contrast to newspaper-sentences, in addition to a significantly different vocabulary. An experiment performed on this POS-tagger by Brox et al.[51] indicated that relevant training data from the clinical domain gives better results for the tagging task in this domain than training the tagger on a corpus from a more general domain. A better alternative might have been the *Oslo-Bergen tagger* which is a pure rule based tagger and has been continuously improved since the late 90's.

14.2.5 Compound words

The lack of a tool recognizing Norwegian compound words is another drawback. Compounds are extremely productive in Norwegian; 10.4% of all words in running text are compound, and any text sample will contain a great number of compounds, which is true for even small samples[73]. The total amount of combinations is huge, hence next to impossible to gather these inside a dictionary. Compounds revealing sensitive information may be challenging to recognize. Here are a few examples:

Indiskfødt (Indian-born)

Drammensgutt (A boy from the town Drammen)

Mattelærer (Maths teacher)

Knespesialist (Knee specialist)

These are examples of sensitive words that will not be recognized by any of our components.

Bibliography

- [1] Ishna Neamatullah, Margaret M Douglass, H Lehman Li-wei, Andrew Reisner, Mauricio Villarroel, William J Long, Peter Szolovits, George B Moody, Roger G Mark, and Gari D Clifford. Automated de-identification of free-text medical records. *BMC medical informatics and decision making*, 8(1):32, 2008.
- [2] Helse og omsorgsdepartementet. Veileder til lov 20. juni 2008 nr. 44 om medisinsk og helsefaglig forskning (helseforskningsloven). URL <http://www.regjeringen.no/upload/HOD/HRA/Veileder%20til%20helseforskningsloven.pdf>.
- [3] Robert A Greenes. *Clinical decision support: the road ahead*. Academic Press, 2011.
- [4] Nasjonalt kunnskapssenter for helsetjenesten. Forskningsprosjekt: Evicare - wp 6 indikatorekstraksjon. URL https://helseforskning.etikkom.no/ikbViewer/page/prosjekterirek/prosjektregister/prosjekt?p_document_id=123256&p_parent_id=125959&_ikbLanguageCode=n.
- [5] DA Dorr, WF Phillips, S Phansalkar, SA Sims, and JF Hurdle. Assessing the difficulty and time cost of de-identification in clinical narratives. *Methods of Information in Medicine-Methodik der Information in der Medizin*, 45(3):246–252, 2006.
- [6] M Douglass, GD Clifford, A Reisner, GB Moody, and RG Mark. Computer-assisted de-identification of free text in the mimic ii database. In *Computers in Cardiology, 2004*, pages 341–344. IEEE, 2004.
- [7] The health research act, 2008-06-20 no. 44. URL <http://www.ub.uio.no/ujur/ulovdata/lov-20080620-044-eng.pdf>. Last accessed 2013.06.09.
- [8] University of Oslo The Faculty of Law Library. Health personnel act, . URL <http://www.lovdatab.no/all/t1-19990702-064-008.html#46>.
- [9] University of Oslo The Faculty of Law Library. Personal data act, . URL <http://www.lovdatab.no/all/n1-20000414-031.html>.

