PhD-FSTM-2023-084

The Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 11/09/2023 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG EN INFORMATIQUE

by

Orlando AMARAL CEJAS

Born on 08 December 1988 in Ciudad de La Habana (Cuba)

ARTIFICIAL INTELLIGENCE-ENABLED AUTOMATION FOR COMPLIANCE CHECKING AGAINST GDPR

Dissertation defense committee

Dr. Sallam Abualhaija, Dissertation Supervisor
Research Scientist, University of Luxembourg

Dr. Lionel Briand, Chairman
Professor, University of Luxembourg

Dr. Domenico Bianculli, Vice Chairman
Professor, University of Luxembourg

Dr. Paola Spoletini, Member
Professor, Kennesaw State University

Dr. Alessio Ferrari, Member
Research Scientist, National Research Council, Italy

Acknowledgement

I'm extremely grateful to my supervisors for their constructive criticism, helpful advice, unwavering support and extensive professional and personal guidance. Without doubt, during these years of intense work, they were crucial to my success as a doctoral researcher and helped me to become a better professional.

I would like to express my deepest appreciation to the committee members for accepting the invitation to my dissertation and for all the insightful suggestions and comments.

I must also extend my sincere gratitude to all my co-authors for their exceptional attitude and brilliant contributions to this dissertation.

I had the great pleasure of working with our industrial partners Linklaters LLP. I must thank them for the great professionalism, extensive expertise, and invaluable help conveyed during our collaboration.

I cannot forget to mention the Luxembourg's National Research Fund (FNR), and the Natural Sciences and Engineering Research Council of Canada that financially supported this dissertation.

I would like to extend my most sincere thanks to all the members of the SVV group in particular and SnT in general with whom I have had the pleasure to interact and exchange ideas during all these years.

Special thanks go to my family, specially my parents and my wife. They are an essential part of my life. The completion of this dissertation would not have been possible without their support. They give me unparalleled support, have a profound belief in me, and provide me with encouragement and patience throughout every day of my life.

Finally, I would like to thank everyone who in any way helped me in my endeavors throughout my life, not only academically but in all spheres of life.

Orlando Amaral Cejas
University of Luxembourg
September 2023

Abstract

Requirements engineering (RE) is concerned with eliciting legal requirements from applicable regulations to enable developing legally compliant software. Current software systems rely heavily on data, some of which can be confidential, personal, or sensitive. To address the growing concerns about data protection and privacy, the general data protection regulation (GDPR) has been introduced in the European Union (EU). Organizations, whether based in the EU or not, must comply with GDPR as long as they collect or process personal data of EU residents. Breaching GDPR can be charged with large fines reaching up to up to billions of euros. Privacy policies (PPs) and data processing agreements (DPAs) are documents regulated by GDPR to ensure, among other things, secure collection and processing of personal data. Such regulated documents can be used to elicit legal requirements that are inline with the organizations' data protection policies. As a prerequisite to elicit a complete set of legal requirements, however, these documents must be compliant with GDPR. Checking the compliance of regulated documents entirely manually is a laborious and error-prone task. As we elaborate below, this dissertation investigates utilizing artificial intelligence (AI) technologies to provide automated support for compliance checking against GDPR.

- **AI-enabled Automation for Compliance Checking of PPs:** PPs are technical documents stating the multiple privacy-related requirements that a system should satisfy in order to help individuals make informed decisions about sharing their personal data. We devise an automated solution that leverages natural language processing (NLP) and machine learning (ML), two sub-fields of AI, for checking the compliance of PPs against the applicable provisions in GDPR. Specifically, we create a comprehensive conceptual model capturing all information types pertinent to PPs and we further define a set of compliance criteria for the automated compliance checking of PPs.

- **NLP-based Automation for Compliance Checking of DPAs:** DPAs are legally binding agreements between different organizations involved in the collection and processing of personal data to ensure that personal data remains protected. Using NLP semantic analysis technologies, we develop an automated solution that checks at phrasal-level the compliance of DPAs against GDPR. Our solution is able to provide not only a compliance assessment, but also detailed recommendations about avoiding GDPR violations.

- **ML-enabled Automation for Compliance Checking of DPAs:** To understand how different representations of GDPR requirements and different enabling technologies fare against one another, we develop an automated solution that utilizes a combination of conceptual modeling and ML. We further empirically compare the resulting solution with our previously proposed solution, which uses natural language to represent GDPR requirements and leverages rules alongside NLP semantic analysis for the automated support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this initial chapter we introduce the context, contributions, and organization of this dissertation.

1.1 Context

Legal requirements refer to the obligations stipulated in the regulations with which organizations and their software systems must comply [1]. Modern software systems are becoming more and more complex and data-driven [2–7], i.e., they collect, process and consume a huge amount of data, some of which (e.g., biometric data) can be confidential and sensitive. The rapid integration of such systems in various industries raised concerns about privacy and data protection, cybersecurity risks, and consumer rights. Regulations are being continuously introduced to address these concerns. In Europe, the general data protection regulation (GDPR) [8] is considered as the benchmark for privacy and data protection. While GDPR provides individuals with far-reaching capabilities, organizations are finding it very difficult and resource-expensive to understand what it means to comply with GDPR and how to implement its legal requirements into their software systems. Any organization, whether based in the European Union (EU) or not, is subject to compliance with GDPR as long as it collects and processes data in the EU. Violating GDPR can result in large fines reaching up to billions of euros. A recent example of violating GDPR is that where Meta Platforms Ireland Limited (Meta IE) was issued a fine of €1.2 billion following an inquiry into its Facebook service. The fine was due to the violation of the personal data transfer requirements in GDPR [9]. GDPR imposes different obligations onto organizations depending on what they do with the personal data. An organization that is subject to compliance with GDPR has to identify itself as either a *data controller* or *data processor*. The controller determines the purpose and means of the processing, whereas the processor acts on the instructions of the data controller. Both organizations should collaborate to ensure the protection of individuals' personal data. The responsibilities of each organization are often outlined in specific legal documents as we will elaborate later in this dissertation.

Requirements engineering (RE) is a sub-field of software engineering that is concerned with eliciting, documenting, and validating requirements which, in turn, are statements that describe what a system-to-be should do and how it should do it [10]. As part of the elicitation activities, requirements engineers have to deal with legal requirements (also referred to as compliance requirements). Specifying and properly implementing legal requirements would enable the development of compliant software systems, avoiding thereby serious consequences. Eliciting legal requirements from regulations like GDPR is challenging due to several factors, mostly bound to the complexity of legal language and multiple legal interpretation [11–13]. This complexity makes regulations difficult to comprehend, hindering thereby the extraction of necessary obligations and constraints by non-experts. Regulations also contain generic rules targeting an entire industry and requiring adaptation to a specific application context. Translating such generic rules into practical requirements customized to software systems can be a complex and nuanced task [13]. These factors emphasize the necessity for collaboration between legal experts and requirements engineers to interpret the regulation and elicit a complete set of requirements that can contribute to developing compliant software systems. Moreover, requirements engineers can rely on regulated documents such as privacy policies to elicit requirements. Unlike general regulations, these documents provide a focused set of requirements that capture the specific policies of organizations, and must therefore be satisfied. Ensuring the compliance of such regulated documents against applicable laws is a prerequisite for using such regulated documents in the elicitation activity in addition to avoiding breaches.

This dissertation investigates automated means that help human analysts (both requirements engineers and legal experts) assess the compliance of regulated documents. Specifically, the dissertation focuses on checking the compliance of two types of documents, namely privacy policies (PPs) and data processing agreements (DPAs), according to the GDPR provisions. We define *compliance checking* throughout this dissertation as the process of analyzing the textual content of a legal document and assessing whether this content complies with what is required in GDPR. The purpose of this analysis as mentioned above is to help requirements engineers elicit a complete set of legal requirements that are essential for developing GDPR compliant software systems. The work described in this dissertation is in collaboration with Linklaters LLP, a multinational law firm headquartered in London with a branch in Luxembourg. Linklaters has top-tier rankings across many legal-practice areas, including funds investment, banking, and finance. The legal experts in Linklaters have long experience in providing legal advisory and compliance services. With their help, we created the different representations of the legal knowledge in GDPR. The reason for focusing on compliance checking of PPs and DPAs is that both documents provide information related to data privacy and personal data processing, major activities in most of current software systems. According to our collaborating experts, these two documents are also very frequently checked for compliance by legal experts. PPs are exposed to the individuals who would be potentially using such software systems. A PP should contain enough information to help individual understand the terms based on which personal data is collected and processed. DPAs are not exposed to individuals, yet they capture the obligations and rights of the organizations that are involved in the data processing activities. DPAs ensure that individuals' personal data remains protected. Consequently, PPs and DPAs contain different sets of legal requirements, which must be adhered to by different actors.

The solutions presented in this dissertation address two main challenges, outlined below.

C1 Creating a machine-analyzable representation of GDPR: GDPR is a complex, 88 pages long legal document composed by several recitals (173), articles (99) and chapters (11). Encoding the GDPR provisions related to privacy and data processing alongside the compliance procedure into a machine-analyzable representation is paramount for building a comprehensive and consistent understanding of GDPR, and for enabling automated compliance analysis.

C2 Develop an automated support for GDPR compliance checking: A fully manual verification to ascertain GDPR compliance is time and effort consuming, and can be error-prone. An automated approach for compliance checking is thus advantageous. Effective automation to check for compliance requirements must be carefully selected taking into account the representation of the legal knowledge, the current state of the art in text processing, and potential future changes that might be introduced in the regulations.

In the RE literature, compliance against GDPR has been widely studied in the past few years [14–18], with a clear emphasis on analyzing privacy policies [19–22]. DPAs, on the other hand, received little attention in RE. Our work in this dissertation builds on existing literature, yet differs in the following aspects. First, we show how to deal with the complexity of legal text through a comprehensive hierarchical conceptual model that describes the information required in the regulation. We further demonstrate the effectiveness of hybrid text classification methods to accurately categorize the textual content of a given legal document with respect to the hierarchical model. Second, we consider DPAs, another essential legal document that regulates the data processing activities in software systems. Third, we propose methods that are capable of analyzing text at phrasal level for checking the compliance of a given document and further providing detailed recommendations towards achieving compliance. Finally, we provide insights based on empirical investigation about the advantages and disadvantages of two different representation methods and two automated techniques (one of which relies primarily on machine learning).

1.2 Contributions and Organization

In this dissertation, we present three automated solutions structured in three Chapters, as shown in Figure 1.1.

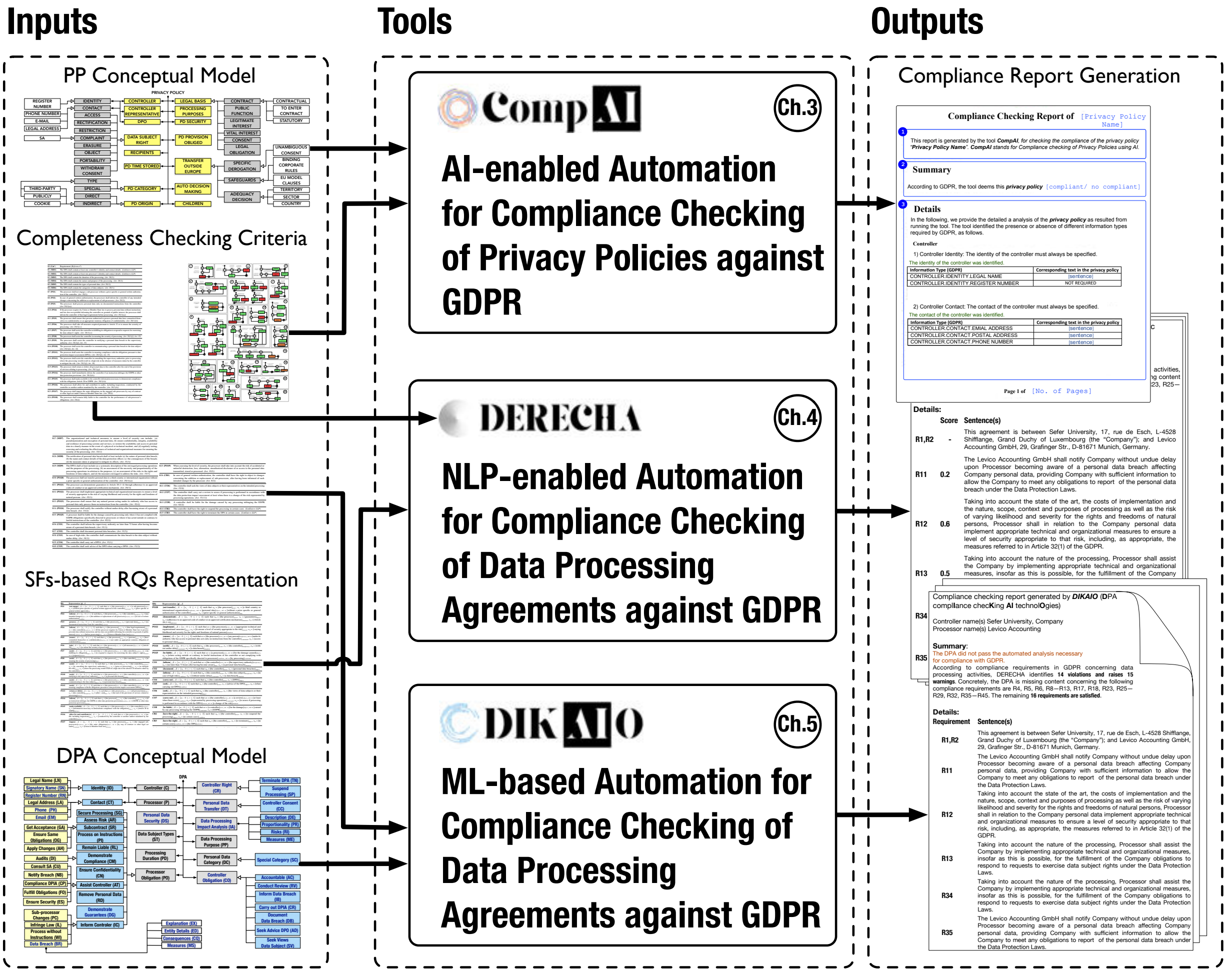

Inputs
Tools
Outputs
PP Conceptual Model
Completeness Checking Criteria
SFs-based RQs Representation
DPA Conceptual Model
CompAI
Ch.3
AI-enabled Automation for Compliance Checking of Privacy Policies against GDPR
DERECHA
Ch.4
NLP-enabled Automation for Compliance Checking of Data Processing Agreements against GDPR
DIKAIO
Ch.5
ML-based Automation for Compliance Checking of Data Processing Agreements against GDPR
Compliance Report Generation
Compliance Checking Report of [Privacy Policy Name]
Summary
Details
Compliance checking report generated by DIKAIO (DPA compliance checKing AI technolOgies)
Summary:
Details:

Figure 1.1: Dissertation overview.

- **Chapter 3: AI-enabled Automation for Compliance Checking of PPs against GDPR.** In this chapter, we propose an AI-based solution for checking the compliance of PPs against the GDPR provisions. Through systematic qualitative methods, we first build two artifacts to characterize the privacy-related provisions of GDPR, namely a conceptual model and a set of compliance criteria. The conceptual model consists of 56 information types, i.e., all GDPR-relevant information content that any PP may contain to be compliant with GDPR. Examples of information types include the rights that individuals have over their personal data such as the right to access and rectify personal data. Using these information types, we define a set of 23 compliance criteria that enable checking whether a PP is compliant according to GDPR, and represent these criteria as activity diagrams. We further develop an automated approach by leveraging a combination of natural language processing (NLP) and supervised machine learning (ML). Our solution, $Comp$liance Checking for PPs using $A\iota$ ($CompA\iota$), identifies in a given PP the information types from the conceptual model. Based on the identified information types in the PP, $CompA\iota$ applies the compliance criteria to detect any possible violation (i.e., missing information type) that can cause non-compliance against GDPR. To evaluate $CompA\iota$, we collected and manually labeled 234 real PPs with the different information types in our conceptual model. Over a set of 48 unseen PPs, $CompA\iota$ correctly detects 300 out of 334 genuine violations, while producing 23 false violations. $CompA\iota$ has a precision of 92.9% and recall of 89.8%. This work has been published in the IEEE Transactions on Software Engineering Journal [23]. Concretely, the contributions of Chapter 3 are the following:
  - We create a comprehensive conceptual model to capture the content of PPs, as stipulated in the GDPR provisions.
  - We create a set of compliance criteria that state when a PP is considered compliant according to GDPR.
  - We develop $CompA\iota$, an automated approach for checking the compliance of PPs using artificial intelligence (AI) technologies.
  - We empirically evaluate our approach using a dataset of 234 real PPs.

- **Chapter 4: NLP-enabled Automation for Compliance Checking of DPAs against GDPR.** In this chapter, we propose an automated solution to check the compliance of a given DPA against GDPR. In close interaction with legal experts, we first build two artifacts: (i) a list of "shall" requirements extracted from the GDPR provisions relevant to DPA compliance and (ii) a glossary table defining the legal concepts in these requirements. Then, we develop an automated solution that leverages NLP technologies to check the compliance of a given DPA against these "shall" requirements. Specifically, our solution DPA SEmantic FRam$E$-based Compliance CHecking Against GDPR (`DERECHA`) automatically generates phrasal-level representations for the textual content of the DPA and compares them against predefined representations of the "shall" requirements. By comparing these two representations, the approach not only assesses whether the DPA is GDPR compliant but further provides recommendations about missing information in the DPA. To evaluate `DERECHA`, we collected and manually labeled 24 real DPAs with the list of "shall" requirements, extracted from GDPR. Over a set of 30 unseen DPAs, `DERECHA` correctly finds 618 out of 750 genuine violations while raising 76 false violations, and further correctly identifies 524 satisfied requirements. `DERECHA` has thus an average precision of 89.1% and a recall of 82.4%. This work has been published in the IEEE Transactions on Software Engineering Journal [24]. Concretely, the contributions of Chapter 4 are the following:

  - We extract, in close interaction with subject-matter experts, a set of 45 compliance rules from the GDPR provisions concerning data processing. We further document these rules as “shall” requirements.
  - We develop an automated approach that leverages NLP technologies for automatically checking the compliance of data processing agreement against GDPR.
  - We empirically evaluate our approach on a dataset of 54 real DPAs.
  - We propose assigning confidence scores to the identified content in the input DPA, thus enabling human analysts to effectively prioritize the content for review based on `DERECHA`’s output.

- **Chapter 5: ML-based Automation for Compliance Checking of DPAs against GDPR.** In this chapter, we propose an automated strategy based primarily on ML for checking GDPR compliance in DPAs. Specifically, we create, based on existing work, a comprehensive conceptual model that describes a total of 63 information types pertinent to DPA compliance. Examples of such information types include the data processor obligation to process personal data only on instructions from the data controller. Using these information types, we define a set of 37 compliance criteria that enable checking whether a DPA is compliant according to GDPR. We then develop an automated approach, thereafter referred to as $D\iota\kappa AIo$, to predict the presence of these information types in a given DPA and detect possible breaches of GDPR accordingly. $D\iota\kappa AIo$ stands for DPA Compl$I$ance Chec$K$ing using AI technol$O$gies. To evaluate $D\iota\kappa AIo$, we collected and manually labeled 180 real DPAs with the different information types in our conceptual model. Over a set of 30 unseen DPAs, $D\iota\kappa AIo$ correctly detects 483 out of 582 genuine violations while introducing 93 false violations, achieving thereby a precision of 83.9% and recall of 83.0%. We empirically compare $D\iota\kappa AIo$ against `DERECHA`. This work has been accepted for publication in the 31st IEEE International Requirements Engineering Conference [25]. Part of the conceptual model used in this work was presented in the 11th IEEE Model-Driven Requirements Engineering Workshop (MoDRE) [26]. Concretely, the contributions of Chapter 5 are the following:

  - We create, building on our previous work, a holistic representation of DPA-related requirements in the form of a conceptual model that contains a total of 63 information types capturing any content to be expected in a GDPR-compliant DPA.
  - We develop an automated approach that is primarily based on ML, to check the compliance of the textual content of data processing agreement against the conceptual model created from the GDPR provisions.
  - We empirically evaluate $D\iota\kappa AIo$ on a dataset, comprised of 180 real DPAs.

Chapter 2

Background

Below, we summarize the necessary background related to the work described in this dissertation.

2.1 The General Data Protection Regulation

The General Data Protection Regulation (GDPR) [8], put into effect in 2018, is considered as the benchmark for privacy and data protection in Europe. It consists of 173 recitals and 99 articles divided into 11 chapters. Every organization, whether Europe-based or not, must comply with GDPR as long as it collects or processes personal data of EU residents. The legal obligations stipulated in GDPR can vary depending on what the organization does with personal data. An organization that is subject to compliance with GDPR has to identify itself as either a data controller or data processor.

The data controller determines the purpose of the data processing, whereas a processor acts according to the instructions of the controller. Processors notably have to: (1) implement adequate technical and organizational measures to keep personal data safe and secure, and, in cases of data breaches, notify the controllers; (2) appoint a statutory data protection officer (if needed) and conduct a formal impact assessment for certain types of high-risk processing; (3) keep records about their data processing; and (4) comply with GDPR restrictions when transferring personal data outside Europe. In comparison to processors, controllers are subject to more provisions. In particular, in addition to having to meet the obligations mentioned above, controllers have to: (1) adhere to six core personal data processing principles, namely, fair and lawful processing, purpose limitation, data minimization, data accuracy, storage limitation, and data security; (2) keep identifiable individuals informed about how their personal data will be used; and (3) preserve the individual rights envisaged by GDPR, e.g., the right to be forgotten and the right to lodge a complaint.

2.1.1 Privacy Policies

A privacy policy (PP) is a legal document that outlines how the data controller collects, uses, processes, and protects personal data of individuals. It is a fundamental component of privacy governance and helps ensure compliance with data protection laws, including GDPR. A PP is a legally binding agreement between the controller and individuals (data subjects) to whom personal data relates. PPs convey to individuals the organization's data protection practices which must be in compliance with GDPR. The legal requirements extracted from a PP represent therefore the obligations of the data controller and their associated software platforms directly used by data subjects. From an RE stand point, a PP can be used as a reference document for capturing privacy-related requirements to ensure that the software system or service aligns with the the organization's privacy commitments and legal obligations.

2.1.2 Data Processing Agreements

A data processing agreement (DPA) refers to a legally binding contract between a data controller and a data processor. It sets out the terms and conditions under which the data processor processes personal data on behalf of the data controller. DPAs define the roles and responsibilities of both parties and establish the safeguards and obligations required for lawful and secure data processing. DPAs are an essential component of GDPR compliance, as they help ensure a lawful and secure processing of personal data and in accordance with the rights of data subjects. From an RE perspective, DPAs provide guidance for capturing the legal requirements for the system being developed with an emphasis on data processing activities. A DPA helps define the technical and organizational measures that must be implemented to meet the data protection requirements outlined in GDPR. Legal requirements extracted from DPAs primarily focus on the obligations of the data processor, and which are in turn distinct from the ones stated in PPs.

2.2 Natural Language Processing

Natural language processing (NLP) is a sub-field of artificial intelligence (AI), which is used for automatically processing natural language data. Examples of NLP applications include machine translation and information extraction [27, 28]. Fig. 2.1 presents a comprehensive NLP pipeline that combines seven modules divided into four categories.

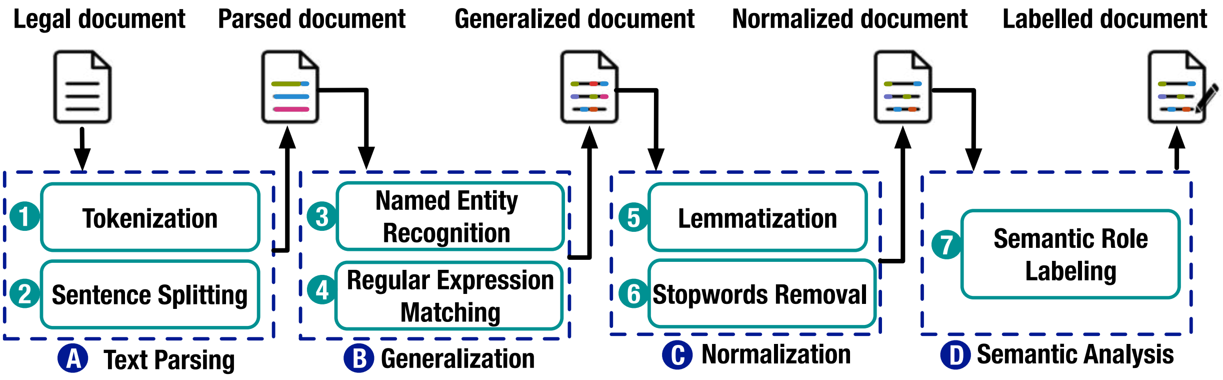


Figure 2.1: A comprehensive NLP pipeline.

The first category in the pipeline, *Text Parsing*, aims at parsing the text of a given legal document. This category includes *Tokenization* for separating out the words and punctuation marks from the running text and *Sentence Splitting* for decomposing the text into coherent sentences based on sentence boundary indicators such as periods, question and exclamation marks [28, 29].

The second category, *Generalization*, is concerned with generalizing the specific entities in the text. We generalize the text in each sentence by replacing specific textual entities with more generic ones. Specifically, this category uses *Named Entity Recognition (NER)*, which is the module of marking the mentions of named entities in a given text with their types [30], e.g., a country name like "Luxembourg" will be annotated with the type *LOCATION*. The entity types, in our work, are limited to *location* and *organization* since these two are expected to appear often in the legal documents analyzed in this dissertation. In addition to the NER module, we use regular expressions [31] to recognize the contact details that are mentioned in a given document, namely *email address*, *postal address*, *telephone numbers* and *websites*. For example, the email address "info@hikari.jp" will be recognized and replaced with its type, *EMAIL*.

The third category, *Normalization*, is concerned with normalizing the text. In particular, the *Lemmatization* module identifies the canonical form of the different words in a text, e.g., the words "deletion", "deleted" and "delete" will be lemmatized to the canonical form "delete". The *Stopwords Removal* modules further removes stopwords, i.e., very frequent words such as prepositions (e.g., "in") and articles (e.g., "a" and "the").

The last category, *Semantic Analysis*, is concerned with analyzing the semantics of the text. Specifically, the semantic role labeling module provides a label to the phrases in a text about its semantic role with respect to a particular event described in the text [32]. For instance, the *purchase event* in the sentence "William purchased a brand new car" contains the semantic roles *buyer* which describes "William" and *purchased item* describing "a brand new car".

2.3 Machine Learning

Machine learning (ML) is another sub-field of AI which describes the automated learning methods used for finding meaningful patterns in data [33–35]. Supervised ML assumes that training examples (input) are provided with their labels (output). Using these training examples, the machine then learns to predict the output of unseen examples. We will refer to the input and its associated output value as a *classification instance*. Text classification (also known as text categorization) is supervised learning for categorizing the text into a set of predefined groups [36], e.g., classifying the text of an email into *spam* and *not spam*. In this dissertation, we experiment with widely used ML algorithms such as decision trees [37], random forests [38], support vector machine [39], logistic regression [40], linear discriminant analysis [41], and neural networks [42]. ML can be used for a wide range of tasks, such as classification, regression, anomaly detection, recommendation systems, and NLP. This last one is the most relevant application for our research work. Supervised ML is widely used in various applications of NLP including sentiment analysis [43, 44], question answering [45, 46], text summarization [47, 48], machine translation [49, 50], named entity recognition [51, 52], and text classification [53, 54].

In this dissertation, we focus on *multiclass multilabel classification*. Multiclass classification is to classify the input examples into three or more predefined classes. A classical example in the ML literature is classifying an iris flower, given its sepal length and width and petal length and width, into one of the three possible types *setosa*, *versicolor*, or *virginica* [34]. Multilabel classification means that the same input example may belong to multiple classes, e.g., classifying movies into one or more genres based on the plot summary, where a movie can belong to *comedy* and *action* at the same time. The multilabel classification problem is often simplified into multiple binary classification problems [36]. A binary classification is a specific case of the multiclass classification with only two target classes. For example, a movie can be classified into genres using multiple learning algorithms, such that each learner predicts whether the movie is from a specific genre (e.g., *comedy*) or not from that genre (e.g., *not comedy*), and so on for the other genres.

2.3.1 Learning Features

Vectorization is a prerequisite step to text classification where the text has to be transformed into a set of feature vectors (i.e., learning features) that describe the text under the different pre-defined classes [36, 55]. Each classification instance is represented by a feature vector. These features can be either manually crafted (e.g., the presence of a first person pronoun like "we") or automatically generated using the words in the text. There are several models to perform vectorization, e.g., bags of words (BoW) that represents words by their frequency of occurrence [56] and TF/IDF (term frequency/inverse document frequency) that uses the frequency of words to determine how relevant those words are to a given document [57]. More advanced representation methods apply embeddings which are mathematical representations (also known as dense vectors) that encapsulate the syntactic and semantic characteristics of the text [58]. Several methods have been proposed for deriving words and sentence embeddings. In our work, we experiment with the pre-trained word embeddings from word2vec [59], GloVe [60], and fasttext [61]. These embeddings are context-independent, i.e., a word has always the same representation irrespective of the context in which it appears. For example, the word "bank" will have the same embedding regardless if it means "a financial institution" or "the side of a river". Moreover, regularities can be observed in the linear relations between word pairs. For example, if a word w is represented by the vector $\vec{w}$, then one can observe the plural relation between the embeddings: $c\vec{a}t - c\vec{at}s \approx ap\vec{p}le - app\vec{l}es$.

Pre-trained embeddings generated by training on extensive text corpora from Wikipedia and the web are widely used in various NLP tasks [62–69]. In modern NLP, pre-trained word embeddings perform better than those learned from scratch [70]. The pre-trained embeddings we apply in our work represent 100-dimensional and 300-dimensional vectors. To illustrate, consider the text segment "Hikari Bank Privacy Policy". Using pre-trained word embeddings, each word is represented as a 100-dimensional vector, e.g., "hikari" is represented as $[0.42192, 0.41032, 0.23888, \ldots]_{100}$. To compute the sentence embeddings we take the average of the words embeddings in that sentence. In our work, we use simple averaging [71–73] because it proved to be effective in text similarity-related tasks. While such representations might be limited in terms of semantic capabilities, they are highly flexible since the embeddings can be efficiently extracted for any text. Compared to more recent technologies for generating text representations like ELMo [74], OpenAI GPT [75] and BERT [76], the pre-trained embeddings used in this work provide context-independent word embeddings (i.e., one-to-one mapping between the words and their vectors) that can be directly used off-the-shelf. Contextualized embeddings are generated for each word depending on the context in which it appears. Such methods rely on recent large-scale language models. In our work, we experiment with the sentence embeddings generated by sentence-BERT (SBERT) [77]. BERT [76] is a language model that has been introduced by Google and it has outperformed the state-of-the-art models and hence dominated (with its variants) the NLP landscape. SBERT, a variant of BERT, is a language model optimized to learn semantically meaningful representations for sentences.

2.3.2 Data Imbalance Handling

Data imbalance occurs when one class has significantly fewer training examples (i.e., under-represented) than the other class. Imbalance can lead to building classifiers that mispredict in favor of the majority class [78]. Inspired by existing work [79], we apply in our work a combination of undersampling and oversampling to accurately predict the minority (under-represented) class. For oversampling the minority class, we apply the widely-used synthetic minority oversampling technique (SMOTE) [80]. In brief, SMOTE creates synthetic examples using the k-nearest-neighbor approach [34]. For removing data points from the majority class we apply Random Undersampling [81].

Chapter 3

AI-enabled Automation for Compliance Checking of Privacy Policies against GDPR

Privacy policies (PPs) play a major role in software development, as they contain privacy-related requirements about how the personal data of individuals will be handled by an organization or a software system (e.g., a web service or an app). For instance, a privacy policy (PP) states among other things information about how the software system collects personal data, what personal data is being collected, for what purpose, and for how long, with whom personal data will be shared, and what rights individuals have over their personal data, etc. A PP is subject to compliance with the general data protection regulation (GDPR) when it involves data collected from EU residents. Non-compliant PPs might result in large fines due to violating GDPR as well as possibly incomplete privacy-related requirements specifications. Checking compliance entirely manually is both time-consuming and error-prone. Providing automated support is thus desirable so that legal experts can focus their effort on more critical tasks.

In this chapter, we propose AI-based automated support for checking the compliance of PPs. Specifically, we devise an approach that leverages natural language processing (NLP) and machine learning (ML) for automatically identifying whether the textual content of a given PP complies with the GDPR provisions. To do so, we first create a comprehensive conceptual model which describes all information types that any PP might contain according to GDPR. We subsequently train ML classifiers to categorize the content of a PP based on these information types. We then define a set of compliance criteria to automatically check whether the PP meets the requirements envisaged by GDPR.

Structure. The remainder of this chapter is structured as follows: Section 3.1 presents the motivation and research contributions of this chapter. Section 3.2 outlines the investigated research questions. Section 3.3 presents the qualitative study we conducted for building our privacy-policy conceptual model. Section 3.4 describes the methods we used to create a set of criteria for checking the compliance of PPs according to GDPR. Section 3.5 explains our proposed AI-based approach for automatically checking the compliance of a given PP. Section 3.6 reports on our empirical evaluation. Section 3.7 positions our work against the related literature. Section 3.8 discusses threats to validity. Section 3.9 presents an interview survey with the subject-matter experts to assess the usefulness of our approach in practice. Section 3.10 describes how we envision our overall approach being replicated beyond GDPR. Finally, Section 3.11 concludes the chapter.

3.1 Motivation and Contributions

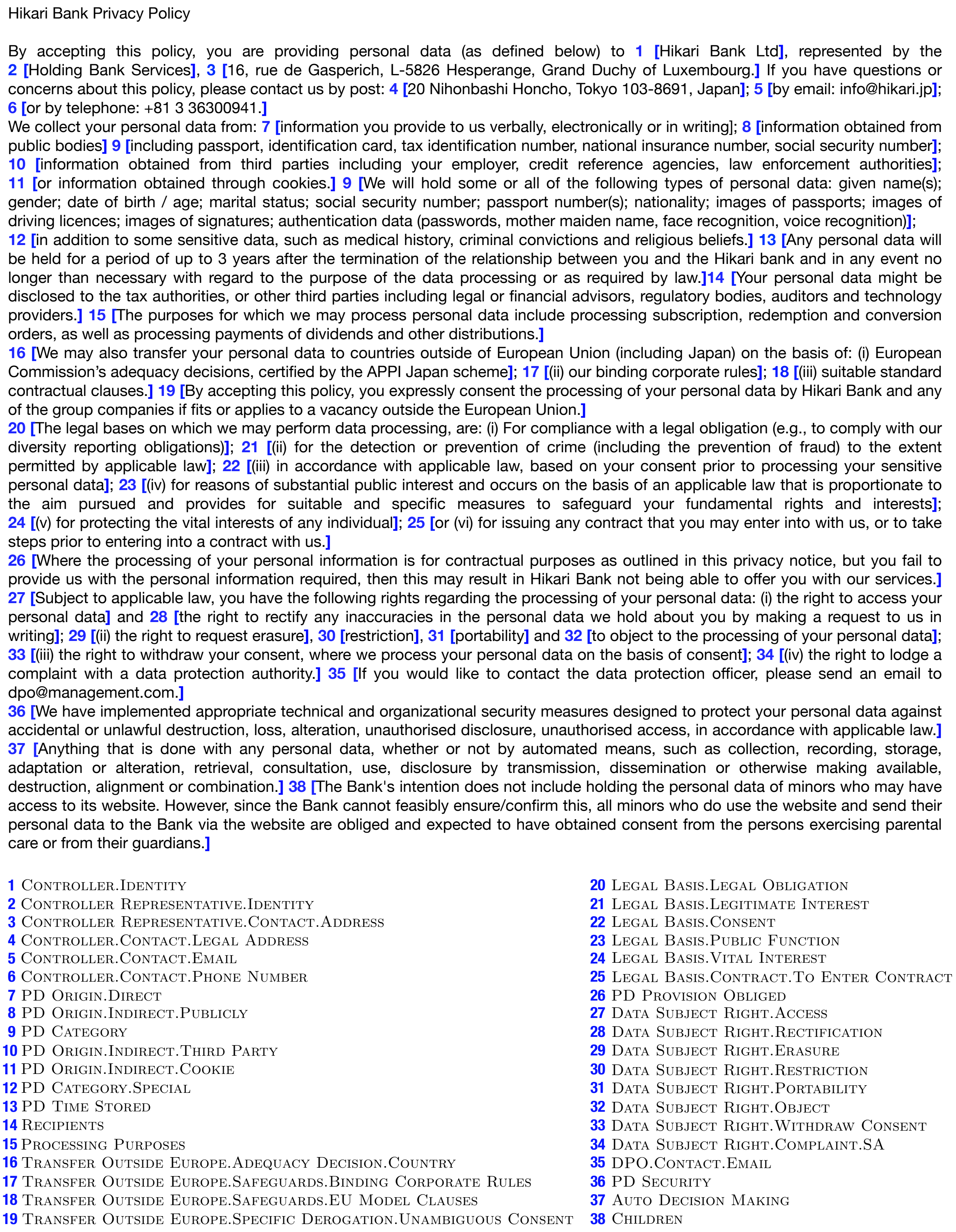
To comply with GDPR, organizations need to take into account the principles of personal data processing set out in the regulation, and to regularly review their measures, practices and processes related to the collection, use and protection of personal data. Compliance also entails that software systems storing or processing personal data should properly implement privacy-related GDPR requirements. To help different organizations better deal with GDPR-relevant privacy considerations, cost-effective methods have been proposed in the literature. For example, Perrera et al. [82] propose systematic guidance to help software engineers develop privacy-aware applications; Torre et al. [83, 84] propose the use of model-driven engineering as a basis for GDPR compliance automation; and Ayala-Rivera and Pasquale [85] present a step-wise approach for eliciting requirements related to GDPR compliance.

