UNIVERSITAT
ROVIRA i VIRGILI

Utility-Preserving Anonymization of Textual Documents

Author:
Fadi HASSAN



DOCTORAL THESIS
2021

Fadi HASSAN

Utility-Preserving Anonymization of Textual Documents

DOCTORAL THESIS

Supervisors:
David SÁNCHEZ
Josep DOMINGO-FERRER

Department of Computer Engineering and Mathematics



UNIVERSITAT ROVIRA i VIRGILI

April, 2021

UNIVERSITAT ROVIRA i VIRGILI

FAIG CONSTAR que aquest treball, titulat “Utility-Preserving Anonymization of Textual Documents”, que presenta Fadi Hassan per a l’obtenció del títol de Doctor, ha estat realitzat sota la meva direcció al Departament d’Enginyeria Informàtica i Matemàtiques d’aquesta universitat.

HAGO CONSTAR que el presente trabajo, titulado “Utility-Preserving Anonymization of Textual Documents”, que presenta Fadi Hassan para la obtención del título de Doctor, ha sido realizado bajo mi dirección en el Departamento de Ingeniería Informática y Matemáticas de esta universidad.

I STATE that the present study, entitled “Utility-Preserving Anonymization of Textual Documents”, presented by Fadi Hassan for the award of the degree of Doctor, has been carried out under my supervision at the Department of Computer Engineering and Mathematics of this university.

Tarragona, 20/05/2021

El/s director/s de la tesi doctoral
El/los director/es de la tesis doctoral
Doctoral Thesis Supervisor/s

David Sánchez Ruenes - DNI 47763566H (SIG)
Firmado digitalmente por David Sánchez Ruenes - DNI 47763566H (SIG)
Fecha: 2021.04.20 16:13:35 +02'00'

DOMINGO FERRER JOSEP - 33890313C
Digitally signed by DOMINGO FERRER JOSEP - 33890313C
Date: 2021.04.20 16:48:47 +02'00'

David Sánchez Ruenes

Josep Domingo Ferrer

Abstract

Every day, people post a significant amount of data on the Internet, such as tweets, reviews, photos, and videos. Organizations collecting these types of data use them to extract information in order to improve their services or for commercial purposes. Yet, if the collected data contain sensitive personal information, they cannot be shared with third parties or released publicly without consent or adequate protection of the data subjects. Privacy-preserving mechanisms provide ways to sanitize data so that identities and/or confidential attributes are not disclosed.

A great variety of mechanisms have been proposed to anonymize structured databases with numerical and categorical attributes; however, automatically protecting unstructured textual data has received much less attention. In general, textual data anonymization requires, first, to detect pieces of text that may disclose sensitive information and, then, to mask those pieces via suppression or generalization.

In this work, we leverage several technologies to anonymize textual documents. We first improve state-of-the-art techniques based on sequence labeling. After that, we extend them to make them more aligned with the notion of privacy risk and the privacy requirements. Finally, we propose a complete framework based on word embedding models that captures a broader notion of data protection and provides flexible protection driven by privacy requirements. We also leverage ontologies to preserve the utility of the masked text, that is, its semantics and readability. Extensive experimental results show that our methods outperform the state

of the art by providing more robust anonymization while reasonably preserving the utility of the protected outcomes.

Resum

Cada dia els éssers humans afegim una gran quantitat de dades a Internet, tals com piulades, opinions, fotos i vídeos. Les organitzacions que recullen aquestes dades tan diverses n'extreuen informació per tal de millorar llurs serveis o bé per a propòsits comercials. Tanmateix, si les dades recollides contenen informació personal sensible, hom no les pot compartir amb tercers ni les pot publicar sense el consentiment o una protecció adequada dels subjectes de les dades. Els mecanismes de preservació de la privadesa forneixen maneres de sanejar les dades per tal que no revelin identitats o atributs confidencials.

S'ha proposat una gran varietat de mecanismes per anonimitzar bases de dades estructurades amb atributs numèrics i categòrics; en canvi, la protecció automàtica de dades textuals no estructurades ha rebut molta menys atenció. En general, l'anonimització de dades textuals exigeix, primer, detectar trossos del text que poden revelar informació sensible i, després, emmascarar aquests trossos mitjançant supressió o generalització.

En aquesta tesi fem servir diverses tecnologies per anonimitzar documents textuals. De primer, millorem les tècniques existents basades en etiquetatge de seqüències. Després, estenem aquestes tècniques per alinear-les millor amb el risc de revelació i amb les exigències de privadesa. Finalment, proposem un marc complet basat en models d'immersió de paraules que captura un concepte més ampli de protecció de dades i que forneix una protecció flexible guiada per les exigències de privadesa. També recorrem a les ontologies per preservar la utilitat del text emmascarat,

és a dir, la seva semàntica i la seva llegibilitat. La nostra experimentació extensa i detallada mostra que els nostres mètodes superen els mètodes existents a l'hora de proporcionar anonimització robusta tot preservant raonablement la utilitat del text protegit.

Resumen

Cada día las personas añadimos una gran cantidad de datos a Internet, tales como tweets, opiniones, fotos y vídeos. Las organizaciones que recogen dichos datos los usan para extraer información para mejorar sus servicios o para propósitos comerciales. Sin embargo, si los datos recogidos contienen información personal sensible, no pueden compartirse ni publicarse sin el consentimiento o una protección adecuada de los sujetos de los datos. Los mecanismos de protección de la privacidad proporcionan maneras de sanear los datos de forma que no revelen identidades ni atributos confidenciales.

Se ha propuesto una gran variedad de mecanismos para anonimizar bases de datos estructuradas con atributos numéricos y categóricos; en cambio, la protección automática de datos textuales no estructurados ha recibido mucha menos atención. En general, la anonimización de datos textuales requiere, primero, detectar trozos de texto que puedan revelar información sensible, para luego enmascarar dichos trozos mediante supresión o generalización.

En este trabajo empleamos varias tecnologías para anonimizar documentos textuales. Primero mejoramos las técnicas existentes basadas en etiquetaje de secuencias. Posteriormente las extendmos para alinearlas mejor con la noción de riesgo de revelación y con los requisitos de privacidad. Finalmente, proponemos un marco completo basado en modelos de inmersión de palabras que captura una noción más amplia de protección de datos y ofrece protección flexible guiada por los requisitos de privacidad. También recurrimos a las ontologías para preservar la utilidad del

texto enmascarado, es decir, su semantica y legibilidad. Nuestra experimentación extensa y detallada muestra que nuestros métodos superan a los existentes a la hora de proporcionar una anonimización más robusta al tiempo que se preserva razonablemente la utilidad del texto protegido.

Acknowledgements

This dissertation would not have been possible without the support of many people.

Foremost, I would like to express my sincere gratitude to my supervisors Dr. David Sánchez and Prof. Josep Domingo-Ferrer for their guidance during the development of this thesis, their patience, their motivation, their enthusiasm, and their immense knowledge. My sincere thanks also go to Dr. Jordi Soria-Comas, who was a co-supervisor during the first year of my doctoral work.

I am also thankful to Dr. Mohammed Jabreel for his helpful comments, support and encouragement.

I am indebted to all CRISES group members as well. Thanks also to my parents, brother and sisters for their unconditional support and love. And to all my friends for always being there.

Finally, I am grateful to Kuan Eeik Tan, Adrian Flanagan and the rest of Huawei Technologies Oy (Finland) Co.Ltd, R&D Center team, for their hospitality during my stay in Helsinki (Finland) in 2020.

This work was partially funded by the European Commission (projects H2020-871042 "SoBigData++" and H2020-101006879 "MobiDataLab"), by the Government of Catalonia (ICREA Acadèmia Prizes to J. Domingo-Ferrer and D. Sánchez, and grant 2017 SGR 705), the Spanish Government (projects RTI2018-095094-B-C21 "Consent" and TIN2016-80250-R "Sec-MCloud") and the Norwegian Research Council (project no. 308904 "CLEANUP").

Contents

Abstract **v**

Resum **vii**

Resumen **ix**

Acknowledgements **xi**

1 Introduction **1**
1.1 Objectives 3
1.2 Thesis structure 4

2 Background and state of the art **7**
2.1 Statistical disclosure control 7
2.2 Privacy models 10
2.2.1 k-Anonymity 10
2.2.2 ϵ-Differential privacy 11
2.3 State of the art in textual document anonymization 11
2.3.1 NER-based approaches 12
2.3.2 SDC methods 15
2.3.3 Methods to protect authorship attribution . . 17
2.4 Conclusion 18

3 Medical document anonymization **21**
3.1 Contributions and plan of this chapter 23
3.2 Background on sequence labeling and related concepts 24
3.2.1 Named-entity recognition 26
3.2.2 Part-of-speech tagging 27
3.2.3 Text chunking 27
3.2.4 Conditional random fields 27
3.2.5 Recurrent neural networks 28
3.2.6 Bidirectional recurrent neural networks . . . 29
3.2.7 Long short-term memory 29
3.2.8 Gated recurrent unit 29
3.2.9 BERT 30
3.3 Data description 31
3.4 First proposed system: ReCRF 32
3.4.1 Text tokenization 32
3.4.2 Rule generation 32
3.4.3 Feature extraction 33
3.4.4 Training the system 34
3.4.5 Using the system 35
3.4.6 Results and discussions 37
3.5 Second proposed system: E2EJ 39
3.5.1 Embedding layer 39
3.5.2 BiLSTM Layer 41
3.5.3 Sensitivity detection sub-model 42
3.5.4 NER type detection sub-model 43
3.5.5 Training 43
3.5.6 Experiments 44
Data set details 44

Hyper-parameters 45
3.5.7 Results 45
3.6 Competition results 47
3.7 Conclusion 47

4 Document anonymization from an SDC perspective 49
4.1 Contributions and plan of this chapter 50
4.2 Our approach 50
4.2.1 Proof of concept 52
4.3 Experimental results 53
4.3.1 Data set 53
4.3.2 Evaluation metrics 55
4.3.3 Results and discussion 56
4.4 Conclusion 57

5 Document anonymization via word embeddings 59
5.1 Contributions and plan of this chapter 60
5.2 Background on word embeddings and ontologies . 62
5.2.1 Word representation 62
Word embeddings 63
5.2.2 Ontologies 65
Types of ontologies 67
WordNet 68
YAGO 68
5.3 Our approach 69
5.3.1 Training the model 71
5.3.2 Detecting quasi-identifying terms 76
5.3.3 Masking quasi-identifying terms 77
5.4 Evaluation 81

5.4.1 Detection phase 81
5.4.2 Masking phase 94
5.4.3 Protection against re-identification 97
5.5 Application scenarios 100
5.6 Conclusion 103

6 Conclusions and Future Work **105**
6.1 Contributions and publications 106
6.2 Future work 108

Bibliography **111**

List of Figures

3.1 Recurrent neural network 28
3.2 Transformer structure 30
3.3 Training the ReCRF system 34
3.4 Using the ReCRF system 36
3.5 E2EJ architecture 40

4.1 Architecture of the named-entity recognition tagger 54

5.1 Example of the one-hot encoded sparse matrix . . . 63
5.2 Visualization of word embedding representations . 64
5.3 Taxonomy example 66
5.4 Overview of the training phase 72
5.5 Overview of the detection phase 78
5.6 Overview of the masking phase 82
5.7 Influence of the value of the similarity threshold t . 86

List of Tables

3.1 Distribution of PHI categories in the training, development and test corpora 31
3.2 ReCRF overall performance at detecting PHI subcategories on the first task 37
3.3 Confusion matrix of ReCRF on the test dataset . . . 38
3.4 ReCRF micro-averaged results on the development and test data sets 38
3.5 Hyper-parameter values chosen for the E2EJ system 45
3.6 Performance of E2EJ compared to various methods. The best value is in bold. 46
3.7 Competition final results for sub-task 1 (NER offset and entity type classification) and sub-task 2 (sensitive token detection (strict spans) on the test data set) 47

4.1 Feature extraction 53
4.2 Evaluation of the model on the test dataset at word level 56
4.3 Evaluation the model on the test dataset at entity level 57

5.1 Average precision, recall and F_1-score for the 50 evaluated documents 89
5.2 Output samples for each method 90
5.3 Average coefficients of variation (CV) for precision, recall and F_1-score 91
5.4 Precision, recall and F_1-score for a document referring to two different individuals 92
5.5 Evaluation figures with several pre-trained word embedding models 94
5.6 Average relative utility preserved by different methods and masking strategies 97
5.7 Percentage of correct predictions for each method . 98

List of Abbreviations

BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
Bi-RNN	**Bi**directional **RNN**
CBOW	**C**ontinuous **B**ag **O**f **W**ords
CRFs	**C**onditional **R**andom **F**ields
EEA	**E**uropean **E**conomic **A**rea
EU	**E**uropean **U**nion
GDPR	The european **G**eneral **D**ata **P**rotection **R**egulation
GRU	**G**ated **R**ecurrent **U**nit
EHR	**E**lectronic **H**ealth **R**ecords
HIPPA	**H**ealth **I**nsurance **P**ortability and **A**ccountability **A**ct
HMM	**H**idden **M**arkov **M**odel
LSTM	**L**ong **S**hort-**T**erm **M**emory
MEDDOCAN	**Me**dical **Do**cument **An**onymization Track
MLM	**M**asked **L**anguage **M**odels
MLP	**M**ulti-**L**ayer **P**erceptron
NER	**N**amed **E**ntity **R**ecognition
NLP	**N**atural **L**anguage **P**rocessing
NSP	**N**ext-**S**entence **P**rediction

NEs	**N**amed **E**ntities
OOV	**O**ut **O**f **V**ocabulary
PHI	**P**rotected **H**ealth **I**nformation
PII	**P**ersonally **I**dentifiable **I**nformation
POS	**P**art-**O**f-**S**peech
PPDP	**P**rivacy-**P**reserving **D**ata **P**ublishing
RNNs	**R**ecurrent **N**eural **N**etworks
RegEx	**Reg**ular **Exp**ression
SDC	**S**tatistical **D**isclosure **C**ontrol
SDL	**S**tatistical **D**isclosure **L**imitation
UKDA	The **U**nited **K**ingdom **D**ata **A**rchive
YAGO	**Y**et **A**nother **G**reat **O**ntology

I would like to dedicate this thesis to
my parents,
for their endless love, support and
encouragement.
And to my girlfriend Mar.
You made my life so much better in so many ways that it is hard to imagine doing this without you.

Chapter 1

Introduction

Text is the most usual way to share information in society. Textual data are therefore a crucial resource for many businesses and researchers. For instance, medical histories and clinical notes are needed in medical and pharmacological research (Meystre et al., 2010), publications in social networks can drive socioeconomic studies (Acemoglu and Ozdaglar, 2011), or written opinions and reviews can be used to improve recommender systems (Jakob et al., 2009). Yet, if textual documents contain personal sensitive information, they cannot be shared with third parties or released in the public sphere without properly protecting the fundamental right to privacy (Rubenfeld, 1989) of the individuals to whom the text refers. Privacy-preserving mechanisms provide ways to sanitize data so that identities and/or confidential attributes are not disclosed. In the last twenty years, a panoply of privacy protection methods have been proposed in the literature (Hundepool et al., 2013), most of them focused on structured data (that is, data that conform to a regular model such as a database schema) and more concretely on numerical attributes (Batet and Sánchez, 2018). However, little attention has been devoted to unstructured

textual data.

This contrasts with the fact that the vast majority of data generated nowadays are unstructured (Economist, 2010; Shilakes and Tylman, 1998). Specifically, unstructured text is the most common form of unstructured data, and it can be found in books, articles, web pages, emails, posts in social networks or clinical reports.

To protect structured databases, attributes are categorized according to their potential disclosure on the individual to whom a record corresponds. An *identifier* is an attribute whose values are enough to re-identify the individual to whom a record corresponds, whereas *quasi-identifiers* are attributes that separately do not allow re-identification but whose combination may. Both types of attributes entail *identity disclosure risk*. On the other hand, confidential attributes are those that may disclose sensitive information on the individual, thereby entailing *attribute disclosure risk*. The usual approach to data protection is to remove identifiers and mask quasi-identifiers (where masking can be enforced via perturbation, generalization or even suppression of values) (Samarati, 2001). While identifier attributes are usually easy to recognize, quasi-identifiers and confidential attributes are not. In general, we should classify as quasi-identifiers any set of attributes whose combined values may be available in an external data source that associates them with an identity.

If dealing with structured data may be challenging, protecting unstructured text is even more complex. First, we no longer have a fixed list of attributes: textual data may contain any information, which varies across documents. Furthermore, deciding what is a quasi-identifier or a confidential value is much more complex than with structured data: for each piece of text we need to judge

whether it can be used for re-identification or may disclose sensitive values. Such a judgment is not easy for a human expert (Bier et al., 2009), let alone for a computer program.

In general, accurate protection of textual documents remains a largely manual process (Bier et al., 2009). At most, (semi)automatic tools based on named entity recognition (NER) have been designed to remove –some– of the burden from the human experts. These tools are configured to pinpoint predefined entity types that are assumed to facilitate the re-identification of individuals (such as names, locations or dates).

1.1 Objectives

In this thesis, we aim to develop methods to automatically anonymize textual data. As such, we introduce the following set of goals:

- To study the privacy threats underlying textual data releases and survey works on data protection framed in the areas of statistical disclosure control (SDC) and privacy-preserving data publishing (PPDP), with a focus on protection methods for unstructured textual data.
- To develop and improve the current machine and deep learning methods (i.e. based on sequence labeling or NER) to tackle the medical document anonymization problem.
- To propose an extension of current NER-based models that is more in line with the notion of privacy as understood in the literature on SDC. To this end, we leverage NER-based

methods to detect identifiers, quasi-identifiers and confidential attributes, and we thereafter protect these in-text attributes using standard masking methods.

- To design and develop an integral approach that captures a broader and more accurate notion of privacy and of privacy requirements. To do this, we delve in state-of-the-art linguistic techniques and more specifically in word embedding models to automatically detect and mask quasi-identifiers in plain text. The goal is to offer a more flexible, robust and utility-preserving protection of unstructured documents.
- To design new metrics to evaluate the robustness of data protection, the potential disclosure risk and the degree of semantics/utility preservation of the masked outputs.

1.2 Thesis structure

- Chapter 2 reviews works on textual document anonymization and highlights their limitations.
- Chapter 3 describes the two systems we proposed to tackle the problem of anonymizing medical documents.
- Chapter 4 describes the application of the privacy protection notion as understood in the SDC literature to the anonymization of textual documents.
- Chapter 5 introduces a complete framework for the anonymization of textual documents by leveraging state-of-the-art word embedding models, linguistic methods and ontologies.

- Chapter 6 summarizes the main contributions of this thesis and presents some lines of future research.

Chapter 2

Background and state of the art in data anonymization

Data anonymization is a "process by which personal data is irreversibly altered in such a way that a data subject can no longer be identified directly or indirectly (*ISO 25237:2017, Health informatics Pseudonymization* 2017)." Data anonymization reduces the risk of unintended disclosure and in certain environments in a manner that enables evaluation and analytics post-anonymization data.

In contrast, re-identification is a process that breaks anonymization by determining the identity of the subject to whom a piece of data corresponds. Re-identification can be attained by linking the anonymized data to other data sources that contain identifiers and share some attributes with the anonymized data.

2.1 Statistical disclosure control

Statistical disclosure control (SDC), also known as statistical disclosure limitation (SDL), is a discipline that provides methods to

transform/mask an original piece of data in such a way that the transformed/masked data are protected against disclosure.

Measuring the risk of disclosure is necessary to decide whether a data set is protected enough for release or sharing. The attributes in a data set can be classified as follows depending on the disclosure risk they entail (Hundepool et al., 2013; Matthias Templ and Kowarik, 2016):

- Identifiers: any attribute that contains information that directly identifies an individual, such as passport no., social security no., full name, etc.
- Quasi-identifiers: attributes which are not identifiers but which together might allow linking a record in the released data with some external data source containing identifiers. As a result, a set of quasi-identifiers might lead to re-identification of the individual to whom a record corresponds. Some examples of quasi-identifiers are gender, age, address, telephone no., etc.
- Confidential attributes: attributes that contain sensitive individual information and that should not be unequivocally linked to an identity, such as religion, medical diagnosis, salary, sexual orientation, etc.
- Other attributes: any other attributes not fitting in any of the previous categories.

A common approach to anonymizing structured data is to remove identifiers and mask quasi-identifiers.

SDC considers several types of disclosure risks (Hundepool et al., 2013):

- Identity disclosure. This type of disclosure occurs when a subject among those to whom the released data correspond is re-identified.

- Attribute disclosure. This type of disclosure occurs when the value of a confidential attribute for a certain subject can be determined more accurately with access to the released data than would otherwise have been possible.