- [10] Personal health data filing system act, act of 18 May 2001 no. 24. URL <http://www.regjeringen.no/en/dep/hod/Subjects/the-department-of-public-health/Act-of-18-May-2001-No-24-on-Personal-Health-Data-Filing-Systems-and-the-Processing.html?id=224129>. Last accessed 2013.06.09.
- [11] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.
- [12] Justisdepartementet og Det juridiske fakultet ved Universitetet i Oslo. Forskrift om innsamling og behandling av helseopplysninger i reseptbasert legemiddelregister (reseptregisteret). URL <http://www.lovdata.no/for/sf/ho/to-20031017-1246-001.html#1-7>.
- [13] Pseudonyme helseregistre. URL <http://www.regjeringen.no/nb/dep/fad/dok/nouer/2009/nou-2009-1/27/2.html?id=542369>. Last accessed 2013.06.09.
- [14] Sumithra Velupillai. *Shades of Certainty: Annotation and Classification of Swedish Medical Records*. PhD thesis, Stockholm, 2012.
- [15] Khaled El Emam, Sam Jabbouri, Scott Sams, Youenn Drouet, and Michael Power. Evaluating common de-identification heuristics for personal health information. *Journal of Medical Internet Research*, 8(4), 2006.
- [16] The U.S Government Printing Office (GPO). Gpo us: 45 c.f.r. § 164 security and privacy. URL http://www.access.gpo.gov/nara/cfr/waisidx_08/45cfr164_08.html.
- [17] Latanya Sweeney. Patient identifiability in pharmaceutical marketing data. *Harvard University, Cambridge, MA, WP-1015*, 2011.
- [18] Oslo Universitetsykehus. Anonymisering og avidentifisering av helseopplysninger, . URL <http://www.oslo-universitetssykehus.no/omoss/personvern/Sider/avidentifisering.aspx>. Last accessed 2013.05.27.
- [19] Oslo Universitetsykehus. Hva er personopplysninger?, . URL <http://www.oslo-universitetssykehus.no/omoss/personvern/Sider/hva-er-personopplysninger.aspx>. Last accessed 2013.05.27.
- [20] Latanya Sweeney. Simple demographics often identify people uniquely. *Health (San Francisco)*, pages 1–34, 2000.
- [21] R Mahesh and T Meyyappan. Anonymization technique through record elimination to preserve privacy of published data. In *Pattern Recognition, Informatics and Medical Engineering (PRIME), 2013 International Conference on*, pages 328–332. IEEE, 2013.
- [22] Ryan Williams and Manuel Blum. K-anonymity, reu summer 2007. URL www.cs.cmu.edu/~jblocki/Slides/K-Anonymity.pdf.

- [23] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.
- [24] Kent A Spackman, Keith E Campbell, RA CÃ, et al. Snomed rt: a reference terminology for health care. In *Proceedings of the AMIA annual fall symposium*, page 640. American Medical Informatics Association, 1997.
- [25] University of Oslo The Faculty of Law Library. Regularions on health records, . URL <http://www.lovddata.no/cgi-wift/ldles?doc=/sf/sf/sf-20001221-1385.html#4>.
- [26] University of Oslo The Faculty of Law Library. Health register act, . URL <http://www.lovddata.no/all/nl-20010518-024.html>.
- [27] Gunnar Flikke Ann Rudinow Sætnan Gro Snortheimsmoen Bergfjord Rune Fløis-bonn Michael Tetzschner Lee A. Bygrave Gisle Hannemyr Sandra Qian Xiao Inge Carlén Mari Bø Haugstad Henriette Sinding Aasen Kjersti Fjørtoft Grete Myhre Kjellbjørg Lunde, Hans Antonsen. Individ og integritet, personvern i det digitale samfunnet. URL <http://www.regjeringen.no/nb/dep/fad/dok/nouer/2009/nou-2009-1.html?id=542049>.
- [28] Norwegian Data Inspectorate. Vedtak om pålegg - uautorisert uthenting av helseopplysninger igjennom leverandørs fjerntilgang., . URL http://www.datatilsynet.no/Global/05_vedtak_saker/2012/Curato.pdf.
- [29] Norwegian Data Inspectorate. Norkar - varsel om vedtak, . URL http://www.datatilsynet.no/Global/05_vedtak_saker/2012/07-00382-12%20Varsel%20om%20overtredelsesgebyr_NORKAR.pdf.
- [30] G Edward Barton, Robert C Berwick, and Eric S Ristad. *Computational complexity and natural language*. MIT press, 1987.
- [31] Jung-Hsien Chiang, Jou-Wei Lin, and Chen-Wei Yang. Automated evaluation of electronic discharge notes to assess quality of care for cardiovascular diseases using medical language extraction and encoding system (medlee). *Journal of the American Medical Informatics Association*, 17(3):245–252, 2010.
- [32] Medical subject headings. URL <http://www.nlm.nih.gov/mesh/>. Last accessed 2013.06.10.
- [33] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270, 2004.
- [34] Louise Deléger, Cyril Grouin, and Pierre Zweigenbaum. Extracting medication information from french clinical texts. *Stud Health Technol Inform*, 160(Pt 2): 949–53, 2010.
- [35] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.