In this chapter, we focus on PPs and their compliance according to GDPR. PPs are usually defined through natural-language statements. Natural language (NL) is an ideal medium for expressing PPs since it is flexible and universal [86]. Though NL is advantageous for establishing a common understanding, processing NL documents is challenging due to common quality issues such as ambiguity, incompleteness and inconsistency [87]. As explained in Section 2.1, a PP can be viewed as a technical document stating the multiple privacy-related requirements that an organization (including processes, services, developed systems) should satisfy in order to help users make informed decisions about the data that this organization may collect and use. In other words, a PP explains how an organization handles personal data and how it applies the principles of GDPR. A PP is considered to be GDPR compliant if it explicitly contains all mandatory content for ensuring data protection and privacy rights, e.g., about the rights individuals have over their personal data.

3.1.1 Practical Scenario

In practice, compliance checking of PPs against GDPR can be beneficial to a diverse group of legal experts, software engineers, and other business stakeholders. The first step in compliance checking is to determine if GDPR-relevant information content is present or not in a given PP. Based on this content analysis, the second step is then to map what is actually present in the PP to what must be present according to the provisions of GDPR. In the rest of this chapter, we will use the term *information type* to describe concepts extracted from the privacy-related provisions of GDPR. Some of these information types are mandatory and thus have a direct impact on compliance. We elaborate how we combine the information types for checking the compliance of a PP in Section 3.4. A comprehensive description of these information types is provided in Section 3.3.

Examples of information types include: PROCESSING PURPOSES to characterize the purposes of the processing for which personal data is being collected, LEGAL BASIS to capture the legal basis for the processing of personal data, and DATA SUBJECT RIGHT to mark the clause(s) giving an individual the rights in relation with their personal data. Under DATA SUBJECT RIGHT, several specializations are listed to describe the different rights an individual has. For instance, DATA SUBJECT RIGHT.ACCESS is concerned with the right to request access to the personal data from the controller. The specializations of the information types are represented, throughout the chapter, with a *dot*. Fig. 3.1 shows a full PP that is annotated with all information types (from Section 3.3). In the figure, we present the information types using numbers (further explained in the legend), and



Hikari Bank Privacy Policy

By accepting this policy, you are providing personal data (as defined below) to **1** **[**Hikari Bank Ltd**]**, represented by the **2** **[**Holding Bank Services**]**, **3** **[**16, rue de Gasperich, L-5826 Hesperange, Grand Duchy of Luxembourg.**]** If you have questions or concerns about this policy, please contact us by post: **4** **[**20 Nihonbashi Honcho, Tokyo 103-8691, Japan**]**; **5** **[**by email: info@hikari.jp**]**; **6** **[**or by telephone: +81 3 36300941.**]**
We collect your personal data from: **7** **[**information you provide to us verbally, electronically or in writing]; **8** **[**information obtained from public bodies**]** **9** **[**including passport, identification card, tax identification number, national insurance number, social security number**]**; **10** **[**information obtained from third parties including your employer, credit reference agencies, law enforcement authorities**]**; **11** **[**or information obtained through cookies.**]** **9** **[**We will hold some or all of the following types of personal data: given name(s); gender; date of birth / age; marital status; social security number; passport number(s); nationality; images of passports; images of driving licences; images of signatures; authentication data (passwords, mother maiden name, face recognition, voice recognition)**]**;
12 **[**in addition to some sensitive data, such as medical history, criminal convictions and religious beliefs.**]** **13** **[**Any personal data will be held for a period of up to 3 years after the termination of the relationship between you and the Hikari bank and in any event no longer than necessary with regard to the purpose of the data processing or as required by law.**]14** **[**Your personal data might be disclosed to the tax authorities, or other third parties including legal or financial advisors, regulatory bodies, auditors and technology providers.**]** **15** **[**The purposes for which we may process personal data include processing subscription, redemption and conversion orders, as well as processing payments of dividends and other distributions.**]**
16 **[**We may also transfer your personal data to countries outside of European Union (including Japan) on the basis of: (i) European Commission's adequacy decisions, certified by the APPI Japan scheme**]**; **17** **[**(ii) our binding corporate rules**]**; **18** **[**(iii) suitable standard contractual clauses.**]** **19** **[**By accepting this policy, you expressly consent the processing of your personal data by Hikari Bank and any of the group companies if fits or applies to a vacancy outside the European Union.**]**
20 **[**The legal bases on which we may perform data processing, are: (i) For compliance with a legal obligation (e.g., to comply with our diversity reporting obligations)**]**; **21** **[**(ii) for the detection or prevention of crime (including the prevention of fraud) to the extent permitted by applicable law**]**; **22** **[**(iii) in accordance with applicable law, based on your consent prior to processing your sensitive personal data**]**; **23** **[**(iv) for reasons of substantial public interest and occurs on the basis of an applicable law that is proportionate to the aim pursued and provides for suitable and specific measures to safeguard your fundamental rights and interests**]**; **24** **[**(v) for protecting the vital interests of any individual**]**; **25** **[**or (vi) for issuing any contract that you may enter into with us, or to take steps prior to entering into a contract with us.**]**
26 **[**Where the processing of your personal information is for contractual purposes as outlined in this privacy notice, but you fail to provide us with the personal information required, then this may result in Hikari Bank not being able to offer you with our services.**]**
27 **[**Subject to applicable law, you have the following rights regarding the processing of your personal data: (i) the right to access your personal data**]** and **28** **[**the right to rectify any inaccuracies in the personal data we hold about you by making a request to us in writing**]**; **29** **[**(ii) the right to request erasure**]**, **30** **[**restriction**]**, **31** **[**portability**]** and **32** **[**to object to the processing of your personal data**]**; **33** **[**(iii) the right to withdraw your consent, where we process your personal data on the basis of consent**]**; **34** **[**(iv) the right to lodge a complaint with a data protection authority.**]** **35** **[**If you would like to contact the data protection officer, please send an email to dpo@management.com.**]**
36 **[**We have implemented appropriate technical and organizational security measures designed to protect your personal data against accidental or unlawful destruction, loss, alteration, unauthorised disclosure, unauthorised access, in accordance with applicable law.**]**
37 **[**Anything that is done with any personal data, whether or not by automated means, such as collection, recording, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, destruction, alignment or combination.**]** **38** **[**The Bank's intention does not include holding the personal data of minors who may have access to its website. However, since the Bank cannot feasibly ensure/confirm this, all minors who do use the website and send their personal data to the Bank via the website are obliged and expected to have obtained consent from the persons exercising parental care or from their guardians.**]**

1 Controller.Identity
2 Controller Representative.Identity
3 Controller Representative.Contact.Address
4 Controller.Contact.Legal Address
5 Controller.Contact.Email
6 Controller.Contact.Phone Number
7 PD Origin.Direct
8 PD Origin.Indirect.Publicly
9 PD Category
10 PD Origin.Indirect.Third Party
11 PD Origin.Indirect.Cookie
12 PD Category.Special
13 PD Time Stored
14 Recipients
15 Processing Purposes
16 Transfer Outside Europe.Adequacy Decision.Country
17 Transfer Outside Europe.Safeguards.Binding Corporate Rules
18 Transfer Outside Europe.Safeguards.EU Model Clauses
19 Transfer Outside Europe.Specific Derogation.Unambiguous Consent
20 Legal Basis.Legal Obligation
21 Legal Basis.Legitimate Interest
22 Legal Basis.Consent
23 Legal Basis.Public Function
24 Legal Basis.Vital Interest
25 Legal Basis.Contract.To Enter Contract
26 PD Provision Obliged
27 Data Subject Right.Access
28 Data Subject Right.Rectification
29 Data Subject Right.Erasure
30 Data Subject Right.Restriction
31 Data Subject Right.Portability
32 Data Subject Right.Object
33 Data Subject Right.Withdraw Consent
34 Data Subject Right.Complaint.SA
35 DPO.Contact.Email
36 PD Security
37 Auto Decision Making
38 Children

Figure 3.1: Example of a fully annotated PP.

square brackets to delineate the text corresponding to the information types. For example, number **15** in Fig. 3.1 refers to the information type Processing Purposes, numbers **20 – 25** refer to different specializations of Legal Basis, and number **27** refers to Data Subject Right.Access.

To deem the example PP in Fig. 3.1 compliant, GDPR requires the presence of multiple mandatory information types, including the ones concerning CONTROLLER, i.e., the organization which collects personal data (GDPR, Art. 13 and Art. 14(f)). In particular, the policy should include the identity (i.e., CONTROLLER.IDENTITY) and contact details of the controller (i.e., CONTROLLER.CONTACT). As we see in Fig. 3.1, these two information types are mentioned respectively in number **1**, and numbers **4** – **6**. Checking the presence of the information types about CONTROLLER is however not sufficient, and verification of other information types is needed in order to make the final decision as to whether the PP is compliant according to GDPR.

Legal provisions in GDPR can contain requirements which depend on one another. Consequently, the presence of certain information types in a PP may necessitate the presence of certain other information types in the policy. For instance, if a PP states that the legal basis for the processing of personal data is based on individual consent (i.e., LEGAL BASIS.CONSENT), then the right to withdraw this consent should be granted in the same policy (i.e., DATA SUBJECT RIGHT.WITHDRAW CONSENT). These information types correspond to two different GDPR articles, Art. 6.1(a) and Art. 13.2(c), respectively. Information related to these types can be found by reviewing paragraphs that are usually located in different parts of the PP. In Fig. 3.1, LEGAL BASIS.CONSENT is mentioned in the text: *in accordance with applicable law, based in your consent [...]*(number **22**), while DATA SUBJECT RIGHT.WITHDRAW CONSENT is mentioned in: *the right to withdraw your consent [...]* (number **33**). If done manually, this back-and-forth reviewing of the text requires a considerable amount of effort and time in practice.

Checking the compliance of a given PP according to GDPR is essential for ensuring the compliance of the privacy-related software requirements induced by the policy. To illustrate, consider our example in Fig. 3.1. Since the CONTROLLER (i.e., Hikari Bank Ltd – number **1**) is located in Japan, it is likely that the personal data of the bank's customers will be transferred outside the Europe. Articles 13.1(a), 13.1(f) and 14.1(f) in GDPR enforce requirements to ensure the protection of personal data, for example when transferred outside Europe. The implications of these articles are then two-fold. On one hand, the PP must provide information about the IDENTITY and CONTACT details of the CONTROLLER REPRESENTATIVE (who has to be located in Europe) – as shown in numbers **2** and **3**, respectively. The policy must also state the legal agreement that is in place for transferring data to Japan such as the Act on the Protection of Personal Information (APPI) – provided in number **16**. On the other hand, the software developed to handle such personal data (e.g., the online banking service) has to comply with Japan's data protection law. APPI-compliant software should provide a response to the individuals' requests in relation with their personal data within two weeks. Otherwise, an individual can sue the controller. PPs missing, for instance, information types related to the location of the controller (in this case, Japan) or to the legal agreement used for transferring data (i.e., APPI) fails to comply with GDPR. The missed information types will remain unknown for a software developer and might lead to developing a non-compliant system. Consequently, the organization could bear significant fines for violating data-protection rules.

More precisely, a PP can be considered as a form of legally binding requirements specification which describes some of the properties and functionalities of a system-to-be. Therefore, compliance checking of PPs, and identifying their information types as a primary step, can be seen as part of a broader solution to ensure legal compliance in information systems. In the software engineering (SE) literature, there have been attempts at mapping the text of a PP to the implementation of a given software application, as a method for detecting GDPR violations [88, 89]. For instance, based on what we argued earlier, the privacy-related requirement about answering an individual's request pertaining to their personal data has to be mapped onto some function in the developed software.

Similarly, other information types identified in PPs can play a major role in software development. Examples include PD SECURITY, PD TIME STORED, DATA SUBJECT RIGHT.ERASURE, LEGAL BASIS.CONSENT, and DATA SUBJECT RIGHT.WITHDRAW CONSENT. In response to PD SECURITY, the controller has to implement appropriate protection mechanisms during software development (e.g., using encryption) to avoid penalty charges for information leakage as stated in GDPR. Further, a software system has to automatically delete collected personal data according to the time limit specified in the PP (PD TIME STORED) or upon an individual's request (DATA SUBJECT RIGHT.ERASURE). When the consent of an individual is required for processing personal data (LEGAL BASIS.CONSENT), a software system has to implement a clear request procedure for consent where the individual takes an action to provide consent, e.g., by checking an "I agree" checkbox. As stipulated by GDPR, the system would also have to provide individuals with the possibility to withdraw this consent (DATA SUBJECT RIGHT.WITHDRAW CONSENT). The above examples show the benefits of compliance checking in different scenarios. Since checking compliance manually is time-consuming and effort-intensive, computer-assisted support for this task is advantageous.

A naive compliance-checking solution is to automatically find certain information types in a PP through searching for keywords that are commonly used to express them. Relying merely on keyword search is problematic due to several reasons. First, there are overlapping keywords among multiple information types. For example, the keyword "protect" can indicate three information types related to security, data protection office, and safeguards for transferring personal data outside of Europe. Second, some information types cannot be captured via keywords. For instance, the information type RECIPIENTS (i.e., the parties with which individual personal data is shared) is usually expressed in the PP as a list of diverse organizations (number **14** in Fig. 3.1). Since each PP can have a different list of RECIPIENTS, using keyword search is infeasible for identifying this information type. To illustrate, let us suppose that "third parties" is used as a keyword for identifying RECIPIENTS. Note that the same keyword can also be used to identify PD ORIGIN.INDIRECT.THIRD-PARTY. Searching for this keyword will result in missing all occurrences of RECIPIENTS that do not contain the keyword and falsely identifying some occurrences due to overlapping keywords. In addition to the limitations of keyword search, the problem of checking compliance raises several other challenges. A particular sentence can discuss one or more information types which can be described in a hierarchy based on the specializations introduced in GDPR. In other words, an automated solution should be able to predict multiple (hierarchical) labels (information types) for a given sentence in the PP. Inter-dependent information types (e.g., CONSENT and WITHDRAW CONSENT – discussed earlier) do not always occur consecutively in the PP. This means that successful compliance checking requires identifying all the related information types accurately.

3.1.2 Contributions

This research makes the following four contributions:

(1) We develop a conceptual model to characterize the content of PPs, as stated in the GDPR provisions. This conceptual model provides an abstract and yet precise set of information types that one can expect to find in PPs according to GDPR.

(2) We create a set of compliance criteria that describe when a PP is considered compliant according to GDPR. For creating these criteria (and also the conceptual model in (1)), we use systematic qualitative methods, as we further explain in this chapter.

(3) We develop an automated compliance checking approach using AI technologies. Specifically, we devise an approach based on NLP and ML for automatically identifying the content of a given PP. To do so, we rely

on the information types in the conceptual model developed in (1) as classification types. Given the identified information types, we subsequently use the compliance criteria created in (2) to automatically check whether a given policy meets the information requirements envisaged by GDPR.

(4) We empirically evaluate our approach using a dataset of 234 PPs. These policies collectively contain 19,847 sentences manually assigned (when applicable) to one or more of the information types from our conceptual model. The large majority (87%) of these assignments have been made by independent, third-party annotators. We use ≈80% of our dataset for developing our proposed solution and the remaining ≈20% for evaluation. On our evaluation set, our AI-based approach yields an average precision of 92.1% and recall of 95.3% in automatically identifying information types. Our compliance checking yields an average precision of 92.9% and an average recall of 89.8%. Compared to a baseline that uses keyword search, our approach leads to an overall average improvement of 24.5% in precision and 38% in recall when checking the compliance of PPs.

3.2 Research Questions

The chapter investigates the following six research questions (RQs):

RQ1: What are the information types required for checking the compliance of a PP according to GDPR? We answer RQ1 by building a conceptual model that specifies GDPR's information requirements for PPs. Our conceptual model, comprised of 56 information types, was developed in close collaboration with subject-matter experts. The concepts in this model are described in a glossary and are further traceable to the articles of GDPR.

RQ2: What are the criteria for checking whether a PP is compliant according to GDPR? Drawing on our conceptual model, to answer RQ2, we define a set of 23 criteria specifying what in a PP should be checked for compliance against GDPR. Violating any of these criteria might lead to non-compliance.

RQ3: How can PPs be automatically checked for compliance against GDPR? To answer RQ3, we use a combination of NLP and ML methods based on word embeddings and semantic similarity to develop an AI-based approach. Our approach identifies the different information types (from our conceptual model in RQ1) that are present in a PP and then checks these information types against the compliance criteria (derived in RQ2) using automated conditional expressions.

RQ4: How accurate is our proposed approach in identifying GDPR-relevant information types in PPs? RQ4 examines the accuracy of our information types identification approach. As we discuss in Section 3.6, we achieve an average precision of 92.1% and average recall of 95.3% on an evaluation set made up of 48 unseen PPs.

RQ5: How accurate is our approach in checking the compliance of PPs? In RQ5, we investigate the accuracy of our automated approach in checking the compliance of PPs according to the provisions of GDPR. Over the evaluation set, our approach successfully finds 300 out of 334 violations of the compliance criteria, while raising false alarms (false positives) in 23 cases. Our approach has thus a precision of 92.9% and a recall of 89.8%.

RQ6: Is our approach worthwhile compared to a simpler solution? In RQ6, we compare our AI-based approach to a baseline that uses only keyword search. Compared to this baseline and over our evaluation set, using AI-technologies improves the information types identification by an average precision of 26.9% and average recall of 5.2%. Our approach significantly improves the overall compliance checking of PPs by an average precision of 24.5% and average recall of 38%.

This chapter extends an existing work by Torre et al. [90] by providing a much more extensive empirical investigation in terms of the research questions, PPs used for evaluation, and information types covered by these policies. In particular, (1) we provide, through a concrete and detailed example, different scenarios where automated compliance checking turned out to be useful to a diverse group of people including lawyers and software engineers; (2) we include two more research questions: RQ2 for addressing the qualitative methods leading to the derivation of compliance criteria and RQ6 for comparing our approach to a simple, intuitive baseline; (3) we apply our AI-based approach for identifying all the 56 information types in a given PP instead of only 20; and (4) we improve our validation method to empirically evaluate our approach on 48 unseen PPs (≈20% of the entire dataset), instead of only 24.

3.3 A Conceptual Model of Privacy-Policy Information Types (RQ1)

In this section, we present the following artifacts to answer RQ1: (1) a conceptual model specifying, in a comprehensive manner, the information types pertinent to GDPR PPs; and (2) a glossary defining all necessary terms to better understand the conceptual model with traceability to GDPR articles. The artifacts were built using an iterative and incremental method following three main steps (see Fig. 3.2): (1) reading the articles of GDPR that address PPs, (2) creating and refining the artifacts, and (3) validating these artifacts with legal experts. The conceptual model (artifact 1) is shown in Fig. 3.3 and an excerpt of the glossary (artifact 2) is presented in Table 3.1. The complete glossary is provided as an online annex [91]. Building the artifacts took four iterations with each iteration requiring, on average, one month. We had several face-to-face and off-line validation sessions with legal experts. The sessions, which lasted between two and three hours each, collectively added up to approximately 30 hours. We conducted our validation sessions with three legal experts, namely (a) a senior lawyer with more than 30 years of experience in European and international laws; (b) a mid-career lawyer with more than 10 years of experience in law with a focus on the data protection and financial domains; and (c) an IT professional with more than 10 years of experience in the legal domain. Each validation session was attended by at least two legal experts. The discussions continued until the experts in attendance agreed that the model correctly reflected their interpretation of GDPR. We observed that the differing viewpoints and thus the deliberations between the legal experts centered primarily around the specializations in the conceptual model (e.g., the sub-information types of LEGAL BASIS.CONTRACT) and about how different information types should be inter-related (e.g., how PD ORIGIN.INDIRECT is related to PD CATEGORY.TYPE).

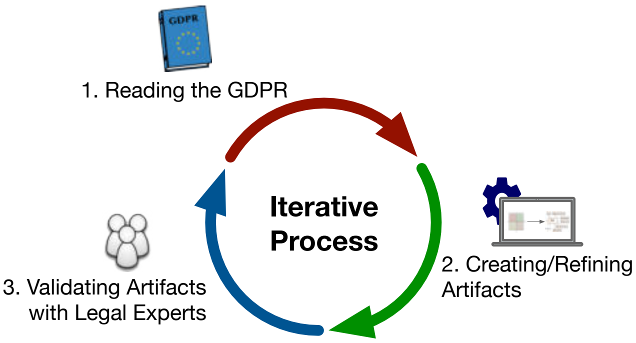

GDPR
1. Reading the GDPR
Iterative Process
2. Creating/Refining Artifacts
3. Validating Artifacts with Legal Experts

Figure 3.2: Iterative process.

Initially, as suggested by our collaborating legal experts from Linklaters, we analyzed Art(icles) 13 and 14 of GDPR, i.e., the main GDPR articles targeting PPs. From these two articles, we extracted important concepts to create the information types and the dependencies between them. Art. 13 focuses on personal data collected directly from data subjects (e.g., filling an online form or an interview), whereas Art. 14 focuses on personal data obtained indirectly from data subjects (e.g., obtained from a public website or public list). We observe that Art. 13.2(e) *"whether the provision of personal data is a statutory or contractual requirement, or a requirement necessary to enter into a contract, as well as whether the data subject is obliged to provide the personal data and of the possible consequences of failure to provide such data"* is related to the direct collection of personal data, while Art. 14.2(f) *"from which source the personal data originate, and if applicable, whether it came from publicly accessible sources"* deals with indirect collection. These observations were considered while building the two artifacts discussed above. Starting from Art. 13 and 14, as per the recommendation of legal experts, we

Table 3.1: Glossary excerpt.

Information Types (**Reference**[1])	Description
CONTROLLER (**Art. 13/14(f)**)	A natural or legal person, public authority, agency or any other body which, alone or jointly with others, determines the purposes and means of the processing of personal data where the purposes and means of such processing are determined by national or EU laws or regulations, the controller or the specific criteria for its nomination may be provided by national or EU law.
IDENTITY (**Art. 13/14(f)**)	The legal name of the company/organization.
CONTACT (**Art. 13/14(f)**)	The method(s) with which the company/organization can be contacted.
CONTROLLER REPRESENTATIVE (**Art. 13/14(f)**)	A natural or legal person established in the Union who is designated by the controller.
DATA PROTECTION OFFICER (DPO) (**Art. 13/14(f)**)	The one who is responsible for overseeing data protection strategy and implementation to ensure compliance with GDPR requirements.
PROCESSING (**Art. 13/14(f)**)	Any operation performed on personal data, whether or not by automated means, including collection, recording, organization, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, restriction, erasure or destruction.
PERSONAL DATA (PD) (**Art.5(f)**)	Any information related to an identified or identifiable natural person.
PROVISION (**Art. 14(f)**)	The action of providing something (i.e., personal data) for use (i.e., to be processed).
PD ORIGIN (**Art. 14.2(f)**)	From which source the personal data originates (i.e., direct or indirect), and if applicable, whether it came from a publicly and/or third-party and/or cookie sources.
INDIRECT (**Art. 14**)	When the personal data are not obtained from the data subject.
THIRD PARTY (**Art. 14**)	When the personal data are obtained from organisations external to the data controller.
PUBLICLY (**Art. 14**)	When the personal data are obtained from public sources (i.e., from a public website).
PROFILING (**Art. 4(f)**)	To analyze or predict aspects concerning a natural person's performance at work.

[1] GDPR-related articles

also examined Art. 6, 9, 21, 37, 46, 47, 49, 55, and 56 by doing a snowball sampling from the cross-references in Art 13 and 14.

Fig. 3.4 illustrates an excerpt of Art. 13 from which we have inferred the hierarchical representation of four information types: CONTROLLER, CONTROLLER REPRESENTATIVE, and their descendants IDENTITY and CONTACT. These information types refer to four distinct concepts: (1) the identity of the data controller (CONTROLLER.IDENTITY), (2) the contact details of the data controller (CONTROLLER.CONTACT), (3) the identity of the data controller's representative (CONTROLLER REPRESENTATIVE.IDENTITY), and (4) the contact details of the data controller's representative (CONTROLLER REPRESENTATIVE.CONTACT). The information types IDENTITY and CONTACT were ultimately specialized with the inclusion of other sub-information types. The former, with LEGAL NAME and REGISTER NUMBER information types and the latter with EMAIL, LEGAL ADDRESS and PHONE NUMBER. Our conceptual model (depicted in Fig. 3.3) is organized into three hierarchical levels: **level-1**, shaded yellow, **level-2**, shaded grey, and **level-3**, shaded white. The colors were introduced to

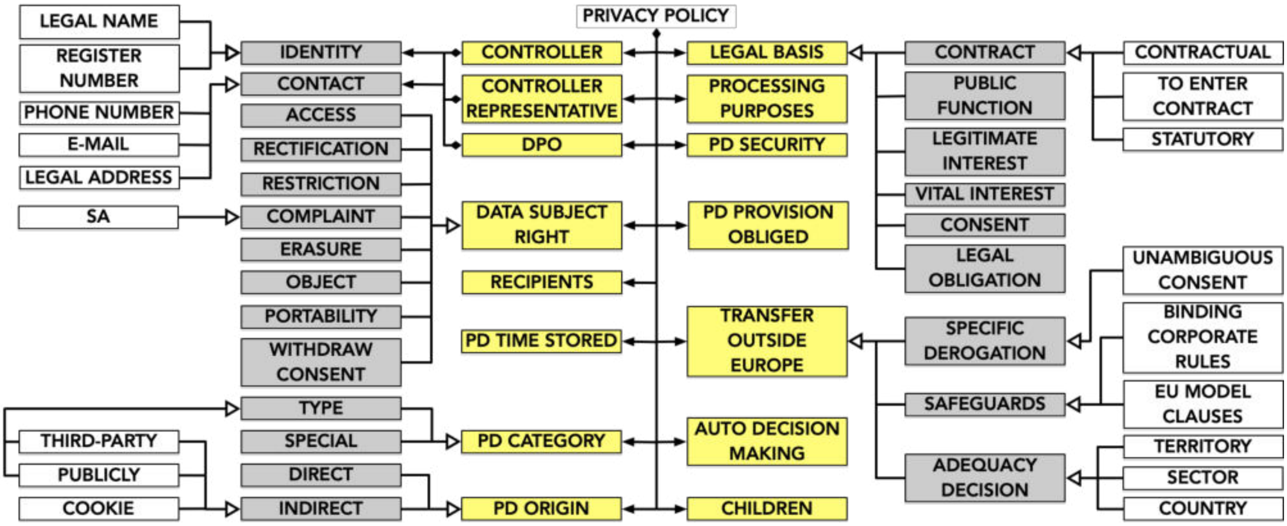
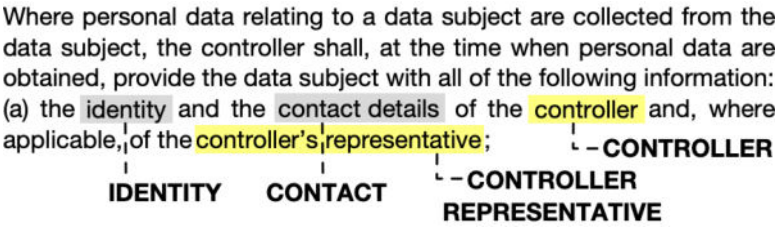


Figure 3.3: Conceptual model of PP information types.

make the model more readable to annotators and legal experts. As presented in Fig. 3.5, the methodology we used for identifying the information types from GDPR and building the conceptual model involved three types of coding: *in-vivo coding, hypothesis coding* and *subcoding* [92].


Where personal data relating to a data subject are collected from the data subject, the controller shall, at the time when personal data are obtained, provide the data subject with all of the following information: (a) the identity and the contact details of the controller and, where applicable, of the controller's representative;

CONTROLLER
IDENTITY
CONTACT
CONTROLLER REPRESENTATIVE

Figure 3.4: Example of coding in the context of GDPR.

1. In-vivo coding: we use this type of coding to identify the core concepts in GDPR and create an initial set of codes. In-vivo coding emphasizes the actual words in the text – in our case the text of GDPR – in order to create codes. The in-vivo approach allowed us to derive the names of the information types directly from the text of GDPR (i.e., the meta documents). Those information types are then used to characterize GDPR-related text found in PPs. An information type, representing a code, is a short phrase that symbolically assigns a summative, salient, essence-capturing, and/or evocative attribute to a particular text in a given PP [92]. For example, the information type CONTROLLER in Fig. 3.4 refers to the text in a given PP that discusses a natural or legal person, public authority, agency or any other body which, alone or jointly with others, determines the purposes and means of personal data processing (see Art. 13.1(a) of GDPR).

2. Hypothesis coding: this type of coding refers to the application of a predetermined set of codes to qualitative data in order to assess researcher-generated hypotheses. The codes are developed from a prediction – in our case, the initial set of codes identified from GDPR with in-vivo coding – about what one would find in the actual data – in our case, PPs – before the data was collected and analyzed. Usually, the application of this coding methodology can range from simple frequency counts to more complex multivariate analyses. In our context, we are interested in the presence or absence of information types in a given PP in order to check its compliance against GDPR. In particular, with the help of legal experts, we manually applied this initial set of codes (obtained with in-vivo coding) via hypothesis coding over 30 PPs (a subset of our training set) in order to ensure that our in-vivo codes are sufficient and at the right level of abstraction.

While applying hypothesis coding to the PPs, for each information type, we collected the keywords that made us decide to associate a given sentence with a specific information type. For example, the combination of keywords *"right to access"* was extracted from sentence number **27** of Fig. 3.1 and included in the list of of keywords associated with DATA SUBJECT RIGHTS.ACCESS. At the end of hypothesis coding, we obtained a list of keywords for each information type as shown in Fig. 3.3.

3. Subcoding: in addition to hypothesis coding, we also use subcoding, which refers to sub-codes as a second-order tag assigned after a primary code, in order to enrich our information types in terms of specificity. For example, in Fig. 3.3, the information type PD ORIGIN (in yellow) is specialized into two sub-information types: DIRECT and INDIRECT (in gray). Then, INDIRECT is further specialized into: THIRD-PARTY, PUBLICLY and COOKIE (in white). The use of subcoding ultimately contributed to the final set of codes represented by the conceptual model of Fig. 3.3.

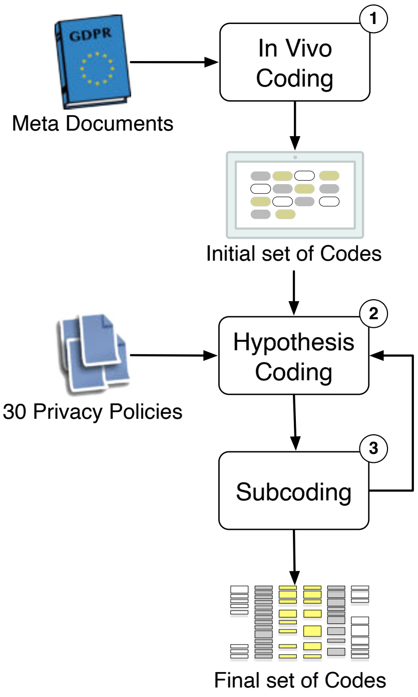


Figure 3.5: Coding methodology.

Based on our interpretation and understanding of GDPR articles, we created an initial version of the information types conceptual model along with their definitions. We kept track of GDPR articles to ensure traceability in our glossary (artifact 2). Table 3.1 presents an excerpt of our glossary.

These (interim) artifacts were then presented to legal experts for feedback. In addition to pointing out issues and omissions, the experts were encouraged to bring to our attention any GDPR article or external documentation/information needed to be considered in the context of PPs. The feedback obtained from legal experts was, by large, concerned with information that was not explicitly included in GDPR (e.g., the European Working Party [93]). For example, Art. 13.1(f) states that *"the controller intends to transfer personal data to a third country or international organization and the existence or absence of an adequacy decision by the Commission, or [...] appropriate or suitable safeguards [...]"*. This article is addressed in Fig. 3.3 by the information type TRANSFER OUTSIDE EUROPE.ADEQUACY DECISION. In response to the legal experts' feedback and by following the external source[1] recommended by them, we created the sub-information types of ADEQUACY DECISION that are not discussed in GDPR. In particular, three sub-information types were added to the conceptual model in Fig. 3.3, namely, TERRITORY, SECTOR, and COUNTRY. These information types refer to the adequacy decisions between the EU and a territory (e.g., Andorra, the Bailiwick of Jersey, etc.), specific sectors (e.g., the commercial organizations from Canada, Argentina, etc.), and a country (i.e., Japan, New Zealand, etc.), respectively. In the same manner, we used another external source[2] to create the level-3 sub-information type EU MODEL CLAUSES.

Once the conceptual model converged to a stable state, we put together a general report including the conceptual model and the glossary table. The conceptual modeling step terminated when the general report was approved by legal experts. The final version of the conceptual model, with a total of 56 information types (see Fig. 3.3), along with a complete glossary table of 60 entries, are provided in an online annex [91].

[1] EU Adequacy Decisions – https://bit.ly/38ciwPU (January 2021)

[2] EU Standard Contractual Clauses – https://bit.ly/3nd6JFt (January 2021)

3.4 Criteria for Compliance Checking of PPs (RQ2)

In this section, we answer RQ2 by presenting the criteria we use to check the compliance of PPs according to GDPR. In particular, we discuss our method for creating a set of 23 criteria for checking the compliance of PPs by analyzing GDPR articles. In order to identify these criteria, we used an iterative three-step method similar to the one we used to create the other two artifacts mentioned in Section 3.3 (see Fig.3.2). We obtained the final set of compliance criteria in six iterations, with each iteration requiring, on average, 15 days. During this process, we combined bi-weekly face-to-face validation sessions and off-line interactions with legal experts. The face-to-face sessions, which lasted between 2 to 3 hours each, collectively added up to approximately 15 hours, plus an additional five hours for off-line interactions.

3.4.1 Transforming GDPR Articles into Criteria

The complete set of criteria discussed in this section uses the information types identified in the conceptual model of Fig. 3.3. We note that some information types are *inter-dependent*, meaning that the presence of an information type requires the presence of another information type. For example, if a PP requires individuals to provide consent for collecting their personal data, then the policy shall also allow individuals to withdraw their consent, i.e., LEGAL BASIS.CONSENT and DATA SUBJECT RIGHT.WITHDRAW CONSENT are inter-dependent. Most of the criteria were extracted from the same GDPR articles from which the information types were also identified (see the external online annex [91] for criteria traceability to the GDPR articles). Based on our interpretation and understanding of these GDPR articles, we identified an initial set of criteria that we formulated as pseudo-code. Each pseudo-code statement is composed of two main parts: 1) a *precondition* (if any) about the identification of one or more information types in a PP or other GDPR-related conditions proposed by the legal experts, and 2) a *postcondition* asserting the identification of one or more information types (different from the one(s) in the *precondition*) in a PP. We use the following template [*precondition*], <*postcondition*> and show below examples of criteria written in pseudo-code; these are derived from the excerpt of Art. 13.1(a) shown in Fig. 3.4:

C1 [ ], <CONTROLLER.IDENTITY.{REGISTER NUMBER *or* LEGAL NAME} must be identified>.

C2 [ ], <CONTACT.{EMAIL or PHONE or LEGAL ADDRESS} must be identified>.

C3 [*if* CONTROLLER is located outside of Europe], <*then* CONTROLLER REPRESENTATIVE.IDENTITY.{REGISTER NUMBER or LEGAL NAME} must be identified>.