Enforcing SDC to protect structured databases with numerical and categorical attributes is a well-studied process. However, protecting unstructured textual data cannot be tackled in the same way. In general, textual data protection requires first detecting pieces of text that may help to disclose sensitive information, classifying them as identifiers, quasi-identifiers, or confidential attributes, and then masking those detected pieces via suppression or generalization.

As we discuss later in this chapter, most current solutions to text anonymization rely on pre-trained classifiers called NER models. NER models can recognize a fixed set of named entities, such as names or locations. Using NER models works in a scenario where the entities to be protected are well defined and have a specific format (*e.g.*, personal health identifiers (PHIs) in the medical domain). However, there also several scenarios where these solutions may fail to give good results. We explain more in detail sequence labeling models in Section 3.2.

2.2 Privacy models

Whereas data protection methods like suppression, generalization, noise addition and microaggregation offer *ex post* privacy guarantees for the data that should be anonymized, privacy models establish *ex ante* conditions that such data must satisfy to guarantee a certain level of anonymity for the individuals. In the sequel, we depict two of the main privacy models proposed in the literature.

2.2.1 k-Anonymity

k-Anonymity is a privacy model for microdata releases focused on preventing the re-identification of the individuals to whom the data refer. Let D be a microdata set consisting of quasi-identifier attributes and confidential attributes. To prevent re-identification, the idea underlying k-anonymity is to make each combination of quasi-identifier attribute values non-unique by transforming the original data set D into a released data set D^* where the combination is shared by at least k records (Samarati and Sweeney, 1998). The set of records in D^* sharing the same combination of values for all the quasi-identifier attributes is named equivalence class. Therefore, an attacker with access to an external non-anonymous dataset that contains the quasi-identifier attributes from the released dataset D^* will not be able to link a specific individual to a specific record in D^*. In this scenario, the attacker will at most be able to identify the k-anonymous class in D^* that contains the target individual. Therefore, the probability of correct re-identification is not greater than $1/k$.

2.2.2 ϵ-Differential privacy

Whereas the k-anonymity model is aimed at microdata releases, ϵ-differential privacy was proposed as a privacy guarantee for queryable databases (Dwork, 2008), where queries (typically count queries) are submitted to a database containing the original individual records (microdata). In this query-answer interactive environment, differential privacy states the conditions that the answers must satisfy so that disclosure risk is under control. The anonymization mechanism to attain differential privacy is called a differentially private sanitizer and sits between the user submitting queries and the database answering them. The principle underlying differential privacy is that the presence or absence of any single individual in the database should be undetectable when analyzing the outcomes of the queries. To that end, the sanitizer must limit the contribution of any single individual to the response to a query. Since differential privacy assumes that each record in the data set refers to a different individual, comparing the outcome of a query before and after an individual has contributed her data to the data set is equivalent to comparing the outcome of that query between data sets that differ in at most one record (neighbor data sets).

2.3 State of the art in textual document anonymization

Statistical disclosure control (SDC) and privacy-reserving data publishing (PPDP) methods protect data sets in order to make them available for secondary use. However, since they require an *ex*

ante classification of attributes as identifiers, quasi-identifiers and confidential attributes, these methods can only be employed on structured data. This contrasts with the fact that most of the personal data currently being gathered (e.g., from social networks, web browsing logs, etc.) that should be subject to anonymization are unstructured textual data. In the following we survey the scarce approaches proposed in the literature to (partially) automate the protection of textual documents.

The task of protecting the private information of the individuals mentioned in text documents is referred to in the literature as *document redaction* (Bier et al., 2009), *sanitization* (Sánchez and Batet, 2016) or *anonymization* (Anandan and Clifton, 2011). Whatever the name, it consists of two steps: (i) detecting (potentially) disclosive pieces of text, and (ii) masking those pieces appropriately.

For many years, textual data protection has been a highly manual process (Agency, 2005), and it still is. Usually, several human experts review the text and mask all items they deem usable to re-identify individuals and/or disclose confidential data on them (Bier et al., 2009).

2.3.1 NER-based approaches

To reduce the burden of human experts, some systems that make use of named entity recognition (NER) have been introduced.

NER was created as a way to extract structured information, like person and organization names, locations, times or dates, from an unstructured text. Early NER systems were based on handcrafted rules or regular expressions. For instance, times can be

identified using the following pattern: "at" + digits + "am"/"pm". Up until 2000, handcrafted rule systems offered the best results. Statistical approaches subsequently took over. In statistical NER systems, models such as HMM (hidden Markov models) or CRF (conditional random fields) are trained to locate a specific type of entity.

With the development of deep learning neural networks, recurrent neural networks (RNN) and extensions of them such as long short-term memory (LSTM) and gated recurrent units (GRUs) surpassed the accuracy of statistical NER systems. Nowadays, the state of the art is based on transformers like BERT (Devlin et al., 2018) and ELMo (Peters et al., 2018). These are pre-trained on large amounts of data and, unlike previous models, they characterize words according to their context. Even though they are general-purpose NLP models, these contextual models can be tailored or fine-tuned to solve multiple tasks including NER, but also sentiment analysis, text generation, question answering, summarization or machine translation.

Training a NER model from scratch or tailoring an NLP model for NER require a considerable amount of tagged data that match the language to which the NER model is to be applied. Well-trained NER models usually have high precision (typically above 80%). Additionally, there are quite a few software packages available to carry out NER tasks, such as spaCy (Honnibal and Montani, 2017) or the Stanford NER (Manning et al., 2014).

Current solutions for textual data protection employ NER because they assume that the named entities (NEs) are the ones that entail the highest disclosure risk, as they refer to real-world entities. Amazon's Macie (*Amazon Macie - Amazon Web Services (AWS)*)

locates several personally identifiable information items (like names, addresses, birth dates, etc.) and classifies documents in several categories according to their risk. Additionally, Macie is capable of detecting many information items that are regarded as confidential (like passwords, bank accounts, etc). Google's Cloud DLP (*Cloud Data Loss Prevention*) also leverages rules and machine learning techniques to detect the presence of confidential and re-identifying pieces of information. Similarly, Symantec's Data Loss Prevention (*Symantec Data Loss Prevention*) uses dictionaries and rules (to detect several types of information items that have a regular structure) as well as machine learning (to detect other types of identifiable and confidential information that lack a regular structure). Microsoft's Presidio tool (Microsoft, 2019) is based on combining regular expression matching, spaCy NER models and Flair (Akbik, Bergmann, and Vollgraf, 2019) with BERT embeddings. It is trained on 80,000 samples generated with data augmentation techniques and can detect 17 (quasi-)identifying and confidential categories.

As evidenced by the number of commercial tools available, NER-based systems are practical enough to be employed in real-world applications. However, since they assume that all (and only) the NEs in a given document should be protected, they suffer from the severe limitations highlighted in the introduction. First, nouns or phrases other than NEs may also be (quasi-)identifying, such as demographic attributes or healthcare conditions. Second, not all the NEs appearing in a document should be protected, perhaps because they are very general entities (such as countries or large cities) or because they do not refer to the individual to be protected (as it may happen when the text refers

to other individuals in addition to the one to be protected). Third, only the NE classes for which the classifier has been trained can be detected; this means that usually a few dozens of NE classes can be detected, whereas there is an unbounded number of potential quasi-identifying information types that may not resemble NEs at all (*e.g.*, demographic information may be quasi-identifying in some contexts). The reason is that training the classifier requires a large collection of manually tagged data for each NE type and language to be supported. In summary, NER-based protection introduces considerable burden (due to the need of training), and typically results in both unnecessary masking and weak protection.

2.3.2 SDC methods

The methods and tools reviewed above solely focus on detecting (quasi-)identifying information or at most they suppress disclosive items or replace them with coarse NE values (like "person", "location", "date"). This falls short of optimizing data protection, which consists in using the minimum amount of masking required to meet the privacy requirements. The analytical utility of the protected outcomes ought to be preserved as much as possible, for them to be usefully shareable.

Generalization is the most common utility-preserving masking technique applied to the protection of text (Sánchez and Batet, 2017). Unlike other methods in the literature, such as entity swapping (Abril, Navarro-Arribas, and Torra, 2011) and noise addition (Feyisetan, Diethe, and Drake, 2019), generalization outputs truthful data (Chakaravarthy et al., 2008; Cumby and Ghani, 2011;

Anandan et al., 2012). The methods in (Chakaravarthy et al., 2008) and (Cumby and Ghani, 2011) use generalization similarly to the way k-anonymity (Samarati, 2001) is employed in structured databases: they assume a large and homogeneous collection of documents and generalize the quasi-identifying terms so that there are at least k identical generalizations in the collection. In this way, each disclosive term becomes indistinguishable from at least $k-1$ other terms in the collection. However, assuming a homogeneous set of documents and protecting them groupwise is quite restrictive.

An approach that can individually sanitize documents is presented in (Anandan et al., 2012). The authors employ a knowledge base to generalize quasi-identifying terms so that at least t plausible versions of the generalized document can be created by combining specializations of the generalized terms. The authors acknowledge that setting the value of t is not intuitive and that it is hard to predict the protection offered by a concrete value because it depends on several factors including the document size, the number of terms to be masked and the detail of the knowledge base used to generalize terms.

The methods cited in the previous paragraph concentrate on masking quasi-identifying terms, but they assume those items have been already detected. An integral approach considering both detection and utility-preserving masking is presented in (Sánchez, Batet, and Viejo, 2013a; Sánchez, Batet, and Viejo, 2013b; Sánchez and Batet, 2016; Sánchez and Batet, 2017). The authors propose a privacy model grounded on information theory that quantifies disclosure risks as a function of the mutual information shared

among the entities referred to in the document. Afterwards, quasi-identifying items are generalized so that the amount of information they disclose on the entity to be protected is sufficiently decreased. Even though this approach is more general than NER-based methods, it suffers from the need to compute accurate conditional probabilities among all the combinations of terms in the document. This hampers scalability to deal with large collections of documents.

2.3.3 Methods to protect authorship attribution

Some authors have recently proposed privacy-preserving methods for text documents that build on word embeddings (Li, Baldwin, and Cohn, 2018; Fernandes, Dras, and McIver, 2019; Feyisetan, Diethe, and Drake, 2019). However, these works focus on obfuscating the authorship of the document, rather than protecting the privacy of the individuals referred to in the text. The authorship of a document and the author's attributes are inferred from the linguistic and stylistic properties/regularities of written text rather than the document's topic or the text semantics. Hence, the approaches to protecting the document author rely on distorting the distribution of words in the text via differentially private noise added to the word embeddings (Fernandes, Dras, and McIver, 2019; Feyisetan, Diethe, and Drake, 2019) or on constraining the training of the embeddings to prevent disclosing certain attributes (Li, Baldwin, and Cohn, 2018). Thus, the outputs of those systems are –distorted– word distributions (*e.g.*, bag-of-words) (Fernandes, Dras, and McIver, 2019) or constrained embedding models (Li, Baldwin, and Cohn, 2018) rather than actual

documents.

As a result, the outputs lose their readability and are only useful for applications that are compatible with this deconstructed representation of documents, such as topic classification. Finally, as discussed above, noise-based approaches in which words are probabilistically replaced by other words do not preserve the truthfulness of the output, unlike generalization-based masking, which is the usual approach to document sanitization.

Employing differential privacy in document releases (even if only word distributions are released) also bears an important limitation: in order to keep the results reasonably useful, a large epsilon (*i.e.*, significantly greater than 1) is needed. For example, in (Fernandes, Dras, and McIver, 2019), epsilon values between 10 and 30 are employed. It is widely acknowledged that the robust privacy guarantee of differential privacy fades away for such large values of epsilon (Domingo-Ferrer, Sánchez, and Blanco-Justicia, 2020; Dwork, Roth, et al., 2014).

2.4 Conclusion

Anonymization of textual data is a challenging task because, unlike for structured data, attributes can hardly be categorized into a finite list of categories. Simplistic approaches to text anonymization assume that only named entities should be (systematically) protected, but this neglects the actual anonymization task, which is to protect against re-identification of a particular entity. As a result, the outcomes are poorly protected because either disclosive information is missed (*i.e.*, that information not falling into the

supported NE types) or unnecessary masking is performed (*i.e.,* for NEs not referring to the actual entity to be protected).

Chapter 3

Medical document anonymization

Patient notes in electronic health records (EHRs) contain critical information that may be useful for medical investigations. However, due to privacy concerns, the vast majority of medical investigators can only access anonymized or de-identified notes to protect the confidentiality of patients(Medicare & Medicaid Services et al., 1996). Anonymization can be either manual or automated. Manual anonymization means that human annotators label protected health information (PHI). This approach has some drawbacks. First, only a limited set of individuals is allowed to access the identified patient notes. Thus, the task cannot be crowdsourced. Second, humans are prone to mistakes. Third, manual anonymization is impractical given the size of EHR databases. Therefore, a reliable automated anonymization system would be of high value (Uzuner, Luo, and Szolovits, 2007; Meystre et al., 2010). In the literature, there are many systems for EHR anonymization, which we can categorize as rule-based, feature-engineering-based, or deep-learning-based approaches.

Starting with a seed collection of sensitive tokens, the idea of rule-based systems is to manually engineer some rules based on regular expressions, syntactic, or dependency structures to expand the collection iteratively (Sweeney, 1996; Neamatullah et al., 2008).

The feature-engineering-based systems aim to train a sequence tagger with rich, hand-crafted features based on linguistic or syntactic information from annotated corpora to predict a label (*e.g.*, O, B- < entity > or I- < entity >) on each token in a sentence (Liu et al., 2015).

Rule-based and feature-engineering-based approaches are labor-intensive for constructing rules or features using linguistic and syntactic information. Despite some promising results, there are two main issues with these approaches. First, the engineering of rules and features is a time-consuming task. Moreover, rules always need to be updated. Second, the systems of these two categories are dependent on some external requirements like a parser analyzing the syntactic and dependency structure of sentences. Therefore, the performances of these systems rely on the quality of the parsing results (Uzuner, Luo, and Szolovits, 2007; Neamatullah et al., 2008). To avoid these issues, deep learning is used to develop systems that learn high-level representations for each token, on which a classifier or sequence tagger can be trained (Liu et al., 2017).

Medical document anonymization (MEDDOCAN) (Marimon et al., 2019) is a challenge in the shared task of IberLEf 2019 dedicated to EHRs in the Spanish language. There are two structured sub-tasks: ”sensitive token detection” and ”NER offset and entity

type classification". The first sub-task aims to identify the sensitive tokens in a document. We can solve this sub-task as a token-level binary classification problem in which we develop a system that takes as input a document and classifies each token as sensitive or not. The second sub-task aims at identifying the type of each token in a document. We can model this problem as a sequence tagging problem. The input is a sequence of tokens, and the output is their corresponding labels.

3.1 Contributions and plan of this chapter

This chapter focuses on developing and improving the state-of-the-art NER-based models to handle the problem of medical document anonymization. We developed two systems, ReCRF and E2EJ. The two proposed systems were submitted to the MEDDOCAN 2019 contest. ReCRF is a hybrid system that automatically detects PHI entities from plain text medical documents. The system consists of an automatically constructed RegEx model and a trained CRF model. The design of the system, which includes using a variety of linguistic and semantic features to increase the accuracy, ensures that it can be generalized well in front of new data. On the other hand, E2EJ is a joint and end-to-end neural network-based system for the two MEDDOCAN sub-tasks. The proposed system provides an end-to-end solution and does not require any parsers or other linguistic resources. Specifically, the proposed system is a multilayer neural network, where the first three layers aim to learn high representations for a sequence of tokens. Then the outputs of the first three layers are passed to two

submodels that are learned interactively. One is for extracting the sensitive tokens, while the other is for identifying their types.

The rest of this chapter structured as follows: Section 3.2 provides background on sequence labeling tasks; Section 3.3 briefly introduces the data to be used; Section 3.4 and Section 3.5 present the two proposed systems, ReCRF and E2EJ, respectively; Section 3.6 shows the final results of the MEDDOCAN 2019 competition; finally, Section 3.7 contains the conclusions.

3.2 Background on sequence labeling and related concepts

In machine learning, sequence labeling is a type of pattern recognition task. The main goal of this task is to assign each member of a sequence of observed values to one or more predefined categorical labels. A common example of a sequence labeling task is the NER task, which seeks to assign a NER tag (*e.g.* PERSON, LOCATION, etc.) to each word in an input sentence or document.

Sequence labeling algorithms are categorized into three main approaches: probabilistic models, deep-learning models and attention-based models. Probabilistic models rely on statistical inference to find the best sequence. The most common probabilistic models in use for sequence labeling are hidden Markov models (HMM) and conditional random fields (CRF). Probabilistic models are very powerful but they have a few drawbacks. One of their best-known weaknesses is their lack of semantic awareness, which causes difficulty to generalized to unseen data. Also, they

have trouble handling long sequential dependency (*i.e.* dependencies of long input sequences are often ignored).

On the other hand, deep-learning models like recurrent neural networks (RNN), long short-term memory (LSTM) and gated recurrent units (GRU) are designed to capture local dependencies and find longer patterns. Deep-learning models have shown great power to learn latent features. The training process is a joint learning of the most representative features and training the best model given these features. This is crucial for model development because it dramatically decreases the development time by saving work on handcrafted features. However, RNN and its derivatives mainly use sequential processing over time (see Figure 3.1). This long-term information sequential travel may cause corrupted outputs due to the multiplication of the input many times by small numbers. This phenomenon is called vanishing gradients. Even though LSTM and GRU have ways to remove some of the vanishing gradient problems, they still suffer of time delays and memory consumption due to their sequential input processing, which makes the training of these types of models very slow.

Finally, with the advance of NLP, transfer learning and pretrained language models like bidirectional encoder representations from transformers (BERT) (Devlin et al., 2018) have become the state of the art to solve most of the NLP tasks, including sequence labeling. In general, transformers have become important for several sequence tasks both in NLP and in other fields. The main idea behind the huge success of transformers is the "self-attention" mechanism. Instead of using recurrent processing of input like RNNs, self-attention uses a token pair-wise scoring system. Thus, given a word in a sequence, self-attention can learn which other

words are to be prioritized for determining the meaning of a word.

3.2.1 Named-entity recognition

Named-entity recognition (NER) is the task of locating and categorizing important terms in a text (Nadeau and Sekine, 2007). Named-entity recognition is a source of information for different natural language processing applications. NER has been used to improve the performance of many applications, such as answering questions (Khalid, Jijkoun, and De Rijke, 2008), automatic text translation (Babych and Hartley, 2003), information retrieval (Sundheim, 1996), and sentiment analysis of tweets (Jabreel, Hassan, and Moreno, 2018).

NER is also useful in the anonymization of unstructured data (*e.g.* free-text documents). In particular, it can detect those terms that might be used to re-identify an individual and those terms that contain sensitive information.

There are many tagging schemes for NER, like IOB2, IOBES. In IOB2 (Sang and Veenstra, 1999), each word in the text is labeled using one of three possible tags: I, O, or B, which indicate if the word is inside, outside, or at the beginning of a named entity. In contrast to IOB2, IOBES (Krishnan and Ganapathy, 2005) tagging schemes distinguish between the beginning and end of a named entity by two more tags E and S, where S is used to represent a named entity containing a single token. Named entities of length greater than or equal to two always start with the B tag and end with the E tag. BILOU sometimes are referred to the same scheme, where L represents last/end and U represents unit/single.

3.2.2 Part-of-speech tagging

Part-of-speech (POS) tagging is a standard sequence labeling task that aims at assigning a correct part-of-speech tag to each lexical item (a.k.a., word) such as noun, verb, adjective. In general, POS can also be viewed as a subclass division of all words in a language, which is thus also called a word class. The tagging system of part-of-speech tags is not usually uniform under different data sets, *e.g.*, (Taylor, Marcus, and Santorini, 2003), which includes 45 different types of POS tags for word classification. For instance, the sentence "Tom Cruise is an American actor and producer.", will be labeled with a sequence like "NNP NNP VBZ DT JJ NN CC NN ." where NN means noun, VB means verb, NNP means Proper noun, DT means determiner, CC means Coordinating conjunction and JJ means adjective.

3.2.3 Text chunking

Text chunking divides the text into syntactically related non-overlapping groups of words (*i.e.* phrases such as noun phrase, verb phrase, etc.). Similar to NER, it can be handled by sequence labeling models and it can also adopt the tagging system as IOB2 or IOBES.

3.2.4 Conditional random fields

In NLP, apart from deep-learning models, there are two common probabilistic models used to solve NER tasks: hidden Markov models (HMMs), used in works such as (Morwal, Jahan, and Chopra, 2012; Zhou and Su, 2002), and conditional random fields (CRFs),

used in works such as (Culotta, Bekkerman, and McCallum, 2004; Ekbal, Haque, and Bandyopadhyay, 2007; Jabreel, Hassan, and Moreno, 2018). NER using CRFs is widely employed and applied and it usually gives good results in many domains.