- [36] Muneo Kushima, Kenji Araki, Muneou Suzuki, Sanae Araki, and Terue Nikama. Text data mining of the electronic medical record of the chronic hepatitis patient. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, 2012.
- [37] Dana Nau. Part-of-speech tagging lecture, 29 april 2010. URL <http://www.cs.umd.edu/~nau/cmsc421/part-of-speech-tagging.pdf>. Last accessed 2013.05.27.
- [38] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the workshop on Speech and Natural Language*, pages 112–116. Association for Computational Linguistics, 1992.
- [39] Stephane Meystre, F Friedlin, Brett South, Shuying Shen, and Matthew Samore. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC medical research methodology*, 10(1):70, 2010.
- [40] Dilip Gupta, Melissa Saul, and John Gilbertson. Evaluation of a deidentification (de-id) software engine to share pathology reports and clinical documents for research. *American Journal of Clinical Pathology*, 121(2):176–186, 2004.
- [41] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.
- [42] the de-id application. URL <http://www.physionet.org/physiotools/deid/>.
- [43] Gari D Clifford, Daniel J Scott, and Mauricio Villarroel. User guide and documentation for the mimic ii database. *MIMIC-II database version, 2*, 2009.
- [44] Kostas Pantazos, Soren Lauesen, and Soren Lippert. De-identifying an ehr database-anonymity, correctness and readability of the medical record. *Proceedings of MIE2011*, 2011.
- [45] Hercules Dalianis and Henrik Boström. Releasing a swedish clinical corpus after removing all words—de-identification experiments with conditional random fields and random forests. In *the Proceedings of the Third Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM 2012) held in conjunction with LREC*, pages 45–48, 2012.
- [46] Özlem Uzuner, Tawanda C Sibanda, Yuan Luo, and Peter Szolovits. A de-identifier for medical discharge summaries. *Artificial intelligence in medicine*, 42(1):13, 2008.
- [47] Informatics for Integrating Biology and the Bedside. 2006 deidentification and smoking challenge. URL <https://i2b2.org/NLP/DataSets/>.
- [48] Oscar Ferrández, Brett R South, Shuying Shen, F Jeffrey Friedlin, Matthew H Samore, and Stéphane M Meystre. Bob, a best-of-breed automated text de-identification system for vha clinical documents. *Journal of the American Medical Informatics Association*, 20(1):77–83, 2013.