C4 [*if* CONTROLLER is located outside of Europe], <*then* CONTROLLER REPRESENTATIVE.CONTACT.{EMAIL or PHONE or LEGAL ADDRESS} must be identified>.

In this step, we transform the text of the relevant GDPR provisions into compliance criteria. For example, considering Fig. 3.4, the word *shall* is translated into a mandatory requirement for including the CONTROLLER.IDENTITY (C1) and CONTROLLER.CONTACT details (C2). On the other hand, the combination of the words *where* and *applicable* suggests that a given criterion should be enforced only if certain precondition(s) are met: the CONTROLLER REPRESENTATIVE.IDENTITY (C3) and CONTROLLER REPRESENTATIVE.CONTACT (C4) need to be checked only if the CONTROLLER is located outside of Europe.

While defining the criteria from the GDPR articles, we realized that some of them should not always be checked. Articles 13.1(a,e,f), 13.2(e), 14.1(a,d,e,f), and 14.2(f) are GDPR articles that apply to PPs only in specific situations. After reviewing these articles with legal experts, they asked us to create a questionnaire that would help them specify, under various situations, the exact content of a given PP for compliance checking. The person who should ideally provide answers to the questionnaire should have expertise in the legal domain as well as extensive knowledge about the company for which the PP analysis is being performed. For example, to determine whether C3 and C4 above should be checked, it is important to know beforehand from the questionnaire that the CONTROLLER is located outside Europe.

The questionnaire contains a set of critical questions whose answers depend on context and are often left tacit in PPs. Nevertheless, these answers carry important implications on what needs to be explicitly covered when checking the compliance. The questionnaire includes the following six questions:

Q1 Who is the CONTROLLER in charge of data processing? *Write name*.

Q2 Do you plan to transfer the collected personal data outside Europe? *Yes/No*.

Q3 Will there be other recipients of the collected personal data besides you? *Yes/No*.

Q4 What is the core of your activities?

□ The processing of personal data is carried out by a public authority or body (except for courts acting in their judicial capacity).

□ The processing of operations which, by nature, scope and/or purposes, require regular and systematic monitoring of data subjects on a large scale.

□ The processing, on a large scale, of personal data relating to sensitive categories (e.g., racial or ethnic origin, political opinions, or religious or philosophical beliefs) or to criminal convictions and offenses.

Q5 Where will the activities carried out by your organization take place?

○ *Inside Europe*

○ *Outside Europe* – if selected, then write the name of CONTROLLER REPRESENTATIVE : ....

Q6 How will the personal data of the data subject be collected?

○ DIRECT ○ INDIRECT ○ *Both*

Question **Q1** is not intended to trigger the checking of any criterion. This first question is used to facilitate the identification of the information type CONTROLLER.IDENTITY.

The main objective of the remaining questions is to determine whether some context-related criteria should be checked. In particular, each of the other five questions (**Q2-Q6**) triggers the search for one or more information types. This leads to checking some specific criteria. A positive answer to question Q2 triggers the verification of criteria C10 – C14. If the answer to Q3 is yes, then criterion C19 is verified. Similarly, Q4 will trigger the verification of criterion C23, if any of the optional answers to this question is checked. The answer to question Q5 activates the verification of criteria C3 and C4, if the activities carried out by the CONTROLLER take place outside Europe. Finally, answering Q6 as "DIRECT" activates checking criterion C22; "INDIRECT" activates checking criteria C15 – C18; and "*Both*" requires checking all the above criteria (i.e., C15 – 18 and C22). The remaining criteria (C1, C2, C5 – C9, C20 and C21) are always verified because they refer to information types that, according to GDPR, must be present in every PP.

At the end of the first step, we created a table with all the information about the criteria set, including an identifier (ID) for each criterion (first column), preconditions (middle column) and postconditions (last column), as shown in Table 3.2. Since C1, C2, C5, C20 and C21 are not triggered by any preconditions, they always need to be checked. The rest of the criteria are triggered by some precondition related to answers to questions Q2 – Q6 (referred to as A2 – A6) from the questionnaire or the presence of some information in the PP.

Table 3.2 presents criteria that *should* be satisfied according to GDPR (ID highlighted in orange) and may lead to a warning, and other criteria that *must* always be satisfied (ID highlighted in red) and may lead to a violation. We further discuss the difference between warnings and violations in the next subsection.

3.4.2 Evaluating the Criteria with Legal Experts

To facilitate the validation of the criteria presented in Table 3.2 with legal experts, we decided to capture them as activity diagrams, following the observation by Soltana et al. [94] that legal experts can understand activity diagrams with relative ease given some basic training. With the help of legal experts, we created a final set of 23 criteria to capture the mechanisms necessary to check the compliance of PPs according to GDPR. Among the 16 criteria shown in Fig. 3.6, C3, C4, C15, C17, C19, and C23 depend on the answers to the questionnaire. Fig. 3.6 and Fig. 3.7 show the 23 criteria to check the compliance of a PP with respect to the information types of the conceptual model presented earlier. Fig. 3.6 contains every possible violation in PPs and Fig. 3.7 all the possible warnings.

The compliance criteria in Fig. 3.6 and Fig. 3.7 use in general three shapes to represent different types of actions or steps in a process: (1) a circle represents the start and endpoint, (2) a diamond indicates a decision, and (3) a rectangle stands for an action representing that (3.1) an information type was correctly identified or not needed in a PP (in green), (3.2) a mandatory information type was entirely missing in a PP (referred to as *violation* – highlighted in red), and (3.3) an information type was only partially identified or, in other words, an information type was identified but some related information is missing (referred to as *warning* – highlighted in orange). An *non-compliance issue* is raised when a criterion returns a violation or warning. A violation corresponds to a direct breach of GDPR, whereas a warning leads to further assessment by the legal expert to finally decide whether there is a breach of GDPR.

Below, we illustrate two criteria, C15 and C16, derived from Art. 14.2(f) of the GDPR (see Fig. 3.6 and Fig. 3.7). These criteria check the compliance of a PP with respect to the information type PD ORIGIN.INDIRECT.

C15 is meant for identifying a violation:

(1) If the answer to **Q6** is INDIRECT or *Both* (recall the questionnaire presented in Section 3.4.1), then go to **(2)**; otherwise PD ORIGIN.INDIRECT is *not needed.*

(2) If the indirect origin of the personal data is mentioned, then PD ORIGIN.INDIRECT is *identified*; otherwise PD ORIGIN.INDIRECT is *missing* – **Violation**.

Table 3.2: Compliance criteria according to GDPR.

ID	Criteria	
	Precondition[1]	Postcondition[2]
C1	-	Controller.Identity
C2	-	Controller.Contact.{Legal Address, Email, *or* Phone Number}
C3	**A5** is a country *outside the EU*	Controller Representative.Identity
C4	**A5** is a country *outside the EU*	Controller Representative .Contact.{Legal Address, Email, *or* Phone Number}
C5	-	Data Subject Right.{Access, Complaint, Rectification, *and* Restriction}
C6	Data Subject Right.Complaint	Data Subject Right.Complaint.SA
C7	Legal Basis.Contract	Data Subject Right.Portability
C8	Legal Basis .{Legitimate Interest *or* Public Function}	Data Subject Right.Object
C9	Legal Basis.Consent	Data Subject Right.{Erasure, Object, Portability, *and* Withdraw Consent}
C10	**A2** is *Yes*	Transfer Outside Europe
C11	Transfer Outside Europe	Transfer Outside Europe.{Adequacy Decision, Safeguards, *or* Specific Derogation}
C12	Transfer Outside Europe .Adequacy Decision	Transfer Outside Europe.Adequacy Decision .{Country, Sector, *or* Territory}
C13	Transfer Outside Europe .Safeguards	Transfer Outside Europe.Safeguards.{EU Model Clauses, *or* Binding Corporate Rules}
C14	Transfer Outside Europe .Specific Derogation	Transfer Outside Europe.Specific Derogation.Unambiguous Consent
C15	**A6** is Indirect or *Both*	PD Origin.Indirect
C16	PD Origin.Indirect	PD Origin.Indirect.{Third Party, *or* Publicly}
C17	**A6** is Indirect or *Both*	PD Category
C18	PD Origin.Indirect .{Third Party, *or* Publicly}	PD Category.Type
C19	**A3** is *Yes*	Recipients
C20	-	PD Time Stored
C21	-	Processing Purposes
C22	**A6** is Direct or *Both* **and** Legal Basis .{Contract.To Enter Contract, *or* Legal Obligation}	PD Provision Obliged
C23	At least one answer in **Q4** is selected	DPO.Contact.{Legal Address, Email *or* Phone Number}

[1] Includes the answers to Q2 – Q6 (**A2 – A6**), or the information types that are present in a PP.

[2] Information types that must / should be present.

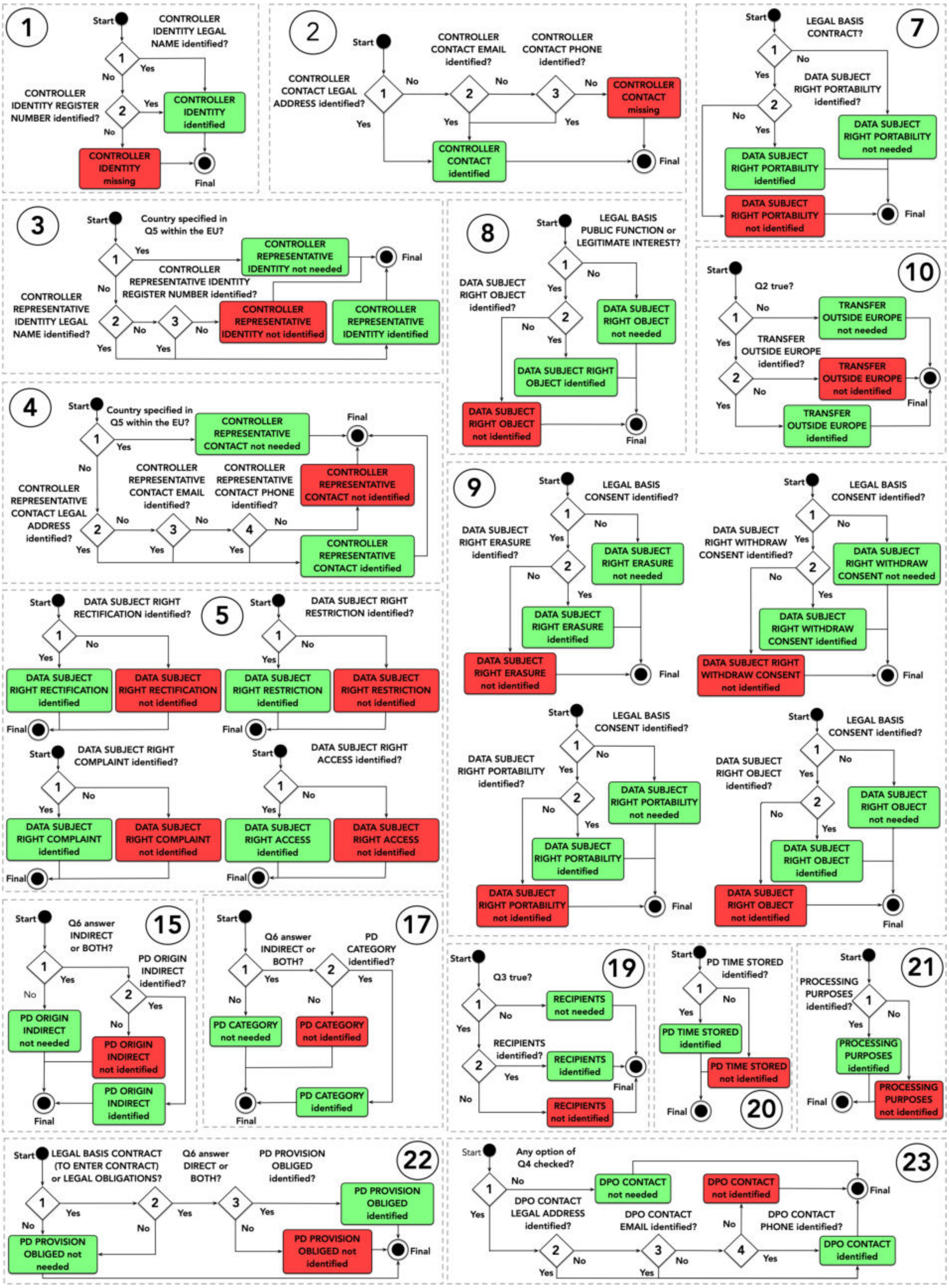

1
Start
CONTROLLER IDENTITY LEGAL NAME identified?
1
Yes
No
CONTROLLER IDENTITY REGISTER NUMBER identified?
2
Yes
No
CONTROLLER IDENTITY identified
CONTROLLER IDENTITY missing
Final
2
Start
CONTROLLER CONTACT LEGAL ADDRESS identified?
CONTROLLER CONTACT EMAIL identified?
CONTROLLER CONTACT PHONE identified?
1
2
3
No
No
No
Yes
Yes
Yes
CONTROLLER CONTACT missing
CONTROLLER CONTACT identified
Final
7
Start
LEGAL BASIS CONTRACT?
1
No
Yes
DATA SUBJECT RIGHT PORTABILITY identified?
2
No
Yes
DATA SUBJECT RIGHT PORTABILITY not needed
DATA SUBJECT RIGHT PORTABILITY identified
DATA SUBJECT RIGHT PORTABILITY not identified
Final
3
Start
Country specified in Q5 within the EU?
1
Yes
No
CONTROLLER REPRESENTATIVE IDENTITY not needed
Final
CONTROLLER REPRESENTATIVE IDENTITY LEGAL NAME identified?
CONTROLLER REPRESENTATIVE IDENTITY REGISTER NUMBER identified?
2
3
No
No
Yes
Yes
CONTROLLER REPRESENTATIVE IDENTITY not identified
CONTROLLER REPRESENTATIVE IDENTITY identified
8
Start
LEGAL BASIS PUBLIC FUNCTION or LEGITIMATE INTEREST?
1
No
Yes
DATA SUBJECT RIGHT OBJECT identified?
2
No
Yes
DATA SUBJECT RIGHT OBJECT not needed
DATA SUBJECT RIGHT OBJECT identified
DATA SUBJECT RIGHT OBJECT not identified
Final
10
Start
Q2 true?
1
No
Yes
TRANSFER OUTSIDE EUROPE not needed
TRANSFER OUTSIDE EUROPE identified?
2
No
Yes
TRANSFER OUTSIDE EUROPE not identified
TRANSFER OUTSIDE EUROPE identified
Final
4
Start
Country specified in Q5 within the EU?
1
Yes
No
CONTROLLER REPRESENTATIVE CONTACT not needed
Final
CONTROLLER REPRESENTATIVE CONTACT LEGAL ADDRESS identified?
CONTROLLER REPRESENTATIVE CONTACT EMAIL identified?
CONTROLLER REPRESENTATIVE CONTACT PHONE identified?
2
3
4
No
No
No
Yes
Yes
Yes
CONTROLLER REPRESENTATIVE CONTACT not identified
CONTROLLER REPRESENTATIVE CONTACT identified
9
Start
LEGAL BASIS CONSENT identified?
1
No
Yes
DATA SUBJECT RIGHT ERASURE identified?
2
No
Yes
DATA SUBJECT RIGHT ERASURE not needed
DATA SUBJECT RIGHT ERASURE identified
DATA SUBJECT RIGHT ERASURE not identified
Final
Start
LEGAL BASIS CONSENT identified?
1
No
Yes
DATA SUBJECT RIGHT WITHDRAW CONSENT identified?
2
No
Yes
DATA SUBJECT RIGHT WITHDRAW CONSENT not needed
DATA SUBJECT RIGHT WITHDRAW CONSENT identified
DATA SUBJECT RIGHT WITHDRAW CONSENT not identified
Final
5
Start
DATA SUBJECT RIGHT RECTIFICATION identified?
1
No
Yes
DATA SUBJECT RIGHT RECTIFICATION identified
DATA SUBJECT RIGHT RECTIFICATION not identified
Final
Start
DATA SUBJECT RIGHT RESTRICTION identified?
1
No
Yes
DATA SUBJECT RIGHT RESTRICTION identified
DATA SUBJECT RIGHT RESTRICTION not identified
Final
Start
DATA SUBJECT RIGHT COMPLAINT identified?
1
No
Yes
DATA SUBJECT RIGHT COMPLAINT identified
DATA SUBJECT RIGHT COMPLAINT not identified
Final
Start
DATA SUBJECT RIGHT ACCESS identified?
1
No
Yes
DATA SUBJECT RIGHT ACCESS identified
DATA SUBJECT RIGHT ACCESS not identified
Final
Start
LEGAL BASIS CONSENT identified?
1
No
Yes
DATA SUBJECT RIGHT PORTABILITY identified?
2
No
Yes
DATA SUBJECT RIGHT PORTABILITY not needed
DATA SUBJECT RIGHT PORTABILITY identified
DATA SUBJECT RIGHT PORTABILITY not identified
Final
Start
LEGAL BASIS CONSENT identified?
1
No
Yes
DATA SUBJECT RIGHT OBJECT identified?
2
No
Yes
DATA SUBJECT RIGHT OBJECT not needed
DATA SUBJECT RIGHT OBJECT identified
DATA SUBJECT RIGHT OBJECT not identified
Final
15
Start
Q6 answer INDIRECT or BOTH?
1
Yes
No
PD ORIGIN INDIRECT identified?
2
Yes
No
PD ORIGIN INDIRECT not needed
PD ORIGIN INDIRECT not identified
PD ORIGIN INDIRECT identified
Final
17
Start
Q6 answer INDIRECT or BOTH?
PD CATEGORY identified?
1
2
Yes
Yes
No
No
PD CATEGORY not needed
PD CATEGORY not identified
PD CATEGORY identified
Final
19
Start
Q3 true?
1
No
Yes
RECIPIENTS not needed
RECIPIENTS identified?
2
Yes
No
RECIPIENTS identified
RECIPIENTS not identified
Final
Start
PD TIME STORED identified?
1
No
Yes
PD TIME STORED identified
PD TIME STORED not identified
Final
20
21
Start
PROCESSING PURPOSES identified?
1
No
Yes
PROCESSING PURPOSES identified
PROCESSING PURPOSES not identified
Final
22
Start
LEGAL BASIS CONTRACT (TO ENTER CONTRACT) or LEGAL OBLIGATIONS?
Q6 answer DIRECT or BOTH?
PD PROVISION OBLIGED identified?
1
2
3
Yes
Yes
Yes
No
No
No
PD PROVISION OBLIGED identified
PD PROVISION OBLIGED not needed
PD PROVISION OBLIGED not identified
Final
23
Start
Any option of Q4 checked?
1
No
Yes
DPO CONTACT not needed
DPO CONTACT not identified
Final
DPO CONTACT LEGAL ADDRESS identified?
DPO CONTACT EMAIL identified?
DPO CONTACT PHONE identified?
2
3
4
No
No
No
Yes
Yes
Yes
DPO CONTACT identified

Figure 3.6: GDPR compliance criteria represented as activity diagrams (violations).

Criterion C16 is meant for identifying a warning:

(1) If PD ORIGIN.INDIRECT is identified in C15, then go to **(2)**; otherwise PD ORIGIN.INDIRECT is *not* *needed.*

(2) If the indirect origin of personal data is from a third-party, then PD ORIGIN.INDIRECT.THIRD-PARTY is *identified*; otherwise go to **(3)**.

(3) If the indirect origin of personal data is from public sources, then PD ORIGIN.INDIRECT.PUBLICLY is *identified*; otherwise PD ORIGIN.INDIRECT is *partially identified* – **Warning**.

Note that C16 in Fig. 3.7 does not refer to COOKIE although COOKIE is a subtype of PD ORIGIN.INDIRECT in the conceptual model of Fig. 3.3. The above-shown criterion strictly follows GDPR, which does not regulate cookies. However, our collaborating legal experts suggested the inclusion of COOKIE in our conceptual model since cookies are often mentioned in PPs and they may become relevant to GDPR in the future.

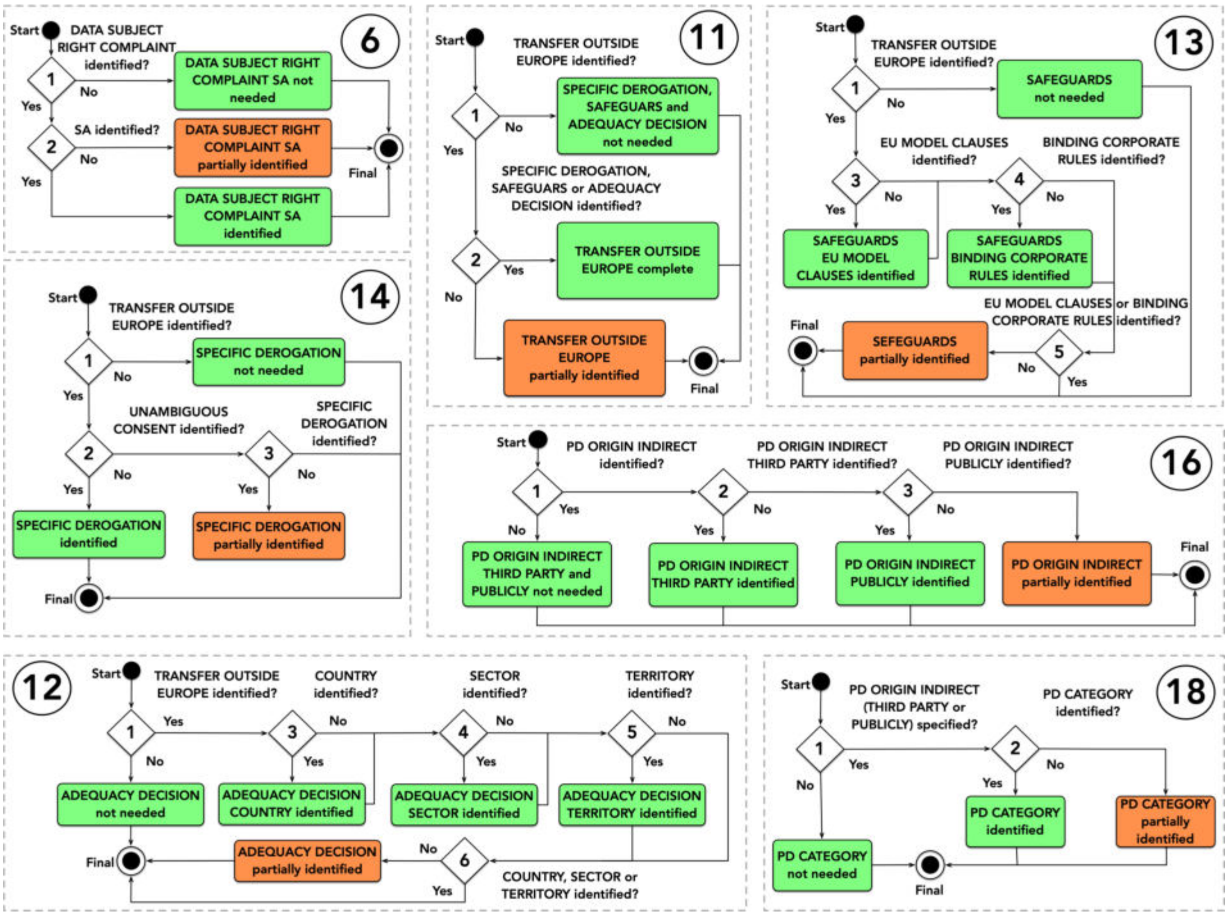

6
Start
DATA SUBJECT RIGHT COMPLAINT identified?
1
No
Yes
DATA SUBJECT RIGHT COMPLAINT SA not needed
SA identified?
2
No
Yes
DATA SUBJECT RIGHT COMPLAINT SA partially identified
Final
DATA SUBJECT RIGHT COMPLAINT SA identified
11
Start
TRANSFER OUTSIDE EUROPE identified?
1
No
Yes
SPECIFIC DEROGATION, SAFEGUARS and ADEQUACY DECISION not needed
SPECIFIC DEROGATION, SAFEGUARS or ADEQUACY DECISION identified?
2
Yes
No
TRANSFER OUTSIDE EUROPE complete
TRANSFER OUTSIDE EUROPE partially identified
Final
13
Start
TRANSFER OUTSIDE EUROPE identified?
1
No
Yes
SAFEGUARDS not needed
EU MODEL CLAUSES identified?
BINDING CORPORATE RULES identified?
3
No
Yes
4
No
Yes
SAFEGUARDS EU MODEL CLAUSES identified
SAFEGUARDS BINDING CORPORATE RULES identified
EU MODEL CLAUSES or BINDING CORPORATE RULES identified?
Final
SEFEGUARDS partially identified
5
No
Yes
14
Start
TRANSFER OUTSIDE EUROPE identified?
1
No
Yes
SPECIFIC DEROGATION not needed
UNAMBIGUOUS CONSENT identified?
SPECIFIC DEROGATION identified?
2
No
Yes
3
No
Yes
SPECIFIC DEROGATION identified
SPECIFIC DEROGATION partially identified
Final
16
Start
PD ORIGIN INDIRECT identified?
PD ORIGIN INDIRECT THIRD PARTY identified?
PD ORIGIN INDIRECT PUBLICLY identified?
1
Yes
No
2
No
Yes
3
No
Yes
PD ORIGIN INDIRECT THIRD PARTY and PUBLICLY not needed
PD ORIGIN INDIRECT THIRD PARTY identified
PD ORIGIN INDIRECT PUBLICLY identified
PD ORIGIN INDIRECT partially identified
Final
12
Start
TRANSFER OUTSIDE EUROPE identified?
COUNTRY identified?
SECTOR identified?
TERRITORY identified?
1
Yes
No
3
No
Yes
4
No
Yes
5
No
Yes
ADEQUACY DECISION not needed
ADEQUACY DECISION COUNTRY identified
ADEQUACY DECISION SECTOR identified
ADEQUACY DECISION TERRITORY identified
Final
ADEQUACY DECISION partially identified
No
6
Yes
COUNTRY, SECTOR or TERRITORY identified?
18
Start
PD ORIGIN INDIRECT (THIRD PARTY or PUBLICLY) specified?
PD CATEGORY identified?
1
Yes
No
2
No
Yes
PD CATEGORY identified
PD CATEGORY partially identified
PD CATEGORY not needed
Final

Figure 3.7: GDPR compliance criteria represented as activity diagrams (warnings).

3.5 Approach (RQ3)

In this section, we address RQ3 and present our AI-based approach for **Compl**iance checking of PPs using **A**rtificial **I**ntelligence against GDPR (thereafter referred to as $CompA\iota$). $CompA\iota$ does not use deep learning (DL) architectures (e.g., LSTM [95]), since we do not have enough data for developing such models with enough accuracy. $CompA\iota$, shown in Fig. 3.8, is composed of two main phases.

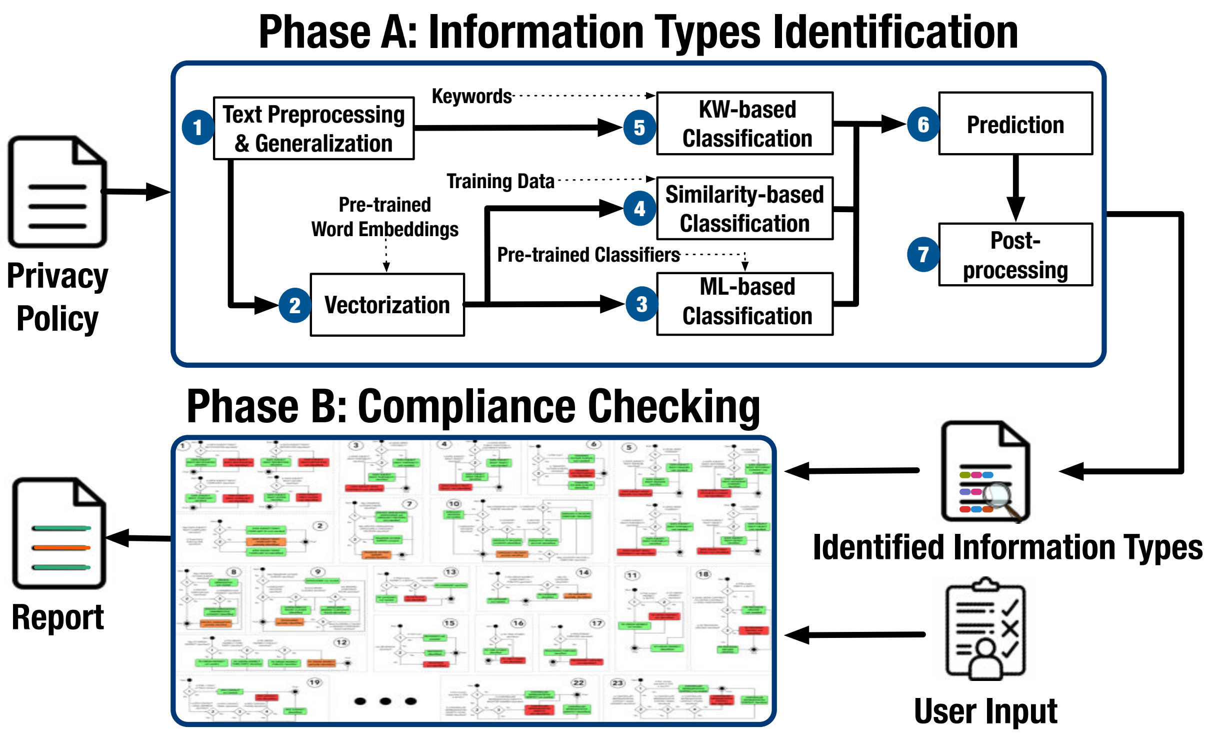

Phase A: Information Types Identification
Privacy Policy
1 Text Preprocessing & Generalization
Keywords
5 KW-based Classification
6 Prediction
Training Data
4 Similarity-based Classification
Pre-trained Word Embeddings
Pre-trained Classifiers
2 Vectorization
3 ML-based Classification
7 Post-processing
Phase B: Compliance Checking
Report
Identified Information Types
User Input

Figure 3.8: Overview of the compliance checking approach ($CompAI$).

Phase A, *information types identification*, takes as an input a PP, and returns the information types that are present in this policy as an intermediary output. More precisely, Phase A results in a binary decision for each information type regarding whether or not it is present in the input PP. Phase B, *compliance checking*, takes as an input the identified information types from Phase A and the user input based on the questionnaire (explained in Section 3.4). Phase B then returns a detailed report about whether the input PP is compliant according to GDPR. We elaborate these phases next.

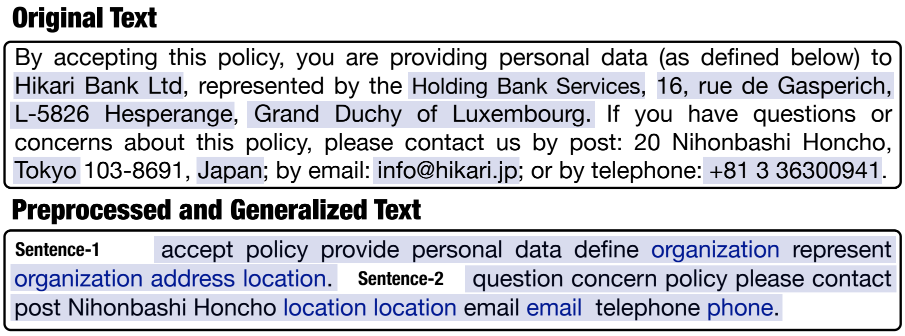
3.5.1 Information Types Identification (Phase A)

Phase A uses a combination of NLP and ML to identify the information types that are present in a given PP. Our information types identification approach aims to solve a hierarchical, multi-label and multi-class classification problem. The nature of the problem is visible from the conceptual model in Fig. 3.3, where most level-1 information types are further specialized into sub-information types (level-2 and level-3). Multi-label classification reflects the fact that a sentence in the PP can discuss one or more information types. Therefore, our solution can predict one or more potential labels (information types) for each sentence in the input PP. Our approach considers a *sentence* as the unit of analysis. A sentence refers to the textual entity that results from applying the sentence splitting module in the NLP pipeline (Fig. 2.1 in Section 2.2), irrespective of whether the sentence identified by this module corresponds to a grammatical sentence. The rationale behind using sentences rather than phrases is that the former are more likely to contain the context necessary for understanding their meaning [96] and thus lead to more accurate classification results.

Phase A is further composed of seven steps. In the first two steps, the text of the input PP is preprocessed, generalized and transformed into a mathematical representation (vectors). In steps 3-5, we classify the sentences of the input PP into one or more information types using three classification methods based on ML, semantic similarity, and keywords. As we will see, relying on these complementary methods is necessary to overcome the complexity of the hierarchical classification problem. In step 6, we combine the results of steps 3, 4, and 5 to predict information types for each sentence in the input PP. In the last step, we refine the results through post-processing. We explain these steps in detail next.

Step 1: Text Preprocessing and Generalization

In step 1, we apply the categories A, B, and C of the NLP pipeline (Fig. 2.1 in Section 2.2) to parse the input PP and obtain the sentences. Using the annotations produced by the NLP pipeline, we generalize the text in each sentence by replacing specific textual entities with more generic ones. Specifically, we replace named entities (as identified by the named entity recognition module) with their types. For example, the entities "Japan" and "Hikari Bank Ltd" in Fig. 3.9 will be replaced with the types *location* and *organization*, respectively. Similarly, we generalize emails, postal addresses, telephone numbers, and websites, e.g., "info@hikari.jp" is replaced with *email*. The intuition behind generalization is to normalize the text such that, despite significant diversity across the PPs used for training (e.g., the mention of different locations), the approach can still learn common patterns and accurately predict information types. The generalized sentences are further normalized through lemmatization and stopword removal, e.g., in Fig. 3.9 "accepting" becomes "accept" and stopwords like "by" are removed.



Original Text

By accepting this policy, you are providing personal data (as defined below) to Hikari Bank Ltd, represented by the Holding Bank Services, 16, rue de Gasperich, L-5826 Hesperange, Grand Duchy of Luxembourg. If you have questions or concerns about this policy, please contact us by post: 20 Nihonbashi Honcho, Tokyo 103-8691, Japan; by email: info@hikari.jp; or by telephone: +81 3 36300941.

Preprocessed and Generalized Text

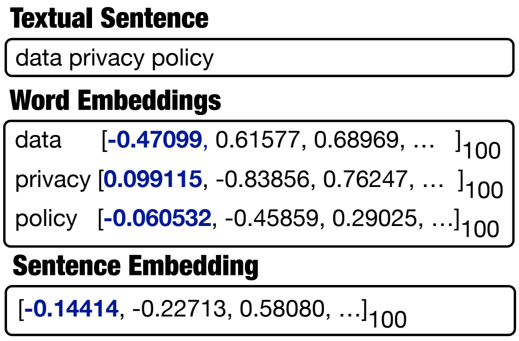
Sentence-1 accept policy provide personal data define organization represent organization address location. **Sentence-2** question concern policy please contact post Nihonbashi Honcho location location email email telephone phone.

Figure 3.9: Example of text preprocessing and generalization.

Step 2: Vectorization

Step 2 transforms the textual sentences resulting from step 1 into embeddings. To do this, we utilize the pre-trained word-vector model of 100-dimensional vectors from GloVe [60] (introduced in Section 2.3.1). Using off-the-shelf, pre-trained and context-independent (i.e., one vector per word regardless of context) word vectors increases the applicability of our approach by making it directly applicable for analyzing new document types.

For computing the sentence embedding, we first retrieve the corresponding embedding for each word in the sentence as given by the pre-trained model. Then, we average over all the word embeddings to get a single vector representing the sentence embedding. For example, the embedding of the sentence "data PP" in Fig. 3.10 is the average of the word embeddings in that sentence, such that the first entry in the sentence embedding (i.e., -0.14414) corresponds to the average of the first entries in the word embeddings of "data", "privacy", and "policy"



Textual Sentence

data privacy policy

Word Embeddings

data $[\mathbf{-0.47099}, 0.61577, 0.68969, \dots]_{100}$

privacy $[\mathbf{0.099115}, -0.83856, 0.76247, \dots]_{100}$

policy $[\mathbf{-0.060532}, -0.45859, 0.29025, \dots]_{100}$

Sentence Embedding

$[\mathbf{-0.14414}, -0.22713, 0.58080, \dots]_{100}$

Figure 3.10: Example of vectorization.

(i.e., -0.47099, 0.099115, and -0.060532), respectively. The objective of the vectorization step is to achieve a representation for measuring text similarity that is effective and fast to train and test. Driven by this objective, we use simple averaging of embeddings because doing so has proven to be efficient for generating sentence embeddings across a broad range of different domains and NLP tasks, including text similarity [71, 73].