CRFs (Lafferty, McCallum, and Pereira, 2001) are conditionally trained undirected graph models often applied in pattern recognition. These models are used to calculate the conditional probability of values on designated output nodes given values assigned to other designated input nodes.

3.2.5 Recurrent neural networks

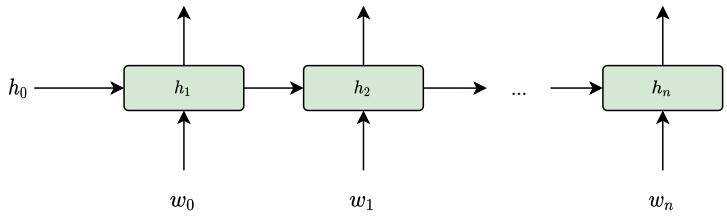


FIGURE 3.1: Recurrent neural network

Recurrent neural networks (RNNs) are one of the best models to handle the problem of sequence labeling. Unlike standard feed-forward neural networks, RNNs have internal memory. The input to the RNN is divided into time steps, and then the RNN recursively performs the same function for every time step, where the output from the previous step is fed as input to the current step. The RNN decision considers the current input and the output that has been learned from the previous input.

3.2.6 Bidirectional recurrent neural networks

The standard RNN reads an input sequence in a forward direction left-to-right or right-to-left. Thus, it processes sequences in temporal order, ignoring the future context. For sequence labeling, it is beneficial to have access to future as well as to past information. Bidirectional RNNs (Bi-RNNs) were proposed to solve that problem by connecting two hidden layers of opposite directions to the same input. This structure allows Bi-RNNs to take into consideration both past and future information.

3.2.7 Long short-term memory

Long short-term memory (LSTM) is an improved version of the standard RNN developed to deal with the vanishing gradient problem. The only difference between RNNs and LSTMs is that LSTM introduces new gates, such as input and forget gates, which allow for better control over the gradient flow and enable better preservation of long-range dependencies.

3.2.8 Gated recurrent unit

Gated recurrent units (GRU) are an improved version of the standard recurrent units. Their gated units allow the network to pass or block information from one time step to the other. These new gates in the GRU structure make it able to keep information around for even longer sequences. GRU are much faster than LSTM because they have a simpler structure.

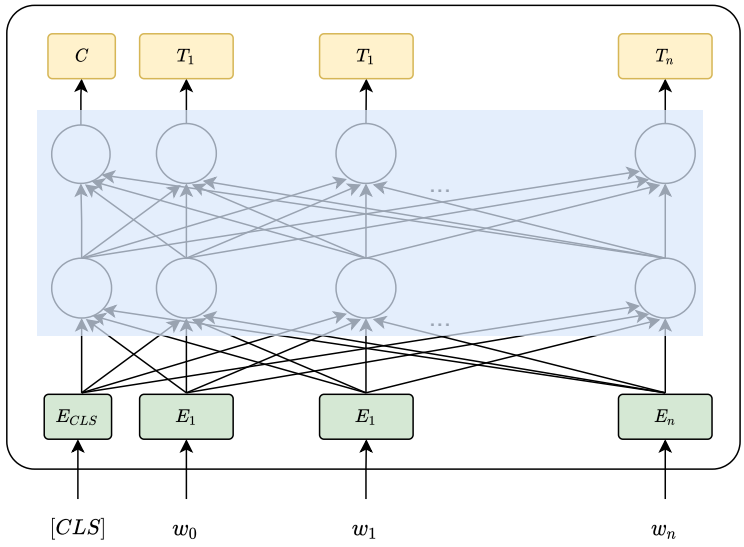


FIGURE 3.2: Transformer structure

3.2.9 BERT

BERT is a new language representation model which has become the state of the art to solve most NLP tasks. Unlike recent language representation models, BERT is designed to be pre-trained on unlabeled text by jointly conditioning both left and right context. The pre-trained BERT model can be fine-tuned by just adding one output layer to create a domain-specific model without the need to modify the architecture of the model. Nowadays, BERT is used in a wide range of tasks, such as sequence labeling, question answering, and machine translation. BERT is empirically robust and characterized by its simplicity. It obtains new state-of-the-art

results on eleven natural language processing tasks, including sequence labeling.

3.3 Data description

The MEDDOCAN challenge task aims at identifying and extracting several types of PHI categories from plain text medical documents. The PHI categories are grouped into eight main categories with 22 sub-categories. The corpora released for the tasks consist of 1,000 documents, divided into: 500 as training data, 250 as development data and 250 as test data. The distributions of PHI categories and sub-categories in the training, development and test data are shown in Table 3.1.

TABLE 3.1: Distribution of PHI categories in the training, development and test corpora

PHI Category	Sub-Category	Training Data	Devloping Data	Test Data
AGE	EDAD_SUJETO_ASISTENCIA	1035	521	518
CONTACT	CORREO_ELECTRONICO	469	241	249
	NUMERO_FAX	15	6	7
	NUMERO_TELEFONO	58	25	26
DATE	FECHAS	1231	724	611
ID	ID_ASEGURAMIENTO	391	194	198
	ID_CONTACTO_ASISTENCIAL	77	32	39
	ID_EMPLEO_PERSONAL_SANITARIO	0	1	0
	ID_SUJETO_ASISTENCIA	567	292	283
	ID_TITULACION_PERSONAL_SANITARIO	471	226	234
LOCATION	CALLE	862	434	413
	CENTRO_SALUD	6	2	6
	HOSPITAL	255	140	130
	INSTITUCION	98	72	67
	PAIS	713	347	363
	TERRITORIO	1875	987	956
NAME	NOMBRE_PERSONAL_SANITARIO	1000	497	501
	NOMBRE_SUJETO_ASISTENCIA	1009	503	502
OTHER	FAMILIARES_SUJETO_ASISTENCIA	243	92	81
	OTROS_SUJETO_ASISTENCIA	9	6	7
	SEXO_SUJETO_ASISTENCIA	925	455	461
PROFESSION	PROFESION	24	4	9
Total:		11 333	5801	5661

3.4 First proposed system: ReCRF

We developed an automatic system to detect PHI categories from Spanish medical documents. The next subsections describe the steps followed to train and use the system.

3.4.1 Text tokenization

In this step, we tokenize the text at two levels: sentence-level and word-level. First, we use a sentence tokenizer that takes a single document as input and produces a list of sentences. Afterwards, we split every single sentence into a list of tokens. The sentence tokenizer is based on a newline delimiter, whereas a manually-crafted regular expression-based tokenizer and a spaCy pre-trained model for Spanish(Honnibal and Montani, 2017) are sequentially used to perform the word-level tokenization.

3.4.2 Rule generation

In this step, we developed a data-driven regular expression generator in order to avoid hand-crafting regular expression rules. This generator analyzes all the appearances of the PHI categories in the training data set and, from them, it generates rules to detect those categories. These rules are later used to extract labeled tokens that are used to guide the CRF tagger in taking the final decision.

3.4.3 Feature extraction

We extract a wide variety of linguistic features, similarly to previous studies (Yang and Garibaldi, 2015; Stubbs, Kotfila, and Uzuner, 2015). These features characterize the semantics of PHI terms. The main types of features are:

- **Lexical features**. They include the target word itself, its prefix and suffix, word lemma, and part-of-speech (POS) tag.
- **Orthographic Features**. They detail word form information, *e.g.* target word length, word shape (CAPITALIZED, ALL_UPPER, ALL_LOWER, MIX), ends with s, contains alpha and contains numbers.
- **RegEx features**. A RegEx model is used as a first-pass recognizer for the PHI entities in the text. We use the output of the RegEx model to detect the location of the token, either at the beginning, middle, end or outside of PHI entity.
- **External Resource Features**. We also consider if a token appears into one or several external resources, which include lists of English and Spanish names of countries and cities, names and abbreviations of time expressions (*e.g.* 'año', 'mes'), or names and abbreviations of places (*e.g.* 'plaza', 'av.'). Additional resources include lists of Spanish last names, Spanish first names, addresses, hospitals, cities and towns, professions, autonomous communities, and provinces.

Extracting these features from just the target word does not consider the context in which the word appears, which may lead to misclassifying tokens due to language ambiguity. To tackle this,

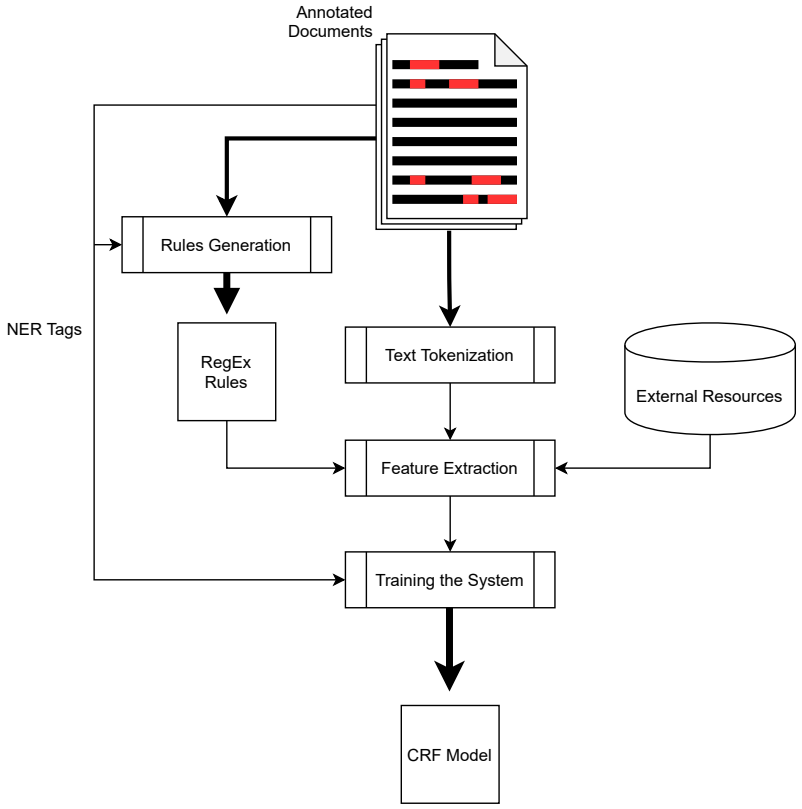


FIGURE 3.3: Training the ReCRF system

we consider a window of 5 words centered at the target word (*i.e.*, the two words on the left and the two words on the right).

3.4.4 Training the system

We used both a set of automatically-crafted rules (RegEx model) and conditional random fields (Lafferty, McCallum, and Pereira,

2001) (CRF model) to identify PHIs in medical documents. The system was implemented using Python 3.7 with sklearn-crfsuite package (*sklearn-crfsuite*) and spaCy package (Honnibal and Montani, 2017) for the tokenization. We also used the BIO tagging scheme to set the labels of the tokens (Sang and Veenstra, 1999). Each word token in the document was labeled using one of three possible tags: B, I, or O, which indicate if the word is at the beginning, at the middle, or outside of a PHI entity.

Fig 3.3 shows that our system has two outputs: the RegEx model and the CRF model. The RegEx model is built by the automatic rule extractor by analyzing the PHI categories that appear in well-structured contexts (e.g. Nombre: Xxxxx., Fecha de nacimiento: dd/mm/yyyy.).

The CRF model is trained by passing all the extracted features from the tokens plus the decision of the RegEx model, which adds extra information and makes the decision easier for the CRF model.

3.4.5 Using the system

Fig 3.4 shows how both RegEx and CRF models are used to make the annotations. Even though the RegEx model is accurate enough to detect well-structured entities, it is not effective in front of small changes in the text format. So, we decided to use the RegEx model to perform a preliminary annotation, which is then passed to the CRF model that will make the final decision.

Fig 3.4 shows how both the RegEx and the CRF models are used to make the prediction decision. The RegEx model is accurate to detect well-structured entities. However, it is not robust

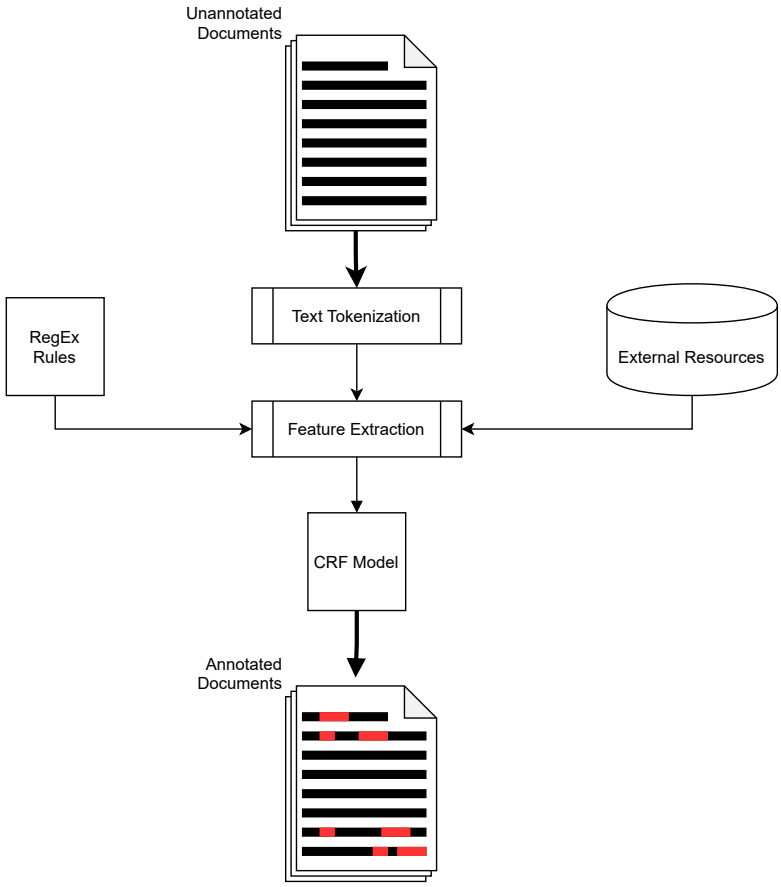


FIGURE 3.4: Using the ReCRF system

against simple changes in the data format. So we decided to use the RegEx model to make an initial decision. This decision goes to the CRF model, which works at the end and makes the final decision.

TABLE 3.2: ReCRF overall performance at detecting PHI sub-categories on the first task

PHI Category	Sub-Category	#Expected	#Predicted	#Correct	Precision	Recall	F1
AGE	ESA	514	516	501	97.09	97.47	97.28
CONTACT	CE	248	249	246	98.80	99.19	98.99
	NF	7	7	5	71.43	71.43	71.43
	NT	26	26	22	84.62	84.62	84.62
DATE	Fech	612	612	603	98.53	98.53	98.53
ID	IA	199	199	197	98.99	98.99	98.99
	ICA	38	39	38	97.44	100.00	98.70
	IEPS	0	0	0	0.00	0.00	0.00
	ISA	277	277	274	98.92	98.92	98.92
	ITPS	234	235	233	99.15	99.57	99.36
LOCATION	Call	412	406	382	94.09	92.72	93.40
	CS	4	6	2	33.33	50.00	40.00
	Hosp	129	120	109	90.83	84.50	87.55
	Inst	58	49	24	48.98	41.38	44.86
	Pais	366	354	348	98.31	95.08	96.67
	Terr	950	945	916	96.93	96.42	96.68
NAME	NPS	499	501	497	99.20	99.60	99.40
	NSA	503	502	502	100.00	99.80	99.90
OTHER	FSA	82	76	59	77.63	71.95	74.68
	OSA	6	2	0	0.00	0.00	0.00
	SSA	461	459	455	99.13	98.70	98.91
PROFESSION	Prof	6	4	3	75.00	50.00	60.00

3.4.6 Results and discussions

The performance of the detection of PHI categories was evaluated using Precision, Recall and F1 scores at the entity level. The results of our system on the test set for the different PHI categories are shown in Table 3.4; the confusion matrix is shown in Table 3.3. Notice that categories that have low frequency in the training dataset have less F1 score (*e.g.* NUMERO_FAX appears only 15 times in the training set and CENTRO_SALUD appears

TABLE 3.3: Confusion matrix of ReCRF on the test dataset

	Output																							
Key	**Call**	**CS**	**CE**	**ESA**	**FSA**	**Fech**	**Hosp**	**IA**	**ICA**	**IEPS**	**ISA**	**ITPS**	**Inst**	**NPS**	**NSA**	**NF**	**NT**	**OSA**	**Pais**	**Prof**	**SSA**	**Terr**	**Miss**	**Total**
Call	382		1																			1	28	412
CS		2																					2	4
CE			246																				2	248
ESA				501	1	1																	11	514
FSA				3	59													1					19	82
Fech						603			1								1					1	6	612
Hosp		1					109																19	129
IA								197				1											1	199
ICA									38															38
IEPS																								
ISA								1			274												2	277
ITPS												233											1	234
Inst		1					1						24										32	58
NPS														497									2	499
NSA					1										502									503
NF																5							2	7
NT																1	22						3	26
OSA											1												5	6
Pais																			348			5	13	366
Prof																				3			3	6
SSA					1													1			455		4	461
Terr						1							2						1			916	30	950
Spur	24	2	2	12	14	7	10	1			2	1	23	4		1	3		5	1	4	22		138
Total	406	6	249	516	76	612	120	199	39		277	235	49	501	502	7	26	2	354	4	459	945	185	5769

Spur=Spurious, Miss=Missing

six times). This result is expected because the model did not get enough examples in order to learn how to accurately detect them.

TABLE 3.4: ReCRF micro-averaged results on the development and test data sets

Sub-Task	**Development Data**			**Test Data**		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Sub-Track 1 (NER)	97.36	95.45	96.40	96.99	95.67	96.33
Sub-Track 2 (Spans strict)	97.78	95.86	96.81	97.53	96.20	96.86
Sub-Track 2 (Spans merged)	98.33	96.58	97.45	98.13	96.89	97.50

The overall result of the ReCRF system for both sub-tasks of the competition on the developing set and test set are shown in Table 3.4. By comparing the results we get on the development

set and the test set regarding the F1 score, one can see that ReCRF causes mild changes in the result, which proves that ReCRF is more robust and resilient about data change.

3.5 Second proposed system: E2EJ

The main distinctive point between our model and the deep-learning literature is the consideration of the interaction between the two tasks of sensitivity detection and token type identification. In this subsection, we introduce E2EJ and its implementation steps in detail. Fig 3.5 depicts the architecture of our model.

3.5.1 Embedding layer

The goal of the embedding layer is to represent each word $w_i \in S$ as a vector $v_i \in R^d$ in a low-dimensional vector space. Here, d is the size of the embedding layer. We use two levels of embedding: word-level and character-level. For the word-level embedding, we replace w_i with its pre-trained Glove word embedding vector v_i^w (Pennington, Socher, and Manning, 2014). We use a single-layer 1-dimensional convolutional neural networks (Conv1D) with max-over-time pooling to represent the word at character level as the following. Suppose that w_i is made up of a sequence of characters $[c_1, c_2, ..., c_n]$, where n is the length of w_i. First, we pass the sequence of characters of the word w_i to a randomly initialized character embedding layer to get the matrix $C_i \in R^{r \times l}$ –that is the character-level representation of w_i. Here, the j-th column corresponds to the character embedding for c_j. After that, we apply a narrow convolution between C_i and a filter (or

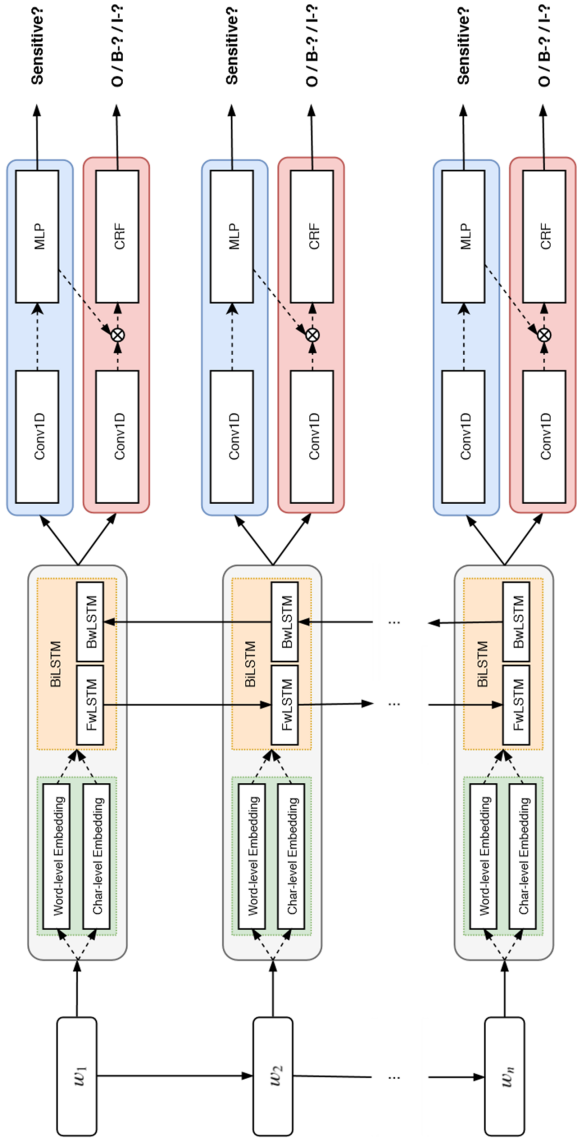


FIGURE 3.5: E2EJ architecture

kernel) $H \in R^{r \times k}$ of width k, after which we add a bias and apply a nonlinear transformation to obtain a feature map $f^i \in R^{n-k+1}$. Specifically, the m-th element of f^i is given by:

$$f^i[m] = tanh(\langle C_i[*, m : m + k - 1], H \rangle + b) \tag{3.1}$$

where $C_i[*, m : m + k - 1]$ is the m-to-$(m + w_1)$-th column of C_i and $\langle A, B \rangle$ is the Frobenius inner product. Finally, we take the max-over-time

$$v_i^c = max_m f^i[m] \tag{3.2}$$

as the feature corresponding to the filter H (when applied to word w_i). A filter basically consists in picking out a character n-gram, where the size of the n-gram corresponds to the filter width. The final representation of the word w_i is given by concatenating the word-level vector and the character-level vector:

$$v_i = [v_i^w; v_i^c]. \tag{3.3}$$

3.5.2 BiLSTM Layer

The goal of the encoder layer is to represent the sequence of word representations, $\{v_1, v_2, ..., v_l\}$, that is obtained from the embedding layer in the higher level of abstraction and model the sequential phenomena. In this work we use a Bi-RNN to design our encoder. A Bi-RNN consists of forward $\overrightarrow{\phi}$ and backward $\overleftarrow{\phi}$ recurrent neural networks. The first RNN reads the input sequence in a forward direction and produces a sequence of forward hidden states $(\overrightarrow{h_1}, ..., \overrightarrow{h_l})$, whereas the second RNN reads the sequence in

the reverse order $(v_{w_l}, ..., v_{w_1})$ resulting in a sequence of backward hidden states $(\overleftarrow{h_l}, ..., \overleftarrow{h_1})$.