- [49] Apache Software Foundation. Apache clinical text analysis and knowledge extraction system, . URL <http://ctakes.apache.org/>.
- [50] Apache Software Foundation. Apache lucene, . URL <http://lucene.apache.org/>.
- [51] Thomas Brox Røst, Ola Huseth, Øystein Nytrø, and Anders Grimsmo. Lessons from developing an annotated corpus of patient histories. *Journal of Computing Science and Engineering*, 2(2):162–179, 2008.
- [52] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [53] International classification of diseases (icd). URL <http://www.who.int/classifications/icd/en/>. Last accessed 2013.06.10.
- [54] UoH sektor/Norges teknisk-naturvitenskape/Det humanistiske fakultet HF. Norsk komputasjonelt leksikon (norkompleks). URL <http://www.forskningsradet.no/servlet/Satellite?c=Prosjekt&cid=1193731511032&pagename=ForskningsradetNorsk/Hovedsidemal&p=1181730334233>.
- [55] The National Library of Norway. Wordnets for norwegian bokmål and nynorsk. URL <http://www.nb.no/English/Collection-and-Services/Spraakbanken/Projects/Wordnets-for-Norwegian-Bokmaal-and-Nynorsk>.
- [56] Felleskatalogen AS. URL <http://www.felleskatalogen.no/medisin/>.
- [57] Anatomical therapeutic chemical (atc). URL http://www.whocc.no/atc/structure_and_principles/. Last accessed 2013.06.01.
- [58] Den internasjonale klassifikasjonen for primærhelsetjenesten (icpc-2). URL <http://www.helsedirektoratet.no/finansiering/medisinsk-koding-og-kodeverk/radiologikodeverket-ncrp/Sider/radiologikodeverket-ncrp.aspx>. Last accessed 2013.06.10.
- [59] Norsk klassifikasjon av medisinske prosedyrer (ncmp). URL http://www.kith.no/templates/kith_WebPage___1138.aspx. Last accessed 2013.06.10.
- [60] Norwegian classification of radiological procedures (ncrp). URL <http://www.helsedirektoratet.no/finansiering/medisinsk-koding-og-kodeverk/radiologikodeverket-ncrp/Sider/radiologikodeverket-ncrp.aspx>. Last accessed 2013.06.10.
- [61] Klassifikasjon av kirurgiske inngrep (ncsp). URL http://www.kith.no/templates/kith_WebPage___1160.aspx. Last accessed 2013.06.10.
- [62] Naoaki Okazaki and Jun'ichi Tsujii. Simple and efficient algorithm for approximate dictionary matching. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 851–859, Beijing, China, August 2010. URL <http://www.aclweb.org/anthology/C10-1096>.

- [63] William W Cohen, Pradeep Ravikumar, Stephen E Fienberg, et al. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*, volume 47, 2003.
- [64] Patrick Ruch, Robert Baud, and Antoine Geissbühler. Using lexical disambiguation and named-entity recognition to improve spelling correction in the electronic patient record. *Artificial intelligence in medicine*, 29(1):169–184, 2003.
- [65] Andres Marzal and Enrique Vidal. Computation of normalized edit distance and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(9):926–932, 1993.
- [66] James Gardner and Li Xiong. Hide: an integrated system for health information de-identification. In *Computer-Based Medical Systems, 2008. CBMS'08. 21st IEEE International Symposium on*, pages 254–259. IEEE, 2008.
- [67] Ben Wellner, Matt Huyck, Scott Mardis, John Aberdeen, Alex Morgan, Leonid Peshkin, Alex Yeh, Janet Hitzeman, and Lynette Hirschman. Rapidly retargetable approaches to de-identification in medical records. *Journal of the American Medical Informatics Association*, 14(5):564–573, 2007.
- [68] A Argraw, Anette Hulth, and B Megyesi. General-purpose text categorization applied to the medical domain.
- [69] James Mayfield and Paul McNamee. Single n-gram stemming. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 415–416. ACM, 2003.
- [70] Nikolaos Nanas, Victoria Uren, Anne De Roeck, and J Domingue. A comparative study of term weighting methods for information filtering, 2003.
- [71] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520, 2004.
- [72] Lois E. Brenneman. Guidelines for writing soap notes, history and physicals. URL <http://www.ravenwood-pa.com/npceu/content/courses/all/doc/guide-shp.pdf>.
- [73] Janne Bondi Johannessen and Helge Hauglin. An automatic analysis of norwegian compounds. In *16th Scandinavian Conference of Linguistics*. Citeseer, 1996.