Step 3: ML-based Classification

In this step, we attempt to solve the multi-class, multi-label classification problem by transforming it into multiple binary classification problems (as explained in the background in Section 2.3). To do so, we apply the pre-trained ML classifiers for predicting the presence of level-1 and level-2 information types in each sentence of the input PP. We restrict the use of ML to level-1 and level-2 information types because the number of positive examples we have in our training set for level-3 information types is not sufficient for building accurate ML classifiers at that level.

Our classifiers are trained on a feature matrix in which each row corresponds to a sentence and the columns are the 100-dimensional sentence embedding computed in step 2. The prediction class for each classifier indicates the presence of a level-1 or level-2 information type in the sentence. For example, the sentence: *Your personal data might be disclosed to the tax authorities, or other third parties including legal or financial advisors, regulatory bodies, auditors and technology providers.* (number **14** in Fig. 3.1) is predicted as RECIPIENTS. We train the classifiers with positive examples representing the sentences that have been annotated with a particular information type (e.g., DATA SUBJECT RIGHT) and negative examples annotated with any other information type *at the same level* (i.e., all but DATA SUBJECT RIGHT). In most of the cases, we obtained imbalanced datasets with positive examples being under-represented. Inspired by Wang and Manning [97], we use a support-vector machine (SVM) classifier with its default hyper-parameters for sentence classification. SVM is widely used for text classification [98]. We address the imbalance problem in our work using under-sampling over negative examples [34].

Our preliminary experiments suggested that using both SVM for text classification and under-sampling for handling imbalanced datasets outperformed alternatives, e.g., using Naïve Bayes classifier or minority over-sampling. Further, as we will discuss in Section 3.6, the high accuracy obtained by our current solution alleviates the need to empirically examine alternatives.

Step 3 uses one pre-trained binary classifier for each level-1 and level-2 information type in the model of Fig. 3.3. The classifier predicts for each sentence in the input PP, using its embedding vector as features, whether it should be labelled with the information type on which the classifier has been trained. For example, the sentence: *the right to request erasure, restriction, portability, and to object to the processing of your personal data* (numbers **29** – **32** in Fig. 3.1) is predicted by the corresponding binary classifiers as the level-2 information types DATA SUBJECT RIGHT.{ERASURE, RESTRICTION, PORTABILITY, AND OBJECT}. If an information type is not predicted by any binary classifier to be present in a sentence, then this information type is deemed as absent. The resulting classifications are passed on to step 6 (Prediction).

Step 4: Similarity-based Classification

In this step, we classify each sentence of the input PP based on how similar it is to the group of sentences, in the training set, that are annotated with a certain level-1 or level-2 information type. Restricting the use of this classification to level-1 and level-2 information types is due to the same reason explained in step 3. Step 4 creates one group for each level-1 and level-2 information type. Similar to step 3, this step characterizes a sentence using the vector representation built in step 2. Since an individual sentence can have multiple information-type annotations, the same sentence embedding can be part of several groups. Each group is represented by a single vector which is computed as the average of all sentence embeddings in that group. To predict whether a sentence (S) should be annotated with a certain information type (t), we compute the *cosine similarity* between the sentence embedding ($\vec{S}$) and the vector capturing the average embedding of the group of sentences annotated by t (i.e., $\vec{t}$) in the training set.

If the similarity is above a pre-specified threshold (0.9), we predict t to be an information type for S. The threshold value was empirically obtained by evaluating the accuracy of the prediction using a range of similarity threshold values between 0.5 and 0.9, with a step of 0.01, on a subset of the PPs in the training set. Threshold values less than 0.5 are not considered because they fail to capture similarity. To illustrate, consider our example in Section 3.1 (Fig. 3.1). The cosine similarity between the group of sentences annotated with PD ORIGIN.INDIRECT (t) and the vector representation ($\vec{S}$) of the sentence: *information obtained from third parties including [...]* (number **10**) is 0.91, while the cosine similarity with $\vec{S'}$ of the sentence: *Your personal data might be disclosed to the tax authorities, or other third parties [...]* (number **14**) is 0.43. As a result, S is classified as t while S' is not. The results of this step are passed on to step 6 (Prediction).

Step 5: Keyword-based Classification

In this step, we conduct a keyword search (from a predefined list) over the (textual) sentences in the input PP. If a sentence S contains one or more of the keywords associated with information type t, then we predict that S should be annotated with t. For example, the sentence (number **16** in Fig. 3.1): *We may also transfer your personal data to countries outside the European Union (including Japan) on the basis of: European Commission's adequacy decisions, certified by the* ***APPI Japan Scheme****.* will be predicted as TRANSFER OUTSIDE EUROPE.ADEQUACY DECISION.COUNTRY, since it contains keywords indicating this information type (highlighted in bold). Another important point in relation to keywords is that, the text generalization performed in step 1 improves the efficacy of keyword search. For example, number **5** in Fig. 3.1 represents an email address that is generalized with *email*. Thus, including *email* as a keyword enables identifying the information type CONTACT.EMAIL. We have collected a list of keywords covering all of the information types in Fig. 3.3. We elaborate in Section 3.6 on how we obtain these keywords. The results of this step are passed on to the next step (Prediction).

Step 6: Prediction

This step combines the classification results produced based on ML (step 3), semantic similarity (step 4) and keyword search (step 5) to produce a final recommendation about which information types should be ascribed to a given sentence. The reason why we use three different classifiers is to overcome the complexity of the hierarchical multi-class classification problem and hence improve the accuracy of predicting the potential labels for each sentence in the PP. Each method alone has some limitations. On the one hand, relying only on keyword search is not sufficient because of the limitations discussed in Section 3.1. ML-based and similarity-based classifications, on the other hand, are restricted to level-1 and level-2 information types and are further more accurate for the former since the number of datapoints gets much smaller at level-2. Thus, ensembling the three classifiers yields accurate predictions as we will show in our empirical evaluation (Section 3.6).

Our strategy for combining the above classification methods is elaborated in Algorithm 1. The algorithm applies ML-based and similarity-based classifiers for predicting both level-1 and level-2 information types.

Despite having keywords for all information types, the use of keyword search in our approach is limited. We use keywords to predict level-3 that is specializing an already-predicted (level-2) information type or to provide supporting evidence for predicting a level-2 information type in case its level-1 cannot be predicted.

The algorithm starts with an initially empty set of labels ($\mathcal{M}$) – line 1. A label can be represented as *level-1.level-2.level-3* for specialized information types, e.g., DATA SUBJECT RIGHT.COMPLAINT.SA. A partial label can also be predicted, e.g., DATA SUBJECT RIGHT.COMPLAINT or CHILDREN, in case the information type has no specialization or there is no evidence that supports predicting a specialization.

Level-1 and Level-2 Information Types. The algorithm predicts a level-1 information type and its corresponding level-2 specializations in two cases, Case 1 (lines 4 – 10) and Case 2 (lines 11 – 17). Case 1 applies when some level-1 information type can be predicted; the algorithm then attempts to predict its level-2 type. Case 2 applies when Case 1 fails to predict a level-1 type but there is strong support for predicting its level-2 type. The rationale behind Case 2 is that when two classifiers jointly predict a level-2 information type (as we elaborate next), then their predictions should compensate for the absence of a level-1 prediction in Case 1. If Case 2 leads to a prediction of a certain level-2 information type (e.g., VITAL INTEREST), then this will be considered as an indirect indication for predicting the level-1 of that information type (e.g., LEGAL BASIS).

Case 1: If a level-1 information type (ℓ_i) is predicted for the sentence (S) via the (level-1) ML-based classifier (cf_1) or by the similarity-based classifier (line 4), then ℓ_i is added to $\mathcal{M}$ (Line 5). If the predicted ℓ_i has any specialization, the algorithm attempts to further predict its level-2 information type (ℓ_j). If ℓ_j is predicted by the (level-2) ML-based (cf_2) or similarity-based classifiers (line 7), then the annotation $\ell_i.\ell_j$ is added to $\mathcal{M}$. Since ℓ_i has been confirmed earlier, it is sufficient to get ℓ_j predicted by one classifier (excluding keyword-based for the reasons mentioned earlier). Regardless of whether or not the algorithm succeeds to predict ℓ_j, ℓ_i is still added to $\mathcal{M}$ (line 5). The rationale is that pinpointing the sentence that discusses ℓ_i helps the legal experts easily locate ℓ_j which is expected to appear in the following sentences.

Case 2: If the level-1 information type (ℓ_i) cannot be directly predicted, the algorithm checks whether its level-2 (ℓ_j) can still be predicted. Case 2 requires ℓ_j to be predicted by two classifiers (line 18). Specifically, the label $\ell_i.\ell_j$ is added to $\mathcal{M}$ if at least one of the following three pre-conditions is satisfied: ℓ_j is predicted by the (level-2) ML-based (cf_2) and similarity-based classifiers (line 13 – first condition). Alternatively, ℓ_j is predicted by either cf_2 or the similarity-based classifier, and ℓ_j is further predicted by keyword search (line 13 – second two conditions). In Case 2, ℓ_i is automatically added to the set of annotations to get the hierarchical label, since there is enough evidence to support the prediction of ℓ_j. To obtain a joint prediction by the three classifiers in Case 2, we considered all possible combinations as described in the set of rules – line 13. These rules include combining the predictions of (i) ML-based with similarity-based, (ii) ML-based with keyword-based, and (iii) similarity-based with keyword-based.

The level-2 information types CONTROLLER.IDENTITY (C_{ID}) and CONTROLLER REPRESENTATIVE.IDENTITY (CR_{ID}) are provided by the user through the questionnaire explained in Section 3.4 (as answers to Q1 and Q5). If C_{ID} (or CR_{ID}) occurs in the sentence S, then CONTROLLER.IDENTITY (or CONTROLLER REPRESENTATIVE.IDENTITY) is added to $\mathcal{M}$ (lines 19 – 23).

Level-3 Information Types. Recall that ML-based and similarity-based classifiers are not applicable to level-3 information types due to the lack of positive examples in our training data. Therefore, we use keyword-based classification only. The algorithm attempts to predict level-3 information types based on any already predicted level-1.level-2 annotation. Specifically, the algorithm considers all level-3 information types that specialize some level-2 information type already in $\mathcal{M}$. For each level-2 information type that is predicted, if its level-3 is predicted by keyword search, then level-3 information type is added to $\mathcal{M}$ (line 27).

Algorithm 1 Information types prediction for a sentence S

```
Require: $\vec{S}$: vector representation of $S$; $cf_1$, $cf_2$: binary classifiers trained on level-1 and level-2 information
    types for $\vec{S}$, respectively; $\vec{av}(t)$: average vector for the group of sentences annotated with information
    type $t$; $\mathcal{K}$: set of information types predicted based on keyword search in $S$; $C_{ID}$, $CR_{ID}$: the values of
    CONTROLLER.IDENTITY and CONTROLLER REPRESENTATIVE.IDENTITY, respectively.
Output: $\mathcal{M}$: a set of information types predicted for $S$
$\mathcal{M} \leftarrow \emptyset$
Let $\mathcal{L}_1$ be the set of level-1 information types
for $\ell_i \in \mathcal{L}_1$ do
    if $cf_1$ predicts $\ell_i$ or sim$(\vec{S}, \vec{av}(\ell_i)) \geq 0.9$ then        // Predict level-1 & level-2 (Case 1)
        Add $\ell_i$ to $\mathcal{M}$
        for $\ell_j$ s.t. $\ell_j$ is a (level-2) specialization of $\ell_i$ do
            if $cf_2$ predicts $\ell_j$ or sim$(\vec{S}, \vec{av}(\ell_j)) \geq 0.9$ then
                Add $\ell_i.\ell_j$ to $\mathcal{M}$
            end if
        end for
    else                                                                              // Predict level-1 & level-2 (Case 2)
        for $\ell_j$ s.t. $\ell_j$ is a (level-2) specialization of $\ell_i$ do
            if ($cf_2$ predicts $\ell_j$ and sim$(\vec{S}, \vec{av}(\ell_j)) \geq 0.9$) or
            ($cf_2$ predicts $\ell_j$ and $\ell_j \in \mathcal{K}$) or
            (sim$(\vec{S}, \vec{av}(\ell_j)) \geq 0.9$ and $\ell_j \in \mathcal{K}$) then
                Add $\ell_i.\ell_j$ to $\mathcal{M}$
            end if
        end for
    end if
end for
if $S$ contains $C_{ID}$ then
    Add CONTROLLER.IDENTITY to $\mathcal{M}$
else if $S$ contains $CR_{ID}$ then
    Add CONTROLLER REPRESENTATIVE.IDENTITY to $\mathcal{M}$
end if
for $\ell_i.\ell_j \in \mathcal{M}$ do                                               // Predict level-3
    for $\ell_q$ s.t. $\ell_q$ is a (level-3) specialization of $\ell_j$ do
        if $\ell_q \in \mathcal{K}$ then
            Add $\ell_i.\ell_j.\ell_q$ to $\mathcal{M}$
        end if
    end for
end for
```

Step 7: Post-processing

In the seventh and final step of our information types identification approach, we refine the results of step 6 by considering the information types predicted for the sentences surrounding a given sentence. The intuition behind this step is the observation that specializations of certain information types are discussed in consecutive sentences of PPs. Based on this observation, when a sentence S is predicted as having a specific information type t, the surrounding context, specifically the preceding and succeeding sentences, can provide a confirmatory measure about whether t is a reliable prediction for S.

We employ several such context-based heuristics for post-processing DATA SUBJECT RIGHT, TRANSFER OUTSIDE EUROPE, and LEGAL BASIS, since these types are often discussed in consecutive sentences in the PP. The heuristic states that if some level-2 information type (ℓ_j) is predicted for a sentence (S), then we look at the n preceding and n succeeding sentences, such that n equals the number of the information types at the same level of ℓ_j. The number n accounts for the possibility to discuss the level-2 of an information type each in a separate sentence. For example, eight sentences before and after a sentence are considered to belong to the context for the level-2 information type DATA SUBJECT RIGHT.PORTABILITY, where the level-2 information types of DATA SUBJECT RIGHT can be listed in eight sentences at most.

If none of these surrounding sentences are predicted to discuss an information type relevant to ℓ_j, then we remove from the annotations for S the predicted label that includes ℓ_j. This is because the context around S lends no support to ℓ_j being a correct annotation for S. An information type ℓ'_j is said to be relevant to ℓ_j if it belongs to the same level-1 information type. To illustrate, let S be sentence number **16** in Fig. 3.1. This sentence can be falsely classified as DATA SUBJECT RIGHT.PORTABILITY, because of the misleading words ("transfer", "personal", "data"). In post-processing, we look at the information types predicted for the *eight* preceding (i.e., **8 – 15**) and *eight* following sentences (i.e., **17 – 24**) to decide if there is enough support to confirm the prediction of S. If none of the predicted information types in the context is relevant to DATA SUBJECT RIGHT.PORTABILITY, then we filter out this prediction assuming it is false.

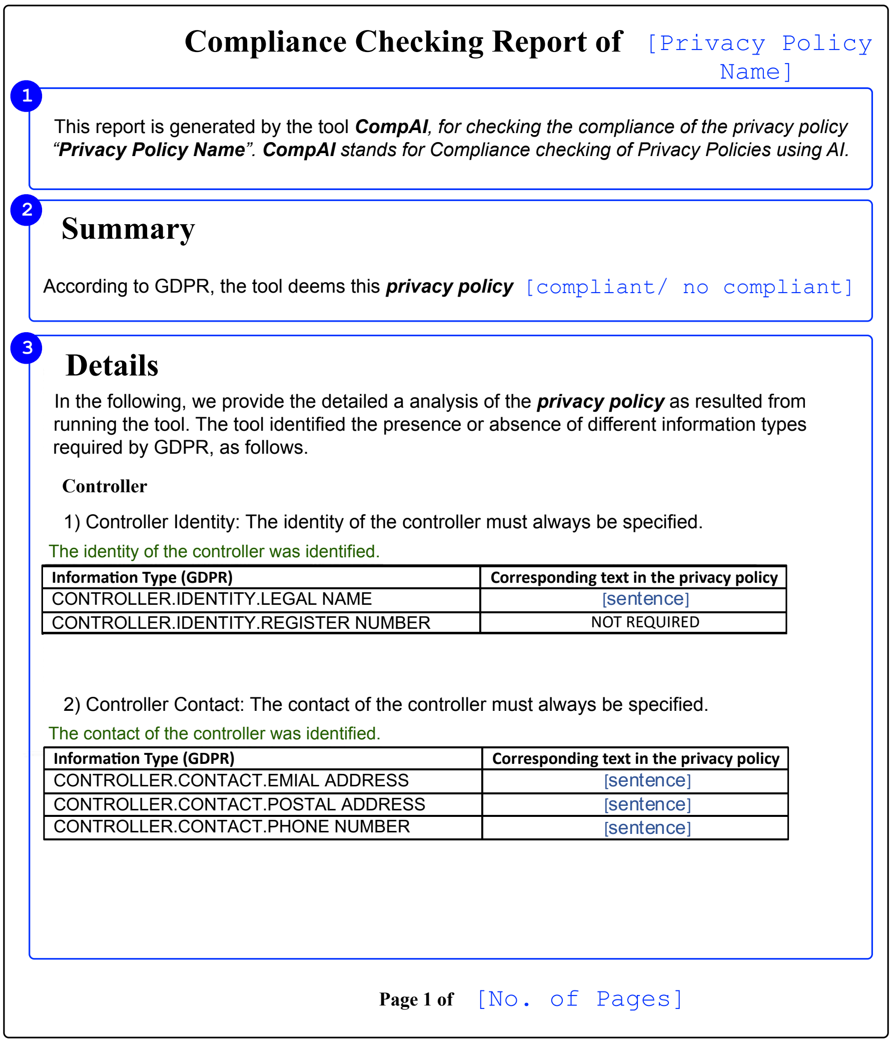
3.5.2 Compliance Checking (Phase B)

Phase B takes as an input: (1) the output of Phase A representing the predicted information types in the input PP, and (2) the answers of the user to the six questions discussed in Section 3.4. Phase B then returns a detailed report on compliance analysis as the final output of our overall approach. Fig. 3.11 shows the template of the report which $CompA\iota$ generates.

The first part is a preamble including the name of the PP. The second part presents a summary about the final decision regarding compliance. The third part shows the details of the identified information types under each compliance criterion. If the information type is not identified in any sentence, the report will show "NOT FOUND" and indicate a violation or warning accordingly. If the information type is not required because the presence of another information type is sufficient, or if the criterion is not applicable based on the answers to the questionnaire, then the report will reflect this through the respective statements "NOT REQUIRED" or "NOT APPLICABLE".

This phase implements the compliance criteria shown in Fig. 3.6 and Fig. 3.7. Using our running example in Fig. 3.1, the expected answers to the questionnaire are the following. The CONTROLLER.IDENTITY is *Hikari Bank Ltd* (**Q1**), personal data will likely be transferred outside the EU (**Q2**), there will be recipients other than the CONTROLLER (**Q3**), the core activities include processing special categories (**Q4**), processing of personal data will take place in Europe (**Q5**), and finally the personal data will be collected both directly and indirectly (**Q6**). The answer to **Q5** requires an additional input from the user about the CONTROLLER REPRESENTATIVE.IDENTITY which is *the Holding Bank Services*.

Based on the answers given above, all compliance criteria (see Section 3.4) need to be checked in this step. For example, the criterion **C22** states that PD PROVISION OBLIGED should be present in a PP when the answer to Q6 is either PD ORIGIN.DIRECT or both, and at the same time the legal basis of processing personal data is either LEGAL BASIS.LEGAL OBLIGATION or LEGAL BASIS.CONTRACT.TO ENTER CONTRACT. A violation of this criterion raises an non-compliance issue. In our example in Fig. 3.1, both the above-mentioned information types are found in the PP, in sentences **20** and **25**, respectively. As a result, we have to find the information type PD PROVISION OBLIGED in the same policy; this comes in sentence **26**. Had this sentence not been correctly identified by phase A, either due to inaccurate prediction or because it is actually missing in the policy, then this criterion would have been violated. The result of Phase B is a set of detected violations and warnings for the 23 criteria due to missing information types in the input PP.



Compliance Checking Report of `[Privacy Policy Name]`

This report is generated by the tool ***CompAI***, *for checking the compliance of the privacy policy "****Privacy Policy Name****". **CompAI** stands for Compliance checking of Privacy Policies using AI.*

Summary

According to GDPR, the tool deems this ***privacy policy*** `[compliant/ no compliant]`

Details

In the following, we provide the detailed a analysis of the ***privacy policy*** as resulted from running the tool. The tool identified the presence or absence of different information types required by GDPR, as follows.

Controller

1) Controller Identity: The identity of the controller must always be specified.

The identity of the controller was identified.

Information Type (GDPR)	Corresponding text in the privacy policy
CONTROLLER.IDENTITY.LEGAL NAME	[sentence]
CONTROLLER.IDENTITY.REGISTER NUMBER	NOT REQUIRED

2) Controller Contact: The contact of the controller must always be specified.

The contact of the controller was identified.

Information Type (GDPR)	Corresponding text in the privacy policy
CONTROLLER.CONTACT.EMIAL ADDRESS	[sentence]
CONTROLLER.CONTACT.POSTAL ADDRESS	[sentence]
CONTROLLER.CONTACT.PHONE NUMBER	[sentence]

Page 1 of `[No. of Pages]`

Figure 3.11: Template of compliance analysis report.

3.6 Empirical Evaluation

This section presents the empirical evaluation for $CompA\iota$.

3.6.1 Implementation and Availability

We have implemented our approach using Java. The implementation has $\approx$ 7,500 lines of code excluding comments and third-party libraries. For the basic NLP pipeline, we use the DKPro toolkit [99]. For text generalization, we use regular expressions available in Java. We transform words into embeddings by utilizing the publicly available pre-trained word embeddings from GloVe [60]. Noting that our implementation is Java-based, we perform operations on word embeddings using Deeplearing4j [100]. Our information types identification approach uses ML-based classification. For classification and handling imbalance in our dataset, we employ WEKA [101, 102]. For computing similarity between two textual entities in the similarity-based classification, we use Cosine Similarity [56]. All non-proprietary material related to this tool is available online [103].

3.6.2 Data Collection Procedure

Our data collection aimed at collecting and annotating PPs according to the conceptual model of Fig. 3.3. Specifically, we collected from the fund domain a total of 234 PPs, of which about 60% were provided to us by Linklaters. For the remaining 40%, we downloaded PPs from companies in the fund registry of Luxembourg, which has a substantial footprint in fund management [104]. We chose the fund domain because it is one of the main domains in which Linklaters is active. Focusing on the fund domain has an impact on the external threat to validity, as we will elaborate in Section 3.8. Nonetheless, the conceptual model described in Section 3.3 is domain-agnostic, noting that it was derived from GDPR and the (domain-independent) knowledge of legal experts about PPs.

Our data collection was performed in two steps. In the first step, a batch of 30 policies was annotated by one of the authors of this research who acquired domain expertise through close interaction with Linklaters. For annotating this first batch, hypothesis coding was applied (as explained in Section 3.3). During this step, we also drafted detailed guidelines with illustrative examples to explain the annotation process. These guidelines were then shared with the external annotators.

The second batch (204 policies) was annotated by four third-party annotators (non-authors). Three of these individuals are graduate students in social sciences; they are native English speakers with considerable prior exposure to legal documents. The fourth annotator is a computer-science graduate student with an excellent command of English and six months of prior internship experience on legal text processing in industry. All four annotators attended two four-hour training sessions, focused on GDPR concepts and the definitions of our information types. The annotators were further provided with the guidelines drafted in the first step, during which the conceptual model was also refined. To obtain an unbiased evaluation, the conceptual model was frozen before the second step started since a subset of the annotations is used for evaluating our approach as we explain later in this section. Thus, the two steps of our annotation process are performed in a strict sequence. During the entire annotation process, the annotators kept track of the keywords that were frequently used to express certain information types.

The annotators were asked to annotate each sentence in the PPs with the information types that they deemed to be present in the sentence. When no information type was present, they classified the sentence as *no information type identified*. To illustrate, consider the example in Fig. 3.1. Numbers **27** – **34** represent one sentence that includes multiple information types related to DATA SUBJECT RIGHT. The annotators would then annotate the sentence with *all* the information types that are present in the sentence. To measure the quality of our dataset, we computed the interrater agreement using Cohen's Kappa (κ) [105]. Specifically, we selected 24 PPs ($\approx$10% of our dataset) using random stratification to ensure that this subset covers some annotations from each of the four third-party annotators. The annotated sentences in the 24 PPs were independently checked by the first author who had done more than half a year of training on the compliance checking of PPs before validating the annotations.

The interrater agreement is computed for level-1 information types only. The agreement obtained at a sentence-level is on average $\kappa = 0.71$ indicating "substantial agreement" [106]. We observed that most of the disagreements occurred over the identification of PROCESSING PURPOSES and LEGAL BASIS. Since these two information types are usually related and often span multiple sentences in a PP, such disagreements are expected. LEGAL BASIS ensures that the processing of personal data for certain purposes (i.e., PROCESSING PURPOSES) is lawful when some conditions are satisfied in line with the sub-information of LEGAL BASIS. At a privacy-policy level, we obtained an average κ score of 0.87, indicating "strong agreement" [106]. This suggests that the annotators strongly agreed on which information types are present in a given PP. We believe this agreement is acceptable in our context given that our automated solution aims at identifying information types at a privacy-policy level.

Table 3.3 shows the results of our document collection. Following best practices, the entire document collection (234 PPs) is split randomly into two subsets containing about 80% and 20% of the policies, respectively used for training and development (186 policies) and for evaluation (48 policies). The first batch used in our annotation for finalizing the model is included in the training set. Hereafter, we refer to the dataset used for training as *T*, and to that used for testing as *E*. We use *E* for answering the research questions (RQs).

Table 3.3: Document collection results.

		Total		Training Data (*T*)		Test Data (*E*)	
Level	**Information Type**	**Number of PPs**	**Sentences**	**Number of PPs**	**Sentences**	**Number of PPs**	**Sentences**
L1	**Controller**	-	-	-	-	-	-
L2	Identity	221	799	175	415	46	384
L2	Contact	151	652	117	466	34	186
L1	**Controller Representative**	-	-	-	-	-	-
L2	Identity	19	36	16	33	3	3
L2	Contact	21	63	18	60	3	3
L1	**DPO**	-	-	-	-	-	-
L2	Contact	125	462	104	404	21	58
L1	**Data Subject Right**	224	3,352	180	2,651	44	701
L2	Access	209	378	167	304	42	74
L2	Rectification	211	345	169	267	42	78
L2	Restriction	170	311	137	247	33	64
L2	Complaint	172	286	137	219	35	67
L3	SA	172	264	138	217	34	47
L2	Erasure	196	386	157	296	39	90
L2	Object	181	484	145	373	36	111
L2	Portability	163	263	131	210	32	53
L2	Withdraw Consent	169	395	136	322	33	73
L1	**Legal Basis**	231	4,511	185	3,238	46	1,273
L2	Contract	161	553	127	366	34	187
L3	Contractual	123	275	89	183	34	92
L3	To Enter Contract	73	105	18	30	55	75
L3	Statutory	20	25	13	16	7	9
L2	Public Function	73	122	51	84	22	38
L2	Legitimate Interest	214	2,424	170	1,846	44	578
L2	Vital Interest	17	24	10	15	7	9
L2	Consent	180	554	141	423	39	131
L2	Legal Obligation	200	1,028	155	704	45	324
L1	**PD Origin**	216	1,904	310	125	45	356
L2	Direct	165	436	125	310	40	126
L2	Indirect	209	1,356	164	1,129	45	227
L3	Third Party	113	294	80	206	33	88
L3	Publicly	79	127	55	80	24	47
L3	Cookie	155	668	123	597	32	71

Table 3.3: Document collection results (continued).

Level	Information Type	Number of PPs	Sentences	Number of PPs	Sentences	Number of PPs	Sentences
L1	**TRANSFER OUTSIDE EUROPE**	178	823	148	707	30	116
L2	ADEQUACY DECISION	47	76	64	109	4	4
L3	COUNTRY	47	76	45	74	2	2
L2	SAFEGUARDS	136	280	113	230	23	50
L3	EU MODEL CLAUSES	96	129	76	100	20	29
L3	BINDING CORPORATE RULES	50	64	46	58	4	6
L2	SPECIFIC DEROGATION	17	20	10	13	7	7
L3	UNAMBIGUOUS CONSENT	16	18	10	12	6	6
L1	**PD CATEGORY**	228	2,209	182	1,860	46	349
L2	SPECIAL	98	265	69	198	29	67
L2	TYPE	33	71	24	60	9	11
L1	**RECIPIENTS**	209	1,599	167	1,369	42	230
L1	**PD TIME STORED**	200	873	162	738	38	135
L1	**PD PROVISION OBLIGED**	128	281	102	230	26	51
L1	**PROCESSING PURPOSES**	158	1,422	112	1,099	46	323
L1	**PD SECURITY**	182	883	140	717	42	166
L1	**AUTO DECISION MAKING**	84	295	63	224	21	71
L1	**CHILDREN**	24	70	19	58	5	12

The table provides statistics about the entire dataset, T and E. Specifically, we provide per information type t: the number of PPs in our document collection where t appears, and the number of sentences that are annotated with t. Further, this information type was annotated in a total of 3,352 sentences across the PPs in our collection. We note that none of the sentences in our dataset is annotated with CONTROLLER, CONTROLLER REPRESENTATIVE or DPO as separate labels. These information types always appear with their specializations, e.g., CONTROLLER.IDENTITY or CONTROLLER.CONTACT.

3.6.3 Evaluation Procedure

We answer RQ4 – RQ6 by conducting the experiments explained next.

EXPI. This experiment answers RQ4. We assess the accuracy of our information types identification approach. To do so, we run our approach and compare the results against manual annotations of the PPs in the test set E (defined in Section 3.6.2). We evaluate, in EXPI, the information types detected by our approach in a given PP. We recall that fo this analysis, an information type is counted only once per PP, even if it appears in multiple sentences. Designing our evaluation around PPs (instead of actual sentences) is driven by our objective, which is compliance checking. To check the compliance criteria, presented in Section 3.4, one needs to ascertain whether or not an information type exists in the PP rather than how many time it appears. To illustrate, consider criterion **C6** as an example. In **C6**, we need to find DATA SUBJECT RIGHT.PORTABILITY in the PP, when the legal ground is based on CONTRACT. If our approach is able to identify the existence of CONTRACT and PORTABILITY, at least in one sentence, then the compliance of the policy can be properly checked.

Let the mention of the information type t_i be represented, in a PP, by $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$, such that $\mathcal{S}$ is the set of sentences that are annotated with t_i according to our ground truth. Our approach deems an information type of t_i as present, if t_i is predicted as a label for at least one sentence in the PP. Following this, we define a *True Positive (TP)* when the approach correctly identifies t_i, i.e., the approach finds at least one sentence $s_j \in \mathcal{S}$. A *False Positive (FP)* is when the approach falsely identifies t_i, i.e., the approach finds a group of sentences $\mathcal{S}'$ to be about t_i, such that either $\mathcal{S}' \nsubseteq \mathcal{S}$ or there is no mention of t_i in the PP according to our ground truth. A *False Negative (FN)* is when the approach misses t_i, i.e., the approach does not find any $s_j \in \mathcal{S}$.

In EXPI, we report the overall *Accuracy (A)*, *Precision (P)*, *Recall (R)*, and the harmonic mean *F-measure (F_β)* for each information type across *E*. We compute these metrics as $A = (TP + FN)/(TP + FP + FN + TN)$, $P = TP/(TP + FP)$, $R = TP/(TP + FN)$, and $F_\beta = (1 + \beta^2) * (P * R)/(\beta^2 * P + R)$. For information types identification, recall is more important than precision, since the information types identified by the approach will be used to check compliance of PPs. This means that, if an information type is falsely introduced by the approach, it can be reviewed and filtered out by an analyst, whereas missing information types will require the analyst to review the entire PP. In EXPI, we report the *F2-measure* (i.e., β = 2) to show the evaluation in favor of recall. We choose F2-measure for two reasons: First, values of $\beta \geq 2$ do not change the reasoning about our evaluation. Second, despite recall being more important, precision still has a great value, a very low precision (too many false positive errors) will require more time and effort in filtering the erroneous findings.

EXPII. This experiment answers RQ5. We evaluate the accuracy of checking the compliance criteria on our test set. The unit of evaluation in EXPII is *a non-compliance issue* resulting from an unsatisfied criterion in a given PP. Correspondingly, we redefine a *TP* as a non-compliance issue found correctly by our approach, a *FP* as a non-compliance issue found by our approach when the criterion is satisfied, a *FN* as a non-compliance issue missed by our approach, and a *TN* when our approach correctly concludes that there is no non-compliance issue in the PP. Similar to EXPI, we report *A*, *P*, *R*, and *F2-measure*.

EXPIII. This experiment answers RQ6. We compare our *AI-based* approach for compliance checking to a simple approach that uses keyword search (hereon, referred to as *KW-based*). The latter predicts the mention of a certain information type t_i, in a given PP, if at least one keyword associated with t_i is present in any sentence in this policy. We note that keyword search is introduced as one of the classifiers in our AI-based approach (Step 5 in Fig. 3.8). To have a fair comparison, the list of keywords used in our approach is the same one used in the baseline. In EXPIII, we compare our approach against KW-based using the same evaluation metrics defined in EXPI for information types identification, and in EXPII for compliance checking.

3.6.4 Results and Discussion

In this section, we describe the results and answer RQ4, RQ5, and RQ6 (stated in Section 3.2).

RQ4. How accurate is our proposed approach in identifying GDPR-relevant information types in PPs?

Table 3.4, on the left-hand side, shows the results of EXPI. As explained in Section 3.6.3, the results are obtained by running our information types identification approach on the test set (*E*) which is comprised of 48 PPs. The table reports the accuracy, precision, recall and F2-measure computed on PPs containing information types. We show in Table 3.3 the total number of PPs containing each information type in *E*. Out of the 56 information types (see Fig. 3.3), we exclude the evaluation of CONTROLLER.IDENTITY and CONTROLLER REPRESENTATIVE.IDENTITY because they are given as input by the user, and are looked up in the PP rather than being identified like the other information types (see Algorithm 1). We also exclude TRANSFER OUTSIDE EUROPE.ADEQUACY DECISION.{TERRITORY *and* SECTOR} because we have no examples in our experimental material (both for training or testing). To summarize the different metrics, we report the *micro average* across the different information types, by computing the metrics on all TPs, FPs, FNs, TNs found across all information types.

Accuracy evaluates how the information identification approach performs in correctly predicting information types in PPs. Apart from the excluded information types (explained above), the table shows that the presence or absence of all information types are identified with an accuracy greater than 80%. The relatively low accuracy in the case of PD ORIGIN.DIRECT and PD ORIGIN.INDIRECT.THIRD PARTY is due to eight PPs in which our approach identifies sentences containing these information types, but the sentences were not the same as the ones in the ground truth. We thus counted these identifications of information types as both FPs and FNs. As for PD PROVISION OBLIGED, the approach produces 12 errors and achieves a relatively low accuracy: ≈75.5%. We note that this information type is usually expressed in a conditional statement, e.g., if the individual fails to provide the personal data as needed, there will be consequences. Conditional sentences in English take multiple forms. Thus, semantic analysis would be required to improve the accuracy of identifying this information type.

Precision reflects how many actual PPs are correctly identified by the approach as containing information types out of the total number of identified PPs. Our approach achieves a precision greater than 80% for 51 out of the 54 information types. In the case of information type LEGAL BASIS.CONTRACT.TO ENTER CONTRACT, at level L3, the reason for achieving low precision is the reliance on keyword search. Keywords can easily introduce false positives as we elaborate later in our analysis under RQ6. The same reasons mentioned above for the low accuracy of PD PROVISION OBLIGED and PD ORIGIN.INDIRECT.THIRD PARTY can also explain the low precision of these two information types. In total, our approach introduces 119 false positives out of 1,449 identified information types.

Recall assesses how many actual information types in the PPs are also correctly identified. The table shows that we achieve a high recall for all information types, except for CONTROLLER REPRESENTATIVE.CONTACT and PD PROVISION OBLIGED. Note that, in *E*, we only have three PPs containing CONTROLLER REPRESENTATIVE.CONTACT, and the low recall (66.7%) is due to missing only one of them. In total, our approach missed 68 information types from a total of 1,448 in *E*.

The answer to RQ4 is that our information types identification approach achieves an average accuracy, precision, recall and F2-measure of 93.4%, 92.1%, 95.3% and 94.9%, respectively.