We obtain a representation for each word v_{w_t} by concatenating the corresponding forward hidden state $\overrightarrow{h_t}$ and the backward one $\overleftarrow{h_t}$. The following equations formalize these ideas:

$$h_t = \overrightarrow{\phi}(v_{w_t}, \overrightarrow{h_{t-1}}) \tag{3.4}$$

$$h_t = \overleftarrow{\phi}(v_{w_t}, \overleftarrow{h_{t-1}}) \tag{3.5}$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{3.6}$$

In practice, RNNs are challenging to train. Gradients may explode or vanish over long sequences(Abril, Navarro-Arribas, and Torra, 2011). To overcome these problems, we use long short-term memory (LSTM) (Schmidhuber and Hochreiter, 1997) networks, that are a more sophisticated variant of regular RNNs.

3.5.3 Sensitivity detection sub-model

The input to this sub-model is the sequence of vectors obtained from the BiLSTM layer, and the output is the probability for each token to be sensitive. As shown in Fig. 3.5, the sub-model comprises two units: Conv1D with a single layer and a multi-layer perceptron (MLP) with one hidden layer and one Sigmoid neuron, *i.e.*, the output layer. The goal of the Conv1D layer is to enrich the representation of each token with information about a fixed-sized context depending on a kernel width of k. Formally, we get the final representation of the input sequences as follows:

$$v_1^2, v_2^2, ..., v_l^2 = Conv1D([v_1, v_2, ..., v_l]), \tag{3.7}$$

where Conv1D refers to the same operations in Equations (3.1) and (3.2), given that, for each, we obtain the final output as

$$X_t^s = \tanh(v_t^s \cdot W_1^s + b_1^s), \tag{3.8}$$

$$\bar{y}_t^s = sigmoid(x_t^s \cdot W_2^s + b_2^s). \tag{3.9}$$

3.5.4 NER type detection sub-model

Similarly, the input to this sub-model is the sequence of vectors obtained from the BiLSTM layer. The output, in this case, is the probability for each token to be sensitive. Formally, let $[v_1^t, v_2^t, ..., v_l^t]$ be the sequence of vectors to be labeled, which is produced by the concatenation of the MLP layer in the Sensitivity Detection sub-model and the output of the Conv1D layer in this sub-model; let $Y^t = [y_1^t, y_2^t, ..., y_l^t]$ be the corresponding tag sequence. Each element y_i^t of y is one of the $B- < entity >$, $I- < entity >$ or O tags. Both H and Y^t are assumed to be random variables, and they are jointly modeled using a conditional random field (CRF).

3.5.5 Training

We train our model to minimize the joint objective function J:

$$J = J_s + J_t, \tag{3.10}$$

where J_s is the sigmoid cross-entropy and J_t is the negative log-probability of the correct tag sequence

$$J_s = y_t^s \times -log(\bar(y)_t^s + (1 - y_t^s) \times -log(1 - \bar(y)_t^s), \tag{3.11}$$

$$J_t = -log(p(Y^t|H)), \tag{3.12}$$

with y_t^s being the golden label and $\bar{y}_t^s$ being the predicted label. The Y^t refers to the sequence of tags. As optimization algorithm, we used Stochastic Gradient Descent (SGD)-based ADAM algorithm (Da, 2014) with learning-rate 0.001. To avoid over-fitting, we used dropout on the embeddings and decoder outputs with a rate 0.3 (Srivastava et al., 2014).

3.5.6 Experiments

In this section, we discuss the data set we used and different experimental settings we devised to evaluate our systems.

Data set details

We trained and fine-tuned our systems respectively on the training and the development sets provided by the organizers of the MEDDOCAN challenge. After that, we submitted the predicted labels of the test set that are produced by our systems to evaluate their performance. The organizers omitted the golden labels of the test. The training set contains 500 documents, and the development and test sets contain 250 documents each.

TABLE 3.5: Hyper-parameter values chosen for the E2EJ system

WordEmbedding	Dimensionsize: 300 Initialization: Glove Trainable: No
CharEmbedding	Dimensionsize: 50 Conv1Dfilters: 100 Kernelwidth: 3 Initialization: Uniform$[-0.1, 0.1]$
BiLSTM	Hiddenunits: 256 Layers: 2
Sub-Model(1)	Conv1Dfilters: 200 Kernelwidth: 3 Hiddensize: 200
Sub-Model(2)	Conv1Dfilters: 200 Kernelwidth: 3

Hyper-parameters

We used grid-search to obtain the best hyper-parameter values based on the development set. We list these values in Table 3.5.

3.5.7 Results

We evaluated the performance of the E2EJ system by comparing it against the following baseline systems:

- RegEx: a rule-based system using only regular expressions.
- CRF: a CRF-based system trained on a set of features such as unigram, part-of-tags, word shape, affixes, etc. (Srivastava et al., 2014).

TABLE 3.6: Performance of E2EJ compared to various methods. The best value is in bold.

System	Sub-Task 1 (NER)			Sub-Task 2 (Spans)		
	P	R	F_1	P	R	F_1
RegEx	91.06	81.01	85.74	91.32	81.24	85.99
CRF	**97.02**	94.93	**95.96**	**97.47**	95.37	96.41
E2E	94.78	93.64	94.21	95.80	94.65	95.22
E2EJ	95.98	**95.69**	95.83	96.76	**96.45**	**96.61**

– E2E-LSTM: a version of our system that is trained to only identify the type of tokens.

Table 3.6 shows the results of our second system (*i.e.*, the E2EJ system) and the baseline systems. The evaluation metrics are precision, recall, and F1 scores. From the reported results, we can note that in general, E2EJ gives comparable performance to the state-of-the-art system CRF. It outperforms all the baseline systems in terms of the recall metric. One remarkable observation is that E2EJ, unlike the other systems, gives a similar performance in all the evaluation metrics, which shows its consistency. Hence, some error analysis and performance inspection can lead to improving the performance of the system. The CRF-based system gives the best performance in terms of precision and F1 metrics with the NER sub-task, and the best performance in terms of the precision score for the Spans detection sub-task. We attribute this to the use of the external MEDDOCAN-Gazetteer resources provided by the organizers of the task.

3.6 Competition results

TABLE 3.7: Competition final results for sub-task 1 (NER offset and entity type classification) and sub-task 2 (sensitive token detection (strict spans) on the test data set)

Team Name	Sub-Task 1 (NER)				Sub-Task 2 (Spans)			
	Team Rank	Precision	Recall	F1	Team Rank	Precision	Recall	F1
lukas.lange	1	96.98	96.94	96.96	1	97.51	97.47	97.49
Fadi (ReCRF)	2	96.99	95.67	96.33	2	97.53	96.20	96.86
nperez	3	96.40	95.64	96.02	3	97.19	96.41	96.80
FSL	4	95.86	96.04	95.95	5	96.31	96.50	96.41
mhjabreel (E2J2)	5	95.98	95.69	95.83	4	96.76	96.47	96.61
lsi uned	6	95.88	92.85	94.34	6	96.41	93.36	94.86
jiangdehuan	7	92.81	95.25	94.01	7	93.36	95.81	94.57
jimblair	8	96.45	91.20	93.75	9	96.78	91.91	94.28
ccolon	9	93.65	92.79	93.22	10	94.70	93.84	94.27
sohrab	10	95.68	90.69	93.12	11	96.09	91.08	93.52
Jordi	11	93.15	90.57	91.84	13	93.73	91.13	92.41
plubeda	12	92.11	88.71	90.38	8	96.17	92.62	94.36
m.domrachev	13	91.10	88.94	90.01	14	91.42	89.26	90.33
vcotik	15	91.41	88.01	89.68	12	94.77	91.24	92.97
VSP	16	85.54	86.49	86.01	16	86.55	87.51	87.03
gauku	17	90.84	58.17	70.92	17	91.42	58.54	71.38
Baseline-VT	-	37.02	50.34	42.67	-	44.17	50.63	47.18
Aspie96	18	18.83	52.96	27.78	18	19.77	55.61	29.17

Table 3.7 shows the final results of the MEDDOCAN 2019 competition. In the first sub-task, our two proposed systems ReCRF and E2J2 ranked second and fifth, respectively, while in the second sub-task they respectively ranked second and fourth.

3.7 Conclusion

Our first proposal is a hybrid system that automatically detects PHI entities from plain text medical documents. The system consists of an automatically constructed RegEx model and a trained

CRF model. The design of the system, which includes using a variety of linguistic and semantic features to increase the accuracy, ensures that it generalizes well in front of new data.

The second system we propose contains two sub-models that are trained jointly. The first one aims to detect the sensitive entities and guides the second sub-model to accurately predict the type of these detected tokens. E2EJ provides an end-to-end solution and does not require any external tools or other linguistic resources.

The effectiveness of the proposed systems has been evaluated by participating in the Medical Document Anonymization challenge for the electronic health records in Spanish language. In that contest, we have obtained results that compare favorably to the state-of-the-art systems and outperform the baseline systems. The reported results show that the proposed systems are stable and consistent.

The systems we have designed try to solve the problem of medical document anonymization from the NLP viewpoint. Even though these types of systems can detect private named entities in medical text like PHI with high accuracy, they are not generalizable in the other domains because of two reasons: 1) entities to be protected may not fall into NE categories, and 2) the content of the document to be protected may not only refer to the single entity to be protected. These limitations derive from the fact that NER-based system do not take the notion of disclosure risk and privacy requirements into account. The following chapter tries to overcome some of the limitations of NER-based methods by relying on a notion of disclosure risk that is more in line with SDC methods.

Chapter 4

Approaching document anonymization from an SDC perspective

There is a substantial amount of literature on SDC for the case of structured data (Hundepool et al., 2013; Drechsler, 2011; Domingo-Ferrer, Sánchez, and Soria-Comas, 2016). Structured data are those that can be described as a set of records each of which corresponds to an individual and contains the values of a fixed set of attributes for that individual. A common approach to anonymize structured data is to remove attributes that are identifiers and then mask quasi-identifier attributes. Alternatively, instead or in addition to masking quasi-identifiers, one can mask the confidential attributes, to introduce uncertainty about the confidential attribute values.

Once a decision has been made on which attributes are quasi-identifiers and which are confidential ones, anonymization of structured data can be fully automated. However, automation of unstructured data anonymization is much more difficult, because

there is no database schema that can be followed to classify the data into identifiers, quasi-identifiers and confidential attributes.

4.1 Contributions and plan of this chapter

The purpose of this chapter is to automate the extraction of quasi-identifier and/or confidential attributes from unstructured textual data by more formally adhering to the privacy notion of SDC methods. For the sake of concreteness, in this work the focus will be on medical diagnosis reports. Once this automatic identification of the relevant attributes is completed, we can apply some of the methods designed for anonymizing structured data. To identify attributes, we will take advantage of a named-entity recognition (NER) tagger (Finkel, Grenager, and Manning, 2005).

In Section 4.2, we describe our proposal. Experiments are presented in Section 4.3 and conclusions and future work ideas are gathered in Section 4.4.

4.2 Our approach

Formally, given a collection of text documents $D_1, \ldots, D_n$, we want to locate supersets of all the privacy-relevant attributes they contain. Specifically, we want to come up with a superset of identifier attributes $\mathcal{ID} = \{ID_1, \ldots, ID_p\}$, a superset of quasi-identifier attributes $\mathcal{QID} = \{QID_1, \ldots, QID_q\}$, and a superset of confidential attributes $\mathcal{C} = \{C_1, \ldots, C_r\}$. The set $\mathcal{ID}$ should contain the identifier attributes that appear in at least one of the documents; for example, $\mathcal{ID}$ will contain "Passport no." if at least one of the documents contains a passport number (even if the other

documents contain no passport number). Similarly, the set $\mathcal{QID}$ should contain the quasi-identifier attributes that appear in at least one document, and the set $\mathcal{C}$ the confidential attributes that appear in at least one document.

Once the above supersets have been determined, the collection of documents can be viewed as a *structured* data set with records $D_1, \ldots D_n$ and attributes that are the elements of $\mathcal{ID} \cup \mathcal{QID} \cup \mathcal{C}$. Obviously, this structured data set is likely to be a sparse one, as not all attributes take values in all documents. To anonymize this data set, we proceed as usual in the case of structured data sets. The values of attributes in $\mathcal{ID}$ should be suppressed from all records/documents and masking should be applied to attributes in $\mathcal{QID}$ and/or $\mathcal{C}$. Depending on the type of masking used, it may be necessary to deal first with the missing attribute values in some documents; imputing them by partial synthesis is a possibility (Drechsler, 2011; Hundepool et al., 2013).

Thus, the problem of anonymizing unstructured data reduces to locating the appearances of the various privacy-relevant attributes in the collection of documents and then anonymizing the resulting structured data set. We can tackle the task of locating attribute appearances by building several machine learning models, each of them recognizing a different type of named entity. For example, a first model to recognize identifier attributes (*e.g.* passport number, social security number, etc.), a second model to recognize quasi-identifier attributes (e.g. location, birth date, age, postal code, etc.), and a third model to recognize confidential attributes (e.g. disease names, etc.).

4.2.1 Proof of concept

As a proof of concept, we focus on locating confidential data within medical diagnoses. We propose a model based on conditional random fields to extract the disease names from a given medical record. For a given text, this model predicts a sequence of corresponding IOB2 tags.

Once we have the predicted sequence of IOB2 tags for every token in the medical record, we can interpret this sequence of labels and extract the “disease” entity. For instance, if we have the sentence "Retinopathy was assessed by ophthalmoscopy" and the corresponding IOB2 tags sequence {B-DIS, O, O, O}, we move through the IOB2 sequence tags and every word corresponding to a B-DIS label is considered as the beginning of a disease entity and every word corresponding to an I-DIS label is considered as being within a disease entity. Thus, a B-DIS word with all directly following I-DIS words forms one disease entity. In fact, B-DIS and I-DIS labels do the same job but B-DIS has the particular job of distinguishing between two consecutive disease entities.

Figure 4.1 shows the structure of the proposed model for disease name recognition. It consists of three steps:

- The first step is the tokenizer, which splits a sentence into tokens.
- The second step is the feature extractor; in this step, we use a window of three words (the current word, the previous word and the next word), and we extract the features of these words. Table 4.1 explains all the features we considered.

TABLE 4.1: Feature extraction

Feature	Explanation
Word stem	E.g. the stem of "illness" is "ill". We extract stems using SnowballStemmer from the nltk library (Bird and Klein, 2009).
Word length	The length of the word
Word shape	The shape of the word, which can be 'lowercase', 'uppercase', 'capitalized', 'mixed'
Word POS	Part of speech for the word. We use the Stanford POS tagger to extract this feature (Toutanova et al., 2003).

- The third step uses a CRF model, which takes the features from the second step and produces a sequence of tags for the whole sentence.

4.3 Experimental results

In this section we describe the experimental results of the above-mentioned proof of concept. We programmed the experiments in Python, and we used sklearn-crfsuite for CRF (*sklearn-crfsuite*) and SnowballStemmer for word stemming (Bird and Klein, 2009).

4.3.1 Data set

In our experiments, we took advantage of medical texts that were labeled to study the relation between diseases and treatments. These files were obtained from MEDLINE 2001 using the first 100 titles and the first 40 abstracts from the 59 files medline01n*.xml, that are available in (Rosario and Hearst, 2004).

These data contain 3,654 labeled sentences. The labels are: "DISONLY", "TREATONLY", "TREAT PREV", "DIS PREV", "TREAT

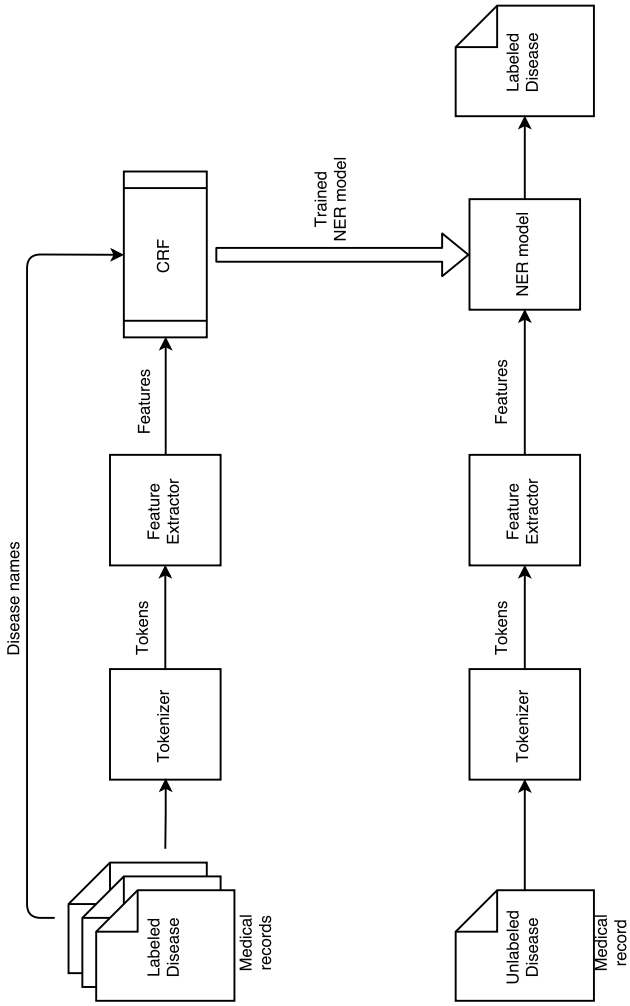


FIGURE 4.1: Architecture of the named-entity recognition tagger

SIDE EFF", "DIS SIDE EFF", "DIS VAG", "TREAT VAG", "TREAT NO" and "DIS NO". As we were only interested in diseases, we only kept the 629 sentences with the "DISONLY" labels.

4.3.2 Evaluation metrics

We used three metrics to evaluate the performance of the proposed model for the recognition of diseases:

- *Precision*. Number of diseases correctly identified by the classifier divided by the total number of identified diseases:

$$\text{Precision} = \frac{|S \cap G|}{|S|},$$

  where S is the set of all diseases identified by the classifier and G is the set of correct diseases according to the original dataset.

- *Recall*. Number of diseases correctly identified by the classifier divided by the number of correct diseases in the original dataset:

$$\text{Recall} = \frac{|S \cap G|}{|G|}.$$

- *F1*. Harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

TABLE 4.2: Evaluation of the model on the test dataset at word level

	Precision	Recall	F1-score
B-DIS	0.766	0.677	0.719
I-DIS	0.789	0.709	0.747
avg / total	0.778	0.693	0.733

4.3.3 Results and discussion

We did the experimental evaluation in two phases: model training and model testing. Out of the 629 samples of labeled sentences, 503 were devoted to model training (80% of the samples), and 126 to model testing (20% of the samples).