Part V

Appendix

A.1 Regulation on health records, §8

Health records shall include following information to the extent they are relevant and necessary:

- a) Tilstrekkelige opplysninger til å kunne identifisere og kontakte pasienten, blant annet pasientens navn, adresse, bostedskommune, fødselsnummer, telefonnummer, sivilstand og yrke.
- b) Opplysninger om hvem som er pasientens nærmeste pårørende, jf. pasientrettighetsloven § 1-3 bokstav b og lov om psykisk helsevern § 1-3, og hvordan vedkommende om nødvendig kan kontaktes.
- c) Dersom pasienten ikke har samtykkekompetanse, skal det nedtegnes hvem som samtykker på vegne av pasienten, jf. pasientrettighetsloven kapittel 4.
- d) Når og hvordan helsehjelp er gitt, for eksempel i forbindelse med ordinær konsultasjon, telefonkontakt, sykebesøk eller opphold i helseinstitusjon. Dato for innleggelse og utskriving.
- e) Bakgrunnen for helsehjelpen, opplysninger om pasientens sykehistorie, og opplysninger om pågående behandling. Beskrivelse av pasientens tilstand, herunder status ved innleggelse og utskriving.
- f) Foreløpig diagnose, observasjoner, funn, undersøkelser, diagnose, behandling, pleie og annen oppfølging som settes i verk og resultatet av dette. Plan eller avtale om videre oppfølging.
- g) Opplysninger som nevnt i § 6 fjerde ledd.
- h) Overveielser som har ledet til tiltak som fraviker fra gjeldende retningslinjer.
- i) Om det er gitt råd og informasjon til pasient og pårørende, og hovedinnholdet i dette, jf. pasientrettighetsloven § 3-2. Pasientens eventuelle reservasjon mot å motta informasjon.

- j) Om pasienten har samtykket til eller motsatt seg nærmere angitt helsehjelp. Pasientens alvorlige overbevisning eller vegring mot helsehjelp, jf. pasientrettighetsloven § 4-9. Pasientens samtykke eller reservasjon vedrørende informasjonsbehandling. Pasientens øvrige reservasjoner, krav eller forutsetninger.
- k) Om det er gjort gjeldende rettigheter som innsyn i journal og krav om retting og sletting, utfallet av dette, ved avslag at pasienten er gjort kjent med klageadgangen, og eventuell klage i slik sak.
- l) Utveksling av informasjon med annet helsepersonell, for eksempel henvisninger, epikriser, innleggelsesbegjæringer, resultater fra rekvirerte undersøkelser, attestkopier m.m.
- m) Pasientens faste lege. Det helsepersonell som har begjært innleggelse eller har henvist pasienten.
- n) Individuell plan etter spesialisthelsetjenesteloven § 2-5, psykisk helsevernloven § 4-1 eller kommunehelsetjenesteloven § 6-2a.
- o) Sykmeldinger og attester.
- p) Uttalelser om pasienten, for eksempel sakkyndige uttalelser.
- q) Om det er gitt opplysninger til politi, barneverntjenesten, helse- og omsorgstjenesten, sosialtjenesten mv., og om samtykke er innhentet fra pasienten eller den som har kompetanse til å avgi samtykke i saken. Det skal angis hvilke opplysninger som er gitt.
- r) Tvangsinnleggelse, annen bruk av tvang, det faktiske og rettslige grunnlaget for slik tvang og eventuelle kontrollkommisjonsvedtak, jf. lov om psykisk helsevern.
- s) En faglig begrunnelse¹ i de tilfellene legen har reservert seg mot apotekets generiske bytterett.²
- t) Opplysninger om hvorvidt pasient med psykisk sykdom, rusmiddelavhengighet eller alvorlig somatisk sykdom eller skade har mindreårige barn.
- u) Opplysninger om foreldrene som har konsekvens for barnets behandlingssituasjon, herunder nødvendige opplysninger om foreldrenes helsetilstand.