RQ5. How accurate is our approach in checking the compliance of PPs?

Table 3.5, shows on top the results of EXPII. We evaluate in RQ5 how well our compliance criteria (see Section 3.4) can detect non-compliance issues, given the information types identified by the approach and evaluated in RQ4. A non-compliance issue can be either a violation or a warning (as defined in Section 3.4). The table reports the number of TPs, FPs, and FNs (redefined for EXPII in Section 3.6.3) in addition to the evaluation metrics, namely accuracy (A), precision (P), recall (R) and F2-measure (F2).

We note that seven criteria lead to warnings, namely C6, C11 – C14, C16 and C18. The remaining criteria lead to violations. We also note that C1, C2, C5, C20, and C21 are concerned with the unconditional presence of mandatory information types, whereas the criteria C3, C4, C10 – C19, C22 and C23 need to be checked only in specific situations based on the answers provided on the questionnaire (explained in Section 3.4). For the latter set of criteria, we assume in our evaluation that they *always* need to be checked. The criteria C6 – C9,

Table 3.4: Results of information types identification.

		AI-based solution (**RQ4**)				KW-based solution (**RQ6**)			
Level	**Information Type**	**A (%)**	**P (%)**	**R (%)**	**F2 (%)**	**A (%)**	**P (%)**	**R (%)**	**F2 (%)**
L1	**Controller**	-	-	-	-	-	-	-	-
L2	Contact	91.8	94.1	94.1	94.1	71.4	71.7	97.1	90.7
L3	Phone Number	95.8	92.9	92.9	92.9	85.4	68.4	92.9	86.7
L3	Email	85.7	90.9	80.0	82.0	62.5	58.1	100	87.4
L3	Legal Address	86.0	88.9	85.7	86.3	49.1	51.1	82.1	73.2
L1	**Controller Representative**	-	-	-	-	-	-	-	-
L2	Contact	97.9	100	66.7	71.4	08.2	04.3	66.7	17.2
L3	Legal Address	97.9	100	66.7	71.4	10.2	04.4	66.7	17.5
L1	**Data Subject Right**	97.9	97.8	100	99.5	91.7	91.7	100	98.2
L2	Access	88.0	90.9	95.2	94.3	84.0	87.0	95.2	93.5
L2	Rectification	100	100	100	100	70.4	81.0	81.0	81.0
L2	Restriction	95.8	94.3	100	98.8	93.8	91.7	100	98.2
L2	Complaint	100	100	100	100	87.5	85.4	100	96.7
L3	SA	100	100	100	100	60.4	67.6	73.5	72.3
L2	Erasure	100	100	100	100	100	100	100	100
L2	Object	97.9	97.3	100	99.4	97.9	97.3	100	99.4
L2	Portability	100	100	100	100	95.9	96.9	96.9	96.9
L2	Withdraw Consent	95.8	100	93.9	95.1	95.8	94.3	100	98.8
L1	**Legal Basis**	97.9	97.9	100	99.6	95.8	95.8	100	99.1
L2	Contract	85.4	82.9	100	96.0	70.8	70.8	100	92.4
L3	Contractual	86.0	88.6	91.2	90.6	83.7	90.6	85.3	86.3
L3	To Enter Contract	83.3	70.8	94.4	88.5	79.2	64.3	100	90.0
L3	Statutory	97.9	100	85.7	88.2	83.3	45.5	71.4	64.1
L2	Public Function	83.7	81.8	81.8	81.8	45.8	45.8	100	80.9
L2	Legitimate Interest	84.0	92.9	88.6	89.4	91.7	91.7	100	98.2
L2	Vital Interest	100	100	100	100	14.6	14.6	100	46.1
L2	Consent	93.8	95.0	97.4	96.9	81.3	81.3	100	95.6
L2	Legal Obligation	93.9	95.7	97.8	97.3	93.8	93.8	100	98.7
L1	**Transfer Outside Europe**	97.9	96.8	100	99.3	73.5	70.7	96.7	90.1
L2	Adequacy Decision	97.9	80.0	100	95.2	66.7	20.0	100	55.6
L3	Country	100	100	100	100	79.2	10.0	50.0	27.8
L2	Safeguards	97.9	95.8	100	99.1	97.9	95.8	100	99.1
L3	Binding Corporate Rules	100	100	100	100	100	100	100	100
L3	EU Model Clauses	97.9	95.2	100	99.0	97.9	95.2	100	99.0
L2	Specific Derogation	97.9	87.5	100	97.2	60.4	20.0	57.1	41.7
L3	Unambiguous Consent	97.9	85.7	100	96.8	58.3	15.0	50.0	34.1
L1	**PD Origin**	93.8	93.8	100	98.7	93.8	93.8	100	98.7
L2	Direct	76.5	80.4	92.5	89.8	83.3	83.3	100	96.2
L2	Indirect	93.9	95.7	97.8	97.3	93.8	93.8	100	98.7
L3	Third Party	71.7	76.5	78.8	78.3	45.8	51.6	48.5	49.1
L3	Publicly	89.6	82.8	100	96.0	83.3	75.0	100	93.8
L3	Cookie	95.8	94.1	100	98.8	89.9	90.9	93.8	93.2
L1	**PD Category**	93.9	95.7	97.8	97.4	62.0	93.5	63.0	67.4
L2	Special	95.8	93.5	100	98.6	95.8	93.5	100	98.6
L2	Type	81.6	50.0	77.8	70.0	81.3	0.0	0.0	0.0
L1	**Recipients**	93.8	93.3	100	96.6	29.0	41.9	42.9	42.7

Table 3.4: Results of information types identification (continued).

Level	Information Type	A (%)	P (%)	R (%)	F2 (%)	A (%)	P (%)	R (%)	F2 (%)
L1	**PD Time Stored**	91.7	94.7	94.7	94.7	86.0	87.8	94.7	93.3
L1	**PD Provision Obliged**	75.5	76.9	76.9	76.9	54.2	54.2	100	85.5
L1	**Processing Purposes**	84.3	91.3	91.3	91.3	40.0	58.1	54.3	55.1
L1	**PD Security**	82.7	88.4	90.5	90.0	83.7	85.4	97.6	94.9
L1	**Auto Decision Making**	93.8	95.0	90.5	91.3	64.6	55.3	100	86.1
L1	**Children**	95.8	80.0	80.0	80.0	77.1	31.3	100	69.4
L1	**DPO**	-	-	-	-	-	-	-	-
L2	Contact	97.9	95.5	100	99.1	47.9	45.7	100	80.8
L3	Phone Number	100	100	100.0	100.0	62.5	5.6	50.0	19.2
L3	Email	97.9	95.0	100	99.0	50.0	44.2	100	79.8
L3	Legal Address	91.8	81.3	92.9	90.3	20.4	17.8	57.1	39.6
	Summary	93.4	92.1	95.3	94.6	70.7	65.2	90.1	83.7

C11 – C14, C16, C18 and C22 are concerned with information types that need to be present only if some other information types are also present in the same PP.

Violations. Out of 285 genuine violations in the test PPs, our compliance criteria correctly detect 261, while introducing 16 false positives. This results in a precision of 94.2% and a recall of 91.6%.

Table 3.5 shows that the approach introduces 40 errors (16 FPs and 24 FNs) that led to violations. We analyzed the reason for having these errors. Out of the 40 errors, 26 are originated from false positives in the information types identification results. The remaining 14 errors are due to missed information types (FNs) across seven criteria. Specifically, one or two missed information types yielded errors in **C02**, **C04**, **C08**, **C09** and **C21**, whereas five missed information types yielded errors in **C22**. The low precision and recall values for the compliance checking of **C22** are in part due to the accuracy of identifying PD Provision Obliged. As we explained in RQ4, this information type requires further analysis to capture the variations of how it is expressed.

Warnings. Out of 49 genuine warnings in the test PPs, our compliance criteria correctly detect 39, while introducing seven false positives. This results in a precision of 84.4% and a recall of 79.6%. A total of 17 errors resulted in warnings, including seven FPs and 10 FNs. All of these errors are due to FPs from information identification, except one that is due to a missed information type in one PP.

Our compliance checking approach generates a report, as an output, that is shared with the analyst. The report includes not only the final decision regarding whether a PP is compliant according to GDPR, but also a structured summary of the identified information types. Specifically, the report lists under each criterion the set of sentences describing the identified information types or "not found" in case no sentence mentioning it is found by our approach. The analyst can then review this summary instead of analyzing a PP in its entirety. The criteria that erroneously result in violations or raise warnings due to false positives in the information identification (33 in total) can be easily filtered out by the analyst. If the analyst reviews the non-compliance issues only instead of the summary, our approach still fares well in identifying about 90% of the actual violations and warnings. In practice, the accuracy of our approach is sufficient to be used by diverse users, including software engineers who might lack legal expertise or legal experts who need assistance to optimize their time and effort.

Based on our assumption that all criteria need to be satisfied in order for a PP to be compliant according to GDPR, all of the 48 policies in the test set are non-compliant. Our approach is able to correctly identify all the PPs with non-compliance issues.

Table 3.5: Results of compliance checking.

AI-based approach (**RQ5**)

Criteria	**TPs**	**FPs**	**FNs**	**TNs**	**A (%)**	**P (%)**	**R (%)**	**F2 (%)**
C01	2	0	0	46	100	100	100	100
C02	13	1	1	33	95.8	92.9	92.9	92.9
C03	45	0	0	3	100	100	100	100
C04	45	1	0	2	97.9	97.8	100	99.6
C05	36	0	4	152	97.9	100	90.0	91.8
C07	7	4	0	37	91.7	63.6	100	89.7
C08	7	1	3	37	91.7	87.5	70.0	72.9
C09	31	1	1	159	99.0	96.0	96.9	96.9
C10	17	0	1	30	97.9	100	94.4	95.5
C15	2	0	1	45	97.9	100	66.7	71.4
C17	1	0	1	46	97.9	100	50.0	55.6
C19	3	0	3	42	93.8	100	50.0	55.6
C20	8	2	2	36	91.7	80.0	80.0	80.0
C21	1	1	1	45	95.8	50.0	50.0	50.0
C22	17	5	5	21	79.2	77.3	77.3	77.3
C23	26	0	1	21	97.9	100	96.3	97.0
C06	1	0	0	47	100	100	100	100
C11	6	0	0	42	100	100	100	100
C12	2	1	0	45	97.9	66.7	100	90.9
C13	3	0	0	45	100	100	100	100
C14	1	0	0	47	100	100	100	100
C16	6	1	3	38	91.7	85.7	66.7	69.8
C18	20	5	7	16	75.0	80.0	74.1	75.2
Summary	300	23	34	1,035	95.9	92.9	89.8	90.4

KW-based baseline (**RQ6**)

Criteria	**TPs**	**FPs**	**FNs**	**TNs**	**A (%)**	**P (%)**	**R (%)**	**F2 (%)**
C01	2	0	0	46	100	100	100	100
C02	2	0	12	34	75.0	100	14.3	17.2
C03	42	1	3	2	91.7	97.7	93.3	94.2
C04	2	0	43	3	10.4	100	04.4	05.5
C05	24	2	15	152	91.1	92.3	61.5	65.9
C07	7	9	0	32	81.3	43.8	100	79.5
C08	9	2	1	36	93.8	81.8	90.0	88.2
C09	30	19	2	141	89.1	61.2	93.8	84.7
C10	7	0	11	30	77.1	100	38.9	44.3
C15	0	0	3	45	93.8	n/a	0.0	n/a
C17	2	15	0	31	68.8	11.8	100	40.0
C19	2	3	4	39	85.4	40.0	33.3	34.5
C20	7	0	3	38	93.8	100	70.0	74.5
C21	1	4	1	42	89.6	20.0	50.0	38.5
C22	0	0	22	26	54.2	n/a	0.0	n/a
C23	2	0	25	21	47.9	100	07.4	9.1
C06	0	4	1	43	89.6	0.0	0.0	n/a
C11	1	0	5	42	89.6	100	16.7	20.0
C12	2	4	0	42	91.7	33.3	100	71.4

Table 3.5: Results of compliance checking (continued).

Criteria	TPs	FPs	FNs	TNs	A (%)	P (%)	R (%)	F2 (%)
C13	3	0	0	45	100	100	100	100
C14	0	0	1	47	97.9	n/a	0.0	n/a
C16	5	4	4	35	83.3	55.6	55.6	55.6
C18	24	15	3	6	62.5	61.5	88.9	81.6
Summary	174	82	159	977	82.7	68.0	52.3	54.8

The answer to RQ5 is that our compliance checking approach can detect non-compliance issues in PPs with an average accuracy, precision, recall and F2-measure of 95.9%, 92.9%, 89.8% and 90.4%, respectively.

RQ6. Is our approach worthwhile compared to a simpler solution?

Tables 3.4 and 3.5 show the results of EXPIII including, on the left-hand side and top, respectively, the results of our AI-based approach as discussed in RQ4 and RQ5, and on the right-hand side and bottom, respectively, the results of the KW-based approach that we introduced in Section 3.6.3.

Information types identification. The table suggests that there are two disadvantages of using the KW-based approach. First, not all of the information types can be accurately identified using keywords, e.g., RECIPIENTS. Recall our discussion in Section 3.1 about this information type that can include a list of organizations. A finite list of predefined keywords cannot possibly cover all organization names that might appear in RECIPIENTS or capture the diverse PROCESSING PURPOSES of personal data. Our ground truth contains a total of 42 PPs containing RECIPIENTS and 46 containing PROCESSING PURPOSES.

We note that the predicted RECIPIENTS and PROCESSING PURPOSES predicted by KW-based are counted as both FPs and FNs in 21 and 17 cases. This is because none of the identified sentences associated with these information types are matching the ones in the ground truth. In contrast, our AI-based approach finds only three cases of PROCESSING PURPOSES with irrelevant sentences. This shows that our approach is more reliable in finding the correct sentences related to the identified information types (35 less such errors).

The second disadvantage of KW-based is that, though it achieves a relatively good recall, this comes at the cost of precision. For example, the recall for identifying the information type TRANSFER OUTSIDE EUROPE.ADEQUACY DECISION using keywords is 100% but the precision is only 20%. Despite such high recall, our AI-based approach achieves an overall better F2-measure, namely +11%. To summarize, our AI-based approach misses in total 68 information types (FNs) and introduces 119 FPs, whereas KW-based misses 144 FNs and produces 697 FPs (76 more FNs and 578 more FPs than our approach). As a result, we achieve a gain of ≈5% in recall and ≈27% in precision.

Compliance checking. The difference in performance becomes even clearer in the compliance checking task, which depends largely on the accuracy of information types identification. The KW-based approach has a respective precision and recall of 71.6% and 48.9% for detecting violations, and 55.7% and 69.4% for detecting warnings. In comparison with the total of 57 errors produced by our approach for violations and warnings, KW-based produces 241 errors (i.e., 184 more errors). Of these 241 errors, 45 are due to missed information types (FNs), 15 are caused by PD CATEGORY (which is hard to capture via keywords), and the remaining 196 are originating from false positives in information types identification. Filtering so many cases out is, compared to the 33 FPs introduced by our approach, much more time-consuming for the analyst. Our approach is therefore advantageous over the KW-based solution in terms of both precision and recall. Specifically, using a combination of NLP and ML leads to a significant improvement of ≈23% in precision and ≈43% in recall for detecting violations. The overall improvement, considering both warnings and violations, is ≈25% higher precision,

≈38% higher recall, and ≈36% higher F2-measure.

The answer to RQ6 is that our AI-based approach presents a significant improvement over merely using keywords, both in information types identification and in compliance checking. In information types identification, our approach outperforms the KW-based solution by an average of 22.7% in accuracy, 26.9% in precision, 5.2% in recall and 11% in F2-measure. This leads to a significant follow-on gain in compliance checking, where our approach outperforms the baseline by 24.5% in precision and 38% in recall.

3.7 Related Work

Our proposed approach for checking the compliance of PPs spans three different tasks. The first task involves the elicitation of privacy-related requirements for GDPR compliance. The second task covers the compliance checking of PPs (with a GDPR focus). The last task is concerned with checking the data handling practices and privacy compliance of software against their associated PPs. This last task enables an implicit compliance checking of the software against the privacy-requirements stated in the provisions (GDPR, in our case). Our work concentrates on providing automation for the first two tasks, noting that the results from these two tasks also serve as an input to the third task, which we do not directly address in this research. Below, we position our work against the related work on (i) identifying privacy-policy requirements, (ii) compliance checking of PPs, and (iii) completeness/compliance checking of software against data protection regulations.

3.7.1 Elicitation of Privacy-related Requirements

Vanezi et al. [107] propose a graphical modeling language for GDPR PPs and a methodology for transforming such graphically-defined PPs into formal definitions. This work focuses on one (namely, PROCESSING PURPOSES) out of the 56 information types we consider in our work. Caramujo et al. [86] target PPs for the web and mobile applications, and propose a domain-specific language along with model transformations for specifying privacy-policy models. Similarly, Pullonen et al. [108] present a multi-level model to be used as an extension of the Business Process Model and Notation to enable the visualization, analysis, and communication of the privacy-policy characteristics of business processes. Finally, Kumar and Shyamasundar [109] explore the suitability of information-flow controls as a tool for specifying and enforcing privacy-policy requirements. These existing works address a rather small subset of the privacy-policy information types considered in this research. In addition, excluding [107], all of the above-mentioned papers focus on providing guidelines that are not strictly based on GDPR. In contrast, we systematically identify the requirements that, according to GDPR, must be met by PPs for compliance.

3.7.2 Compliance/Completeness Checking of PPs

Sánchez et al. [110] check the compliance of PPs with respect to six data protection goals as stated by GDPR, including lawfulness, purpose limitation, data minimisation, accuracy, storage limitation, and integrity and confidentiality. The authors use four PPs to train binary classifiers for deciding whether a PP is compliant with respect to each of the six goals. These goals cover only 15 out of the 56 information types we handle. Nejad et al. [111] present three different models for classifying the paragraphs of PPs into pre-defined categories using supervised machine learning. To train their models, the authors use a dataset containing 115 PPs from various US companies. The authors consider 12 high-level categories for their classification. All these categories are included in our set of information types. Tesfay et al. [20] propose a ML-based method for classifying the content of PPs across 10 categories using predefined keywords. Those categories are all covered by our information types, except for one category, Policy Change, which is orthogonal to our purposes.

Bhatia et al. [19] develop a semi-automated framework for extracting privacy goals from PPs through crowdsourcing and NLP. Similar crowdsourcing initiatives have been proposed by others as well, e.g., by Liu et al. [112] and Wilson et al. [113], where PPs are manually annotated in order to match their text segments against privacy issues of interest. Guerriero et al. [114] propose a framework for specifying, enforcing and checking PPs in data-intensive applications. Bhatia et al. [21] present a semantic frame-based representation for privacy statements that can be used to identify incompleteness in four categories of data action: collection, retention, usage, and transfer. Lippi et al. [115] present 33 information types for GDPR PPs and provide automatic support for vagueness detection based on manually crafted rules and ML classifiers built using the exact terminology of the policies as learning features.

In summary, in comparison to the above-cited works, we have a different analytical focus, namely compliance checking. In terms of the information types, our proposed 56 types cover all the ones identified by others, except – as noted before – one metatadata type, Policy Change [20], which is orthogonal to compliance checking. Furthermore, the existing approaches outlined above rely to a large extent on the exact phrasing of the policies to be able to extract and classify information. They do not present a thorough conceptualization of the content expected in PPs. The scope of application of these approaches is thus limited and, where automation is provided, the accuracy is not high enough for industrial use. In this chapter, we addressed the above limitations by considering a wider set of information types and using a combination of advanced NLP and ML for automated support.

3.7.3 Compliance Checking of Software

Fan et al. [89] check for the compliance of mobile health applications against GDPR. To do so, the authors propose an automated system for detecting three types of violations: incompleteness of the app PP, inconsistency of data collection, and non-secure data transmission. For incompleteness checking of PPs, the authors define six categories of privacy-related information that need to be present in a PP. They apply ML-based binary classifiers on bag-of-words representations of the sentences in a given policy to predict whether any of the categories is present in the policy. Specifically, they apply random forest (RF), decision trees (DT) and naïve Bayes (NB). Based on 10-fold cross-validation over 100 PPs (1,284 labelled sentences), RF performed the best in four of the six categories, and DT performed the best in the other two. The best reported precision and recall are on average 92.5% and 93.3%, respectively. In contrast with our work, the authors consider only six out of the 56 information types that we present in this chapter. Moreover, and from an ML standpoint, our solution architecture is different: we use embeddings to create the representations of the text in a given policy and apply an ensemble classification approach.

The COVID-19 pandemic has heightened privacy concerns for individuals, as seen, for example, in the analysis of app reviews for COVID-19 contact-tracing apps [116]. Hatamian et al. [117] analyze the privacy and security aspects of COVID-19 contact tracing apps. In their analysis, they consider 12 information types derived from different GDPR articles, including children protection, data retention and others. The authors collect the data access intentions from the permissions an Android app is given, e.g., the access to data such as call logs and contact lists. Through manual incompleteness checking of the PPs of 28 COVID-19 contact tracing apps, the authors assess the extent to which the policies cover the 12 GDPR principles. Subsequently, the authors check whether the apps fulfill the provisions in their respective PPs. Nine of the privacy principles addressed by

Hatamian et al. are pertinent to privacy-policy completeness checking and are thus tackled in our work. However, we provide a more elaborate treatment of these principles (information types). Moreover, we devise an AI-based solution to automatically identify these information types in the PPs and thereby analyze compliance.

Kununka et al. [118] assess the compliance of Android and iOS apps with their PPs. The basis for selecting which apps to analyze is the number of third-party domains that the apps transfer sensitive data to. In total, 30 apps are selected. The authors first analyze the categories of personal data transferred to a third-party. They then manually identify information types in the PPs of these apps, focusing only on the collection, use and transfer of personal data. Finally, the authors check for the compliance of the data practices in the apps against what is stated in the policies. Compared to our automated approach, their information identification from both the apps and their PPs, as well as the compliance checking, are done entirely manually. Further, they consider only two information types (i,e., PD CATEGORY and its specialization SPECIAL) from what we present in this research.

3.8 Threats to Validity

Below, we discuss threats to the validity of our empirical results and what we did to mitigate these threats.

Internal Validity. Bias was an important concern in relation to internal validity. To mitigate bias, we curated most ($\approx$90%) of the manual annotations through third-parties (non-authors). Another potential threat to internal validity is that the authors interpreted the text of GDPR provisions in order to create the PP conceptual model presented in Fig. 3.3. To minimize the threat posed by a subjective interpretation, this phase was done in close collaboration with three independent legal experts from Linklaters, who have expertise in European and international laws with a focus on the data protection and financial domains. While we cannot entirely rule out subjectivity, we provide our interpretation in a precise and explicit form. In addition, our model is publicly available and thus open to scrutiny. Another threat to internal validity is our reliance on a static set of keywords. Changing this set might have an impact on the results of our automated solution. However, we believe that our set of keywords is reasonably adequate and exhaustive since we manually created the keywords during our qualitative study in close collaboration with legal experts.

External Validity. The qualitative study through which we built our conceptual model of PPs is domain-agnostic: the study was rooted in GDPR and further enhanced by feedback from legal experts who had familiarity with data protection in a variety of domains. This provides a fair degree of confidence about our conceptual model being generalizable. As for our evaluation of automation accuracy of our compliance checking approach (see Section 3.6.4), and more specifically, whether the accuracy levels observed would generalize beyond the fund domain, we note that certain information types were rare in PPs from the fund domain. Furthermore, we have not yet conducted a multi-domain evaluation of our information types identification and compliance checking approaches. For these reasons, it would be premature to make claims about how our accuracy results would carry over to other domains. That said, we believe that the core components of our automation approach, notably, our hybridized use of word embeddings, ML-based classification, similarity analysis and keyword search, provides a versatile basis for the future development of a more broadly applicable solution to check the compliance of PPs.

3.9 Expert Interview Survey

To assess the usefulness of our tool ($CompA\iota$) in practice, we conducted an interview survey with legal experts from our collaborating partner at Linklaters LLP. Their expertise spans several areas including corporate laws, commercial litigation, data protection laws and financial regulations, and information technology within the legal domain. They specially have long experience in providing GDPR advisory and compliance services. The survey was conducted in a single session, with the participation of two legal experts from Linklaters LLP and all the members of our research team. The survey material consists of two privacy policies (PP1 – PP2), automatically analyzed by $CompA\iota$. Table 3.6 lists the details of the analyzed PPs. For each PP, the table reports the total number of pages, the number of pages marked by $CompA\iota$ as containing information types as well as the total number of information types identified. Using Likert scales [119], our survey aimed at collecting feedback from the experts on the four questions and two follow-up questions listed in the PP-related questionnaire.

Table 3.6: Survey material details.

Privacy policy	Pages	Pages with information types	Information types found by $CompA\iota$
PP 1	10	10	66
PP 2	8	6	38
Summary	18	16	104

At the beginning of the survey, we explained all the questions in the questionnaire to the legal experts along with examples. We then asked the experts to separately respond to all the questions. The experts' feedback for Questions 1 – 4 (**Q1** – **Q4** below) was collected on each page in the PPs. To ensure the understanding of the questions ratings and to properly collect the rationale behind their answers, we asked the legal experts to verbalize their reasoning and further discuss their rationale whenever they disagreed. To mitigate fatigue, we conducted the survey for a duration of approximately two hours. Given the limited availability of experts and time restrictions, both experts were provided with the survey material, i.e., the two PPs with the analysis generated by $CompA\iota$, one week in advance and were asked to familiarize themselves with the content.

Questions 1 and 2 are concerned with identifying *false negatives* and *false positives*, respectively. In other words, which information types $CompA\iota$ missed and which ones it falsely introduced. For Questions 1a, 2a, 3 and 4, the experts rated the questions on a five-point Likert scale [119]. The possible options were: "Strongly Agree", "Agree", "Neutral", "Disagree", and "Strongly Disagree". In some cases when Questions 1-a and 2-a were not required (e.g., no information types are identified on a page), the experts were provided with the additional option "Not Relevant". The results from these cases have been excluded from the analysis.

Questionnaire. Our questionnaire included the following questions:

Q1- On this page, indicate all the information types that have not been identified by $CompA\iota$? *Highlight all.*

a. The information types found by $CompA\iota$ helped me to easily spot the missed information types. (Asked for each missed information type)

○ *Strongly agree* ○ *Agree* ○ *Neutral* ○ *Disagree* ○ *Strongly disagree* ○ *Not relevant*

Q2 On this page, indicate all information types found by $CompA\iota$ that are not information types? *Highlight all.*

a. The information type found by $CompA\iota$ is not an information type, but it provides useful information that would trigger further discussion. (Asked for each information type marked as false by experts)

○ *Strongly agree* ○ *Agree* ○ *Neutral* ○ *Disagree* ○ *Strongly disagree* ○ *Not relevant*

Q3 On this page, I would perform the compliance analysis faster and more efficiently with the help of $CompA\iota$ than without it.

○ *Strongly agree* ○ *Agree* ○ *Neutral* ○ *Disagree* ○ *Strongly disagree* ○ *Not relevant*

Q4 On this page, given my time budget in daily practice, it is likely that I would have missed some important information if I had done the compliance analysis fully manually.

○ *Strongly agree* ○ *Agree* ○ *Neutral* ○ *Disagree* ○ *Strongly disagree* ○ *Not relevant*

3.9.1 Survey Results

Table 3.7 summarizes the results from the expert interview survey, which we conducted according to the procedure described above. The table provides overall statistics from the survey, showing for each PP the number of information types found by $CompA\iota$, the number of information types marked as correct by experts (true positives or TPs), the number of information types marked as wrong by experts (false positives or FPs), the number of information types missed by $CompA\iota$ (false negatives or FNs), and the corresponding precision and recall metrics.

Table 3.7: Interview survey results.

Privacy policy	Information types found by $CompA\iota$	TPs	FPs	FNs	P(%)	R(%)
PP1	66	62	4	5	93.9	92.5
PP2	38	33	5	0	86.8	100
Summary	104	95	9	5	91.3	95.0

With regard to Question 1, the experts identified a total of five FNs. Further, the experts marked as correct 95 out of 104 information types found by $CompA\iota$. Thus, the average recall of $CompA\iota$ is 95.0%. For each of the five FNs, the experts answered the follow-up Question 1a from the PPs-related questionnaire. Notice that in occasions, these FNs were found in the surrounding text of FPs. For instance, when an information type is present in a given sentence but $CompA\iota$ finds the information type in a sentence close to the previous one. Both experts provided positive answers to Question 1-a for all FNs, that is the experts “Strongly Agree” that the findings of $CompA\iota$ did help in identifying the FNs.

With regard to Question 2, the experts marked as wrong a total of nine information types found by $CompA\iota$ (i.e., FPs). Thus, the average precision of $CompA\iota$ is 91.3%. In each case, for the follow-up Question 2a, the answers chosen by both experts were “Agree” in six cases and “Neutral” in the remaining three. This indicates that the text wrongly classified as containing information types often points out some useful information for the compliance checking process.

With regard to Question 3, we had 16 (10 + 6) responses, one response per each page containing information types. The 56.2% of the experts responses used the answer “Strongly Agree”. The remaining 43.8% used the category "Agree". The experts agreed that $CompA\iota$ helps them check the compliance of PPs more efficiently. With regard to Question 4, similar to Question 3, we had 16 responses in total. The 68.7% of the experts responses used the category "Strongly Agree". The remaining 31.3% used the category "Agree". The experts agreed that $CompA\iota$ helps them locate information types that they might otherwise overlook in a daily basis, within the given time budget.

Overall, for Questions 3 and 4, the 62.5% of the responses used the category "Strongly Agree", the remaining 37.5% used the category "Agree". These results show that automated support is highly beneficial for assisting human analysts in checking the compliance of PPs.

3.10 Replicating our Methodology

Our proposed compliance checking process is not limited to PPs and GDPR, and can be instantiated for checking the compliance of any given document type (*D*) according to any given regulation (*R*). In our context, *D* represents a PP, and *R* is GDPR. Reusing our approach can be done by replicating the same methodology as described in this chapter. Specifically, one must first conduct a qualitative study over those of *R*'s provisions that are relevant to checking the compliance of *D*. Such a qualitative study should aim at building a conceptual model and a set of compliance checking criteria. Subsequently, one must develop automation for compliance checking. When supervised machine learning is used for automation, one will need to (manually) create a labeled dataset covering a relatively large number of documents of type *D*. This will be followed by the development of classification methods, potentially alongside prediction rules and post-processing steps.

The effort required to replicate our methodology for other regulations and document types depends on several factors, including the number and complexity of the provisions that need to be considered in the qualitative study, the background and expertise in conceptual modeling and AI, the size of the evaluation data and the complexity of the classification algorithms used in the work. Based on our experience, we anticipate that 30-40% of the effort would go towards building a conceptual model and compliance criteria and the remaining 60-70% would go towards developing an automated solution.

3.11 Summary

In this chapter, we proposed an AI-enabled approach for compliance checking of PPs according to the GDPR. We first developed a conceptual model aimed at providing a thorough characterization of the content of PPs. Based on this conceptual model, we devised criteria describing how a PP should be checked for compliance against GDPR. Second, using NLP and ML, we developed automated support for classifying the content of PPs and thus identifying the information types necessary for checking privacy-policy compliance.

To evaluate our approach, we curated a considerable number of real PPs (234 policies in total), with the majority of the annotation work performed by third-parties. On a test set of 48 PPs, our information types identification approach achieved an average precision of 92% with an average recall of 95% for identifying the information types across the test PPs. Applying the compliance criteria over the identified information types, our compliance checking approach detected 300 out of 334 genuine non-compliance issues, while producing 23 false positives. Our compliance checking approach thus had a precision of 93% and a recall of 90% over our test set. Compared to an intuitive automated solution based on keyword search, our AI-based approach leads to a significant improvement in precision and recall of 27% and 5% for information types identification and of 24.5% and 38% in compliance checking, respectively.

In the future, we plan to enhance our compliance criteria so that they consider not only the presence/absence of information types but also the meaning of the sentences containing the information types. Another important direction for future work is to go beyond our current case-study domain (funds) in order to assess the generalizability of our approach.

Chapter 4

NLP-based Automation for Compliance Checking of Data Processing Agreements against GDPR

When the entity processing personal data (the processor) differs from the one collecting personal data (the controller), processing personal data is regulated in the EU by the general data protection regulation (GDPR) through data processing agreements (DPAs). Checking the compliance of DPAs contributes to the compliance checking of software systems as these documents are an important source of requirements for software development involving the processing of personal data. However, manually checking whether a given data processing agreement (DPA) complies with GDPR is challenging as it requires significant time and effort for understanding and identifying DPA-relevant compliance requirements in GDPR and then checking these requirements in the DPA. Legal texts introduce additional complexity due to convoluted language and inherent ambiguity leading to potential misunderstandings. Thus, an automated solution for checking the compliance of a given DPA against GDPR is highly desirable.

In this chapter, we propose an automated approach for checking the compliance of DPAs using natural language processing (NLP) technologies. Specifically, approach leverages semantic analysis for automatically identifying whether the textual content of a given DPA complies with the GDPR provisions. For this, we first represent the applicable provisions of GDPR as "shall" requirements. We then analyze at phrasal-level the text in a given DPA to identify the breaches according to what is required by the shall requirements.

Structure. The remainder of this chapter is structured as follows. Section 4.1 presents the motivation and contributions of this chapter. Section 4.2 presents the background related to the chapter. Section 4.3 presents the research questions. Section 4.4 presents our method for extracting the DPA-related requirements from GDPR. Section 4.5 describes our automated approach. Section 4.6 reports on our empirical evaluation. Section 4.7 reviews related work. Section 4.8 discusses threats to validity. Section 4.9 describes how our methodology can be adapted beyond GDPR. Section 4.10 presents an experts interview survey. Finally, Section 4.11 concludes the chapter.

4.1 Motivation and Contributions

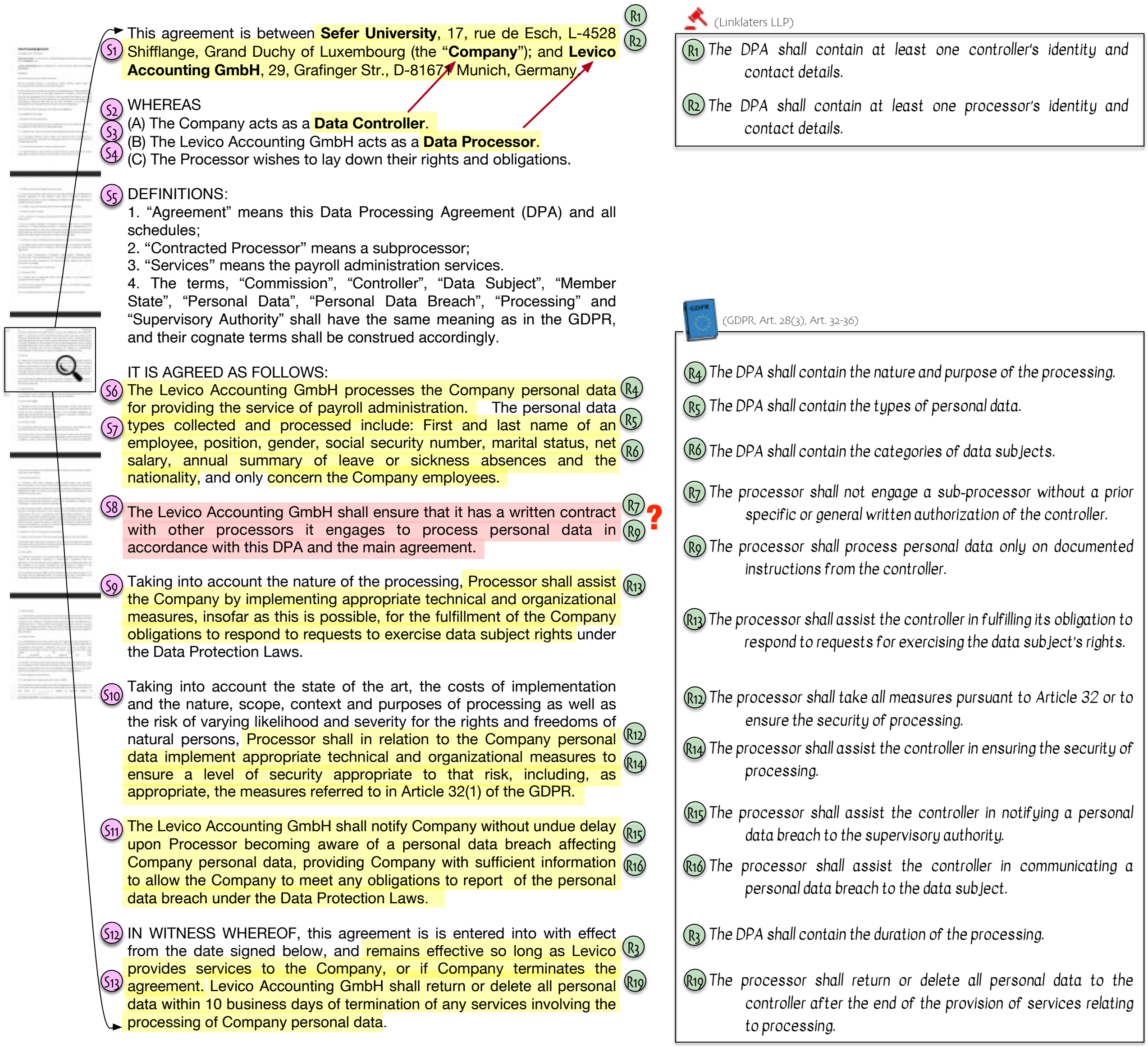
The advances in AI technologies led to increasingly integrated modern software systems (e.g., mobile applications) in everyday life activities. Applications are often developed to provide convenient services or assistance to individuals (e.g., Amazon Alexa). However, they also entail challenges regarding compliance to privacy standards and data protection requirements, e.g., when sharing personal data with third parties [120]. Individuals (i.e., the applications' end-users) who provide explicit consent to these applications for allowing the processing of their personal data must have guarantees that their data remains protected, also when shared with third parties [121]. Worldwide, GDPR obliges organizations to provide such data protection guarantees when handling personal data of EU residents. Violating GDPR can levy penalty fines that could exceed one billion euros [122].