The training phase was performed via 10-fold cross-validation, as follows. We partitioned the training data set into 10 equal-size subsamples. Out of the 10 subsamples, a single subsample was retained as validation data for testing the model while in the training phase, and the remaining 9 subsamples were used in training.

While most words in the data set were labeled as O (outside disease), we were interested in words labeled as B-DIS (beginning of disease) and I-DIS (in disease). Thus, we computed the precision, the recall and the F1 score only for B-DIS and I-DIS. For example, if we have the sentence "Diagnostic evaluation of the patient with high blood pressure", its word tokens are {"Diagnostic", "evaluation", "of", "the", "patient", "with", "high", "blood", "pressure"} and the corresponding labels are {O, O, O, O, O, O, B-DIS, I-DIS, I-DIS}. The named entity here contains three words

TABLE 4.3: Evaluation the model on the test dataset at entity level

	Precision	Recall	F1-score
Disease Entity	0.742	0.660	0.698

"high blood pressure". Table 4.2 shows the evaluation of the predicted tags against the correct tags at the word level (separately for each word). In contrast, Table 4.3 reports the same evaluation metrics for whole entities. That is, in the previous example, Table 4.2 would separately refer to the three words "high", "blood" and "pressure", while Table 4.3 would refer to the entity "high blood pressure"; in the latter case, unless *all three* words of the entity were correctly labeled, the whole entity would be considered as misclassified.

According to Table 4.3, our model performed significantly better regarding the precision than regarding the recall. It is very likely that the recall can be increased by using more training samples.

We consider the above results to be promising, as they are quite close to the manual labeling of the data we used.

4.4 Conclusion

In this chapter, we have dealt with the anonymization of unstructured textual data from the viewpoint of SDC. As a proof of concept, we have focused on locating disease names (*i.e.* sensitive

attributes) in medical records. Once located, these sensitive attributes can be protected using common SDC techniques for structured data.

Even though this approach enforces a more formal notion of anonymization than plain NER-based methods, it is only able to protect homogeneous collections of documents (*i.e.* datasets contain a set of documents describing similar entities, e.g. medical records describing patient history). In the next chapter we extend this approach to provide a more general and flexible approach to anonymize text documents independently and in a more robust way.

Chapter 5

Utility-preserving protection of documents via word embeddings

As discussed in previous chapters, NER-based techniques are the current solution to deal with textual document anonymization. However, NER-based techniques have important limitations. First, for the more sophisticated NER techniques one needs to train the classifiers, and this requires a large amount of manually tagged training data that match the language of the text to be protected. Tagging a sufficient volume of training data may become a considerable effort. Second, NER-based methods are unable to discern whether the pinpointed entities refer to the individual to be protected or not. Hence, systematic masking (for example suppression) of those entities often degrades the text semantics (and therefore its utility) without a corresponding reduction of risk. Finally, while NER systems detect a fixed set of entity types, there are unlimited ways of referring to (quasi-)identifying information

in a text. As a result, NER-based methods are usually characterized by a low detection recall, which yields poorly protected outcomes.

5.1 Contributions and plan of this chapter

In this chapter, we overcome the above-mentioned limitations of NER-based techniques by proposing a more general and flexible method that better captures the notion of disclosure risk as understood in the literature on data privacy (Westin, 1967). We characterize the semantic relationships between the textual entities appearing in a document by leveraging word embeddings (Mikolov et al., 2013a). Word embeddings learn detailed vector representations of linguistic terms that convey the semantics of such terms. We make use of these vectors to measure the semantic relatedness and, from it, the extent to which the terms appearing in the text document disclose the entity to be protected. The latter can be either an individual's identity (*e.g.*, a name) or a confidential attribute (*e.g.*, a sensitive disease). Thus, our method naturally encompasses the notions of identity and attribute disclosure, and it automatically classifies the textual terms as being disclosive or not disclosive; that is, it automatically determines which terms act as (quasi-)identifiers of the entity to be protected. This delivers a more powerful solution to protect textual documents than NER-based methods, because our solution is not restricted to detecting predefined (quasi-)identifying types (*e.g.*, names or locations) and it can limit the protection only to the terms referring to the entity to be protected, whatever it is. The empirical results we report

show that our method offers more robust protection than NER-based approaches. Regarding ease of use and deployment, our solution is mostly *language-agnostic* and does *not require manual tagging* of training data.

Beyond accurately meeting the privacy requirements, we also improve the masking of quasi-identifying terms in order to increase utility preservation. In contrast to approaches that simply suppress quasi-identifying terms, we propose a generalization-based masking procedure that preserves their underlying semantics as much as allowed by the privacy requirements, which state the maximum level of allowed disclosure. To this end, we rely on the taxonomies contained in structured knowledge bases, like ontologies, which model the domains to which the entities appearing in the document belong. As we also show in the empirical work, with this approach we significantly reduce the information loss incurred by data masking in comparison with approaches based on data removal or NER-based classification of entities.

The rest of this chapter is organized as follows. In Section 5.2 we review related concepts from NLP. In Section 5.3 we present our approach to document protection based on word embeddings. Section 5.4 contains an empirical evaluation of our method and a comparison against related contributions. In Section 5.5 we discuss several practical applications of our method that sustain its generality. Finally, Section 5.6 concludes the chapter.

5.2 Background on word embeddings and ontologies

In this section, we provide an overview of the relevant concepts from the literature of NLP on word embeddings.

5.2.1 Word representation

A word in the text can be represented as a sequence of characters. However, this representation is difficult to be used directly in most essential NLP tasks. Not long ago, words were represented as integers (*i.e.*, each word in a corpus was represented by a unique positive integer number). This representation had several advantages, like less memory consumption because every word was stored in the same amount of memory. The main disadvantage of this representation that these integers themselves did not mean anything; word similarity could not be calculated, and semantic relationships between words could not be extracted.

Treating a word as a single integer number is not sufficient to represent the semantic relationship between the words, so it is useful to consider words as vectors. Word vectorization is another way to represent words; usually, these vectors can be integers or real numbers. The simplest word vectorization method is called one-hot representation. In this method, each word is represented as a vector with all 0's and one 1 at the position of that word in the sorted vocabulary. The one-hot vector representation method treats every word as equidistant from every other. However, the relationship among words cannot be inferred, and usually, this representation leads to data sparsity, see Figure 5.1.

More recently, word embedding was proposed to overcome some of the previous drawbacks. With word embeddings, words are represented as real-valued vectors that encode the word meanings in such a way that words with similar meanings should have similar representations in the vector space.

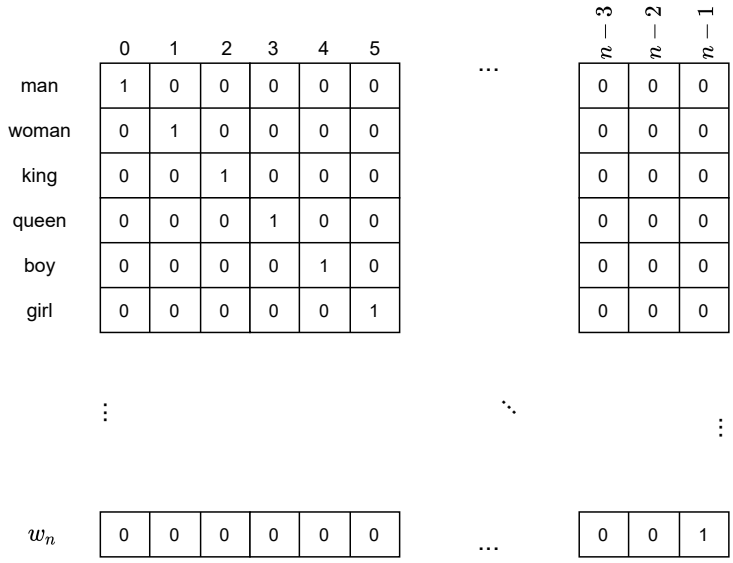


FIGURE 5.1: Example of the one-hot encoded sparse matrix

Word embeddings

Word embeddings map words into high-dimensional numerical vectors capturing their semantics (see Fig. 5.2).

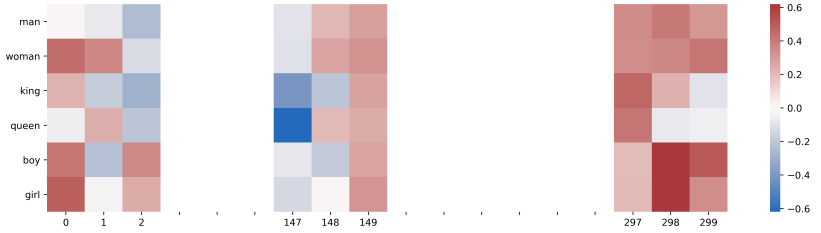


FIGURE 5.2: Visualization of word embedding representations

Word embedding models can be categorized into two main types: static embedding and dynamic/contextual embedding. Models like word2vec (Mikolov et al., 2013b), fastText (Bojanowski et al., 2017) and GloVe (Pennington, Socher, and Manning, 2014) are static and context-independent: they build vector representations of words that do not depend on the context in which words appear. Word2vec uses a neural network (Mikolov et al., 2013b) trained either to predict the current word from a window of neighboring words (continuous bag-of-words model) or to predict neighboring words based on the current word (skip-gram model). FastText (Bojanowski et al., 2017) also works on the same idea, but the main difference is that fastText takes care of the out-of-vocabulary (OOV) problem for n-grams (*i.e.*, the inability to characterize an n-gram because it was not found in the training data). FastText mitigates this problem by taking into consideration the subword information through embedding of subword n-grams. Finally,

GloVe (Pennington, Socher, and Manning, 2014) uses two methods to generate word representations: local context window information and aggregated global word-word co-occurrence statistics from the pre-training corpus. Unlike word2vec and fastText, GloVe does not rely just on local context window information, but incorporates global statistics to obtain a more accurate word representation.

However, the current state of the art is contextual/dynamic embedding with models like BERT (Devlin et al., 2018) or ELMo (Peters et al., 2018). These models are built using transformer-based self-supervised architectures that are pre-trained for language understanding. The key idea of these models is that the pre-training task is designed to be a generic form that can be tailored to solve any specific problem in NLP. Pre-training can be performed by using masked language models (MLM) or next-sentence prediction (NSP). MLM randomly masks some of the tokens from the input and the objective is to predict the original words, whereas NSP consists in identifying consecutive sentences. Both tasks aim at steering the model into taking the context of a word into consideration.

5.2.2 Ontologies

An ontology is a structured knowledge source that explicitly and consensually represents the concepts and the semantic interrelations of a domain of knowledge (Guarino, 1998). According to the formal definition proposed in (Wu and Palmer, 1994), an ontology O is composed of a set of concepts or classes C, and a set of relation types R. The set of concepts represents the real-world

entities of the area of knowledge being modeled. For example, in a medical ontology, the concepts can be types of diseases, medical procedures or clinical findings; *i.e.*, single units of thought with a distinct clinical meaning. R represents types of semantic relations between concepts, such as taxonomic relationships, *e.g.*, hyponymy and hypernymy (is-a links), and non-taxonomic relationships, *e.g.*, meronymy and holonymy (part-of links).

Taxonomies are the backbone of ontologies. Taxonomies are organized by supertype-subtype relationships, also called generalization-specialization relationships, or less formally, parent-child relationships. Once a taxonomy tree has been created, all the items in the tree are tagged as belonging to one or more specific taxonomy categories. See Fig. 5.3 for an example of taxonomy.

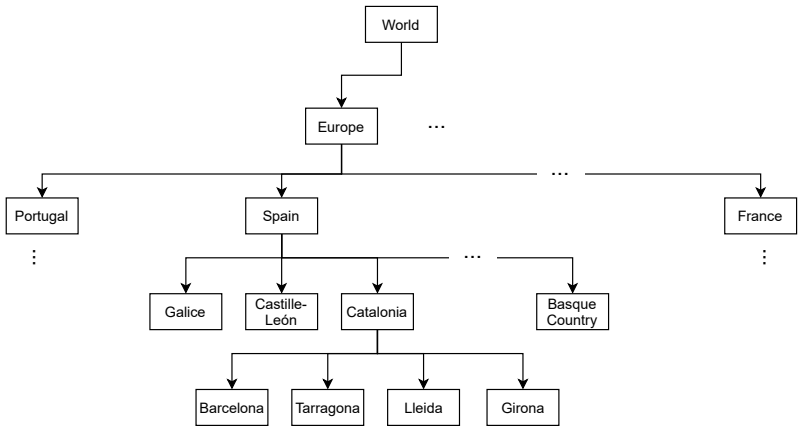


FIGURE 5.3: Taxonomy example

Types of ontologies

As it is proposed by (Guarino, 1998), ontologies can be classified according to their level of dependence on a particular task or point of view:

- *Top-level ontologies*. They describe general concepts which are independent of a particular problem or domain. Examples of top-level ontologies are WordNet (Fellbaum, 1998) or Yago (Suchanek, Kasneci, and Weikum, 2007), which try to model knowledge of the world.

- *Domain-ontologies*. They describe the vocabulary related to a generic domain by specializing the concepts introduced in the top-level ontologies. An example of domain ontology is SNOMED-CT (Spackman, 2004), which models biomedical knowledge.

- *Task ontologies*. They describe the vocabulary related to a generic task or activity by specializing the top-level ontologies.

- *Application ontologies*. These are the most specific ones. Concepts often correspond to roles played by domain entities. They have limited reusability as they depend on the particular scope and requirements of a specific application. Those ontologies are typically developed *ad hoc* by the application designers.

We briefly explain the two ontologies we use in this work in the following sub-sections.

WordNet

WordNet (Fellbaum, 1998) is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each one expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations.

WordNet superficially resembles a thesaurus, in that it groups words together based on their meanings. However, there are some important distinctions. First, WordNet interlinks not just word forms—strings of letters—but specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the grouping of words in a thesaurus does not follow any explicit pattern other than meaning similarity.

YAGO

Yet Another Great Ontology (YAGO) (Rebele et al., 2016) is a large knowledge base that is built automatically from Wikipedia, WordNet (Fellbaum, 1998) and GeoNames (*The GeoNames geographical database covers all countries* 2006). YAGO contains general knowledge about the real world. It contains both entities (such as movies, people, cities, countries, etc.) and facts about these entities (who played in which movie, which city is located in which country, etc.). Overall, YAGO contains about 10 million entities and 120 million facts.

5.3 Our approach

The most widely accepted definition of privacy rests on the notion of informational self-determination, that is, "the claim of individuals, groups or organizations to determine for themselves when, how, and to what extent information about them is communicated to others" (Westin, 1967). Following this definition, the crux of protecting data releases is the ability to detect (and subsequently remove or mask) the information that refers to a single entity and to no other entity. In other words, protecting one entity should not encroach on how another entity is protected. This is exactly the goal that our approach sets out to achieve. As discussed in the previous section, approaches based on NER fail in this respect because they implicitly assume the entire content of each document refers to a single entity.

To reach our goal, we need a way to characterize the textual terms according to the information they disclose on the entity to be protected. A common metric to quantify this "amount of disclosure" is the semantic relatedness between the terms in the document and the entity to be protected (Sánchez and Batet, 2016). Traditionally, the semantic relatedness between linguistic entities has been assessed using distributional (Mohammad and Hirst, 2012) or probabilistic models (Sánchez et al., 2010), which require accurate statistics on the (co-)occurrence of words. A recent trend in computational linguistics to measure the relatedness between words is to use word embedding models.

From the perspective of *distributional semantics*, words that are

likely to co-occur in a context (or, otherwise put, those with similar contexts) tend to be semantically related (Sahlgren, 2008). Therefore, after training a word embedding model with a collection of word contexts, semantically related words will have similar word embedding vectors.

The distributional semantics captured by word embedding models also encompasses a very broad notion of relatedness. Moreover, the larger the amount of data used, the more general the resulting distributional model (Boleda, 2020); in fact, word embedding models owe their success to the massive amount of data they use for training. On the other hand, a strong semantic relatedness between the words appearing in a text and the entity to be protected is what enables the semantic inferences that may lead to disclosure (Sánchez and Batet, 2016; Anandan and Clifton, 2011). Therefore, we propose to measure the disclosure risk caused by each term appearing in a document w.r.t. an entity to be protected as a function of the similarity between their word embedding vectors.

Our approach consists of three phases. In the first phase, we use a large corpus to train a word embedding model tailored to capture the semantic relationships that may cause disclosure. The trained model has learned the relationships (and, therefore, the pairwise disclosure risks) between all the terms appearing in the collection of documents. In the second phase, for a given document D, an entity to be protected e and a threshold t stating the maximum level of allowed disclosure, we use the trained model to detect the terms in D that may act as (quasi-)identifiers of e. Both e and t define the privacy requirements. In the third phase we mask the quasi-identifying terms we detected in the second

phase. Masking is performed by replacing those terms by generalizations extracted from structured knowledge bases modeling the concepts of the domain. The generalizations are picked so that they are the most specific ones that are 'safe' (*i.e.*, non-disclosive enough) according to the risk criterion employed in the second phase. In this way, we protect privacy while retaining the semantics (and, therefore, the utility and readability) of document D as much as possible.

5.3.1 Training the model

The first phase of our method is depicted in Figure 5.4. It consists of the following steps, which are explained further below:

- Data collection and pre-processing;
- Model training.

To train a word embedding model that accurately characterizes the disclosure-enabling semantic relationships affecting an entity or a set of entities, we need a representative "core" corpus of documents that describe those entities.

Ideally, the corpus ought to contain all the documents that shall be protected (for instance, a collection of medical records). In this way we ensure that all the terms in such documents appear in the model's vocabulary and get accurate vectors. If this "core" corpus is small, the collection of documents can be expanded with more general corpora that will provide additional evidences on the co-distribution of words and thereby mitigate the data scarcity. An alternative would be to use an embedding

model pre-trained on large corpora (such as BERT) and fine-tune it with the corpus of documents to be protected.

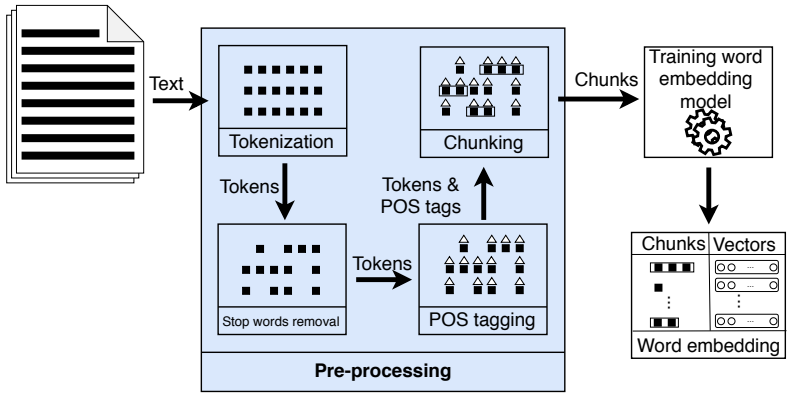


FIGURE 5.4: Overview of the training phase

Since we aim at protecting entities and semantic relationships occur at a conceptual level (rather than at a word level), we introduce a pre-processing step to create a meaningful vocabulary of concepts (rather than isolated words) for the word embedding model.

Specifically, concepts and entities are referred to in a text via noun phrases. For example, the noun phrase "New York Times" refers to a sole specific entity that is completely different from the individual meaning of its words "New", "York" and "Times". To properly evaluate disclosure risks, we need the vector representation of the *concepts* referred to by the text (*e.g.*, "New York Times"), rather than the representations of isolated words. For this purpose, in the pre-processing step we extract the noun phrases (or n-grams) and feed them as atomic elements to the word embedding model for training.

Even though some word embedding models, such as fastText, can construct representations of unseen noun phrases by combining representations of constituent subwords, the approach described in the previous paragraph yields more accurate vectors.

As shown in Figure 5.4, the pre-processing step consists of a pipeline of syntactic analyses: tokenization, part-of-speech tagging and chunking (Morton et al., 2005). As a result, noun, verb and prepositional phrases are obtained. Also, to minimize the lexical variability of the noun phrases, stop words are removed during the tokenization step; in this way, the occurrences of n-grams like "the New York Times" and "New York Times" will contribute to the same vocabulary entry/word vector.

In addition to improving the characterization of the entities appearing in the documents, this pre-processing helps reduce the size of the vocabulary and, therefore, the training runtime.

Our method is not tied to a particular embedding model: it only needs accurate and exhaustive vector representations of all the n-grams appearing in the documents to be protected. In the sequel we illustrate on word2vec (Mikolov et al., 2013b) the training process of a word embedding model tailored to our needs, even though, as we show in the evaluation section, other embedding models can be employed. Word2vec can be trained either to predict the current word from a window of neighboring words (continuous bag-of-words model) or to predict neighboring words based on the current word (skip-gram model). To this end, the neural network uses a collection of sentences as training data, and builds a vocabulary with the words appearing in the collection. The weights of the hidden layer of neurons that results from training the neural network for each word in the vocabulary

are used as the vector associated with that word. In this way, the number of neurons in the hidden layer (which can be configured), corresponds to the dimensionality of the vectors.