B.1 Regular Expressions and IDF

Token IDF	Sentence IDF	Recall	Precision	Fallout	F-measure
0.50	2.00	0.85	0.14	0.65	0.25
1.00	2.00	0.84	0.15	0.59	0.26
1.50	2.00	0.83	0.19	0.46	0.30
2.00	2.00	0.78	0.24	0.31	0.37
2.50	2.00	0.68	0.28	0.22	0.40
0.50	4.00	0.66	0.14	0.54	0.22
1.00	4.00	0.66	0.15	0.49	0.24
1.50	4.00	0.66	0.18	0.39	0.28
2.00	4.00	0.62	0.23	0.23	0.33
2.50	4.00	0.55	0.27	0.19	0.36

Table B.1: The statistical component's results on the entire reference standard

Token IDF	Sentence IDF	Recall	Precision	Fallout	F-measure
0.50	2.00	0.95	0.15	0.70	0.26
1.00	2.00	0.94	0.16	0.65	0.27
1.50	2.00	0.92	0.18	0.53	0.31
2.00	2.00	0.87	0.23	0.37	0.37
2.50	2.00	0.77	0.27	0.27	0.40
0.50	4.00	0.74	0.14	0.58	0.24
1.00	4.00	0.74	0.15	0.54	0.25
1.50	4.00	0.73	0.18	0.44	0.28
2.00	4.00	0.70	0.22	0.31	0.34
2.50	4.00	0.63	0.26	0.24	0.36

Table B.2: The statistical component's results on discharge summaries

Token IDF	Sentence IDF	Recall	Precision	Fallout	F-measure
0.50	2.00	0.63	0.12	0.48	0.21
1.00	2.00	0.63	0.14	0.40	0.23
1.50	2.00	0.61	0.19	0.28	0.29
2.00	2.00	0.56	0.26	0.16	0.36
2.50	2.00	0.45	0.33	0.10	0.38
0.50	4.00	0.43	0.11	0.37	0.17
1.00	4.00	0.43	0.13	0.31	0.20
1.50	4.00	0.43	0.17	0.22	0.25
2.00	4.00	0.40	0.24	0.13	0.30
2.50	4.00	0.31	0.28	0.09	0.29

Table B.3: The statistical component’s results on nursing notes

Token IDF	Sentence IDF	Recall	Precision	Fallout	F-measure
0.50	2.00	0.71	0.15	0.68	0.25
1.00	2.00	0.71	0.16	0.63	0.26
1.50	2.00	0.70	0.20	0.47	0.31
2.00	2.00	0.67	0.26	0.33	0.37
2.50	2.00	0.60	0.33	0.20	0.43
0.50	3.00	0.70	0.15	0.67	0.25
1.00	3.00	0.70	0.16	0.62	0.26
1.50	3.00	0.70	0.20	0.47	0.31
2.00	3.00	0.67	0.26	0.33	0.37
2.50	3.00	0.60	0.33	0.20	0.43
0.50	4.00	0.63	0.14	0.64	0.23
1.00	4.00	0.63	0.15	0.59	0.24
1.50	4.00	0.63	0.19	0.45	0.29
2.00	4.00	0.60	0.24	0.31	0.35
2.50	4.00	0.54	0.32	0.19	0.40

Table B.4: The statistical component’s results on record notes

B.2 Pattern Matching and IDF

I:L	I:U	I:S	P	SP	Recall	Precision	Fallout	F-measure
1.00	4.00	0.80	Exact	0.80	0.87	0.41	0.16	0.55
1.00	4.00	3.00	LD3	0.80	0.75	0.41	0.14	0.53
1.50	4.00	3.00	LD3	0.80	0.75	0.44	0.12	0.55
1.50	4.00	2.00	LD3	0.80	0.78	0.43	0.13	0.56
1.50	4.00	1.70	LD3	0.80	0.83	0.43	0.14	0.57
1.50	5.00	1.70	LD3	0.80	0.83	0.44	0.14	0.57
1.50	5.00	1.70	LD3	0.60	0.84	0.41	0.16	0.55
1.50	5.00	1.70	LD3	1.00	0.83	0.44	0.14	0.58
1.50	5.00	1.60	LD3	1.00	0.85	0.43	0.14	0.57
1.00	5.00	1.70	LD3	1.00	0.84	0.41	0.15	0.55
1.70	5.00	1.70	LD3	1.00	0.83	0.45	0.13	0.58
2.00	5.00	1.70	LD3	1.00	0.83	0.49	0.11	0.62
2.50	5.00	1.70	LD3	1.00	0.82	0.52	0.10	0.64
3.00	5.00	1.70	LD3	1.00	0.81	0.55	0.08	0.66
5.00	5.00	1.70	LD3	1.00	0.77	0.68	0.05	0.72
4.00	5.00	1.70	LD3	1.00	0.79	0.61	0.06	0.69
3.50	5.00	1.70	LD3	1.00	0.81	0.58	0.08	0.67
4.40	5.00	1.70	LD3	1.00	0.78	0.65	0.05	0.71