Software systems which typically involve processing or sharing personal data are subject to compliance with GDPR. Developing GDPR-compliant software requires taking into account the legally-binding agreements signed between different organizations involved in collecting and processing personal data. Examples of such agreements include PPs between organizations and individuals (often signed by individuals) as well as DPAs between two organizations involved in collecting and processing personal data. Individuals are not necessarily aware of DPAs. We focus in our work on checking the compliance of DPAs against the provisions of GDPR. From an RE standpoint, compliance checking is a prerequisite for capturing a complete set of *privacy-related requirements* covered in the DPAs to be implemented in software systems for processing personal data.

Data processing normally involves an individual (i.e., *data subject*) who willingly shares personal data, an organization (i.e., *data controller*) that collects and in many cases further shares personal data, and another organization (i.e., *data processor*) that processes personal data on behalf of the controller. A data processor can share personal data again with yet another organization (i.e., *sub-processor*) to perform some data processing services on its behalf. Individuals in this chain are often aware of the terms based on which their personal data is collected and handled as described in privacy notices of the data controller. All subsequent sharing of individual's personal data with processors (and sub-processors) is however not directly visible to individuals. According to GDPR, both the controller and processor should share the responsibility of protecting personal data of individuals. Consequently, a DPA listing privacy-related requirements should be established between the controller and the data processor(s) [123]. To be deemed GDPR-compliant, a DPA must explicitly cover all the criteria imposed by the GDPR provisions concerning data processing. Establishing a DPA includes setting terms for how data is used, stored, protected, and accessed as well as defining the obligations and rights of the controller and processor. By signing a DPA, the processor is obliged to ensure that any software system deployed for processing personal data has to also comply with GDPR.

4.1.1 Running Example

Let's consider the scenario of an educational institution (called as *Sefer University*) which shares personal data of its employees with a third-party service provider (e.g., accounting office) for a particular purpose (e.g., payroll administration). Examples of shared personal data include: first and last name, birth date, marital status, annual summary of leave or sickness absences and a scanned copy of a valid personal identification. The data controller in this scenario is Sefer University which collects the personal data of the employees (data subjects). The data processor is the service provider (in our example, the accounting office) performing financial services on behalf of the controller. Details concerning this service provider (e.g., the name and address of the hired accounting office) might not be shared with employees. To ensure that their personal data is sufficiently protected, GDPR



S1 This agreement is between **Sefer University**, 17, rue de Esch, L-4528 Shifflange, Grand Duchy of Luxembourg (the "**Company**"); and **Levico Accounting GmbH**, 29, Grafinger Str., D-81671 Munich, Germany R1 R2

S2 WHEREAS
S3 (A) The Company acts as a **Data Controller**.
S4 (B) The Levico Accounting GmbH acts as a **Data Processor**.
(C) The Processor wishes to lay down their rights and obligations.

S5 DEFINITIONS:
1. "Agreement" means this Data Processing Agreement (DPA) and all schedules;
2. "Contracted Processor" means a subprocessor;
3. "Services" means the payroll administration services.
4. The terms, "Commission", "Controller", "Data Subject", "Member State", "Personal Data", "Personal Data Breach", "Processing" and "Supervisory Authority" shall have the same meaning as in the GDPR, and their cognate terms shall be construed accordingly.

IT IS AGREED AS FOLLOWS:
S6 The Levico Accounting GmbH processes the Company personal data for providing the service of payroll administration. R4
S7 The personal data types collected and processed include: First and last name of an employee, position, gender, social security number, marital status, net salary, annual summary of leave or sickness absences and the nationality, and only concern the Company employees. R5 R6

S8 The Levico Accounting GmbH shall ensure that it has a written contract with other processors it engages to process personal data in accordance with this DPA and the main agreement. R7 ? R9

S9 Taking into account the nature of the processing, Processor shall assist the Company by implementing appropriate technical and organizational measures, insofar as this is possible, for the fulfillment of the Company obligations to respond to requests to exercise data subject rights under the Data Protection Laws. R13

S10 Taking into account the state of the art, the costs of implementation and the nature, scope, context and purposes of processing as well as the risk of varying likelihood and severity for the rights and freedoms of natural persons, Processor shall in relation to the Company personal data implement appropriate technical and organizational measures to ensure a level of security appropriate to that risk, including, as appropriate, the measures referred to in Article 32(1) of the GDPR. R12 R14

S11 The Levico Accounting GmbH shall notify Company without undue delay upon Processor becoming aware of a personal data breach affecting Company personal data, providing Company with sufficient information to allow the Company to meet any obligations to report of the personal data breach under the Data Protection Laws. R15 R16

S12 IN WITNESS WHEREOF, this agreement is is entered into with effect from the date signed below, and remains effective so long as Levico provides services to the Company, or if Company terminates the agreement. R3
S13 Levico Accounting GmbH shall return or delete all personal data within 10 business days of termination of any services involving the processing of Company personal data. R19

(Linklaters LLP)

R1 The DPA shall contain at least one controller's identity and contact details.

R2 The DPA shall contain at least one processor's identity and contact details.

(GDPR, Art. 28(3), Art. 32-36)

R4 The DPA shall contain the nature and purpose of the processing.

R5 The DPA shall contain the types of personal data.

R6 The DPA shall contain the categories of data subjects.

R7 The processor shall not engage a sub-processor without a prior specific or general written authorization of the controller.

R9 The processor shall process personal data only on documented instructions from the controller.

R13 The processor shall assist the controller in fulfilling its obligation to respond to requests for exercising the data subject's rights.

R12 The processor shall take all measures pursuant to Article 32 or to ensure the security of processing.

R14 The processor shall assist the controller in ensuring the security of processing.

R15 The processor shall assist the controller in notifying a personal data breach to the supervisory authority.

R16 The processor shall assist the controller in communicating a personal data breach to the data subject.

R3 The DPA shall contain the duration of the processing.

R19 The processor shall return or delete all personal data to the controller after the end of the provision of services relating to processing.

Figure 4.1: Excerpt from a DPA verified against GDPR requirements.

requires Sefer university to sign a DPA with the accounting office. Fig. 4.1 shows on the left-hand side an excerpt from a DPA that addresses this scenario, and on the right-hand side a set of DPA-related requirements extracted from GDPR, ordered by their occurrence in the DPA. In Section 4.4, we elaborate on the complete list of GDPR requirements. In the remainder of the chapter, we use *requirements* to refer to the privacy-related requirements extracted from GDPR concerning data processing, and *statements* to refer to the textual content of a given DPA. For simplicity, a statement is regarded as equivalent to one sentence in the DPA.

Checking the compliance of the DPA in Fig. 4.1 requires mapping the statements (marked as S1 – S13) in the DPA against the requirements envisaged by GDPR (marked as R1 – R7, R9, R12 – R16, and R19). Note that the requirements numbering is not sequential and is defined by the complete list presented in Table 4.2. The most immediate straight-forward solution that comes to mind for matching statements with requirements is by applying semantic similarity measures, e.g., cosine similarity [28]. In brief, cosine similarity is a measure that assigns a score between 0 and 1 to a pair of text sequences. The higher the score, the more semantically similar the two text sequences. Following this, a statement is satisfying a given requirement when the cosine similarity is sufficiently high. Below, we discuss the limitations of such a simple solution.

The example shows that S1 satisfies both R1 and R2 since it clearly identifies the data controller (Sefer

University) and processor (Levico Accounting GmbH). However, computing the semantic similarity between S1 and R1 and R2 results in scores equal to 0.02 and 0.14, respectively. Low similarity scores in this case should be expected since S1 contains specific entities in place of the identity and contact details of the controller and processor (required by GDPR). Without considering S2 and S3, one cannot distinguish between the data controller and processor fully automatically. Since this is key information that has a significant impact on compliance checking, we assume in our work that the name(s) of the data controller and processor are provided as input by the human analyst. To properly check the compliance of a given DPA, the requirements have to be checked for the right entity. For example, the processor's obligations in R7, R9, R12-R16, and R19 in Fig. 4.1 must be associated with the processor. On a similar basis, S12 has a similarity score of 0.3 with R3, S6 0.38 with R4, S7 0.53 with R5 and 0.33 with R6. All these semantic similarity scores are too low to enable a decision about compliance. Using semantic similarity is not suitable for checking the compliance against R1 – R6 since they require content that varies across DPAs, e.g., a list of collected personal data or the processing purpose.

The example further shows that S8 satisfies neither R7 nor R9, yet it has a semantic similarity of 0.52 and 0.57, respectively. S8 is about having a written agreement with sub-processors, while R7 concerns acquiring a written approval from the controller before engaging sub-processors. R9 is related to processing personal data only based on documented instructions from the controller. Though there is some resemblance in the wording of S8 and that of R7 and R9, it is still clear that S8 does not comply with either requirements.

The example illustrates that statements in DPAs are often verbose, i.e., include more content that is needed for GDPR compliance. For example, S9 satisfies R13 with a similarity score of 0.87, whereas S10 satisfies both R12 and R14 with a similarity score of 0.52. This indicates that irrelevant content in S10 reduces the overall semantic similarity, and thus hinders compliance checking. Computing the similarity with only the relevant part of S10 (shaded yellow) leads to higher scores: 0.71 with R12 and 0.60 with R14.

Further, statements can satisfy multiple requirements. For example, S11 satisfies R15 and R16. The similarity scores between S11 and R15 and R16 are 0.82 and 0.79, respectively. However, S11 does not specify stating to whom data breaches shall be notified (that is the supervisory authority in R15 or the data subject in R16). In this case, S11 can be regarded as *partially* satisfying both requirements. We see that a GDPR requirement can contain different key elements which must be verified in the DPA statement. For instance, key elements in R15 include *the processor*, *assist in notifying*, *the controller*, *personal data breach*, and *supervisory authority*. Checking all key elements is necessary to understand the missing content in a statement. To become compliant, S11 can be improved by adding missing information content.

The above discussion highlights that the seemingly simple task of mapping statements in a given DPA to GDPR requirements cannot be addressed accurately through straight-forward automation based on semantic similarity. Alternatively, manually analyzing the content of DPAs for compliance against GDPR is a very tedious and time-consuming task for legal experts and requirements engineers alike. In this chapter, we propose an NLP-based automation for checking the compliance of DPAs against GDPR. We develop our solution in close collaboration with legal experts from our industry partner (Linklaters LLP).

Ensuring that the DPA excerpt in Fig. 4.1 is compliant with GDPR, and consequently provides a complete list of requirements, has an impact on building a GDPR-compliant software system. For example, a system deployed by the processor shall not process or store any personal data outside the list specified in S7. The system shall further implement: (i) appropriate protection procedures (e.g., encryption, data anonymization) to ensure the required level of security in accordance with S10, and (ii) a data breach notification procedure, corresponding to S11, which shall send a text message to the controller and also collect an acknowledgment of receipt.

As we elaborate in Section 4.7, much work has been done in the RE literature on automating regulatory

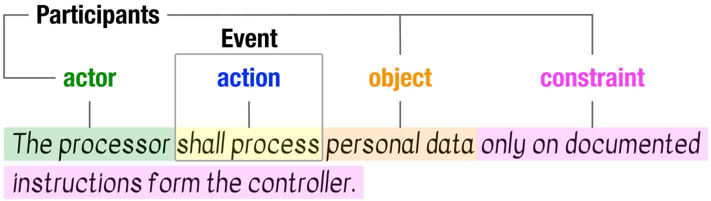


Figure 4.2: Illustration of SF-based representation.

compliance. Despite being an essential source for compliance requirements of data processing activities in software systems, automated compliance checking of DPAs against GDPR has not been previously studied in RE. Moreover, none of the existing work leverages the key elements of requirements in GDPR (e.g., the case of R15 and R16 discussed above) for a phrasal-level, automated compliance checking. Analyzing the content of DPAs at a phrasal level is beneficial for understanding the exact missing information content in the DPA and hence recommending remedies to prevent non-compliance with GDPR.

In our work, we propose using semantic frames (see Section 4.2) for generating an intermediate representation to characterize the information content in each GDPR requirement and thus enable a meaningful, phrasal-level decomposition of that requirement. As we elaborate in Section 4.2, a *semantic frame (SF)* of a given sentence describes an event and a set of participants, where each participant has a specific *semantic role (SR)* in that event. Fig. 4.2 illustrates SF elements describing a *processing* event in a requirement example extracted from GDPR Art. 28(3)(a) – corresponding to R9 in Fig. 4.1. The event *processing*, identified through the SR *action*, evokes three participants: “the processor” who performs processing (i.e., *actor*), ”personal data” on which processing is performed (i.e., *object*) and “only on documented instructions from the controller” that restricts processing (i.e., *constraint*).

4.1.2 Contributions

We take steps toward addressing the limitations outlined above. Concretely, our contributions are as follows:

(1) We extract in close interaction with subject-matter experts, a list of 45 compliance requirements from the GDPR provisions concerning data processing. We further simplify and document these compliance requirements as “shall” requirements. Our motivation is to increase readability for requirements engineers by using a familiar format, while maintaining usability for legal experts since these requirements rely on GDPR terminology.

(2) We devise an automated approach that leverages NLP technologies for automatically checking the compliance of DPAs against GDPR. More specifically, we rely on syntactic and semantic parsing to generate intermediate SF-based representations that are then compared for compliance checking. Our proposed SF-based representation is based on a set of semantic roles with intuitive descriptions. Future changes in compliance requirements is a common concern for legal experts and requirements engineers. With enough training, additional compliance requirements can be added to our automated approach by defining their respective SF-based representations. The SF-based representation is used as an enabler for automated compliance checking.

(3) We empirically evaluate our approach on 54 real DPAs. We randomly split the DPAs into two mutually exclusive subsets as follows. The fist subset containing 24 DPAs was used for creating the SF-based representations and developing our approach, whereas the second subset, composed of the remaining 30 DPAs, was used exclusively for evaluation. The evaluation set collectively contains a total of 7,048 statements manually reviewed and analyzed for compliance by third-party annotators. On this evaluation set, our approach yields a precision of 89.1%, and a recall of 82.4%. Our approach outperforms a straight-forward baseline that employs NLP off-the-shelf tools with a percentage point gain of ≈17 in both precision and recall.

(4) We propose computing and assigning scores to requirements found to be satisfied in a given DPA. Such score indicates the maximum matching degree between a requirement and any DPA statement satisfying it. The matching degree represents the ratio of key elements that are present in a DPA statement compared to the elements to be expected given the requirement. We discuss how matching degrees are computed in detail in Section 4.5. These scores are intended to facilitate the review and validation of findings from our automated approach by a human analyst. For example, our empirical evaluation shows that the analyst can improve the overall accuracy of our automated approach by 6.4 percentage points when reviewing statements satisfying requirements with the maximum matching degree scores as long as the scores are less than or equal to a threshold of 0.5. The proportion of these statements, on average, is $\approx$6% (corresponding to an average of 12 statements in a DPA). This accuracy improvement comes with a gain of 1.2 and 11.5 percentage points in precision and recall, respectively.

4.2 Background

Semantic frames (SFs), refer to a linguistic concept that represents the underlying structure of meaning in language. A semantic frame is a cognitive structure that captures the knowledge and expectations associated with a particular concept or scenario [28]. It provides a structured representation of the roles, attributes, and relationships that are typically associated with a specific situation or event. SFs are used in various NLP tasks, such as information extraction [124, 125], text understanding [126, 127], and machine translation [128, 129]. By using SFs, NLP systems can better understand the relationships and roles of different words in a given context. Therefore, their use leads to a better understanding of the language and interpretation of sentences beyond their surface-level representation.

In this chapter, we use SFs to describe the set of elements that participate in a particular event. Each element in the frame is a text span that is labeled with a semantic role (SR), denoted as **[span]$_{\boldsymbol{role}}$**. For instance, the SF about a *purchase event* in the sentence “[William]$_{buyer}$ purchased [a brand new car]$_{purchased\ item}$” contains the SRs *buyer* and *purchased item* spanning “William” and “a brand new car”, respectively. SFs are often represented as *predicate-arguments structure*, where a predicate describes an event represented by the text span of the verb, and each argument describes a participant in the event labeled with a specific SR. A predicate evokes a set of expected arguments based on the event. For example, the above sentence contains the arguments *buyer* and *purchased item*, but it could also contain *seller* and *price*. The NLP community has long been creating machine-readable lexical resources like FrameNet [130] and PropBank [131] that contain manually-labeled predicates and arguments.

One way to identify the SF of a given sentence is through semantic role labeling (SRL), which is the task of identifying the different SRs occurring in the sentence [132]. In SRL, one identifies the text spans that provide answers to questions about who (i.e., *actor*) did what (i.e., *action*) to whom (*beneficiary*), when (i.e., *time*) and why (i.e., *reason*) [133, 134]. SRL can be employed to improve the solutions of many downstream NLP tasks such as machine translation [135], question answering [136, 137] and relation extraction [138, 139]. For instance, question answering can be improved by aligning the SRs in the question with those in the likely answer. If we identify the SRs in the question “[What]$_{purchased\ item}$ did [Mazda]$_{seller}$ [sell]$_{action}$ to [William]$_{buyer}$?”, extracting the answer from the above example (i.e., “a brand new car”) becomes straightforward. Inspired by the NLP literature [137, 140, 141], we check the compliance of a given DPA against GDPR requirements by generating SF-based representations. In particular, we use the SR *action* to define the predicate of an SF, and a set of other SRs (e.g., *actor* and *reason*) to define the arguments as we elaborate in Section 4.5.

4.3 Research Questions

This chapter investigates four Research Questions (RQs):

RQ1: What are the requirements for checking the compliance of a DPA according to GDPR? We answer RQ1 by defining a set of 45 "shall" requirements concerning the compliance of a DPA. Checking these requirements in DPAs ensures that the DPAs contain a complete set of necessary legal requirements, to be implemented in software systems, concerning data processing activities. We further create a glossary table that defines the legal concepts related to DPA compliance and provide traceable links to GDPR articles. We present the compliance requirements in Section 4.4.

RQ2: How can a DPA be automatically checked for compliance against GDPR at a phrasal level? To address RQ2, we first manually define SF-based representations for the 45 compliance requirements from RQ1. Then, we devise an automated approach that leverages various NLP technologies to automatically generate SF-based representations for the textual content of a given DPA. Our approach then verifies the compliance of the DPA by aligning its respective SF-based representations against the representation of each GDPR requirement. We elaborate our approach in Section 4.5.

RQ3: How accurately can we check the compliance of a given DPA? RQ3 assesses the accuracy of our approach in checking the compliance of DPAs according to GDPR provisions. Over an evaluation set of 30 real DPAs, our approach successfully detects 618 out of 750 violations and introduces false violations in 76 cases. Our approach has thus a precision of 89.1% and a recall of 82.4%. Compared to a baseline that employs off-the-shelf NLP tools, our approach yields on average a gain of ≈20 percentage points in accuracy, and ≈17 percentage points in both precision and recall. We discuss RQ3 and the subsequent RQs in Section 4.6.

RQ4: How efficient is our approach in terms of execution time? Based on our experience, reviewing entirely manually the content of a given DPA containing about 200 statements takes on average 30 minutes if one is intimately familiar with the GDPR articles relevant to data processing activities. In contrast, our approach automatically analyzes a DPA of the same size in ≈2.5 minutes, and further produces as output a detailed report which summarizes the results, including the compliance decision together with the missing information and the respective recommendations to achieve compliance.

4.4 Compliance Requirements for Data Processing in GDPR (RQ1)

In this section, we present two artifacts to answer **RQ1**, namely (1) a total of 45 compliance requirements for DPA according to GDPR; and (2) a glossary table defining the legal concepts in these requirements, including traceability to GDPR articles. The artifacts are provided in Tables 4.2, 4.3, and 4.1.

In collaboration with legal experts from Linklaters, we extracted from GDPR a total of 45 compliance requirements that regulate DPAs. A GDPR-compliant DPA provides assurance about the exhaustiveness of the requirements concerning data processing activities, which are essential for developing compliant software systems. According to the experts' feedback, there are 26 mandatory and 19 optional requirements. The different requirements' types were discussed and agreed upon during several sessions.

Mandatory requirements (R1 – R26, listed in Table 4.2) are explicitly stipulated in GDPR, except R1 and R2 which were strongly recommended by legal experts. R1 and R2 provide key information concerning the identity and contact details of the data controller and processor based on which the compliance checking is performed. Optional requirements (R27 – R45) in Table 4.3 are derived from good practice in writing DPAs according to

Table 4.1: Glossary terms for DPA compliance requirements in GDPR.

Concept (*Art.* [1])	Definition
Personal Data (*Art. 4(1)*)	Any information relating to an identified or identifiable natural person (data subject).
Processing (*Art. 4(2)*)	Any operation or set of operations which is performed on personal data or on sets of personal data, whether or not by automated means, such as collection, recording, organization, structuring, storage, adaptation or alteration, retrieval, consultation, use, disclosure by transmission, dissemination or otherwise making available, alignment or combination, restriction, erasure or destruction.
Pseudonymization (*Art. 4(5)*)	The processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organizational measures to ensure that the personal data are not attributed to an identified or identifiable natural person.
Data Controller (*Art. 4(7)*)	A natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data; where the purposes and means of such processing are determined by Union or Member State law, the controller or the specific criteria for its nomination may be provided for by Union or Member State law.
Data Processor (*Art. 4(8)*)	A natural or legal person, public authority, agency or other body which processes personal data on behalf of the controller.
Personal Data Breach (*Art. 4(12)*)	A breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access to, personal data transmitted, stored or otherwise processed.
Supervisory Authority (*Art. 4(22)*)	A supervisory authority which is concerned by the processing of personal data because the controller or processor is established on the territory of the Member State of that supervisory authority; data subjects residing in the Member State of that supervisory authority are substantially affected or likely to be substantially affected by the processing; or a complaint has been lodged with that supervisory authority.
Objection (*Art. 4(24)*)	An objection to a draft decision as to whether there is an infringement of this Regulation, or whether envisaged action in relation to the controller or processor complies with this Regulation, which clearly demonstrates the significance of the risks posed by the draft decision as regards the fundamental rights and freedoms of data subjects and, where applicable, the free flow of personal data within the Union.
International Organization (*Art. 4(26)*)	An organization and its subordinate bodies governed by public international law, or any other body which is set up by, or on the basis of, an agreement between two or more countries.
Sub-processor (*Art. 28(4)*)	A natural person or organization engaged by a Processor for carrying out specific processing activities on behalf of the Controller.
Security of Processing (*Art. 32(1)*)	Taking into account the state of the art, the costs of implementation and the nature, scope, context and purposes of processing as well as the risk of varying likelihood and severity for the rights and freedoms of natural persons, the controller and the processor shall implement appropriate technical and organizational measures to ensure a level of security appropriate to the risk.
Data Protection Impact Assessment (*Art. 35(7)*)	An assessment of the necessity and proportionality of the processing operations in relation to the purposes, considering the risks to the rights, freedoms and legitimate interests of data subjects (abbreviated as DPIA).

Table 4.1: Glossary terms for DPA compliance requirements in GDPR (continued).

Codes of Conduct (*Art. 40(1)*)	Are intended to contribute to the proper application of this Regulation, taking account of the specific features of the various processing sectors and the specific needs of micro, small and medium-sized enterprises.
Certification (*Art. 42(1)*)	The establishment of data protection certification mechanisms and of data protection seals and marks, for the purpose of demonstrating compliance with this Regulation of processing operations by controllers and processors. The specific needs of micro, small and medium-sized enterprises shall be taken into account.
Personal Data Transfer (*Art. 45(1)*)	A transfer of personal data to a third country or an international organization may take place where the Commission has decided that the third country, a territory or one or more specified sectors within that third country, or the international organization in question ensures an adequate level of protection. If agreed by the parties, for the transfer of personal data processed under this agreement from a third country or international organization, they can rely on different mechanisms, including the prior consent of the controller.

[1] GDPR-related articles.

legal experts. Our automated support results in a non-compliance decision when a mandatory requirement is violated, and raises a warning when an optional requirement is violated.

We further group the compliance requirements into four categories, namely nine *metadata* requirements (denoted as MD), 25 requirements concerning the *processor's obligation* (denoted as PO), three requirements about the *controller's rights* (denoted as CR), and eight about the *controller's obligations* (denoted as CO). As shown in Table 4.2, the majority of mandatory requirements describe the processor's obligations (corresponding to 19/25).

To build our artifacts, we followed an iterative method that includes three steps. The first step was reading the articles of GDPR related to DPA. In the second step, we created the artifacts. Finally, we validated these artifacts with legal experts. Through a close interaction with these experts, we incrementally improved and refined the artifacts to get them in their final form. To mitigate fatigue, we conducted each validation session for a duration of approximately two hours adding up to an overall of 46 hours for all sessions. Three legal experts participated in these joint sessions. Their expertise spans several areas including corporate laws, commercial litigation, data protection laws and financial regulations, and information technology within the legal domain. During these sessions, we refined the compliance requirements according to the legal experts' interpretation, thereby alleviating any ambiguity in the GDPR. During these steps, we created the glossary terms (Table 4.1) with traceability links to GDPR articles to facilitate the understanding of these requirements.

Following experts' recommendations, we started by analyzing Art. 28 in GDPR as it is directly relevant to the compliance of DPAs. We then read through all referenced articles in Art. 28, namely Articles 32–36, 40, 42, 43, 63, 82–84, and 93(2). While reading the articles, we (1) transformed the text in GDPR articles into "shall" requirements and (2) divided compound statements into simple requirements. For example, we extracted from Art. 28(3), which states that *"Processing by a processor shall be governed by a contract [...] that sets out the subject-matter and duration of the processing, the nature and purpose of the processing, the type of personal data and categories of data subjects [...]"*, four requirements related to the metadata content of a DPA, marked as R3 – R6 in Table 4.2. We adopted the GDPR text as-is when it resembled a "shall" requirement such as R34 in Table 4.3, extracted from Art. 33(2). Using the resulting list of 45 compliance requirements, we manually checked the compliance of 24 DPAs. Specifically, we identified for each requirement whether it is satisfied by any statement in the DPA. As we elaborate in Section 4.6.2, our manual analysis aimed at establishing in-depth

Table 4.2: Mandatory DPA compliance requirements in GDPR.

ID (**Cat**[1])	Requirement (*Reference*[2])
R1 (**MD1**)	The DPA shall contain at least one controller's identity and contact details. (*Linklaters LLP*)
R2 (**MD2**)	The DPA shall contain at least one processor's identity and contact details. (*Linklaters LLP*)
R3 (**MD3**)	The DPA shall contain the duration of the processing. (*Art. 28(3)*)
R4 (**MD4**)	The DPA shall contain the nature and purpose of the processing. (*Art. 28(3)*)
R5 (**MD5**)	The DPA shall contain the types of personal data. (*Art. 28(3)*)
R6 (**MD6**)	The DPA shall contain the categories of data subjects. (*Art. 28(3)*)
R7 (**PO1**)	The processor shall not engage a sub-processor without a prior specific or general written authorization of the controller. (*Art. 28(2)*)
R8 (**PO2**)	In case of general written authorization, the processor shall inform the controller of any intended changes concerning the addition or replacement of sub-processors. (*Art. 28(2)*)
R9 (**PO3**)	The processor shall process personal data only on documented instructions from the controller. (*Art. 28(3)(a)*)
R10 (**PO4**)	If the processor requires by Union or Member State law to process personal data without instructions and law does not prohibit informing the controller on grounds of public interest, the processor shall inform the controller of that legal requirement before processing. (*Art. 28(3)(a)*)
R11 (**PO5**)	The processor shall ensure that persons authorized to process personal data have committed themselves to confidentiality or an appropriate statutory obligation of confidentiality. (*Art. 28(3)(b)*)
R12 (**PO6**)	The processor shall take all measures required pursuant to Article 32 or to ensure the security of processing. (*Art. 28(3)(c)*)
R13 (**PO7**)	The processor shall assist the controller in fulfilling its obligation to respond to requests for exercising the data subject's rights. (*Art. 28(3)(e)*)
R14 (**PO8**)	The processor shall assist the controller in ensuring the security of processing. (*Art. 28(3)(f), Art. 32*)
R15 (**PO9**)	The processor shall assist the controller in notifying a personal data breach to the supervisory authority. (*Art. 28(3)(f), Art. 33*)
R16 (**PO10**)	The processor shall assist the controller in communicating a personal data breach to the data subject. (*Art. 28(3)(f), Art. 34*)
R17 (**PO11**)	The processor shall assist the controller in ensuring compliance with the obligations pursuant to data protection impact assessment (DPIA). (*Art. 28(3)(f), Art. 35*)
R18 (**PO12**)	The processor shall assist the controller in consulting the supervisory authorities prior to processing where the processing would result in a high risk in the absence of measures taken by the controller to mitigate the risk. (*Art. 28(3)(f), Art. 36*)
R19 (**PO13**)	The processor shall return or delete all personal data to the controller after the end of the provision of services relating to processing. (*Art. 28(3)(g)*)
R20 (**PO14**)	The processor shall immediately inform the controller if an instruction infringes the GDPR or other data protection provisions. (*Art. 28(3)(h)*)
R21 (**PO15**)	The processor shall make available to the controller information necessary to demonstrate compliance with the obligations Article 28 in GDPR. (*Art. 28(3)(h)*)
R22 (**PO16**)	The processor shall allow for and contribute to audits, including inspections, conducted by the controller or another auditor mandated by the controller. (*Art. 28(3)(h)*)
R23 (**PO17**)	The processor shall impose the same obligations on the engaged sub-processors by way of contract or other legal act under Union or Member State law. (*Art. 28(4)*)
R24 (**PO18**)	The processor shall remain fully liable to the controller for the performance of sub-processor's obligations. (*Art. 28(4)*)

Table 4.2: Mandatory DPA compliance requirements in GDPR (continued).

R25 (**PO19**)	When assessing the level of security, the processor shall take into account the risk of accidental or unlawful destruction, loss, alternation, unauthorized disclosure of or access to the personal data transmitted, stored or processed. (*Art. 32(2)*)
R26 (**CR1**)	In case of general written authorization, the controller shall have the right to object to changes concerning the addition or replacement of sub-processors, after having been informed of such intended changes by the processor. (*Art. 8(2)*)

[1] Cat is the category of the requirement, namely metadata (MD), processor's obligation (PO), controller's right (CR), and controller's obligation (CO).

[2] GDPR-related articles.

Table 4.3: Optional DPA compliance requirements in GDPR.

ID (**Cat**[1])	Requirement (*Reference*[2])
R27 (**MD7**)	The organizational and technical measures to ensure a level of security can include: (a) pseudonymization and encryption of personal data, (b) ensure confidentiality, integrity, availability and resilience of processing systems and services, (c) restore the availability and access to personal data in a timely manner in the event of a physical or technical incident, and (d) regularly testing, assessing and evaluating the effectiveness of technical and organizational measures for ensuring the security of the processing. (*Art. 32(1)*)
R28 (**MD8**)	The notification of personal data breach shall at least include (a) the nature of personal data breach; (b) the name and contact details of the data protection officer; (c) the consequences of the breach; (d) the measures taken or proposed to mitigate its effects. (*Art. 33(3)*)
R29 (**MD9**)	The DPIA shall at least include (a) a systematic description of the envisaged processing operations and the purposes of the processing, (b) an assessment of the necessity and proportionality of the processing operations in relation to the purposes, (c) an assessment of the risks to the rights and freedoms of data subjects, and (d) the measures envisaged to address the risks. (*Art. 35(7)*)
R30 (**PO20**)	The processor shall not transfer personal data to a third country or international organization without a prior specific or general authorization of the controller. (*Art. 28(3)(a)*)
R31 (**PO21**)	The processor can demonstrate guarantees to Article 28 (1–4) through adherence to an approved codes of conduct or an approved certification mechanism. (*Art. 28(5)*)
R32 (**PO22**)	The processor shall implement appropriate technical and organizational measures to ensure a level of security appropriate to the risk of varying likelihood and severity for the rights and freedoms of natural persons. (*Art. 32(1)*)
R33 (**PO23**)	The processor shall ensure that any natural person acting under its authority who has access to personal data only process them on instructions from the controller. (*Art. 32(4)*)
R34 (**PO24**)	The processor shall notify the controller without undue delay after becoming aware of a personal data breach. (*Art. 33(2)*)
R35 (**PO25**)	A processor shall be liable for the damage caused by processing only where it has not complied with GDPR obligations specifically directed to processors or where it has acted outside or contrary to lawful instructions of the controller. (*Art. 82(2)*)
R36 (**CO1**)	The controller shall inform the supervisory authority no later than 72 hours after having become aware of a personal data breach. (*Art. 33(1)*)
R37 (**CO2**)	The controller shall document personal data breaches. (*Art. 33(5)*)
R38 (**CO3**)	In case of high risks, the controller shall communicate the data breach to the data subject without undue delay. (*Art. 34(1)*)
R39 (**CO4**)	The controller shall carry out a DPIA. (*Art. 35(1)*)
R40 (**CO5**)	The controller shall seek advice of the DPO when carrying a DPIA. (*Art. 35(2)*)

Table 4.3: Optional DPA compliance requirements in GDPR (continued).

R41 (**CO6**)	The controller shall seek the views of data subjects or their representatives on the intended processing. (*Art. 35(9)*)
R42 (**CO7**)	The controller shall carry out a review to assess if processing is performed in accordance with the data protection impact assessment at least when there is a change of the risk represented by processing operations. (*Art. 35(11)*)
R43 (**CO8**)	A controller shall be liable for the damage caused by any processing infringing the GDPR. (*Art. 82(2)*)
R44 (**CR2**)	The controller shall have the right to suspend the processing in certain cases. (*Linklaters LLP*)
R45 (**CR3**)	The controller shall have the right to terminate the DPA in certain cases. (*Linklaters LLP*)

[1] Cat is the category of the requirement, namely metadata (MD), processor's obligation (PO), controller's right (CR), and controller's obligation (CO).

[2] GDPR-related articles.

understanding of the compliance requirements before outsourcing the annotation to external annotators.

4.5 Approach (RQ2)

To address **RQ2**, we devise an NLP-based approach for checking the compliance of DPAs against GDPR at a phrasal level (DERECHA). The current version of DERECHA does not use machine learning or deep learning architectures as, in our application context, we typically do not have enough labeled data to train robust models with accurate predictions. Recent advances in the NLP literature show that fine-tuning large-scale language models like BERT [76] often yields accurate models for many downstream tasks [142–144]. To this end, we note that such language models are integrated into currently available off-the-shelf semantic role labeling (SRL) tools which DERECHA does not directly use as we explain later in this section. Instead, we employ these off-the-shelf NLP tools as a baseline solution and compare its performance in Section 4.6.3 against the performance of DERECHA. That said, we believe that extending our dataset and experimenting with machine learning (implemented in Chapter 5) or deep learning models (left for future work) are important to refine our conclusions.