Regarding the learning model, the skip-gram model yields more accurate vectors (Mikolov et al., 2013b). More specifically, it uses as input a binary vector in which each position corresponds to a word w_i in the vocabulary V. The position of the current word (w_c) is set to '1', whereas the remaining positions are set to '0'. The output layer is a Softmax classifier with as many neurons as words in the vocabulary, where the i-th neuron provides the probability that the word at a randomly chosen nearby position of the current word w_c is w_i. The neural network is trained with nearby word pairs from the input collection of sentences. Context windows are employed to restrict the neighborhood of words that are considered to be nearby and to build the training samples. Once the training is complete for w_c, the weights of the hidden layer of neurons —which embed the tendency to co-occur between w_c and all the other words in the vocabulary— are employed as the vector representation of w_c.

The training of the skip-gram model depends on several configuration parameters. In what follows we discuss such parameters and argue which values are appropriate in the context of document protection.

As said above, the skip-gram model predicts the probability that words in the vocabulary appear in the neighborhood of the input word. To this end, it uses training samples of word pairs that co-occur within a context. This context (and, therefore, the co-occurring word pairs) can be restricted according to a *window size*. The window size is usually set to encompass complete sentences,

say between 5 and 10 words each, because words appearing in the same sentence are assumed to be closely related. Larger window sizes require more iterations, because more word pairs are evaluated during the learning process; as a matter of fact, doubling the window size increases the training runtime by around 50%. We also set the window size to include sentences but considering that our linguistic units are n-grams rather than isolated words.

Another relevant parameter is the *dimensionality of vectors*. In principle, the greater the dimensionality, the more accurate the results, because the adaptability of the model is proportional to the vector size. However, since the dimensionality is equal to the number of neurons in the hidden layer of the network, a greater dimensionality significantly increases the training runtime. Again, doubling the size of the vectors implies increasing the runtime by around 50%. Even though there is no fixed rule to tune the dimensionality, a value 300 is suggested in (Mikolov et al., 2013b) because larger values do not significantly improve accuracy.

Finally, it is possible to set a minimum number of appearances as a *cutting threshold* below which words in the input collection of documents will not be used for training. Since word embedding is usually employed to guide general semantic similarity assessments, it makes sense to discard words that occur too rarely because the evidences of co-occurrence they provide are too weak to derive robust statistics. Moreover, filtering out outliers significantly reduces the vocabulary size and, therefore, the training runtime. However, in the context of document protection rare words (such as names or particular addresses, which may appear only once together with the entity they refer to) are usually those that entail the greatest risk because they often refer to

very specific (quasi-)identifying information (Sánchez, Batet, and Viejo, 2013a). For this reason, we do not use any cutting threshold for rare words. For such words, the model may learn a strongly biased relationship w.r.t. the entity to be protected, which is the only one they co-occur with in the training data. This is however beneficial from the point of view of privacy protection, because in this way these rare words will be characterized for sure as quasi-identifying terms.

Training a word embedding model on a large collection of documents can be costly. Nonetheless, once trained, the model can be efficiently reused to protect any number of documents as long as their content is covered by the vocabulary of the trained model. Also, in the event that a new document to be protected contains terms that are not in the vocabulary, the previously trained model can be efficiently updated (without re-initializing the training) with new vocabulary entries via vocabulary expansion techniques (Kiros et al., 2015).

5.3.2 Detecting quasi-identifying terms

Once the model is built, we obtain a vector representation of each phrase in the input collection of documents. If two n-grams had similar contexts in the training data (and, therefore, are semantically related (Sahlgren, 2008)) they will also have similar vectors. The standard way of measuring the similarity between vectors is the cosine similarity. We employ this similarity to assess how disclosive/similar are the terms in a document w.r.t. the entity to be protected and, in this way, we detect those terms that act as quasi-identifiers of that entity.

The second phase of our method is depicted in Figure 5.5. Given a document D and a particular entity e whose privacy is to be protected (where e can be an identity or a confidential value), we iteratively evaluate how disclosive about e each phrase p_i in D is. We do this by measuring the cosine similarity between the vector representations of p_i and e, which we denote by $sim(v(p_i), v(e))$. Prior to that, we pre-process D as described in Section 5.3.1, so that the contents of D are evaluated at a conceptual level rather than at a word level. If the similarity of a certain p_i is above a threshold t, then p_i is deemed a quasi-identifier and will undergo masking in the third phase. Thus, t defines the maximum level of tolerated disclosure for the (masked) terms appearing in the protected output, and it allows balancing the trade-off between disclosure protection and utility preservation. As it happens with other generalization-based methods (Chakaravarthy et al., 2008; Cumby and Ghani, 2011; Anandan et al., 2012), which also rely on privacy/utility thresholds, the specific value of t should be set according to the needs of the application scenario: higher values for better protection (and less utility) or lower values for better utility (and less protection).

5.3.3 Masking quasi-identifying terms

As discussed in Chapter 2, different strategies can be employed to mask quasi-identifiers, of which the most common are suppression and generalization. The former strategy is straightforward and is usually employed in document redaction (Sánchez, Batet, and Viejo, 2013b). On the other hand, term generalization,

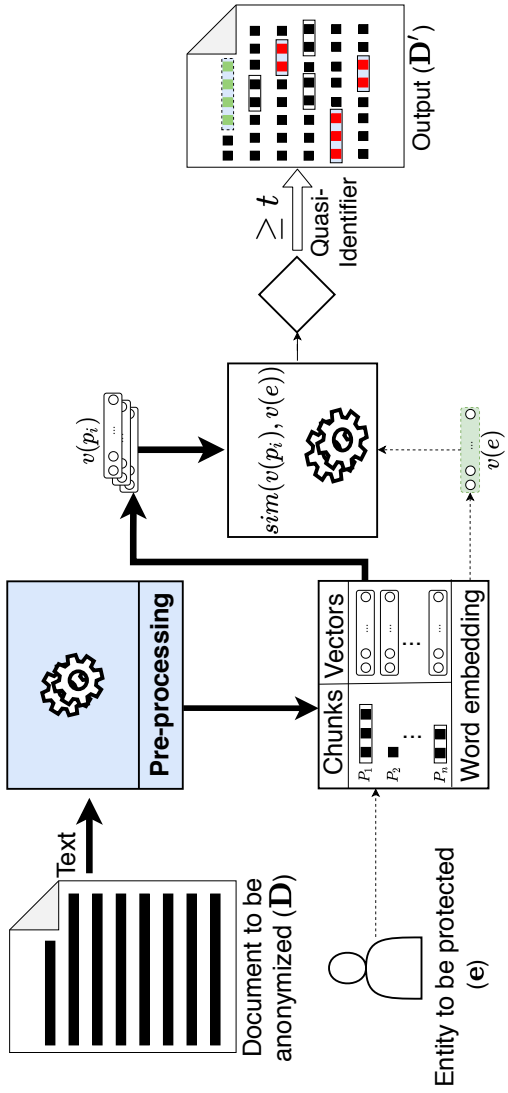


FIGURE 5.5: Overview of the detection phase

which consists in replacing specific terms by less detailed generalizations, does a better job at preserving the semantics and the readability of the protected document. Since by definition generalizations encompass a subset of the semantics of their respective specializations, generalization-based replacements preserve a subset of the original document semantics.

Generalizing requires detailed taxonomies from which suitable generalizations of disclosive terms can be obtained. Taxonomies suitable for non-specialized text can be obtained from general-purpose ontologies, such as WordNet (Fellbaum, 1998) or YAGO (Suchanek, Kasneci, and Weikum, 2007). More specifically, WordNet models the semantic relationships between 175,979 concepts, which are taxonomically organized under the common abstraction "entity". YAGO enriches WordNet's taxonomy by adding Wikipedia categories and articles; as a result, YAGO includes more than 10 million entities. These knowledge bases can be expected to cover most of the entities appearing in text. For specialized documents such as medical records, domain-specific knowledge bases can be used; for example, SNOMED-CT (Spackman, 2004) models more than 311,000 clinical terms within several taxonomies.

Masking quasi-identifying terms is performed as follows. For each quasi-identifying phrase s_i detected in the second phase, we obtain an ordered set of generalizations $G(s_i)$ from an ontology by matching s_i to concept labels in the ontology. If s_i matches more than one concept (due to its being polysemic), we map it to its most probable sense/concept, based on the probability of occurrence available in the sources we use for generalization. If s_i is not found in the ontology, we look for simpler forms of the noun phrase by iteratively removing adjectives and nouns starting with

the leftmost ones (*e.g.*, "metastatic breast cancer" $\rightarrow$ "breast cancer" $\rightarrow$ "cancer"). These simpler forms of s_i are also added in the first positions of $G(s_i)$ because they are actual generalizations of s_i. In this way, $G(s_i)$ contains generalizations of s_i ordered from most specific to most general. If s_i is a very specific entity, such as the name of an individual, we may not find it in any ontology. In this case, we use the most abstract concept in the ontology (*e.g.*, "entity") as its generalization.

As shown in Figure 5.6, the most suitable generalization g_i to mask s_i is the most specific generalization in $G(s_i)$ such that $sim(v(g_i), v(e)) < t$, that is, the first generalization in $G(s_i)$ that brings the disclosure on e below threshold t. To calculate $sim(v(g_i), v(e))$ we also need the vectors corresponding to all the generalizations of all s_i. Since g_i are generalizations of s_i, they are likely to have already appeared in the input collection of documents, in which case $v(g_i)$ has already been calculated. Otherwise, we need to update the model by feeding new documents that contain the missing g_i. Training data could be Wikipedia articles covering g_i, which are already associated with concepts in ontologies such as YAGO, and are a common training source of general-purpose embedding models (Bojanowski et al., 2017). As discussed in Section 5.3.1, to efficiently re-train the model with new samples (and vocabulary elements) we can use vocabulary expansion techniques. If the amount of documents and entities to be protected is large, a more efficient approach would be to first train a complete model covering all the entities contained in the ontology by using, *e.g.*, their corresponding Wikipedia articles as training data, and second, to expand the model by considering the specific contents of the documents to be protected. In this way, we

ensure that all possible generalizations are already covered by the model when we reach the masking phase.

5.4 Evaluation

In this section we report a performance evaluation of our approach from three perspectives: (i) the accuracy of the detection of quasi-identifying terms, (ii) the utility preserved by the protected document after masking such terms, and (iii) the effectiveness of the protection against a simulated re-identification attack. We have evaluated our method under different conditions and we have compared our results against several tools based on named entity recognition.

5.4.1 Detection phase

Our evaluation considered a scenario similar to that used in related works on document protection (Sánchez, Batet, and Viejo, 2013b; Sánchez, Batet, and Viejo, 2013a; Chow, Golle, and Staddon, 2008). In these works, the evaluation data consist of a set of Wikipedia articles describing real-world entities of different domains. Our goal was to protect each article so that the outcome did not unequivocally disclose the entity described by the article. To obtain the ground truth, we manually examined the contents of the articles to identify the terms that might disclose the described entity. Wikipedia articles were used because of their high informativeness and tight discourses, which constitute a challenging scenario for document protection.

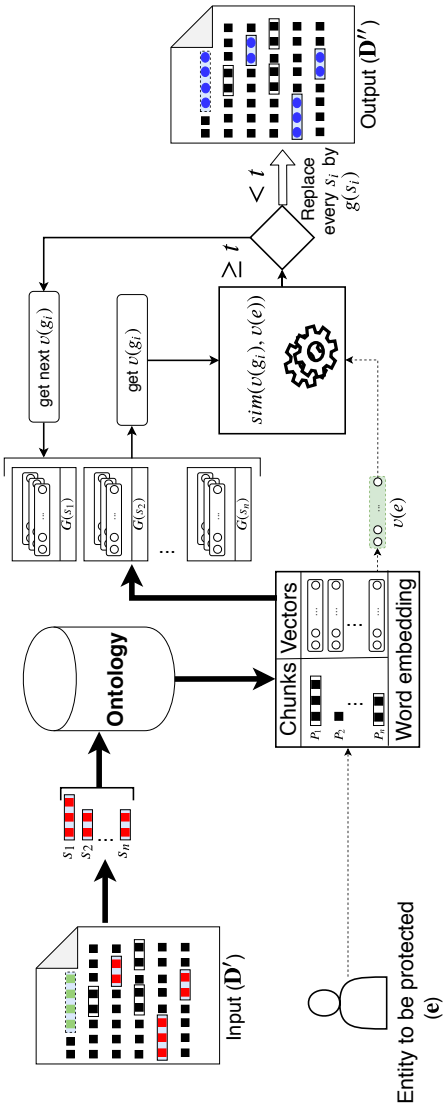


FIGURE 5.6: Overview of the masking phase

More specifically, we used a collection of English Wikipedia articles corresponding to movie actors from several countries. First, we collected the abstracts of 19,000 articles under the "20^{th} century actors" Wikipedia category. These were used to train the word embedding model as detailed in Section 5.3.1. The model was built using word2vec (Mikolov et al., 2013a). Training was configured with the parameters discussed in Section 5.3.1: window size 10, vector dimension 300 and no filtering of rare words.

As an evaluation test bed, we randomly picked 50 summaries from the collection and we tagged them manually to identify words and n-grams that might disclose the actor's identity. We used the following annotation guidelines, which are inspired by how (quasi-)identifying attributes are selected in structured databases (Hundepool et al., 2013):

- *Identifiers*: any information that can directly and unequivocally identify an individual. This includes the actor's name and also direct family members such as father, mother, brothers, children, husband/wife, etc. We also considered the movie characters' name he/she have played.

- *Quasi-identifiers*: publicly available information that, in isolation, does not identify the individual but whose combination may. There is an unbounded number of information types that may act as quasi-identifiers, but they mainly boil down to demographic and spatiotemporal attributes such as age, date of birth, place of living, received awards, names and dates of the movies he/she has started, etc.

The annotation was independently carried out by the author of this Ph.D. thesis and his two supervisors. A final annotation

was thereafter agreed upon via majority voting. The inter-annotator agreement among the three of us was Fleiss' kappa = 0.869, which shows a very strong agreement. As a result of the annotation, 2,655 words or around 30% of the content were tagged.

The evaluation metrics we employed were the standard precision, recall and F_1-score measures, which we next summarize. Precision is defined as

$$Precision = \frac{\#detected\ tagged\ terms}{\#detected\ terms},$$

where *detected terms* is the set of terms detected as quasi-identifiers through the process detailed in Section 5.3.2 and *detected tagged terms* is the subset of terms detected as quasi-identifiers that contain one or more tagged words. The higher the precision, the lower the number of false positives, that is, of over-protected terms. A high precision implies that the document's semantics and readability, that is, its utility, are better preserved by the protection process. Regarding recall, it is defined as

$$Recall = \frac{\#detected\ tagged\ terms}{\#tagged\ terms},$$

where *tagged terms* is the set of terms manually tagged as quasi-identifiers. The higher the recall, the more robust the protection, because a greater amount of (quasi)-identifiers have been correctly detected. Finally, the F_1-score is defined as

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall},$$

which corresponds to the harmonic mean of precision and recall and can be viewed as a performance summary of the detection

phase when the same weight is given to precision and recall. Notice that, even though a high precision is always positive, a high recall is usually more desirable because undetected quasi-identifiers may render the protection useless.

We empirically set the similarity threshold t employed to detect quasi-identifiers so that the F_1-score was maximized on average across the evaluated documents. The selected value was $t = 0.25$. Notice that, rather than being a hyperparameter to be optimized, the threshold t is a privacy requirement, *i.e.*, it allows tailoring the privacy/utility trade-off, and its value can be set by the user according to his/her protection needs. We show in Figure 5.7 how the threshold influences precision/recall/F_1 for values within the $[0.01, \ldots, 1]$ range. We can see that different values yield different balances between protection (recall) and utility preservation (precision), with $t = 0.25$ offering the best balance.

We evaluated two versions of our method: the first one included the pre-processing detailed in Section 5.3.1 whereas the second one did not. In the first version the model vocabulary consisted of 651,835 n-grams, whereas in the second version it comprised 1,084,189 individual words. Model learning took 177 seconds with the first version and 232 seconds with the second version, in both cases on an AMD Athlon X4 860K CPU with 24GB RAM. Notice that document pre-processing is the only phase of our method that is language-dependent. Therefore, by measuring the influence of the linguistic pre-processing on the results, we were able to quantify the benefits brought by this additional analysis and the penalty incurred if the linguistic tools required to analyze a (minority) language were not available.

We then compared the evaluation figures obtained by our method

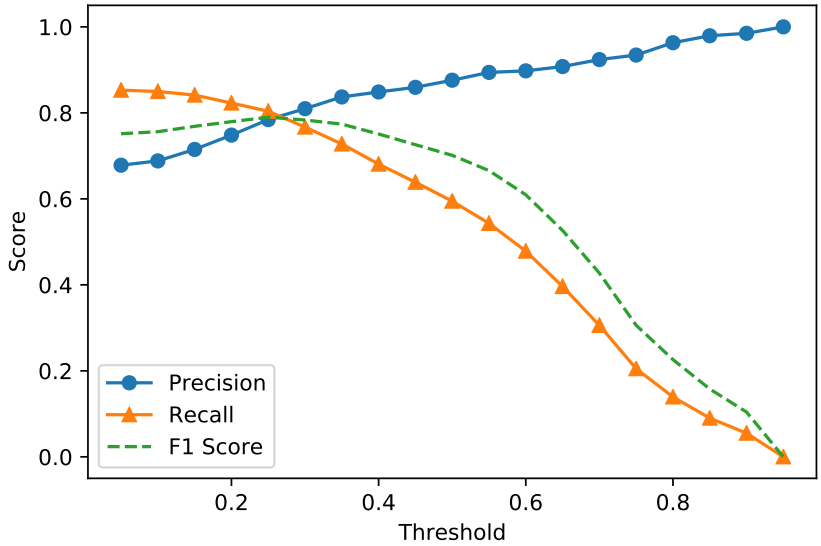


FIGURE 5.7: Influence of the value of the similarity threshold t

against those achieved by several NER-based tools. In addition of NER being the most common approach to document protection, it is also the only method among those discussed in Chapter 2 that can compare with our approach in terms of practicality for real-world tasks. In particular we used the Stanford Named Entity Recognizer software (Manning et al., 2014), which provides 3 pre-trained NER models for English (NER3, NER4 and NER7), and Microsoft Presidio, which tailors NER towards privacy protection:

- NER3: detects and categorizes named entities of ORGANIZATION, LOCATION and PERSON types.

- NER4: detects and categorizes named entities of ORGANIZATION, LOCATION, PERSON and MISC types.
- NER7: detects and categorizes named entities of LOCATION, ORGANIZATION, DATE, MONEY, PERSON, PERCENT and TIME types.
- Presidio: detects and categorizes named entities of CREDIT_CARD, CRYPTO, DATE_TIME, DOMAIN_NAME, EMAIL_ADDRESS, IBAN_CODE, IP_ADDRESS, LOCATION, PERSON, NRP, PHONE_NUMBER, UK_NHS, US_BANK_NUMBER, US_DRIVER_LICENSE, US_ITIN, US_PASSPORT and US_SSN types.

Table 5.1 reports the evaluation figures of the different methods on average for the 50 documents under consideration. It is clear that our method improves on the NER-based approach very significantly, regardless of the NER model in use. In particular, the recall is more than doubled, which results in a much higher F_1-score. This illustrates the main limitation of NER-based methods: named entities are not the sole source of disclosure. This limitation tends to yield under-protected documents in which, for example, identities may be disclosed by correlating several non-protected facts or personal features that do not fall into the predefined types of named entities. By comparing the last column of Table 5.1 with Figure 5.7, we also see that our method provides significantly better F_1 scores than NER tools for a wide range of threshold values (those below 0.5). This shows that the user enjoys some freedom to tailor the threshold to his/her needs, while still getting a better protection-utility balance than with NER-based methods.

Regarding the differences among the three NER models, we see that NER3 produces the worst recall because it has been trained to detect the least number of entity types. NER7 and Presidio improve on the results of NER3, mainly because they can detect dates such as birthdates, which are quite common in biographies. Finally, whereas NER3, NER7 and Presidio have been trained with specific named entity types, NER4 adds the MISC type, which encompasses a variety of named entities such as nationalities. No significant differences in precision are visible across NER tools, regardless of the different models they use to detect NEs, *i.e.*, CRF for Standford and BERT-based NER for Presidio. From the privacy point of view, the low recall resulting from the limited amount of supported NE types has a much greater influence than precision.