Table B.5: Results on all notes

I:L	I:U	I:S	P	SP	Recall	Precision	Fallout	F-measure
1.00	4.00	0.80	Exact	0.80	0.87	0.38	0.18	0.53
1.00	4.00	3.00	LD3	0.80	0.83	0.41	0.15	0.55
1.50	4.00	3.00	LD3	0.80	0.83	0.44	0.13	0.58
1.50	4.00	2.00	LD3	0.80	0.86	0.44	0.14	0.58
1.50	4.00	1.70	LD3	0.80	0.86	0.42	0.15	0.56
1.50	5.00	1.70	LD3	0.80	0.86	0.42	0.15	0.57
1.50	5.00	1.70	LD3	0.60	0.87	0.40	0.17	0.55
1.50	5.00	1.70	LD3	1.00	0.86	0.43	0.15	0.57
1.50	5.00	1.60	LD3	1.00	0.86	0.41	0.16	0.56
1.00	5.00	1.70	LD3	1.00	0.87	0.40	0.17	0.55
1.70	5.00	1.70	LD3	1.00	0.85	0.43	0.14	0.57
2.00	5.00	1.70	LD3	1.00	0.85	0.49	0.11	0.62
2.50	5.00	1.70	LD3	1.00	0.84	0.51	0.10	0.64
3.00	5.00	1.70	LD3	1.00	0.84	0.54	0.09	0.66
5.00	5.00	1.70	LD3	1.00	0.78	0.68	0.05	0.72
4.00	5.00	1.70	LD3	1.00	0.81	0.60	0.07	0.69
3.50	5.00	1.70	LD3	1.00	0.83	0.56	0.08	0.67
4.40	5.00	1.70	LD3	1.00	0.78	0.65	0.05	0.71

Table B.6: Results on discharge summaries

I:L	I:U	I:S	P	SP	Recall	Precision	Fallout	F-measure
1.00	4.00	0.80	Exact	0.80	0.83	0.42	0.12	0.56
1.00	4.00	3.00	LD3	0.80	0.52	0.36	0.10	0.42
1.50	4.00	3.00	LD3	0.80	0.52	0.39	0.09	0.44
1.50	4.00	2.00	LD3	0.80	0.60	0.38	0.10	0.46
1.50	4.00	1.70	LD3	0.80	0.67	0.40	0.11	0.50
1.50	5.00	1.70	LD3	0.80	0.67	0.40	0.10	0.50
1.50	5.00	1.70	LD3	0.60	0.67	0.37	0.12	0.48
1.50	5.00	1.70	LD3	1.00	0.67	0.41	0.10	0.51
1.50	5.00	1.60	LD3	1.00	0.72	0.42	0.10	0.53
1.00	5.00	1.70	LD3	1.00	0.67	0.37	0.12	0.48
1.70	5.00	1.70	LD3	1.00	0.67	0.42	0.10	0.52
2.00	5.00	1.70	LD3	1.00	0.66	0.44	0.09	0.53
2.50	5.00	1.70	LD3	1.00	0.65	0.48	0.07	0.55
3.00	5.00	1.70	LD3	1.00	0.64	0.52	0.06	0.57
5.00	5.00	1.70	LD3	1.00	0.63	0.60	0.04	0.61
4.00	5.00	1.70	LD3	1.00	0.63	0.56	0.05	0.60
3.50	5.00	1.70	LD3	1.00	0.63	0.54	0.06	0.58
4.40	5.00	1.70	LD3	1.00	0.63	0.58	0.05	0.60