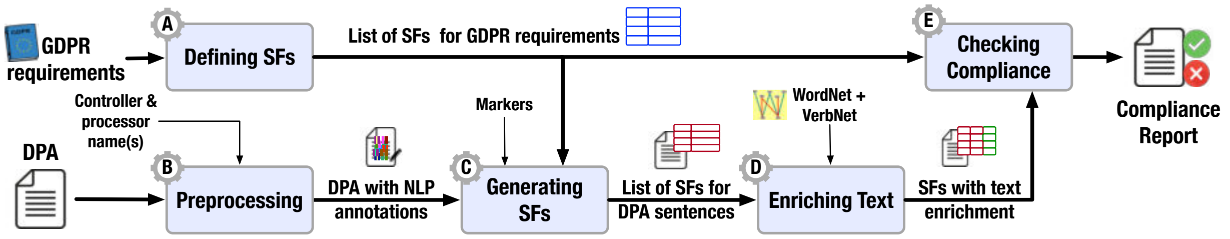


Figure 4.3: Overview of our DPA compliance checking approach (DERECHA).

Fig. 4.3 presents an overview of DERECHA, composed of five steps. The input to DERECHA includes a DPA, the name(s) of the controller and processor, and a set of requirements. The output is then a compliance report which summarizes the findings of DERECHA. In step A, we manually create SF-based representations of the compliance requirements extracted in Section 4.4. Note that this step is performed only once and can be reused across DPAs. In practice, from a user perspective, these SF-based representations can be adopted as-is. In step B, we apply the NLP pipeline presented in Figure 2.1 (Section 2.2) to parse the DPA and preprocess the text. In step C, we automatically generate SF-based representations for the input DPA. In step D, we enrich the textual content in the DPA with semantically-related text, e.g., the text "engage" will be enriched with "hire" and “employ”. Finally, in step E, we use the SF-based representations created in steps A and D to check the compliance of the input DPA against GDPR. Finally, we generate a detailed report about the compliance decision. We elaborate all these steps next.

4.5.1 Step A: Defining SFs in Compliance Requirements

Here we define an SF-based representation to characterize the information content of GDPR requirements. This representation aims at decomposing the requirements and thus enabling compliance checking at a phrasal level. As can be seen in Table 4.2 in Section 4.4, the metadata requirements describe the content that a DPA has to cover, e.g., the duration of the agreement. In contrast, the requirements under the other categories (e.g., processor’s obligations) are expected to be precisely stated in the DPA. While the exact terminology is not required by GDPR, the semantics of these requirements are. This has an implication on the compliance checking process. For example, a statement in a given DPA satisfying R5 (**MD5**) will unlikely mention “the DPA” or “shall contain”. On the contrary, a statement satisfying R9 (**PO3**) has to explicitly cover the different elements of its SF-based representation illustrated in Fig. 4.2 in Section 4.4. Consequently, we define such representations only for the requirements concerning processor’s obligations, controller’s rights and controller’s obligations. We elaborate on how we handle the compliance checking against the metadata requirements (R1 – R6 and R27 – R29) in Section 4.5.5. The SF-based representations for all 45 requirements are provided in Tables 4.4 and 4.5.

To define the SF-based representation, we first identify the SRs in each requirement. Inspired by existing literature [21, 132, 145], we collate a list of 10 SRs pertinent to DPA compliance requirements in GDPR. Four SRs are illustrated in Fig. 4.2, including *action* to describe a particular event, and *actor*, *object*, and *constraint* to describe the SRs of the participants in that event. The remaining six SRs, exemplified on R18 – R20 in Table 4.2, are: *Beneficiary* – an entity that benefits from an action (e.g., [the controller]$_{beneficiary}$ in R18); *reason* – justifying why the action is performed (e.g., [to mitigate the risk]$_{reason}$ in R18); *time* – a temporal aspect (e.g., [prior to processing]$_{time}$ in R18); *condition* – the cases where the action is performed (e.g., [If an instruction infringes GDPR or other data protection provisions]$_{condition}$ in R20); *situation* – something that happened or can happen (e.g., [the end of provision of services related to processing]$_{situation}$ in R19); and finally *reference* – the mention of legal entities (e.g., [GDPR or other data protection provision]$_{reference}$ in R20).

Recall from Section 4.2 that an SF is represented as a predicate-arguments structure. For each requirement, we therefore define a predicate and a set of arguments evoked by that predicate. The predicate highlights the main action a data processor needs to perform to be compliant with GDPR. Example predicates include ⟨not engage⟩ in R7, ⟨process⟩ in R9, and ⟨return or delete⟩ in R19. The arguments are the different phrases in the requirement which constitute the participants associated with the predicate. We manually identify the arguments by applying the list of SRs explained above on the requirement text. Different requirements can share the same predicate when they refer to the same action, e.g., six requirements share the predicate ⟨assist⟩, namely R13 – R18. Nonetheless, we define one representation for each requirement, since the required arguments might be different. Doing so is essential for capturing missing information content in DPA statements according to what is expected by each requirement. Finally, we have discussed the SF-based representations with our collaborating experts from Linklaters and received feedback that the underlying SRs are intuitive and easy to understand. This suggests that extending our approach with more compliance requirements alongside their respective SF-based representations is practically feasible.

The first step results in: (i) a list of SRs collated and refined from the existing literature which is passed on to step C to be further used in automatically generating SF-based representations for the input DPA; and

Table 4.4: SF-based representations of mandatory GDPR requirements (R7 – R26 in Table 4.2)

ID§	Representation $\langle \mathbf{p} \rangle$, $\mathcal{A}$
PO1	$\langle$**not engage**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [a sub-processor]$_{object}$, a_2 = [without prior specific or general written approval of the controller]$_{constraint}$, a_3 = [prior specific or general written approval]$_{time}$
PO2	$\langle$**inform**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 4\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the controller]$_{beneficiary}$, a_2 = [any intended changes]$_{object}$, a_3 = [the addition or replacement of sub-processors]$_{situation}$, a_4 = [in case of written authorization]$_{condition}$
PO3	$\langle$**process**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 2\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [personal data]$_{object}$, a_2 = [on documented instructions from the controller]$_{constraint}$
PO4	$\langle$**inform**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 5\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [that legal requirement]$_{object}$, a_2 = [the controller]$_{beneficiary}$, a_3 = [If the processor requires by Union or Member State law to process personal data without instructions and law does not prohibit informing the controller on grounds of public interest]$_{condition}$, a_4 = [before processing]$_{time}$, , a_5 = [Union or Member State law]$_{reference}$
PO5	$\langle$**ensure**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [persons]$_{object}$, a_2 = [have committed themselves to confidentiality]$_{situation}$, a_3 = [are under an appropriate statutory obligation of confidentiality]$_{constraint}$
PO6	$\langle$**take**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [all measures]$_{object}$, a_2 = [Article 32]$_{reference}$, a_3 = [to ensure the security of processing]$_{reason}$
PO7	$\langle$**assist**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 4\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the controller]$_{beneficiary}$, a_2 = [in fulfilling its obligation]$_{object}$, a_3 = [to respond to requests for exercising the data subject's rights]$_{reason}$, a_4 = [requests]$_{situation}$
PO8	$\langle$**assist**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 2\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the controller]$_{beneficiary}$, a_2 = [in ensuring the security of processing]$_{object}$
PO9	$\langle$**assist**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 5\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the controller]$_{beneficiary}$, a_2 = [in consulting the supervisory authorities]$_{object}$, a_3 = [prior to processing]$_{time}$, a_4 = [to mitigate the risk]$_{reason}$, a_5 = [where the processing would result in a high risk in the absence of measures taken by the controller]$_{constraint}$
PO10	$\langle$**assist**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the controller]$_{beneficiary}$, a_2 = [in notifying to the supervisory authority]$_{object}$, a_3 = [a personal data breach]$_{situation}$
PO11	$\langle$**assist**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the controller]$_{beneficiary}$, a_2 = [in communicating to the data subject]$_{object}$, a_3 = [a personal data breach]$_{situation}$
PO12	$\langle$**assist**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 2\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the controller]$_{beneficiary}$, a_2 = [in ensuring compliance with the obligations pursuant to data protection impact assessment (DPIA)]$_{object}$
PO13	$\langle$**return or delete**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 4\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [all personal data]$_{object}$, a_2 = [the controller]$_{beneficiary}$, a_3 = [after $\mid$ end]$_{time}$, a_4 = [the end of the provision of services related to processing]$_{situation}$
PO14	$\langle$**inform**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the controller]$_{beneficiary}$, a_2 = [if an instruction infringes the GDPR or other data protection provisions]$_{condition}$, a_3 = [GDPR or other data protection provisions]$_{reference}$
PO15	$\langle$**make available**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 2\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the controller]$_{beneficiary}$, a_2 = [information necessary to demonstrate compliance with the obligations]$_{object}$, a_2 = [Article 28 in GDPR]$_{reference}$
PO16	$\langle$**allow for and contribute to**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 2\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [audits including inspections]$_{object}$, a_2 = [conducted by the controller or another auditor mandated by the controller]$_{situation}$
PO17	$\langle$**impose**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 4\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the engaged sub-processors]$_{beneficiary}$, a_2 = [the same obligations]$_{object}$, a_3 = [by way of contract or other legal act under]$_{constraint}$, a_4 = [Union or Member State law]$_{reference}$

Table 4.4: SF-based representations of mandatory GDPR requirements (R7 – R26 in Table 4.2) (continued).

PO18	$\langle$**remain fully liable**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 2\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the controller]$_{beneficiary}$, a_2 = [for the performance of sub-processor's obligations]$_{object}$
PO19	$\langle$**take into account**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 2\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the risk of]$_{object}$, a_2 = [accidental or unlawful destruction, loss, alternation, unauthorized disclosure of or access to the personal data transmitted, stored or processed]$_{situation}$
CR1	$\langle$**have the right**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 5\}$ such that a_0 = [the controller]$_{actor}$, a_1 = [to object to changes]$_{object}$, a_2 = [in case of written authorization]$_{condition}$, a_3 = [by the processor]$_{constraint}$, a_4 = [after having been informed of such intended changes]$_{time}$, a_5 = [addition or replacement of sub-processors]$_{situation}$

(ii) SF-based representations for the requirements (R7 – R25 and R30 – R45) passed on to step E. These representations enable automated compliance checking of the statements at a phrasal level. An example of such representation for **PO6** (R12 in Table 4.2) contains the predicate $\langle$**take**$\rangle$, and a set of four arguments $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [all measures]$_{object}$, a_2 = [Article 32]$_{reference}$, a_3 = [to ensure the security of processing]$_{reason}$.

4.5.2 Step B: Preprocessing

Given a DPA as input, this step applies an NLP pipeline composed of seven modules. Recall from Figure 2.1 (Section 2.2): (1) *tokenization*, splits the text into tokens; (2) *sentence splitting*, identifies the different sentences in the text using different heuristics, e.g., a sentence ends with a period; (3) *lemmatization*, maps the words to their canonical forms, e.g., "instructions" and "requested" are mapped to "instruction" and "request"; (4) *POS tagging* assigns a part-of-speech (POS) tag to each token, e.g., noun, verb, and possessive pronoun; (5) *chunking*, groups the words that form a syntactic unit, e.g., "the processor" is a noun phrase; (6) *dependency parsing*, describes the grammatical relations in a sentence, e.g., "the processor" and "personal data" in the sentence "the processor transfers personal data" are the subject and the object of the verb "transfers", respectively; and finally (7) *semantic parsing*, defines the meanings of the different constituents in a sentence. Each module in this pipeline generates NLP annotations on parts of the text in the input DPA. The resulting NLP annotations are passed on to the next steps.

Notice that for semantic parsing, we apply two widely-known lexical resources, namely WordNet [146, 147] and VerbNet [148, 149]. WordNet is a lexical database for English. It groups words into sets of synonyms called *synsets* and further connects the synsets via semantic relations. For example, *is-a* connects a hyponym (more specific synset) to a hypernym (more general synset) [150]. VerbNet groups verbs into classes according to their structure, e.g., the verbs under the *motion* class include run, escape, and leave [151].

4.5.3 Step C: Generating SFs Automatically

In step C, we automatically generate for each statement in the input DPA an SF-based representation. To do so, we first identify the SRs in the DPA statement. Subsequently, we use the SR *action* to generate the predicate and the remaining SRs to generate the arguments. Recall from Section 4.1 that a statement in a DPA corresponds to a sentence.

In this step, we design our own method for identifying SRs instead of directly using the results of available NLP tools for SRL. First, the NLP tools have been trained on a large body of generic text. This can result in missing some of the SRs in our list which are pertinent to compliance checking against GDPR, e.g., *reference* and *situation*. Second, a typical NLP semantic role labeler would generate a predicate for each verb in a given

Table 4.5: SF-based representations of optional GDPR requirements (R27 – R45 in Table 4.3)

ID§	Representation $\langle \mathbf{p} \rangle$, $\mathcal{A}$
PO20	$\langle$**not transfer**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 4\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [a third country or international organization]$_{beneficiary}$, a_2 = [personal data]$_{object}$, a_3 = [without a prior specific or general authorization of the controller]$_{constraint}$, a_4 = [prior specific or general authorization]$_{time}$
PO21	$\langle$**demonstrate**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [guarantees]$_{object}$, a_2 = [adherence to an approved code of conduct or an approved certification mechanism]$_{situation}$, a_3 = [Article 28 (1-4)]$_{reference}$
PO22	$\langle$**implement**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [appropriate technical and organizational measures]$_{object}$, a_2 = [to ensure a level of security appropriate to the risk]$_{reason}$, a_3 = [varying likelihood and severity for the rights and freedoms of natural persons]$_{situation}$
PO23	$\langle$**ensure**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [any person]$_{beneficiary}$, a_2 = [under its authority who has access to personal data acts only on instructions from the controller]$_{constraint}$, a_3 = [access to personal data]$_{situation}$
PO24	$\langle$**notify**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the processor]$_{actor}$, a_1 = [the controller]$_{beneficiary}$, a_2 = [without undue delay]$_{constraint}$, a_3 = [a data breach]$_{situation}$
PO25	$\langle$**be liable**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [a processor]$_{actor}$, a_1 = [for the damage caused]$_{object}$, a_2 = [where acting outside or contrary to lawful instructions of the controller or not complying with obligations of the GDPR specifically directed to processors]$_{condition}$, a_3 = [by processing]$_{constraint}$
CO1	$\langle$**inform**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the controller]$_{actor}$, a_1 = [the supervisory authority]$_{beneficiary}$, a_2 = [no later than 72 hours after having become aware]$_{time}$, a_3 = [a personal data breach]$_{situation}$
CO2	$\langle$**document**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 1\}$ such that a_0 = [the controller]$_{actor}$, a_1 = [personal data breaches]$_{object}$
CO3	$\langle$**inform**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 4\}$ such that a_0 = [the controller]$_{actor}$, a_1 = [the data subject]$_{beneficiary}$, a_2 = [in case of high risks]$_{condition}$, a_3 = [without undue delay]$_{constraint}$, a_4 = [a data breach]$_{situation}$
CO4	$\langle$**carry out**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 1\}$ such that a_0 = [the controller]$_{actor}$, a_1 = [DPIA]$_{object}$
CO5	$\langle$**seek**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 2\}$ such that a_0 = [the controller]$_{actor}$, a_1 = [advice of the DPO]$_{object}$, a_2 = [when carrying out DPIA]$_{condition}$
CO6	$\langle$**seek**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 1\}$ such that a_0 = [the controller]$_{actor}$, a_1 = [the views of data subjects or their representatives on the intended processing]$_{object}$
CO7	$\langle$**carry out**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 5\}$ such that a_0 = [the controller]$_{actor}$, a_1 = [a review]$_{object}$, a_2 = [at least when there exists]$_{condition}$, a_3 = [represented by processing operations]$_{constraint}$, a_4 = [to assess if processing is performed in accordance with the DPIA]$_{reason}$, a_5 = [a change of the risk]$_{situation}$
CO8	$\langle$**be liable**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [a controller]$_{actor}$, a_1 = [for the damage]$_{object}$, a_2 = [caused by any processing infringing the GDPR]$_{constraint}$, a_3 = [GDPR]$_{reference}$
CR2	$\langle$**have the right**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 2\}$ such that a_0 = [the controller]$_{actor}$, a_1 = [to suspend the processing]$_{object}$, a_2 = [in certain cases]$_{condition}$
CR3	$\langle$**have the right**$\rangle$, $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that a_0 = [the controller]$_{actor}$, a_1 = [to terminate]$_{object}$, a_2 = [in certain cases]$_{condition}$, a_3 = [the DPA]$_{reference}$

statement. In contrast, our representation is based on generating one predicate for each statement in DPA, i.e., extracting one SR *action* to which other SRs in the statement should be related. Generating one predicate is essential for the purpose of our analysis since the SF-based representations of GDPR requirements are centered around one predicate. Our goal in this step is to decompose each statement in the DPA into meaningful phrases that are labeled with similar SRs as in the requirements. In addition, comparing multiple predicates for each statement in the DPA against one predicate of each requirement is far too time consuming as reported in Section 4.6.4. Further, this would lead to multiple answers, most of them being irrelevant, as explained in Section 4.6.3. To empirically assess the above limitations with respect to the performance of our approach, we define a baseline that works the same way as DERECHA with the exception that it employs off-the-shelf NLP tools for generating the SF-based representations in this step. Further details are provided about this baseline in Section 4.6.3. Generating an SF-based representation in DERECHA involves two sub-steps, as explained below. ***C1) SRs identification.*** To identify SRs in a given statement, we utilize the NLP annotations from step A (Section 4.5.2). In particular, we apply a set of rules (listed in Table 4.6) over the syntactic information of the statement. We rely mostly on the grammatical relations produced by the dependency parser. For instance, the *action* which will be regarded as the predicate is the main verb in the statement. Consequently, the *actor* should be associated with the action as its subject. We note that any statement without a root verb (e.g., the title of a section) will not have a predicate (i.e., action) and is thus not represented by our method.

Table 4.6: Extraction rules of semantic roles.

SR	Description **(D)** and Example **(E)** from Fig. 4.1
Action	**(D)** The *action* is the VP that contains the root verb retrieved from the dependency parsing tree. The action is the main verb in a statement. We note that a parsing tree contains only one root node, and this is inline with our goal of identifying one predicate for each statement. **(E)** The *action* in S6 is “processes”.
Actor	**(D)** The *actor* is the NP containing the subject of the root verb in the statement (i.e., the *action*). **(E)** The *actor* in S6 is “Levico Accounting GmbH”
Object	**(D)** The *object* is either the NP that contains the object of the root verb *AND* the *action* does not contain any *beneficiary* marker; *OR* the VP that starts with a preposition and is associated with the root verb. Our rule accounts for cases where the root verb is directly followed by a PP (e.g., “assist with”). To differentiate actions that evoke the SR *object* (e.g., “process”) from other actions that evoke *beneficiary* (e.g., “inform”), we define markers. **(E)** The *object* in S6 is “Company personal data”
Beneficiary	**(D)** Similar to *object*, the *beneficiary* is either the NP that contains a preposition linked to the root verb *AND* the *action* contains a *beneficiary* marker, *OR* the NP that contains a subject of a verb that is different from the root verb. Since the *beneficiary* is an entity that benefits from the *action*, it can be associated directly with the action given that the *action* has a marker that pinpoints the necessity of a *beneficiary*. Alternatively, the *beneficiary* can be an entity associated with another action in the statement. **(E)** The *beneficiary* in S11 is “Company”
Condition	**(D)** The *condition* is the PP, *OR* ADVP, *OR* any phrase annotated as subordinate clause by the dependency parser given that it contains a *condition* marker. **(E)** The *condition* in S12 is “if Company terminates the agreement”.
Constraint	**(D)** The *constraint* is identified in similar manner as *condition* except that the phrase should contain a *constraint* marker. The *constraint* in S11 is “without undue delay”
Time	**(D)** The *time* is the NP, *OR* PP, *OR* ADVP, *OR* any phrase annotated as conjunctive preposition by the dependency parser given that it contains a *time* marker. **(E)** The *time* in S12 is “remains effective so long as Levico provides sevices”
Reason	**(D)** *The reason* is either the VP containing an auxiliary and is linked to a verb other than the root, *OR* the VP containing an open clausal complement and is linked to the root verb. **(E)** The *reason* in S6 is “for providing the service of payroll administration”.
Situation	**(D)** The *situation* is the NP *OR VP* that contains a *situation* marker. **(E)** The *situation* in S13 is “termination of any services”.
Reference	**(D)** The *reference* is the NP that contains a *reference* marker. **(E)** The *reference* in S10 is “Article 32(1)” and “GDPR”.

VP: verb phrase, NP: noun phrase, PP: prepositional phrase, ADVP: adverbial phrase.
The extraction rules are adapted from the RE literature [21, 145].

Our rules incorporate a set of markers that help identify the SRs. Examples of markers are shown in Table 4.7. Except for *situation* and *reference*, the markers we use originate from the RE literature [21, 145]. We improved

Table 4.7: Example markers for identifying semantic roles.

SR	Markers
Beneficiary	inform, report, assist, help, aid, support, remain, notify, provide, give, supply
Condition	if, once, in case, where, when
Constraint	without, on, in accordance with, according to, along, by, under, unless
Time	after, later, prior, before, earlier, as long as, as soon as
Situation	access, addition, destruction, loss, disclosure, adherence, modification, termination, expiration
Reference	gdpr, dpa, law, jurisprudence, legislation, agreement, article, contract

these markers in two ways. First, we included the nouns and/or verbs that occur in the respective SRs according to our SF-based representations defined for GDPR requirements in step A. The intuition is to cover what is expected by the GDPR requirements for compliance. Second, we enrich the resulting markers (including the ones reported in the literature and the ones we adapted from GDPR) by including their synonyms from WordNet and VerbNet. This way, our markers will likely cover the different wordings applied in the DPA text.

C2) Text spans demarcation. Once an SR is identified, we demarcate the text span to which this SR should be assigned. Specifically, we use the NLP annotations produced by the text chunking to find the entire phrase where the SR is located. For instance, the SR *action* is assigned to the verb "employ" in the statement displayed in Table 4.8. We then find the verb phrase in the chunking results which contains "employ", i.e., "can employ". This way, we ensure capturing the semantics of the statement in a proper way. Except for *action* which has to be identified in a statement only once, other SRs can be identified in the statement multiple times. In this case, we group the n text spans under that SR to get the argument [span$_1$ | . . . | span$_n$]$_{role}$. For example, the argument [after | end]$_{time}$ contains two spans related to *time*.

As a result of the sub-steps ***C1*** and ***C2*** explained above, each statement in the input DPA is sliced into a set of phrases, each one with an SR label. These phrases constitute our SF-based representation for the statement which is further processed in the next step.

4.5.4 Step D: Enriching Text of Input DPA

In this step, we enrich the DPA text by extending the text spans demarcated in the previous step with semantically-related words. The rationale is to increase the likelihood of finding an overlap when comparing the DPA text against GDPR requirements. We retrieve semantically-related words from the annotations of the semantic parsing module in Section 4.5.2. We extract from WordNet synonyms for each word in a text span labeled with any SR in the statement. For example, an *object* spanning "intended changes" will be enriched with the related words "planned, alteration, modification". To enrich the text span of *action*, in addition to WordNet, we use the verbs provided in VerbNet under the same class, e.g., the *action* spanning "can employ" will be enriched from VerbNet with the verbs "engage" and "hire", since they are grouped under the same class. To select the synonyms of the words in this step, we opted for the most frequent sense (MFS) of the word. While disambiguating the words can be beneficial, it is outside the scope of this research work. We note that MFS has been widely used as a baseline in the NLP literature and has often shown sufficient performance [152].

In this step, we further replace the actual names of the controller and processor with the generic place holders "the controller" and "the processor", e.g., *Levico Accounting GmbH* in the example in Fig. 4.1 is replaced with "the processor". As explained in Section 4.1, we assume in our work that the name(s) of the controller and processor are given as input by the user. Replacing those named entities helps normalize the text and improve the overall compliance checking.

4.5.5 Step E: Compliance Checking

In step E, we check the compliance of the input DPA against GDPR.

Metadata Requirements. To check whether the DPA satisfies the metadata requirements, we do the following. R1 and R2 are concerned with the identity and contact details of the controller and the processor. Given the names as input by the user, we look for statements in the DPA that contain these names. To find the contact details, we create regular expressions that identify entities including phone numbers, email and postal addresses. Then, we check whether these entities are present in the same statements containing the names of controller the and the processor.

For checking the remaining requirements (R3 – R6 and R27 – R29), we apply a method inspired by Lesk algorithm [153], one of the traditional methods applied in NLP in the context of word sense disambiguation. The original Lesk algorithm compares multiple senses (meanings) of two words for the purpose of identifying their respective meanings. For example, to disambiguate the word "cone" in the phrase "the pine cones", Lesk algorithm compares the definition of each sense of the word "pine" with the definition of each sense of the word "cone". The algorithm then computes the number of overlapping words between the definitions of each two senses. The senses with the highest overlap are the ones selected to disambiguate the two words. We adapt this algorithm to our compliance checking process. Specifically, for each metadata requirement (among R3 – R6 and R27 – R29), we look for a statement in the DPA that has overlapping words with the requirement. During our manual analysis of the 24 DPAs (discussed in Section 4.4), we observed that DPA statements often apply the same words when expressing the requirements R3 – R6 and R27 – R29. Thus, we believe that using Lesk algorithm in this case is sufficient.

Table 4.8: Example of compliance checking using SF-based representations against R7.

GDPR	**R7.** The processor shall not engage a sub-processor without a prior written authorization of the controller.						
	actor	⟨*p*⟩	*object*	NOT REQUIRED	*constraint*		
						time	
	the processor	not engage	a sub-processor		without	a prior	written authorization of the controller.
DPA*	Levico Accounting GmbH can employ sub-contractors to perform the service of Levico's obligations.						
	actor	⟨*p*⟩	*object*	*reason*	*constraint*		
						time	
	Levico Accounting GmbH **processor**	can employ **hire** **engage**	sub-contractors **sub-processor**	to perform Levico's obligations. **processor duty**	MISSING		

* Enriched text from step D of our approach is highlighted in bold.

Remaining Requirements. To check whether a requirement is satisfied in the DPA, we compare the SF-based representation of the requirement against that of each statement in the DPA. Table 4.8 shows a simplified example of checking compliance between a DPA statement and requirement R7. Our compliance checking method using SFs comprises four steps, as elaborated in Algorithm 2. First, we check the predicates in the two representations. Only if the predicates are sufficiently similar, we subsequently match the arguments, and further compute a score

Algorithm 2 SF-based compliance checking of DPA statements against a requirement r

```
Require: $\mathcal{S}$: Statements in the input DPA.
 1: score ← 0                                        // Set matching degree score of r to zero
 2: degree$_i$ ← 0
 3: for $s_i \in \mathcal{S}$ do
 4:   Let $\langle p_r \rangle$, $\langle p_i \rangle$ be the predicates of the $r$ and $s_i$, respectively.
 5:   Let $\mathcal{A}_r$, $\mathcal{A}_i$ be the respective set of arguments of $r$ and $s_i$, respectively.
 6:   if $\langle p_r \rangle \cap \langle p_i \rangle \neq \phi$ or sim($\langle p_r \rangle, \langle p_i \rangle$) $> \theta_p$ then  // Check compliance only if the two predicates match
 7:     for each argument $a_{rj}$ in $\mathcal{A}_r$ do
 8:       found ← 0                                  // Look for matching arguments
 9:       let $\ell_{rj}$ be the SR of $a_{rj}$, and $t_{rj}$ be the text span of $a_{rj}$.
10:       for each argument $a_{ik}$ in $\mathcal{A}_i$ do
11:         let $\ell_{ik}$ be the SR of $a_{ik}$, and $t_{ik}$ be the text span of $a_{ik}$.
12:         enrich($t_{ik}$)                         // Enrich DPA text with synonyms
13:         if ($\ell_i$ equals $\ell_r$) & (($t_i \cap t_r \neq \phi$) or sim($t_i, t_r$) $> \theta_a$)) then
14:           found ← found + 1
15:         end if
16:       end for
17:       degree$_i$ ← (found$_i$+1) / (len($\mathcal{A}_r$)+1)   // Compute matching degree of $s_i$
18:     end for
19:   end if
20:   if score < degree$_i$ then
21:     score ← degree$_i$                           // Compute the maximum matching degree score for $r$
22:   end if
23: end for
24: if score $\neq$ 0 then                           // Compliance decision
25:   $r$ is satisfied
26: else
27:   $r$ is violated
28: end if
```

indicating the matching degree. Finally, we conclude the compliance decision for the DPA. We explain each step next.

E.1) Matching predicates. For matching two predicates, we specifically consider the verbs occurring in these predicates. We then deem the predicates to be *similar* if at least one of the two conditions below is met (line 6 in Algorithm 2).

Condition 1 returns *true* if the two predicates have some overlapping verbs.

Condition 2 returns *true* if the semantic similarity between the verb in the requirement and at least one verb in the statement is greater than or equal to a threshold θ_p. In our work, we apply the similarity metric (*WuP*) proposed by Wu and Palmer [154] and widely-applied in NLP applications [155–157]. In brief, WuP is a WordNet-based metric that computes the similarity between two words (or concepts) by finding the path length from the first hypernym in the hierarchy that is shared between the synsets of these concepts [158]. WuP returns a score between 0.0 (dissimilar) and 1.0 (identical).

To optimize the efficiency of DERECHA, only statements having similar predicates to the requirements will be further checked for compliance. A statement that meets none of the conditions above is likely discussing an action that is irrelevant to the one required by GDPR.

E.2) Matching arguments. This is done based on the arguments in the compliance requirement, since they represent required information by GDPR (lines 7 – 18 in Algorithm 2). In particular, we align each argument in the requirement to an *equivalent argument* sharing the same SR in the statement (e.g., *actor* in Table 4.8).

Any argument whose SR exists in the statement but not in GDPR is considered as *not required*. Conversely, an argument is *missing* if its SR is present in GDPR but not in the statement. For example, the argument with the SR *reason* in Table 4.8 (Section 4.5.3) is not required by **R7**, whereas the one related to *constraint* (including *time*) is missing from the statement. For each pair of equivalent arguments, we check for overlap between the two text spans in a similar way to condition 1 and condition 2 above. For condition 2, we compute the Jaro-Winkler distance instead of WuP that we applied for matching predicates. The reason is that Jaro-Winkler distance is often applied for comparing documents' similarities [159–161]. Thus, Jaro-Winkler distance is more suitable in the case of matching arguments since longer text sequences are expected unlike the case of predicates. Finally, equivalent arguments are deemed to be *matching* if there is any overlap between the text spans or when the similarity between the text spans is greater than a threshold θ_a. The arguments are deemed *not matching* otherwise.

Following the findings, we empirically set θ_p to 0.9 and θ_a to 0.7. We note that for θ_p, unlike θ_a, we experimented with high values (> 0.7). The rationale is that arguments are only checked when the predicates are deemed similar, i.e., the similarity value between the two predicates passes the threshold θ_p. Considering lower threshold values for θ_p would entail significant execution time for performing more comparisons involving arguments matching. To avoid excessive computations and further ensure a high level of similarity, we set the first threshold (θ_p) to be relatively high. Once the similarity between two predicates exceeds the threshold (i.e., $>$0.9), we match the arguments. The second threshold (θ_a) is set a bit lower since we already have evidence about the similarity of the predicates.

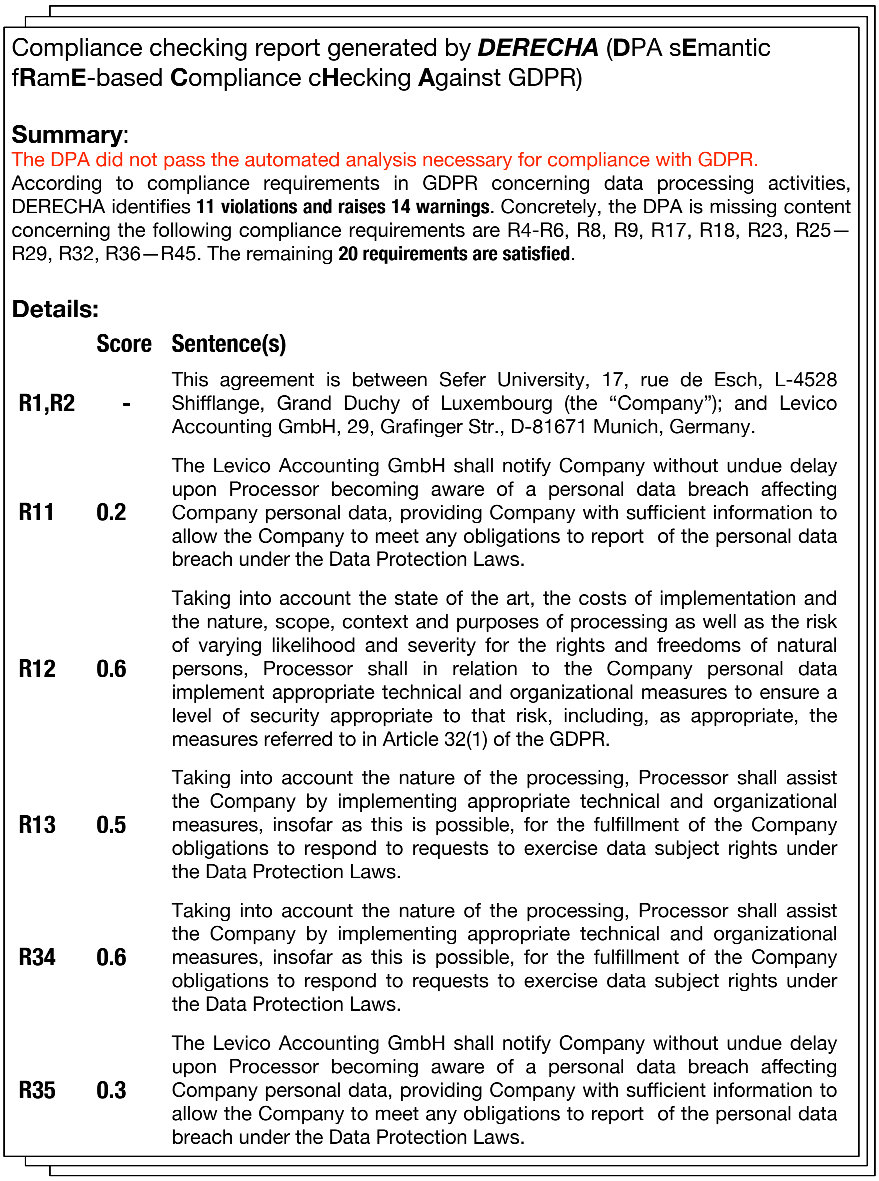
In the example shown in Table 4.8, the predicates are matching since both predicates contain the overlapping verb "engage". For the sake of providing an example, though not needed here as condition 1 is met, for condition 2, we compute WuP between the verb in the predicate of R7 and each verb in the enriched predicate of the statement. This results in WuP scores of 0.4 between "engage" and "hire", 0.4 between "engage" and "employ", and 1.0 between "engage" and "engage". In a similar manner, we match the arguments of R7 with the ones in the statement. Given the overlap in the text spans of the *actor* and *object*, they are considered to be matching. The argument related to *reason* in the statement is skipped since it is not part of the requirement. The argument related to *constraint* has no equivalent argument in the statement and is thus considered not matching.

E.3) Computing matching degree. `DERECHA` further computes a score for each compliance requirement indicating the maximum matching degree of any statement satisfying that requirement in the DPA (line 21 in Algorithm 2). Any statement that does not satisfy any of the two predicate matching conditions above directly receives a score of 0: the statement does not match the requirement at all. Otherwise, if a statement satisfies one predicate matching condition, then the matching degree is computed as a fraction where the numerator equals the total number of arguments found to be matching between the statement and the requirement, and the denominator is the total number of arguments expected by the requirement.

We add one to both the numerator and denominator corresponding to the matching predicate since the predicate is an essential element of the SF. Including the predicates when computing the matching degree accounts for the cases when a statement matches only the predicate but none of the arguments in the requirement. In this case the statement should still receive some low matching degree that is greater than zero. For example, **R7** in Table 4.8 expects a total of four arguments, namely *actor*, *object*, *constraint*, and *time*. The matching degree between the statement and R7 is 3/5 (i.e., 0.6), since the statement successfully matches the predicate and two expected arguments of the requirement.

A requirement can be satisfied by multiple statements in a DPA. In this case, we assign to the requirement the maximum matching degree score across statements. Statements with low matching degrees are due to the lack of matching arguments. However, since these statements have similar predicates to a requirement, they can still be relevant. For such statements, we therefore consider the matching degree as an indicator of the confidence level at which DERECHA predicts that any requirement is satisfied or not in the DPA. This can guide the analyst in checking and possibly correcting decisions when the confidence level is low. We elaborate on this in Section 4.6.4.