Disabling the pre-processing in our method decreases the recall from 81.24% to 59.79%. Even though this penalty is large, the decreased recall is still significantly higher than the recall of the best NER model. On the one hand, this illustrates the benefits of analyzing the content of documents at a conceptual level, rather than at a word level. On the other hand, it can be seen that the results of our approach with a language-agnostic implementation (*i.e.*, without language-dependent tools) are still significantly better than those of NER-based methods, which nonetheless require language-specific tagged training data.

The behavior of the different methods is illustrated in Table 5.2, which contains an extract of the input text of one of the evaluated documents and compares the manual tagging with the entities detected by the different approaches. We can see that NER-based methods failed to detect pieces of information that are relevant

TABLE 5.1: Average precision, recall and F_1-score for the 50 evaluated documents

Method	*Precision*	*Recall*	F_1
NER3	96.09%	19.59%	32.07%
NER4	97.59%	34.25%	49.72%
NER7	**98.32%**	27.89%	42.77%
Presidio	98.06%	27.07%	41.12%
Our method	82.69%	**81.24%**	**81.66%**
Our method (no pre-process)	83.48%	59.79%	69.00%

to re-identify the actor, such as her/his birth date (for NER3 and NER4) or the title of the movies or TV series she/he appeared in. NER7 is particularly worrying, because it missed the actor's name, which is a direct identifier. In contrast, our approach only missed the actor's profession (due to its being very general), and only incurred over-protection for the term "the action drama".

In fact, it takes more than providing good average results for a method to be useful: a good method has to yield good enough results in all cases. Table 5.3 reports the coefficient of variation (a measure of dispersion computed as the ratio of the standard deviation to the mean) of the results given in Table 5.1. We can see that our approach provides the most consistent results, with a variation of the F_1-score just 0.31%.

Precision is the only metric for which the NER-based approach achieved better figures. Indeed, NER has an inherently high accuracy in a pure NER task. Moreover, the evaluation scenario we consider is quite favorable to NER because most of the text in each document is highly related to the individual to be protected (the

TABLE 5.2: Output samples for each method

Manual annotation	Thomas Cruise Mapother IV (born July 3, 1962) is an American actor and producer. He started his career at age 19 in the film Endless Love (1981), before making his breakthrough in the comedy Risky Business (1983) and receiving widespread attention for starring in the action drama Top Gun (1986) as Lieutenant Pete "Maverick" Mitchell.
NER3	Thomas Cruise Mapother IV (born July 3, 1962) is an American actor and producer. He started his career at age 19 in the film Endless Love (1981), before making his breakthrough in the comedy Risky Business (1983) and receiving widespread attention for starring in the action drama Top Gun (1986) as Lieutenant Pete "Maverick" Mitchell.
NER4	Thomas Cruise Mapother IV (born July 3, 1962) is an American actor and producer. He started his career at age 19 in the film Endless Love (1981), before making his breakthrough in the comedy Risky Business (1983) and receiving widespread attention for starring in the action drama Top Gun (1986) as Lieutenant Pete "Maverick" Mitchell.
NER7	Thomas Cruise Mapother IV (born July 3, 1962) is an American actor and producer. He started his career at age 19 in the film Endless Love (1981), before making his breakthrough in the comedy Risky Business (1983) and receiving widespread attention for starring in the action drama Top Gun (1986) as Lieutenant Pete "Maverick" Mitchell.
Presidio	Thomas Cruise Mapother IV (born July 3, 1962) is an American actor and producer. He started his career at age 19 in the film Endless Love (1981), before making his breakthrough in the comedy Risky Business (1983) and receiving widespread attention for starring in the action drama Top Gun (1986) as Lieutenant Pete "Maverick" Mitchell.
Our method	Thomas Cruise Mapother IV (born July 3, 1962) is an American actor and producer. He started his career at age 19 in the film Endless Love (1981), before making his breakthrough in the comedy Risky Business (1983) and receiving widespread attention for starring in the action drama Top Gun (1986) as Lieutenant Pete "Maverick" Mitchell.

TABLE 5.3: Average coefficients of variation (CV) for precision, recall and F_1-score

Method	*Precision CV*	*Recall CV*	F_1 *CV*
NER3	0.48%	2.14%	2.41%
NER4	0.16%	3.67%	2.97%
NER7	**0.09%**	2.65%	2.34%
Presidio	0.12%	4.75%	5.02%
Our method	0.52%	**0.67%**	**0.31%**

biographee). Therefore, if a named entity appeared in the text and was properly identified by the NER method, then this named entity was very likely to refer to the biographee and, therefore, to be disclosive. In a less favorable scenario, in which the content of a document could refer to different people, the precision of the NER-based approach would significantly decrease, because not all the named entities in the document would refer to the individual to be protected. We simulated this setting by putting together the biographies of two related actors (both American and acting in the same TV series) and manually tagging only the terms that may be disclosive on one of them. In this case, the system had to detect not only those terms that exclusively related to the actor to be protected, but also the information he or she had in common with the other actor also referred to in the text. The results of this experiment are reported in Table 5.4.

As expected, the precision of the NER-based methods is significantly lower in this two-actor setting, even though we see relevant differences among the different NER models. The problem was not only to detect the NEs, but to distinguish which actor an

TABLE 5.4: Precision, recall and F_1-score for a document referring to two different individuals

Method	*Precision*	*Recall*	F_1
NER3	55.55%	22.72%	32.25%
NER4	**77.27%**	59.09%	66.96%
NER7	60.0%	27.27%	37.5%
Presidio	66.67%	38.1%	48.48%
Our method	68.0%	**81.81%**	**74.27%**

NE referred to. Some significant false positives of NER tools involved tagging the birth place and birth date of the actor *not* to be protected; this was a mistake that our method avoided. Although the false positive rate of our method also increased with respect to the single-actor setting, the increase was smaller than for NER-based methods; besides, the recall of our method stayed at the same level as in the single-actor setting.

So far, we have examined the performance of our method on word2vec and with an excellent training data set that perfectly matches the contents of the evaluated documents. However, gathering large and suitable training data may be difficult in some domains. On the other hand, our method is not tied to any particular embedding model and may benefit from advances in embedding techniques. To assess the generality of our approach, we also experimented with the following word embedding models trained on general-purpose data:

- *Pre-trained word2vec* (Mikolov et al., 2013a): an off-the-shelf word2vec model trained on the Google News data set. The

model has a vocabulary of 3 million words/terms.

- *FastText* (Bojanowski et al., 2017): a library for word embedding learning created by Facebook's AI Research lab (see Section 5.2.1 for more details). Two pre-trained models were considered: the first model (*wiki1*) has a vocabulary of 2 million words/terms trained on the Common Crawl data set, which is an archive of web data collected since 2011; the second model (*wiki2*), has a vocabulary of 1 million words/terms trained on the 2017 Wikipedia snapshot, the UMBC webbase corpus and the statmt.org news data set.

- *BERT (base-cased)* (Devlin et al., 2018): a BERT model with 12 encoders with 12 bidirectional self-attention heads trained from data extracted from the BookCorpus with 800M words and the English Wikipedia with 2,500M words.

When the training data do not perfectly match the contents of the document to which the model is applied, the document may contain out-of-vocabulary (OOV) terms. For the models trained with fastText this does not occur because it approximates OOV vectors from subword information. However, since word2vec does not do this, many of the complex n-grams we extracted from the documents to be evaluated were not found in the model's vocabulary. For the Google News model to provide usable results, we had to disable pre-processing so that the content of the document was evaluated at a word level. The evaluation figures obtained with the pre-trained models are reported in Table 5.5.

It is interesting to see that the models trained with fastText and BERT produced results comparable to those obtained with

TABLE 5.5: Evaluation figures with several pre-trained word embedding models

Model	*Precision*	*Recall*	F_1
FastText (wiki1)	82.38%	71.84%	75.93%
FastText (wiki2)	83.06%	71.85%	76.20%
Word2Vec (Google News)	68.58%	24.80%	35.58%
BERT (base-cased)	81.84%	72.31%	75.95%

our domain-specific training corpora. This shows that in the absence of such domain-specific corpora, large general-purpose corpora and pre-trained models may be employed with reasonably good results. However, when the pre-processing applied to pre-train the model does not match the pre-processing used to evaluate new documents and OOV terms are not handled properly, results are much worse, as it was the case for the Google News model. Recall was especially bad because many of the n-grams that ought to have been detected as quasi-identifiers were either not found in the vocabulary or were only partially detected, which we also counted as a miss.

5.4.2 Masking phase

In this section we report on the performance of the masking strategy presented in Section 5.3.3. We measured the relative utility preserved by the protected document after masking via ontology-based generalization the quasi-identifiers detected in the previous phase. Generalizations for quasi-identifiers were obtained from WordNet and YAGO. The vectors of such generalizations were

computed by re-training the model with the Wikipedia articles corresponding to those generalizations.

Similarly to related works (Sánchez and Batet, 2016), we measured the relative utility of the protected document (D'') as the aggregation of the semantics it conveys w.r.t. the semantics of the original document (D). This yields

$$Utility_preservation(D'') = \frac{Semantics(D'')}{Semantics(D)} \cdot 100,$$

where $Semantics(D)$ is the sum of the information content $IC(p_i)$ of each phrase p_i in D, that is,

$$Semantics(D) = \sum_{i=1}^{n} IC(p_i),$$

with n being the number of phrases in D.

In information-theoretic terms, $IC(p_i)$ is the inverse logarithm of the probability of occurrence of p_i:

$$IC(p_i) = -log_2 \Pr(p_i).$$

In this way, specific terms that rarely occur are considered more informative (and therefore semantically richer) than general terms that appear frequently and can be assumed to have more general meanings. Employing the information content of terms as a semantic metric is a common approach in computational linguistics (Resnik, 1995).

To obtain representative probabilities of occurrence, we queried p_i in the Bing search engine and divided the number of results it provides by the total number of resources indexed by the search

engine, as done in (Sánchez and Batet, 2016).

The utility metric we use captures the fact that the generalizations used for masking carry less information than their respective specializations. The information content is a metric commonly used to quantify the semantic content of terms in computational linguistics (Resnik, 1995). Moreover, in the literature on data privacy (Hundepool et al., 2013), utility preservation is usually measured as a function of the information loss incurred by masking, which is exactly what our utility metric does. We focus on the total information content lost as a result of the replacements rather on the number of such replacements.

Table 5.6 depicts the average relative utility preserved by different methods and masking strategies for the 50 evaluated documents. We compared our method against the NER approaches discussed in the previous section when replacing the detected named entities by their types (*e.g.*, "Tom Cruise" $\rightarrow$ PERSON). We also report the relative utility that remained when quasi-identifiers (in our case) and named entities (for NER-based methods) were suppressed, as usually done in document redaction (Sánchez, Batet, and Viejo, 2013b).

As expected, plain suppression produced protected outcomes that retained significantly less utility than generalization. Add to this that blacking out pieces of text hampers document readability and makes potential attackers aware of the document sensitivity (Bier et al., 2009). Generalization, either via ontologies or via named entity types, preserved much more utility. For NER models, utility figures were inversely proportional to the number of masked terms (shown in the last column). In contrast, our approach yielded the second best utility value while masking the

TABLE 5.6: Average relative utility preserved by different methods and masking strategies

Method	*Suppression*	*Generalization*	*Avg. masked terms*
NER3	**66.98%**	**85.44%**	33.8
NER4	39.70%	74.73%	64.08
NER7	54.00%	81.29%	54.16
Presidio	54.04%	81.00%	61.12
Our Method	48.48%	85.01%	**86.04**

largest number of terms (resulting from the best recall in the detection phase). The good utility was due to the use of fine-grained ontological generalizations rather than coarse NE types. Thus, our method achieved the best balance between privacy protection and utility preservation.

5.4.3 Protection against re-identification

So far we have evaluated detection and masking in isolation. To measure the practical effectiveness of protection as a whole, we implemented a re-identification attack that is inspired by the evaluation framework proposed in works like (Fernandes, Dras, and McIver, 2019) for authorship attribution. The general idea of this experiment is to check the ability of a machine learning classifier to correctly predict the entity from the protected output of each method.

Specifically, we took the 50 articles we had manually annotated and we fine-tuned the BERT base-cased model to predict the actor's name by training it on the post-summary text, that is,

TABLE 5.7: Percentage of correct predictions for each method

Input	*Correct predictions*
Original summary	84.67%
Manual annotation	2.00%
NER3	18.00%
NER4	16.00%
NER7	37.33%
Presidio	18.67%
Our Method	10.00%

all of the article's text except the part we manually annotated. We split the post-summary text into sentences, each one labeled with the name of the actor. We used 80% of the sentences to train the model and the remaining 20% to validate it. Then we tested the classifier on the summary text, which is the part that we manually annotated and that we protected. Predictions were evaluated by checking whether the majority-predicted class of the sentences in the summary matched the actual actor. The classifier was tested on the original unchanged summaries, the manually annotated summaries (by just replacing the tagged text by the label SENSITIVE) and the masked outputs of the different protection methods. The percentage of correct predictions is reported in Table 5.7. Due to the non-deterministic behavior of the BERT model in tensorflow, which sightly varies for every run, we report the average results of three runs.

First of all, it is important to highlight that this setting is very favorable for re-identification. On the one hand, the number of

individuals/classes to be predicted is very limited in comparison with the size of the population of personal data sets (accounting for thousands or millions of individuals). On the other hand, the text used for prediction (summary) bears a lot of similarities to the training data, not only regarding content, but also regarding the linguistic structure of the sentences. As a matter of fact, sentences in the summary also appear quite frequently in the post-summary text, and this gives an 'unfair' advantage to the classifier. Despite all the above, we see in Table 5.7 that the prediction accuracy for manual annotation was at the level of random guess (2%, that is, 1/50). Yet the protection achieved by manual annotation came at a high utility cost, because the masking in this case was equivalent to text suppression and suppression was shown in Table 5.6 to significantly damage the utility of the document. Table 5.7 also shows that our method is the one that offers the best protection, closest to the protection level offered by manual annotation and with much less utility loss (due to the use of ontological generalization).

The results in Table 5.7 show some discrepancy with respect to the recall-based protection reported in Table 5.1 for some NER models, especially NER7. Even though NER7 yielded a higher recall than NER3 due to the former considering a larger variety of NE types, its protection against re-identification was less effective, mainly because NER7 failed to detect some family names that NER3 did not miss, as we mentioned above. This illustrates that recall figures do not give a complete view on the robustness of protection, because the nature of the terms missed by a method (*e.g.*, highly disclosive direct identifiers such as family names or less risky circumstantial quasi-identifiers such as the

year an event happened) may be more influential on the success of re-identification attacks than the number of identified terms.

5.5 Application scenarios

The approach we present in this chapter is remarkably versatile and unconstrained. In particular, it does not require manually tagged data, it works reasonably well with general-purpose pre-trained models and, except for the optional pre-processing, it is language-agnostic. As a result, our method can be immediately applied to a variety of real-word scenarios.

The most natural application of text protection is document declassification, which consists in releasing documents that used to be classified as confidential. Declassification is oftentimes motivated by transparency principles and open data initiatives. To make transparency compatible with data protection and other interests at stake, parts of the declassified documents that may refer to non-public individuals, facts or places need to be sanitized by redacting (blacking out or deleting) them. Redaction is also employed for selective disclosure of information. For example, when a document is subpoenaed in a court case, information not relevant to the case is often redacted. Similarly, US legislations on the privacy of medical data mandate hospitals to redact all direct or indirect references to sensitive diseases (such as sexually transmitted diseases or AIDS) before releasing patient records to insurance companies or in response to worker's compensation or motor vehicle accident claims (Bier et al., 2009). As discussed in Chapter 2, redaction has traditionally been performed manually

by following certain rules or guidelines (Agency, 2005). However, manual approaches are time-consuming (Dorr et al., 2006) and error-prone, and they usually require the coordinated effort of several human experts (Bier et al., 2009). Our method perfectly fits the needs of document redaction: given a set of entities to be protected (identities, locations or confidential values such as sensitive diseases), our technique can be iteratively applied to each entity in a given document so that any references, either direct or indirect, to those entities are detected and subsequently redacted.

In a different context, the well-known Snowden and Wikileaks scandals have made companies more aware of the damage that may be caused by insiders who gradually gain access to more and more confidential data. To mitigate this threat, companies have started to implement risk management policies, whereby the contents of corporate files are characterized according to their risk, and accounting is enforced on employees by continuously monitoring their accesses to such files. Then, metrics such as *misuseability scores* (Harel et al., 2012) can be developed to quantify the harm that might be inflicted by an employee in a hypothetical data leakage as a function of the accumulated sensitive data he or she has accessed. These metrics enable early detection and prevention of data leakage or misuse by insiders, for example, by implementing dynamic access control policies to decide whether or not access to new content should be granted to specific employees, or by detecting individuals with unusually high scores. A variety of commercial software packages are available to enforce risk assessment on corporate files, such as the aforementioned Amazon's Macie (*Amazon Macie - Amazon Web Services (AWS)*), Google's DLP (*Cloud Data Loss Prevention*) or Symantec's Data Loss

Prevention (*Symantec Data Loss Prevention*). However, all those packages characterize risk based on the (limited set of) named entity types they can detect by means of regular expressions and pre-trained classifiers. Thus, they suffer from the limitations discussed in Chapter 2. In this respect, as shown in Section 5.4, our approach can offer a much more accurate risk characterization, which can also be tailored to the specific privacy requirements of the organization.

A similar approach can also be employed to measure the exposure level of users of social networks and therefore their privacy risks. Proposals in the literature compute privacy risk scores of social network users as the sum of attributes disclosed by their profiles (Srivastava and Geethakumari, 2013; Liu and Terzi, 2009). However, messages posted by users provide much more detailed and up-to-date information on the users' preferences or demography than static attributes, thereby entailing higher risk (Sánchez, Domingo-Ferrer, and Martínez, 2019). Our method can be applied (trained) on the users' data and be enforced on the topics that current regulations (such as GDPR) regard as sensitive, such as religion, sexuality or ethnicity. As a consequence, the user can be made aware of the level of exposure his or her publications entail on such sensitive topics and by that means he or she can make informed decisions on whether to publish certain data. User awareness and empowerment regarding privacy are in fact pillars of the modern outlook on privacy protection (Sánchez and Viejo, 2017).

5.6 Conclusion

In this chapter, we have presented an automatic method to protect text documents that leverages word embeddings to measure disclosure risk and masks disclosive terms via utility-preserving generalizations. Our approach is more general and, at the same time, more flexible than methods based on NER. On the one hand, we do not restrict the disclosure assessment to predefined entity types, because doing so typically incurs under-protection, as we have shown in our evaluation.

On the other hand, our method drives protection according to privacy requirements focused on the entity or entities on whom information should not be disclosed by the sanitized text. This behavior is more similar to the way human experts tackle manual sanitization (Bier et al., 2009) and to the way privacy models enforce *ex-ante* privacy guarantees in structured databases (Domingo-Ferrer, Sánchez, and Soria-Comas, 2016).

As a result, the protection afforded by our method is consistent with the privacy requirements and, at the same time, more robust and utility-preserving than the protection of NER-based methods. Finally, even though our method relies on machine learning, it does not require tagged data and model building is language-agnostic. Therefore, no manual effort is required during the whole lifecycle of the protection process, which makes our method suitable for managing large amounts of textual data.

Chapter 6

Conclusions and Future Work

This thesis has dealt with anonymization methods for unstructured textual data. First, we have focused on improving the current sequence labeling mechanisms (*i.e.* NER models). Even though our methods outperform the current state of the art in specific tasks of medical document anonymization, they are hampered by the inherent limitations of NER methods applied to data anonymization.

Next, we have shown that, provided that collections of textual documents can be transformed to structured lists of (quasi-)identifiers, standard SDC methods can be applied to enforce more robust anonymization. The detection of (quasi-)identifiers is, however, very challenging for textual documents and, again, relying on NER-based methods severely limits the generality of the approach.

To overcome the shortcomings of NER-based methods, we leveraged the notion of semantic relatedness via word embeddings and the structured knowledge modeled in ontologies. In this way,

we were able to build a complete automated framework for textual data anonymization. The empirical work we carried out on real textual data supported our starting hypothesis: by relying on sound semantic tools and resources, textual data can be protected while preserving their utility significantly better than with naive methods like NER-based models.

6.1 Contributions and publications

Chapter 3 focused on medical document anonymization. To tackle the problem of anonymizing medical documents in the Spanish language, we developed two systems, ReCRF and E2EJ. Both systems were submitted to the MEDDOCAN 2019 contest, where they scored the second and the fifth positions, respectively. ReCRF is a combination of hand-crafted features and automatically generated regular expressions, while E2EJ is an end-to-end model based on deep learning methods. This work resulted in the following publications:

- Fadi Hassan, Mohammed Jabreel, Najlaa Maaroof, David Sánchez, Josep Domingo-Ferrer, and Antonio Moreno. "ReCRF: Spanish Medical Document Anonymization using Automatically-crafted Rules and CRF." In *Proceedings of IberLEF@ SEPLN*, pp. 727-734. 2019.