Table B.7: Results on nursing notes

I:L	I:U	I:S	P	SP	Recall	Precision	Fallout	F-measure
1.00	4.00	0.80	Exact	0.80	0.97	0.54	0.14	0.70
1.00	4.00	3.00	LD3	0.80	0.69	0.45	0.14	0.54
1.50	4.00	3.00	LD3	0.80	0.69	0.46	0.14	0.55
1.50	4.00	2.00	LD3	0.80	0.71	0.46	0.14	0.56
1.50	4.00	1.70	LD3	0.80	0.96	0.53	0.14	0.68
1.50	5.00	1.70	LD3	0.80	0.96	0.53	0.14	0.69
1.50	5.00	1.70	LD3	0.60	0.96	0.47	0.18	0.63
1.50	5.00	1.70	LD3	1.00	0.96	0.54	0.14	0.69
1.50	5.00	1.60	LD3	1.00	0.97	0.54	0.14	0.70
1.00	5.00	1.70	LD3	1.00	0.96	0.53	0.14	0.68
1.70	5.00	1.70	LD3	1.00	0.96	0.56	0.13	0.70
2.00	5.00	1.70	LD3	1.00	0.95	0.57	0.12	0.72
2.50	5.00	1.70	LD3	1.00	0.94	0.61	0.10	0.74
3.00	5.00	1.70	LD3	1.00	0.94	0.66	0.08	0.77
5.00	5.00	1.70	LD3	1.00	0.92	0.79	0.04	0.85
4.00	5.00	1.70	LD3	1.00	0.93	0.72	0.06	0.81
3.50	5.00	1.70	LD3	1.00	0.93	0.69	0.07	0.79
4.40	5.00	1.70	LD3	1.00	0.92	0.75	0.05	0.83

Table B.8: Results on record notes



Unicode Whitespace Table

Code	Name of the Character
U+0020	SPACE
U+00A0	NO-BREAK SPACE
U+1680	OGHAM SPACE MARK
U+180E	MONGOLIAN VOWEL SEPARATOR
U+2000	EN QUAD
U+2001	EM QUAD
U+2002	EN SPACE
U+2003	EM SPACE
U+2004	THREE-PER-EM SPACE
U+2005	FOUR-PER-EM SPACE
U+2006	SIX-PER-EM SPACE
U+2007	FIGURE SPACE
U+2008	PUNCTUATION SPACE
U+2009	THIN SPACE
U+200A	HAIR SPACE
U+200B	ZERO WIDTH SPACE
U+202F	NARROW NO-BREAK SPACE
U+205F	MEDIUM MATHEMATICAL SPACE
U+3000	IDEOGRAPHIC SPACE
U+FEFF	ZERO WIDTH NO-BREAK SPACE

Table C.1: Unicode space characters



Input-Output Example

Dato: <DATE>

Godkjent av: <GEOGRAPHICRELATION> <NATIONALITY>

Dokumentnummer: <TLFNUMBER>

INNKOMSTJOURNAL

Innleggende lege: <GEOGRAPHICRELATION>

Pasient: <NAME> <GEOGRAPHICRELATION>

Alder: 53 <GEOGRAPHICRELATION>

Diagnose: Brystsmerter

Tidligere <ORGANIZATIONNAME>

Pasienten hadde hjerteinfarkt i <YEAR>

Aktuelt:

Smerter og nummenhet som startet i venstre hånd og spredte seg oppover og mot hjertet ca kl 14 i dag. Kvalm, ingen effekt av nitro. Ikke økt dyspnoe, ikke vært kaldsvett. Får nitro og morfin i ambulansen, og innkommer smertefri

Medikamenter:

Fragmin 2500 IE x 1

Panodil 500 mg x 4

Nexium 40 mg x 1

Figure D.1: The output produced by our de-identification application on the fictitious clinical note presented in the introduction 1.1