- Mohammed Jabreel, Fadi Hassan, Najlaa Maaroof, David Sánchez, Josep Domingo-Ferrer, and Antonio Moreno. "E2EJ: Anonymization of Spanish Medical Records using End-to-End Joint Neural Networks." In *Proceedings of IberLEF@ SEPLN*, pp. 712-719. 2019.

The work "ReCRF: Spanish Medical Document Anonymization using Automatically-crafted Rules and CRF" received two prizes in the Medical Document Anonymization Track (MEDDOCAN) 2019, as the second-best system in the two sub-tasks (NER sub-task and Spans sub-task).

Chapter 4 presented a first approach applying the notion of disclosure risk as understood in the literature on SDC to textual documents. The proposal leverages NER-based models to detect quasi-identifiers and/or confidential terms in these documents. Once these terms have been located, we can build a structured representation of the sensitive information contained in the document, which can be anonymized through standard SDC methods (*e.g.* generalization, suppression, etc.) to keep the disclosure risk under control. This work resulted in the following publication:

- Fadi Hassan, Josep Domingo-Ferrer, and Jordi Soria-Comas. "Anonymization of Unstructured Data via Named-Entity Recognition." In *Proceedings of International Conference on Modeling Decisions for Artificial Intelligence – MDAI 2018*, pp. 296-305. Springer, Cham, 2018. CORE ranking: B.

In Chapter 5, we introduced a complete framework for document anonymization that leverages word embedding models and ontologies to provide robust and utility-preserving anonymization of textual documents. The presented approach is more general and, at the same time, more flexible than methods based on NER models. The experiments show that the proposed model significantly outperforms NER models. The work in this chapter resulted in the following publications:

- Fadi Hassan, David Sánchez, Jordi Soria-Comas, and Josep Domingo-Ferrer. "Automatic Anonymization of Textual Documents: Detecting Sensitive Information via Word Embeddings." In *Proceedings of 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 358-365. IEEE, 2019. CORE ranking: A.

- Fadi Hassan, David Sánchez, and Josep Domingo-Ferrer. "Utility-Preserving Privacy Protection of Textual Documents via Word Embeddings." *IEEE Transactions on Knowledge and Data Engineering*. Available online 01 May 2021, ISSN 1041-4347, http://dx.doi.org/10.1109/TKDE.2021.3076632. Impact Factor: 4.935 (1st quartile).

The paper "Automatic Anonymization of Textual Documents: Detecting Sensitive Information via Word Embeddings" received the *Best Privacy Track Paper Award* of the TrustCom/BigDataSE 2019 conference.

6.2 Future work

This thesis opens new ground for research. The following topics can be pursued to continue the work:

- In Chapter 3, we developed two systems to identify sensitive and personal data in medical texts. In this respect, we plan to use attention-based models, called transformers (Vaswani et al., 2017; Wolf et al., 2019), which have been

developed to solve many natural language processing tasks. Transformers are showing impressive results in many tasks, including NER. We aim to adapt these models to develop a more accurate system to detect sensitive information in the medical domain.

- The work presented in Chapter 5 opens several avenues for future research:
  - Applying the proposed approach in other scenarios considering i) domain-specific documents (*e.g.*, healthcare-related) and ontologies (such as SNOMED-CT), and ii) a variety of privacy requirements including identities and confidential attributes.
  - Tailoring contextual embedding models like BERT to our domain. As shown in the evaluation, pre-trained BERT was able to obtain results comparable to a word2vec model trained on domain-specific data. Hence, BERT trained on domain-specific data might offer even better results. Moreover, thanks to the contextual embeddings provided by BERT, language ambiguity will be minimized without requiring complex semantic disambiguation methods.

Bibliography

Abril, Daniel, Guillermo Navarro-Arribas, and Vicenç Torra (2011). "On the declassification of confidential documents". In: *Proceedings of Modeling Decisions for Artificial Intelligence, MDAI 2011*, pp. 235–246.

Acemoglu, Daron and Asuman Ozdaglar (2011). "Opinion Dynamics and Learning in Social Networks". In: *Dynamic Games and Applications* 1.1, pp. 3–49.

Agency, National Security (2005). "Redacting with Confidence: How to Safely Publish Sanitized Reports Converted from Word to Pdf". In:

Akbik, Alan, Tanja Bergmann, and Roland Vollgraf (2019). "Pooled Contextualized Embeddings for Named Entity Recognition". In: *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 724–728.

Amazon Macie - Amazon Web Services (AWS). `https://aws.amazon.com/macie/`. last accessed: 24-Jan-2020.

Anandan, Balamurugan and Chris Clifton (2011). "Significance of term relationships on anonymization". In: *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03*. IEEE Computer Society, pp. 253–256.

Anandan, Balamurugan, Chris Clifton, Wei Jiang, Mummoorthy Murugesan, Pedro Pastrana-Camacho, and Luo Si (2012). "t-Plausibility: Generalizing Words to Desensitize Text." In: *Trans. Data Privacy* 5.3, pp. 505–534.

Babych, Bogdan and Anthony Hartley (2003). "Improving machine translation quality with automatic named entity recognition". In: *Proceedings of the 7th International EAMT Workshop on MT and Other Language Technology Tools, Improving MT Through Other Language Technology Tools: Resources and Tools for Building MT*. EAMT '03. Budapest, Hungary: Association for Computational Linguistics, pp. 1–8. URL: http://dl.acm.org/citation.cfm?id=1609822.1609823.

Batet, Montserrat and David Sánchez (2018). "Semantic Disclosure Control: semantics meets data privacy". In: *Online Information Review* 42.3, pp. 290–303.

Bier, Eric, Richard Chow, Philippe Gollé, Tracy Holloway King, and Jessica Staddon (2009). "The rules of redaction: Identify, protect, review (and repeat)". In: *IEEE Security & Privacy* 7.6, pp. 46–53.

Bird Steven, Edward Loper and Ewan Klein (2009). *Natural Language Toolkit (NLTK)*. https://www.nltk.org/. Natural Language Processing with Python, O'Reilly Media Inc.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2017). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. ISSN: 2307-387X.

Boleda, Gemma (2020). "Distributional semantics and linguistic theory". In: *Annual Review of Linguistics* 6, pp. 213–234.

Chakaravarthy, Venkatesan T, Himanshu Gupta, Prasan Roy, and Mukesh K Mohania (2008). "Efficient techniques for document sanitization". In: *Proceedings of the ACM Confernece on Information and Knowledge Management*, 843—852.

Chow, Richard, Philippe Golle, and Jessica Staddon (2008). "Detecting privacy leaks using corpus-based association rules". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 893–901.

Cloud Data Loss Prevention. `https://cloud.google.com/dlp/`. last accessed: 24-Jan-2020.

Culotta, Aron, Ron Bekkerman, and Andrew McCallum (2004). "Extracting social networks and contact information from email and the web". In: *Computer Science Department Faculty Publication Series*, p. 33.

Cumby, Chad and Rayid Ghani (2011). "A machine learning based system for semi-automatically redacting documents". In: *Proceedings of the 23rd Innovative Applications of Artificial Intelligence Conference*, pp. 1628–1635.

Da, Kingma (2014). "A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.

Domingo-Ferrer, Josep, David Sánchez, and Alberto Blanco-Justicia (2020). "The limits of differential privacy (and its misuse in data release and machine learning)". In: *arXiv preprint arXiv: 2011.02352*.

Domingo-Ferrer, Josep, David Sánchez, and Jordi Soria-Comas (2016). "Database anonymization: privacy models, data utility, and

microaggregation-based inter-model connections". In: *Synthesis Lectures on Information Security, Privacy, & Trust* 8.1, pp. 1–136.

Dorr, David A, WF Phillips, Shobha Phansalkar, Shannon A Sims, and John Franklin Hurdle (2006). "Assessing the difficulty and time cost of de-identification in clinical narratives". In: *Methods of Information in Medicine* 45.3, pp. 246–252.

Drechsler, Jörg (2011). *Synthetic datasets for statistical disclosure control*. Vol. 201. Lecture Notes in Statistics. Springer-Verlag New York. DOI: 10.1007/978-1-4614-0326-5.

Dwork, Cynthia (2008). "Differential privacy: A survey of results". In: *International conference on theory and applications of models of computation*. Springer, pp. 1–19.

Dwork, Cynthia, Aaron Roth, et al. (2014). "The algorithmic foundations of differential privacy." In: *Foundations and Trends in Theoretical Computer Science* 9.3-4, pp. 211–407.

Economist, The (2010). "Data, data everywhere: A special report on managing information". In: *The Economist*.

Ekbal, Asif, Rejwanul Haque, and Sivaji Bandyopadhyay (2007). "Bengali part of speech tagging using conditional random field". In: *Proceedings of the seventh International Symposium on Natural Language Processing, SNLP-2007*.

Fellbaum, Christiane (1998). *WordNet: An Electronic Lexical Database*. Cambridge, Massachusetts.

Fernandes, Natasha, Mark Dras, and Annabelle McIver (2019). "Generalised differential privacy for text document processing". In: *International Conference on Principles of Security and Trust*. Springer, Cham, pp. 123–148.

Feyisetan, Oluwaseyi, Tom Diethe, and Thomas Drake (2019). "Leveraging hierarchical representations for preserving privacy and utility in text". In: *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, pp. 210–219.

Finkel, Jenny Rose, Trond Grenager, and Christopher D Manning (2005). "Incorporating non-local information into information extraction systems by Gibbs sampling". In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. ACL '05. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 363–370. DOI: 10.3115/1219840.1219885. URL: \url{https://doi.org/10.3115/1219840.1219885}.

Guarino, Nicola (1998). *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*. Vol. 46. IOS press.

Harel, Amir, Asaf Shabtai, Lior Rokach, and Yuval Elovici (2012). "M-Score: A Misuseability Weight Measure". In: *IEEE Transactions on Dependable and Secure Computing* 9.3, pp. 414–428.

Honnibal, Matthew and Ines Montani (2017). "spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing". In: *To appear*.

Hundepool, Anco, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Eric Schulte Nordholt, Keith Spicer, and Peter-Paul De Wolf (2013). *Statistical Disclosure Control*. Wiley.

ISO 25237:2017, Health informatics Pseudonymization (2017). https://www.iso.org/standard/63553.html. last accessed: 09-Mar-2021.

Jabreel, Mohammed, Fadi Hassan, and Antonio Moreno (2018). "Target-Dependent Sentiment Analysis of Tweets Using Bidirectional Gated Recurrent Neural Networks". In: *Advances in Hybridization of Intelligent Methods*. Springer, pp. 39–55.

Jakob, Niklas, Stefan Hagen Weber, Mark Christoph Müller, and Iryna Gurevych (2009). "Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations". In: *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, pp. 57–64.

Khalid, Mahboob Alam, Valentin Jijkoun, and Maarten De Rijke (2008). "The impact of named entity normalization on information retrieval for question answering". In: *Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval*. ECIR'08. Glasgow, UK: Springer-Verlag, pp. 705–710. ISBN: 3-540-78645-7, 978-3-540-78645-0. URL: http://dl.acm.org/citation.cfm?id=1793274.1793371.

Kiros, Ryan, Yukun Zhu, Ruslan Salakhutdinov, Richard S Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler (2015). "Skip-Thought Vectors". In: *Proceedings of Advances in Neural Information Processing Systems (NIPS 2015)*.

Korobov, Mikhail. *sklearn-crfsuite*. https://sklearn-crfsuite.readthedocs.io/en/latest/index.html. last accessed: 31-May-2019.

Krishnan, Vijay and Vignesh Ganapathy (2005). *Named entity recognition*.

Lafferty, John, Andrew McCallum, and Fernando CN Pereira (2001). "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". In:

Li, Yitong, Timothy Baldwin, and Trevor Cohn (2018). "Towards robust and privacy-preserving text representations". In: *arXiv preprint arXiv:1805.06093*.

Liu, Kun and Evimaria Terzi (2009). "A framework for computing the privacy scores of users in online social networks". In: *Proceedings of ICDM 2009-The 9th IEEE International Confeirence on Data Mining*. IEEE Computer Society, pp. 288–297.

Liu, Zengjian, Yangxin Chen, Buzhou Tang, Xiaolong Wang, Qingcai Chen, Haodi Li, Jingfeng Wang, Qiwen Deng, and Suisong Zhu (2015). "Automatic de-identification of electronic medical records using token-level and character-level conditional random fields". In: *Journal of biomedical informatics* 58, S47–S52.

Liu, Zengjian, Buzhou Tang, Xiaolong Wang, and Qingcai Chen (2017). "De-identification of clinical notes via recurrent neural network and conditional random field". In: *Journal of biomedical informatics* 75, S34–S42.

Manning, Christopher D, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky (2014). "The Stanford CoreNLP natural language processing toolkit". In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pp. 55–60.

Marimon, Montserrat, Aitor Gonzalez-Agirre, Ander Intxaurrondo, Heidy Rodriguez, Jose Lopez Martin, Marta Villegas, and Martin Krallinger (2019). "Automatic De-identification of Medical Texts in Spanish: the MEDDOCAN Track, Corpus, Guidelines, Methods and Evaluation of Results." In: *IberLEF@ SEPLN*, pp. 618–638.

Matthias Templ, Bernhard Meindl and Alexander Kowarik (2016). "Tutorial for sdcMicroGUI (and sdcMicro), December 6, 2016, Vienna". In:

Medicare & Medicaid Services, Centers for et al. (1996). "The health insurance portability and accountability act of 1996 (HIPAA)". In: *Online at http://www. cms. hhs. gov/hipaa*.

Meystre, Stephane M, F Jeffrey Friedlin, Brett R South, Shuying Shen, and Matthew H Samore (2010). "Automatic de-identification of textual documents in the electronic health record: a review of recent research". In: *BMC Medical Research Methodology* 10.70.

Microsoft (2019). *Presidio - Data Protection API*. `https://github.com/microsoft/presidio`.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013a). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean (2013b). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*, pp. 3111–3119.

Mohammad, Saif M and Graeme Hirst (2012). "Distributional measures of semantic distance: A survey". In: *arXiv preprint arXiv: 1203.1858*.

Morton, Thomas, Joern Kottmann, Jason Baldridge, and Gann Bierner (2005). "Opennlp: A java-based nlp toolkit". In: *Proc. EACL*.

Morwal, Sudha, Nusrat Jahan, and Deepti Chopra (2012). "Named entity recognition using hidden Markov model (HMM)". In: *International Journal on Natural Language Computing (IJNLC)* 1.4, pp. 15–23.

Nadeau, David and Satoshi Sekine (2007). "A survey of named entity recognition and classification". In: *Linguisticae Investigationes* 30.1. Publisher: John Benjamins Publishing Company, pp. 3–26.

Neamatullah, Ishna, Margaret M. Douglass, Li-wei H. Lehman, Andrew Reisner, Mauricio Villarroel, William J. Long, Peter Szolovits, George B. Moody, Roger G. Mark, and Gari D. Clifford (2008). "Automated de-identification of free-text medical records". In: *BMC Medical Informatics and Decision Making* 8.1, p. 32. ISSN: 1472-6947. DOI: 10.1186/1472-6947-8-32. URL: \url{https://doi.org/10.1186/1472-6947-8-32}.

Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Peters, Matthew E, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). "Deep contextualized word representations". In: *arXiv preprint arXiv:1802.05365*.

Rebele, Thomas, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum (2016). "YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames". In: *International semantic web conference*. Springer, pp. 177–185.

Resnik, Philip (1995). "Using information content to evaluate semantic similarity in a taxonomy". In: *arXiv preprint cmp-lg/9511007*.

Rosario, Barbara and Marti A. Hearst (2004). "Classifying semantic relations in bioscience texts". In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. ACL

'04. Barcelona, Spain: Association for Computational Linguistics. DOI: 10.3115/1218955.1219010. URL: {}https://doi.org/10.3115/1218955.1219010.

Rubenfeld, Jed (1989). "The right of privacy". In: *Harvard Law Review*, pp. 737–807.

Sahlgren, Magnus (2008). "The distributional hypothesis". In: *Italian Journal of Disability Studies* 20, pp. 33–53.

Samarati, Pierangela (2001). "Protecting Respondents' Identities in Microdata Release". In: *IEEE Transactions on Knowledge and Data Engineering* 13.8, pp. 1010–1027.

Samarati, Pierangela and Latanya Sweeney (1998). "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression". In:

Sánchez, David and Montserrat Batet (2016). "C-sanitized: A privacy model for document redaction and sanitization". In: *Journal of the Association for Information Science and Technology* 67.1, pp. 148–163.

Sánchez, David and Montserrat Batet (2017). "Toward sensitive document release with privacy guarantees". In: *Engineering Applications of Artificial Intelligence* 59, pp. 23–34.

Sánchez, David, Montserrat Batet, Aida Valls, and Karina Gibert (2010). "Ontology-driven web-based semantic similarity". In: *Journal of Intelligent Information Systems* 35.3, pp. 383–413.

Sánchez, David, Montserrat Batet, and Alexandre Viejo (2013a). "Automatic general-purpose sanitization of textual documents". In: *IEEE Transactions on Information Forensics and Security* 8.6, pp. 853–862.

Sánchez, David, Montserrat Batet, and Alexandre Viejo (2013b). "Minimizing the disclosure risk of semantic correlations in document sanitization". In: *Information Sciences* 249, pp. 110–123.

Sánchez, David, Josep Domingo-Ferrer, and Sergio Martínez (2019). "Co-utile disclosure of private data in social networks". In: *Information Sciences* 441, pp. 50–65.

Sánchez, David and Alexandre Viejo (2017). "Personalized privacy in open data sharing scenarios". In: *Online Information Review* 41.3, pp. 298–310.

Sang, Erik F and Jorn Veenstra (1999). "Representing text chunks". In: *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 173–179.

Schmidhuber, Jürgen and Sepp Hochreiter (1997). "Long short-term memory". In: *Neural Comput* 9.8, pp. 1735–1780.

Shilakes, Christopher C and Julie Tylman (1998). "Enterprise Information Portals, Merrill Lynch". In: *Inc., New York, NY*.

Spackman, Kent A (2004). "SNOMED CT milestones: endorsements are added to already-impressive standards credentials". In: *Healthcare Informatics* 21 (9), pp. 54–56.

Srivastava, Agrima and G Geethakumari (2013). "Measuring privacy leaks in online social networks". In: *2013 International Conference on Advances in Computing Communications and Informatics*. IEEE.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Stubbs, Amber, Christopher Kotfila, and Özlem Uzuner (2015). "Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/UTHealth shared task Track 1". In: *Journal of biomedical informatics* 58, S11–S19.

Suchanek, Fabian M, Gjergji Kasneci, and Gerhard Weikum (2007). "Yago - A Core of Semantic Knowledge Unifiying WordNet and Wikipedia". In: *Proceedings of the 16th International World Wide Web conference (WWW 2007)*, pp. 697–706.

Sundheim, Beth M. (1996). "Overview of the results of MUC-6 evaluation". In: *Proceedings of the TIPSTER Text Program: Phase II*. Vienna, Virginia, USA: Association for Computational Linguistics, pp. 423–442. DOI: 10.3115/1119018.1119073. URL: http://www.aclweb.org/anthology/X96-1048.

Sweeney, Latanya (1996). "Replacing personally-identifying information in medical records, the Scrub system". In: *Proceedings of the AMIA Annual Fall Symposium*.

Symantec Data Loss Prevention. https://www.symantec.com/products/dlp. last accessed: 24-Jan-2020.

Taylor, Ann, Mitchell Marcus, and Beatrice Santorini (2003). "The Penn treebank: an overview". In: *Treebanks*, pp. 5–22.

The GeoNames geographical database covers all countries (2006). https://www.geonames.org. last accessed: 04-Apr-2021.

Toutanova, Kristina, Dan Klein, Christopher D Manning, and Yoram Singer (2003). "Feature-rich part-of-speech tagging with a cyclic dependency network". In: *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 252–259.

Uzuner, Özlem, Yuan Luo, and Peter Szolovits (2007). "Evaluating the state-of-the-art in automatic de-identification". In: *Journal of the American Medical Informatics Association* 14.5, pp. 550–563.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *arXiv preprint arXiv:1706.03762*.

Westin, Alan F (1967). "Privacy and freedom Atheneum". In: *New York* 7.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. (2019). "HuggingFace's Transformers: State-of-the-art natural language processing". In: *arXiv preprint arXiv:1910.03771*.

Wu, Zhibiao and Martha Palmer (1994). "Verb semantics and lexical selection". In: *arXiv preprint cmp-lg/9406033*.

Yang, Hui and Jonathan M Garibaldi (2015). "Automatic detection of protected health information from clinic narratives". In: *Journal of biomedical informatics* 58, S30–S38.

Zhou, GuoDong and Jian Su (2002). "Named entity recognition using an HMM-based chunk tagger". In: *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 473–480.



UNIVERSITAT ROVIRA i VIRGILI