E.4) Compliance decision. Finally, we make the compliance decision at the DPA level. A requirement is marked as satisfied in the DPA if there is at least one statement satisfying the requirement. Otherwise, the requirement is marked as violated. DERECHA recommends that the DPA should be considered as *not compliant* when at least one mandatory requirement is violated. When optional requirements are violated, DERECHA raises a warning to draw the attention of the analyst to missing common practices. Fig. 4.4 shows the detailed report corresponding to the DPA example in Fig. 4.1 as generated by DERECHA.



Compliance checking report generated by ***DERECHA*** (**D**PA s**E**mantic f**R**am**E**-based **C**ompliance c**H**ecking **A**gainst GDPR)

Summary:
The DPA did not pass the automated analysis necessary for compliance with GDPR.
According to compliance requirements in GDPR concerning data processing activities, DERECHA identifies **11 violations and raises 14 warnings**. Concretely, the DPA is missing content concerning the following compliance requirements are R4-R6, R8, R9, R17, R18, R23, R25—R29, R32, R36—R45. The remaining **20 requirements are satisfied**.

Details:

	Score	**Sentence(s)**
R1,R2	**-**	This agreement is between Sefer University, 17, rue de Esch, L-4528 Shifflange, Grand Duchy of Luxembourg (the “Company”); and Levico Accounting GmbH, 29, Grafinger Str., D-81671 Munich, Germany.
R11	**0.2**	The Levico Accounting GmbH shall notify Company without undue delay upon Processor becoming aware of a personal data breach affecting Company personal data, providing Company with sufficient information to allow the Company to meet any obligations to report of the personal data breach under the Data Protection Laws.
R12	**0.6**	Taking into account the state of the art, the costs of implementation and the nature, scope, context and purposes of processing as well as the risk of varying likelihood and severity for the rights and freedoms of natural persons, Processor shall in relation to the Company personal data implement appropriate technical and organizational measures to ensure a level of security appropriate to that risk, including, as appropriate, the measures referred to in Article 32(1) of the GDPR.
R13	**0.5**	Taking into account the nature of the processing, Processor shall assist the Company by implementing appropriate technical and organizational measures, insofar as this is possible, for the fulfillment of the Company obligations to respond to requests to exercise data subject rights under the Data Protection Laws.
R34	**0.6**	Taking into account the nature of the processing, Processor shall assist the Company by implementing appropriate technical and organizational measures, insofar as this is possible, for the fulfillment of the Company obligations to respond to requests to exercise data subject rights under the Data Protection Laws.
R35	**0.3**	The Levico Accounting GmbH shall notify Company without undue delay upon Processor becoming aware of a personal data breach affecting Company personal data, providing Company with sufficient information to allow the Company to meet any obligations to report of the personal data breach under the Data Protection Laws.

Figure 4.4: Example report generated by DERECHA.

4.6 Evaluation

This section presents the empirical evaluation for DERECHA.

4.6.1 Implementation and Availability

We have implemented our approach in Java. We extract the textual content of a DPA (provided as MS Word document) using Apache POI 3.17 [162]. For operationalizing the NLP pipeline applied in step B of Fig. 4.3, we use the DKPro 1.10 toolkit [99]. In particular, our operationalization employs the OpenNLP tokenizer, POS-tagger and text chunker [163], Mate lemmatizer [164] and Stanford dependency parser [165]. For enriching the textual content of DPA statements (step D), we rely on the semantic parsing results produced by the NLP pipeline. In particular, we apply extJWNL 1.2 [166] for accessing WordNet, and JVerbNet 1.2 [167] for accessing VerbNet. To compute semantic similarity needed in step E in our approach, we use the *Jaro-Winkler* distance [168] and the *wup* metric [154] as implemented by the WS4J 1.0 library [169]. All non-proprietary material related to this tool is available online [170].

4.6.2 Data Collection Procedure

To conduct this study, we have collected a total of 54 DPAs covering diverse sectors and services such as telecommunication, banking, healthcare, postal services, data analytics, cloud and web hosting services. Among them, 34 DPAs have been provided by our industry collaborator (Linklaters LLP) and the rest have been collected from online resources. The goal of our data collection is to manually check whether a given DPA complies with GDPR requirements and more specifically what statements satisfy which requirements. Our data collection was performed in two phases described next.

In the first phase, three authors of this research manually analyzed 24 DPAs for compliance with GDPR. This analysis spanned two working weeks. We started by independently analyzing five DPAs. Specifically, we checked each sentence in the DPAs against the 45 compliance requirements defined in Section 4.4 and further labeled the sentence with the requirement(s) it satisfied. We then met and discussed our findings and observations in several joint sessions. Once we reached an agreement, we split the remaining DPAs such that each DPA was analyzed by two authors. Finally, we carefully discussed with our collaborating experts the statements that were not straightforward. Our interactions with the legal experts were divided into four sessions over the course of one month, where each session lasted two hours. During this phase, we acquired knowledge about the compliance checking of DPAs and devised annotation guidelines for the second phase.

The second phase had three third-party annotators read through and examine the remaining 30 DPAs for compliance against GDPR. All of the annotators are law students and went through a half-day training on GDPR compliance. We shared with the annotators the guidelines from the first phase, the DPAs to be analyzed and the list of compliance requirements. The annotators were instructed to examine each statement in the DPA and select all requirements which they deem the statement to satisfy. The annotators produced their annotations over a one-month span, during which they declared an average of 40 working hours. To mitigate fatigue, the annotators were recommended to limit their periods of work to a maximum of two hours per day.

To measure the quality of the annotations in this phase, we computed the interrater agreement using Cohen's Kappa (κ) [105] on five of the 30 DPAs. The average κ value across pair-wise agreements of the annotations, split by categories *metadata*, *processor's obligations*, *controller's obligations*, and *controller's rights*, are 0.77, 0.72, 0.82, and 0.79, respectively. κ value for the category *controller's obligations* is "almost perfect agreement" [171], whereas the rest of the three categories indicate “moderate agreement” among the annotators. The disagreements were discussed and resolved by the annotators in a subsequent session.

Table 4.9: Document collection results.

		Total		Dev Data		Test Data (Ev)	
Cat[†]	**ID**	$\mathcal{D}$	$\mathcal{S}$	$\mathcal{D}$	$\mathcal{S}$	$\mathcal{D}$	$\mathcal{S}$
MD1	R1	22	29	7	10	15	19
MD2	R2	40	93	13	50	27	43
MD3	R3	33	45	15	22	18	23
MD4	R4	38	133	11	58	27	75
MD5	R5	38	270	11	118	27	152
MD6	R6	36	126	10	31	26	95
MD7	R27	31	914	9	391	22	523
MD8	R28	21	99	5	22	16	77
MD9	R29	8	9	5	5	3	4
PO1	R7	34	62	16	24	18	38
PO2	R8	34	52	17	31	17	21
PO3	R9	43	83	14	30	29	53
PO4	R10	27	55	12	24	15	31
PO5	R11	44	64	18	29	26	35
PO6	R12	36	206	14	49	22	157
PO7	R13	37	85	14	50	23	35
PO8	R14	15	22	12	17	3	5
PO9	R15	18	25	11	14	3	3
PO10	R16	20	27	15	16	7	9
PO11	R17	32	34	11	22	5	5
PO12	R18	15	17	12	12	21	22
PO13	R19	39	67	13	28	26	39
PO14	R20	23	25	8	9	15	16
PO15	R21	29	48	17	29	12	19
PO16	R22	35	49	14	15	21	34
PO17	R23	38	59	16	29	22	30
PO18	R24	31	53	14	27	17	26
PO19	R25	28	36	13	17	15	19
PO20	R30	26	29	14	17	12	12
PO21	R31	15	19	7	8	8	11
PO22	R32	24	35	12	21	12	14
PO23	R33	17	23	9	15	8	8
PO24	R34	37	67	10	25	27	42
PO25	R35	18	25	6	8	12	17
CR1	R26	24	34	13	22	11	12
CR2	R44	4	5	3	3	1	2
CR3	R45	8	21	5	16	3	5
CO1	R36	1	1	0	0	1	1
CO2	R37	2	3	2	3	0	0
CO3	R38	1	1	0	0	1	1
CO4	R39	1	1	0	0	1	1
CO5	R40	1	1	1	1	0	0
CO6	R41	1	1	1	1	0	0
CO7	R42	1	1	0	0	1	1
CO8	R43	9	14	5	7	4	7

[†] Cat: Requirement’s category; mandatory requirements are in **bold**.

The overall document collection resulted in analyzing a total of 7,048 statements in both phases, out of which 1,742 (≈25%) are marked as satisfying at least one requirement. The DPAs annotated during the first phase were the basis for developing our approach (thereafter referred to as Dev dataset), whereas we used the DPAs annotated by the third-party annotators (thereafter referred to as Ev) for addressing our research questions (RQs) and evaluating the performance our approach. Table 4.9 shows the results of our document collection considering the entire dataset, Dev and Ev. The compliance requirements are sorted by category (with mandatory requirements in boldface) and, for each requirement, the table reports the total number of DPAs ($\mathcal{D}$) in which the requirement was found to be satisfied across the 54 DPAs and the total number of statements ($\mathcal{S}$) that satisfy the requirement. For example, R9 describes the processor's obligation (**PO3**) concerning processing personal data only on documented instructions from the controller. In our dataset, R9 was satisfied in 43 DPAs (out of 54 DPAs), and a total of 83 statements were used across the DPAs to comply with R9.

We observe from the table that mandatory requirements are often satisfied by DPAs while the optional ones are not. According to our collaborating experts, requirements concerning controller's obligations are not necessary for compliance, since GDPR provisions related to DPA focus mostly on the metadata requirements and processor's obligations. We note that, as stated earlier, despite our extensive data collection in terms of DPAs, Table 4.9 confirms that the small number of statements we have, for each requirement, prevents us from developing DERECHA based on machine learning.

4.6.3 Evaluation Procedure

To answer **RQ3** and **RQ4**, we conduct the experiments explained below.

EXPI. This experiment addresses **RQ3**. EXPI evaluates whether our approach accurately identifies the violations of compliance requirements in a given DPA. Specifically, we compare the results per requirement in each DPA against the manual annotations in Ev. We define for each requirement (R_i) a *true positive (TP)* when R_i is violated in the DPA and correctly marked as violated by DERECHA, a *false positive (FP)* when R_i is satisfied (i.e., not violated) in the DPA, but falsely marked R_i as violated, a *false negative (FN)* when R_i is violated in the DPA but falsely marked as satisfied, and finally a *true negative (TN)* when R_i is satisfied in the DPA and correctly marked as such. Following this, we report the *accuracy (A)*, *precision (P)*, and *recall (R)*, computed as *A = (TP+TN)/(TP+FP+FN+TN)*, *P = TP/(TP+FP)*, and *R = TP/(TP+FN)*, respectively. In addition, we report the *F-score* that weighs either precision or recall more highly, computed as (1+β^2)*(P*R)/(β^2*P+R). In EXPI, we choose to report $F_{0.5}$, where $\beta = \frac{1}{2}$, indicating that precision is more important than recall in the context of our study as we explain later in this section.

Baseline. In EXPI, we further compare our approach against a baseline (referred to as **B1**), which utilizes off-the-shelf NLP tools for producing SF-Based representations for both the compliance requirements and the DPA statements. More specifically, **B1** applies existing NLP tools for extracting the SRs and generating the SF-based representations. The definitions of the SF-based representations for compliance requirements are not applicable for the baseline. The reason is that checking compliance requires aligning SF-based representations which are generated over the same set of SRs.

In B1, we apply the SRL module provided by AllenNLP [139] which is based on the BERT language model [76]. Since NLP tools generate a predicate-argument structure for each verb in the sentence, **B1** generates automatically multiple SF-based representations for each statement in the DPA. **B1** automatically generates as well the corresponding SF-based representations for each compliance requirement instead of applying our predefined representations. As a result, **B1** checks the DPA compliance by comparing all SF-based representations generated by AllenNLP for a statement against those generated for a compliance requirement. For example, for the requirement "The processor shall not engage a sub-processor without a prior specific or general written authorization of the controller." (R7 in Table 4.2), **B1** generates two SF-based representations. The first one contains $\langle$**engage**$\rangle$, $\mathcal{A} = \{$[the processor]$_{argument0}$, [a sub-processor]$_{argument1}\}$, and the second one is $\langle$**written**$\rangle$, $\mathcal{A} = \{$[authorization]$_{argument1}\}$. **B1** further enriches the DPA text similar to DERECHA. However, **B1** limits matching predicates and aligned arguments to exact overlapping text due to efficiency concerns. Finally, **B1** computes the proportion of predicates in a statement that are matching the predicates of the requirement. If this proportion is greater than a certain threshold θ_{B1}, then **B1** concludes that the statement satisfies the requirement. In our experiments, we empirically tuned θ_{B1} to 0.30. For the reasons explained in Section 4.5.1, SF-based representation does not apply for metadata requirements. Thus, **B1** is applied only for the three other requirements categories.

EXPII. This experiment addresses **RQ4**. We run our approach on a laptop with a 2.3 GHz CPU and 32GB of memory. Our goal is to assess whether execution times suggest that DERECHA is applicable in practice.

4.6.4 Results and Discussion

RQ3: How accurately can we check the compliance of a given DPA?

Table 4.10 shows the results of EXPI; on the left-hand side, the results of DERECHA, and on the right-hand side the results of **B1** introduced in Section 4.6.3. As explained in Section 4.6.2, the results are obtained by running DERECHA and **B1** on the evaluation set (Ev), which is comprised of 30 DPAs (7048 statements) on which we check the compliance of our 45 requirements. DERECHA correctly finds 618 requirement violations (out of 750) and 524 satisfied requirements, but also leads to 76 false violations. As pinpointed in section 4.6.3, **B1** is applied only to the categories of requirements PO, CO, and CR. Hence, we compare the performance of **B1** against DERECHA considering only these categories. Specifically, DERECHA correctly identifies 550 violated requirements (out of 661) and 348 satisfied requirements (out of 419), while introducing 71 false violations. In comparison, **B1** finds 436 violations, 243 satisfied requirements and introduces 175 false violations. To summarize, 182 errors (FPs+FNs) are produced by DERECHA for both mandatory and optional requirements in these three categories, whereas **B1** produces 401 (219 more errors). Overall, DERECHA outperforms **B1** by an average of ≈20 percentage points (pp) in accuracy, ≈17 pp in precision, recall, and $\mathrm{F}_{0.5}$.

In the context of our study, we favor high precision over high recall. In the case of unsatisfactory recall, the human analyst needs to go through the subset of statements that are marked by DERECHA as satisfying a requirement in order to determine whether this is actually the case. Unsatisfactory precision, on the other hand, indicates that some predicted violations cannot be trusted with a high level of confidence. In such a case, the human analyst would have no choice but to check all statements in the DPA to decide whether at least one of them actually satisfies a requirement. However, doing such a thing would defeat the very purpose of DERECHA. In contrast, addressing unsatisfactory recall can be done with relative ease as the number of statements satisfying any requirement usually represent a small percentage. We discuss next the trade-off between additional manual validation and accuracy improvement.

FNs entail that our approach can mispredict a requirement as being satisfied by at least one statement in the DPA. We carefully checked the matching degrees (scores defined in Section 4.5.5) of these FNs and observed that, in 86 out of 132 cases, such requirements are assigned matching degree scores ≤0.5. By reviewing a fraction of statements, which are predicted to satisfy any requirement with a score below some predefined threshold, the analyst can identify statements that actually do not satisfy any requirement. One possible heuristic that was observed to work well is, for each requirement, to only analyze the statement with maximum score if it is

Table 4.10: Results of compliance checking.

		DERECHA								**B1**							
ID	**Cat†**	**TPs**	**FPs**	**FNs**	**TNs**	**A**	**P**	**R**	**$F_{0.5}$**	**TPs**	**FPs**	**FNs**	**TNs**	**A**	**P**	**R**	**$F_{0.5}$**
R1	**MD1**	15	3	0	12	90.0	83.3	100	86.2	-	-	-	-	-	-	-	-
R2	**MD2**	3	0	0	27	100	100	100	100	-	-	-	-	-	-	-	-
R3	**MD3**	5	0	7	18	76.7	100	41.7	78.1	-	-	-	-	-	-	-	-
R4	**MD4**	1	0	2	27	93.3	100	33.3	71.4	-	-	-	-	-	-	-	-
R5	**MD5**	2	0	1	27	96.7	100	66.7	90.9	-	-	-	-	-	-	-	-
R6	**MD6**	2	0	2	26	93.3	100	50	83.3	-	-	-	-	-	-	-	-
R7	**PO1**	4	1	8	17	70.0	80.0	33.3	62.5	5	3	7	15	66.7	62.5	41.7	56.8
R8	**PO2**	8	1	5	16	80.0	88.9	61.5	81.6	12	16	1	1	43.3	42.9	92.3	48.0
R9	**PO3**	1	1	0	28	96.7	50.0	100	55.6	0	1	1	28	93.3	0.0	0.0	0.0
R10	**PO4**	5	2	10	13	60.0	71.4	33.3	58.1	10	12	5	3	43.3	45.5	66.7	48.6
R11	**PO5**	4	1	0	25	96.7	80.0	100	83.3	3	19	1	7	33.3	13.6	75.0	16.3
R12	**PO6**	2	0	6	22	80.0	100	25.0	62.5	3	8	5	14	56.7	27.3	37.5	28.9
R13	**PO7**	3	1	4	22	83.3	75.0	42.9	65.2	5	22	2	1	20.0	18.5	71.4	21.7
R14	**PO8**	27	3	0	0	90.0	90.0	100	91.8	3	0	23	4	23.3	100	11.5	39.4
R15	**PO9**	17	3	6	4	70.0	85.0	73.9	82.5	22	5	1	2	80.0	81.5	95.7	84.0
R16	**P1O**	18	3	7	2	66.7	85.7	72.0	82.6	15	1	10	4	63.3	93.8	60.0	84.3
R17	**PO11**	3	3	6	18	70.0	50.0	33.3	54.5	1	1	8	20	70.0	50.0	11.1	29.4
R18	**PO12**	27	2	0	1	93.3	93.1	100	94.4	23	2	4	1	80.0	92.0	85.2	90.6
R19	**PO13**	2	1	2	25	90.0	66.7	50.0	62.5	0	2	4	24	80.0	0.0	0.0	0.0
R20	**PO14**	12	5	3	10	73.3	70.6	80.0	72.3	11	12	4	3	46.7	47.8	73.3	51.4
R21	**PO15**	13	3	5	9	73.3	81.3	72.2	79.3	2	0	16	12	46.7	100	11.1	38.4
R22	**PO16**	4	1	5	20	80.0	80.0	44.4	69.0	8	9	1	12	66.7	47.1	88.9	52.0
R23	**PO17**	4	3	4	19	76.7	57.1	50.0	55.6	4	12	4	10	46.7	25.0	50.0	27.8
R24	**PO18**	13	3	0	14	90.0	81.3	100	84.4	4	1	9	16	66.7	80.0	30.8	60.6
R25	**PO19**	10	4	5	11	70.0	71.4	66.7	70.4	7	5	9	9	53.3	58.3	43.8	54.7
R26	**CR1**	16	3	6	5	70.0	84.2	72.7	81.6	18	10	1	1	63.3	64.3	94.7	68.7
R27	MD7	6	2	2	20	86.7	75.0	75.0	75.0	-	-	-	-	-	-	-	-
R28	MD8	6	0	7	17	76.7	100	46.2	81.1	-	-	-	-	-	-	-	-
R29	MD9	28	0	0	2	100	100	100	100	-	-	-	-	-	-	-	-
R30	PO20	3	3	7	17	76.7	50.0	30	44.1	4	4	6	16	66.7	50.0	40.0	47.6
R31	PO21	30	0	0	0	100	100	100	100	29	0	1	0	96.7	100	96.7	99.3
R32	PO22	15	4	3	8	76.7	78.9	83.3	79.8	11	8	7	4	50.0	57.9	61.1	58.5
R33	PO23	16	6	3	5	70.0	72.7	84.2	74.8	3	0	19	8	36.7	100	13.6	44.0
R34	PO24	3	0	0	27	100	100	100	100	2	3	1	24	86.7	40.0	66.7	43.5
R35	PO25	12	5	6	7	63.3	70.6	66.7	69.8	17	12	1	0	56.7	58.6	94.4	63.4
R36	CO1	29	1	0	0	96.7	96.7	100	97.3	14	0	15	1	50.0	100	48.3	82.4
R37	CO2	30	0	0	0	100	100	100	100	26	0	4	0	86.7	100	86.7	97.0
R38	CO3	28	0	1	1	96.7	100	96.6	99.3	25	0	5	0	83.3	100	83.3	96.1
R39	CO4	29	1	0	0	96.7	96.7	100	97.3	13	0	16	1	46.7	100	44.8	80.2
R40	CO5	30	0	0	0	100	100	100	100	29	0	1	0	96.7	100	96.7	99.3
R41	CO6	30	0	0	0	100	100	100	100	13	0	17	0	43.3	100	43.3	79.2
R42	CO7	29	1	0	0	96.7	96.7	100	97.3	26	1	3	0	86.7	96.3	89.7	94.9
R43	CO8	20	2	6	2	73.3	90.9	76.9	76.2	24	4	2	0	80.0	85.7	92.3	86.9
R44	CR2	29	1	0	0	96.7	96.7	100	97.3	24	1	5	0	80.0	96.0	82.8	93.0
R45	CR3	24	3	3	0	80.0	88.9	88.9	88.9	20	1	7	2	73.3	95.2	74.1	90.1
Summary		**618**	**76**	**132**	**524**	**84.6**	**89.1**	**82.4**	**87.6**	**436**	**175**	**226**	**243**	**62.9**	**71.4**	**65.9**	**70.2**

† Cat: Requirement's category; mandatory requirements are in **bold**.

below the selected threshold. Assuming we apply the above heuristic, Fig. 4.5 shows the gain in recall achieved, for instance, when considering a threshold of 0.5: recall can be improved from 82.4% to 93.9% at the cost of reviewing ≈6% of the statements. For an average-sized DPA with 200 statements, this percentage corresponds

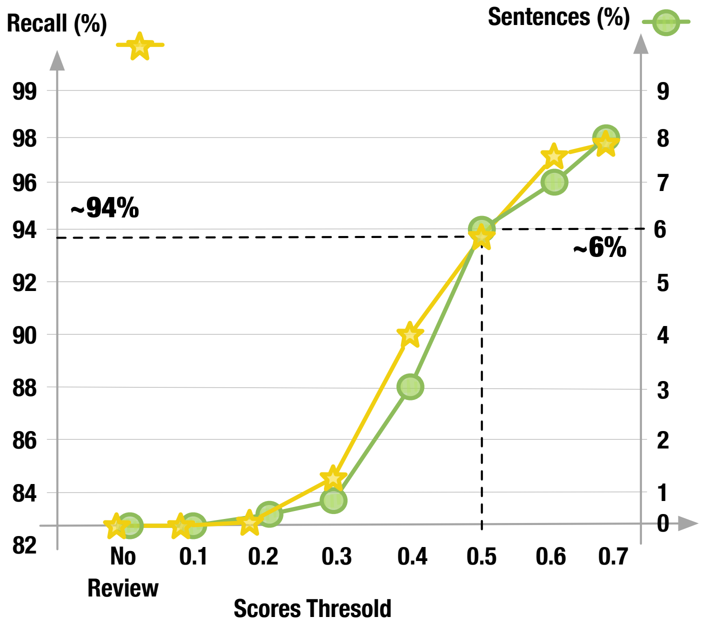


Figure 4.5: Validation-accuracy trade-off.

to approximately 12 statements. Further increases in recall are possible but are associated with higher reviewing effort and the right trade-off depends on context. Eliminating FNs through manual reviews also slightly improves precision and accuracy. For example, the manual review mentioned above increases precision to 90.3% and accuracy to 91.0%. Selecting a threshold at which the user reviews DERECHA's output (matching statements) in order to improve accuracy depends on the resources available and therefore the number of statements that can realistically be reviewed using our heuristic. Users who can afford it might opt for reviewing all of the matching statements with maximum scores for each requirement to achieve a near perfect recall (with ≈90.6% precision and ≈93.1% accuracy). In our DPAs, doing so would result in reviewing about 9% of statements on average across DPAs.

DERECHA achieves an average precision of 81.5% and 93.9 % when identifying violations of mandatory (R1 – R26) and optional requirements (R27 – R45), respectively. In other words, DERECHA introduces false violations (FPs) in a total of 76 cases, out of which 50 are related to mandatory requirements. We analyzed these cases to determine the root causes, which we explain below. We further illustrate each cause with an example from our data collection, and report the number of FPs attributable to that cause.

C1. *Inconsistent reference to the controller and/or processor:* 22 FPs are caused by this error. We observed that in addition to using a concrete named entity representing the controller or the processor (e.g., Levico in Fig. 4.1), many DPAs use other words to refer to the controller or processor. For example, the following statement refers to the processor as *data importer* and to the controller as *data exporter*: “The data importer agrees at the request of the data exporter to submit its data processing facilities for inspection of the processing activities covered by the clauses which shall be carried out by the data exporter”. This statement should be predicted as satisfying R22 (see Table 4.2). However, it is unlikely for DERECHA to identify this statement correctly due to the way it is expressed and the additional unconventional names used in place of processor and controller. As an attempt to reduce such errors, we can modify DERECHA in a way such that the user of our tool can input multiple names for the controller or the processor accounting for both the named entity (e.g., Levico GmbH) and references (e.g., service provider). The user might be familiar with some commonly used names, e.g., *contractor*, *client*, *customer*, or *operator*.

C2. *Loss of context:* 39 FPs are caused by this error. In our work, we use the NLP pipeline to automatically demarcate statements in a given DPA, where each statement corresponds to one sentence as produced by the sentence splitter. Statements included in itemized or enumerated lists can suffer from loss of context when the statements cover grammatically incomplete sentences (marked as **S1** – **S5**). For example, the following list spans five statements "Where Processor uses subprocessors for the processing of personal data on behalf of Controller, Processor shall: **[S1]**

- impose the same obligations on the engaged subprocessors; **[S2]**
- transfer only the data that is reasonably necessary for the purposes of providing the services; **[S3]**
- ensure all the time the security of the shared personal data, using encryption strategies; **[S5]**
- remain fully liable to the Controller for the performance of subprocessors." **[S5]**

In the above example, the combination of the first two statements (**S1** and **S2**) should be predicted as satisfying R23. However, considering these two statements separately leads to falsely predicting a violation of R23 instead.

C3. *Noisy text in DPA:* 15 FPs are caused by this error. Statements in a given DPA are often verbose, i.e., containing content that is considered irrelevant regarding compliance with requirements. Such cases can result in poor performance in the NLP syntactic parsing on which our SR extraction rules depend. For example, the following statement is long and contains multiple clauses that are hard to accurately parse: "Controller will permit Processor to take reasonable steps, according to a reasonable notice and during normal business hours, at Processor's cost to assess compliance by Controller with its obligations under this DPA, including by inspecting Controller's data processing facilities, procedures and documentation (limited to a maximum of one inspection in any twelve month period, or such further occasions as may be required by Privacy Law or if there is an actual Personal Data Breach by Controller)".

Handling errors caused by **C2** and **C3** could be achieved with heuristics to identify and paraphrase itemized lists, or filter out noisy content from DPA statements.

Table 4.10 further suggests that the baseline **B1** has two disadvantages. First, compared to our approach, **B1** checks the compliance of a statement in DPA against a compliance requirement considering multiple verbs (predicates). This not only increases the processing time exponentially but also leads to poor results. Some of these predicates include very common verbs that can be found often in statements such as "mean", "take", and "write". Having multiple predicates increases the likelihood of concluding a match between statements and requirements, consequently leading to more FNs (e.g., R14 and R33). Second, **B1** relies fully on the NLP tool for SRL. In some cases, **B1** could not identify any statement satisfying a given requirement since the SRL tool could not accurately parse the statement. This led to producing more FPs (e.g. R11 and R13). In contrast, DERECHA employs a set of rules that are likely to decompose the statement into meaningful phrases and thus conclude more accurately about their compliance against requirements. Our analysis shows that off-the-shelf NLP tools are unable to identify SRs in a legal text and are thus not adequate for checking the compliance of DPAs.

The answer to RQ3 is that DERECHA is 84.6% accurate in identifying both satisfied and violated GDPR requirements in DPAs. Specifically, DERECHA identifies violated requirements with a precision of 89.1% and a recall of 82.4%. Compared to a baseline that relies on existing NLP tools, DERECHA provides an accuracy gain of ≈20 pp. Our approach also computes matching scores that allow the analyst to review a small percentage of statements with low scores and thus eases the identification of FNs to further increase accuracy.

RQ4: How efficient is our approach in terms of the execution time?

Recall that our approach consists of five steps as shown in Fig. 4.3. Step A is concerned with manually defining SF-based representations for the requirements in GDPR. Step A took about a month, including reviewing related work to define the set of SRs. This step is a one-off and performed only occasionally upon changes in the regulation, i.e., new requirements become relevant to DPA compliance. For analyzing the complete evaluation set of 30 DPAs, running our approach requires ≈90 minutes (corresponding to an average of ≈0.77 second for processing a given statement). The detailed execution time required by each step of our approach is as follows. Preprocessing (step B) requires a total of 1 minute (≈0.009 second per statement). Generating the SF-based representations automatically (step C) is dominated by identifying SRs. This step takes in total about 52 minutes (≈0.44 second per statement). We note that step C consumes most of the time in our approach since our rules are built over syntax parsing. Finally, steps D and E require 37 minutes (≈0.32 second per statement).

For efficiency, we implement step D (text enrichment) as part of step E (compliance checking), i.e., we perform text enrichment of arguments only when the statements in DPA have passed predicate matching. Our approach runs ≈10 times faster than the baseline **B1**. The reason is, **B1** compares all the predicates in a given statement against all the predicates in a requirement leading to a significantly longer computational time.

The answer to RQ4 is that our approach has a running time enabling its application in practice. Running our approach on an average-sized DPA (with 200 statements) takes ≈2.5 minutes to check its compliance against all 45 requirements.

4.7 Related Work

Regulatory compliance is a long-standing research topic in the RE literature [12, 13, 172–177]. While little attention has been given to the compliance of DPAs, recent research explores GDPR compliance of data sharing. We therefore discuss current research on data sharing platforms (e.g., cloud service providers) since a GDPR-compliant DPA provides a complete set of compliance requirements that must be addressed in data processing activities. In addition, we address below two other broad topics that are closely related to our proposed approach for checking compliance, namely elicitation of compliance requirements from regulation, and automated support for compliance checking of requirements and software systems. Naturally, we focus our discussions in this section on GDPR.

4.7.1 GDPR Compliance of Data Processing Activities

Several studies analyze the impact of GDPR on data sharing platforms [178, 179]. Urban et al. [180] study the behavior of online advertising companies in data tracking and information sharing with respect to the GDPR. Alhazmi and Arachchilage [181] investigate the obstacles that developers encounter when implementing the privacy provisions in GDPR, including lack of familiarity with legal text. In a follow-up work, the authors [182] propose a game design framework to help software developers better understand and implement privacy-related requirements in GDPR. Shastri et al. [183] introduce anti-patterns, referring to practices of cloud-scale systems that serve their originally intended purpose well but hinder their compliance with GDPR. For example, such systems are developed to store personal data without having a clear timeline for deleting and sharing them with other applications. While this feature is naturally beneficial in terms of revenues from the system, it violates the storage and purpose limitations of GDPR.

We observe from the above-mentioned work that implementing the GDPR requirements in practice is challenging for developers. Even developers that are familiar with GDPR still struggle with identifying the complete set of requirements that is most relevant to their work. A compliant DPA contains a complete set of legally binding requirements that must be adhered to by both the controller and processor. Recall our example in Section 4.1 which describes an accounting office (the processor) that provides payroll administration services to Sefer University (the controller). In this case, the engineers working in the accounting office should specify explicit requirements to demonstrate that their payroll software complies with GDPR. Since the DPA contains statements specific to the context of this service, such requirements can be defined based on the DPA provided, assuming it has been properly verified and found to be GDPR-compliant. For example, the engineers should define a communication procedure, satisfying time requirements, in case of personal data breach (pursuant to S11 in Fig. 4.1), a module that enables rectifying personal data of Sefer University's employees (data subjects) upon request (pursuant to S9), and an authorization module that allows only authorized users to access personal data (pursuant to S10). Our approach, presented in this chapter, provides automated assistance to the developers and requirements engineers to check the compliance of a DPA prior to relying on it for specifying compliance requirements for developing GDPR-compliant systems.

4.7.2 Elicitation of Compliance Requirements from Regulation

In the RE literature, representing requirements in regulations (including GDPR) relies heavily on model-driven engineering [1,23,26,107,108,184–189]. Other methods for representing compliance requirements have however been explored. Ayala-Rivera and Pasquale [85] propose GuideMe, a systematic approach that facilitates the elicitation of requirements linking GDPR data protection obligations with privacy concerns that should be implemented in a software system. In an early work, Breaux et al. [190] extract and prioritize rights and obligations from the US healthcare regulation and represent them by applying semantic parameterization, a method that enables expressing rights and obligations in restricted natural language statements. The authors derive from the regulation constraints and obligations where each phrase is attributed to an element in their representation. Following this, Breaux and Anton's [191] propose a frame-based requirements analysis method that extracts semi-formal representations of requirements from regulations. More recently, formal languages [192–195] and predefined templates [196] have also been introduced in the literature to represent compliance requirements. To our knowledge, the only work to date that addresses the compliance of DPAs against GDPR is the work by Amaral et al. [26]. The authors create a conceptual model that characterizes the DPA-compliance related information content according to GDPR provisions.

Compared with Amaral et al.'s work, we go one step further and present automated support for checking the compliance of DPAs against GDPR. Another major difference (also compared to the above listed work) is that we define DPA-related compliance requirements as "shall" requirements. Our choice is motivated by the familiarity of both requirements engineers and legal experts with this format. Requirements engineers know from the common templates in RE (e.g., Rupp's [197] and EARS [198]) that a modal verb ("shall" in this case) is an essential element indicating the importance of that requirement. Legal experts are also familiar with this format since "shall" is often used in drafting requirements in regulations [199].

The use of SRs (also referred to as semantic metadata in the RE literature) has been also discussed in the context of regulatory compliance. Semantic metadata is a prerequisite for deriving compliance requirements [19, 172, 184]. It further facilitates transforming legal text to formal specifications [200]. Several strands of work define legal concepts pertinent to requirements in regulations as semantic metadata types. For a detailed discussion on semantic metadata in the RE literature, we refer the reader to the work of Sleimi et al. [145]. In our work, we refine and adapt a set of 10 SRs from literature [175, 189, 201–204].

4.7.3 Automated Support for Compliance Checking

Automated and semi-automated approaches using ML (in combination with NLP) [20, 23, 115, 205–208], as well as manual approaches through crowd-sourcing [19, 22, 112, 113], have been studied in the RE literature for classifying the textual content according to privacy regulations. With respect to GDPR, automated compliance has been widely studied [14–18]. Several approaches rely on semantic web as an enabler [209, 210]. Li et al. [211] identify a set of operationalized GDPR principles and further develop a tool to test these derived GDPR privacy requirements. The authors apply design science and their findings reveal that GDPR can be operationalized and tested through automated means. Nazarenko et al. [212] propose enriching legal text with semantic information both at a sentence-level (e.g., annotating that a sentence contains an exception) and at a phrasal-level (e.g., annotating that a phrase represents a legal entity) to facilitate semantic search and querying legal text. The authors apply their proposed approach on the French version of GDPR. Sleimi et al. [145] propose an automated rule-based method over the NLP syntax parsing to extract legal semantic metadata, e.g., an actor or a sanction, from regulations. They evaluate their approach on traffic law. Bhatia et al. [21] propose using semantic frames to detect incompleteness in PPs. Specifically, the authors manually analyze 15 PPs and conclude a set of 17 semantic roles that are expected to be present in other policies from the same domain. The authors use these SRs to further study the effect of an incomplete privacy statement (i.e., a statement that misses semantic roles) on the user's perspective of a privacy risk.

The approaches of Sleimi et al. [145] and Bhatia et al. [21] are the closest to our work. In contrast, our work utilizes automatically-generated semantic frames for enabling automated compliance checking of DPAs at a phrasal level.

Another strand of research focuses on checking the compliance of websites and mobile applications against applicable laws. Extensive analyses of popular websites show that the variance in how organizations interpret privacy and data protection standards can lead to legal documents (e.g., PPs) which are not informative for individuals [213, 214]. Relevant to this, the lack of templates for drafting such legal documents can cause discrepancies between the data processing activities and what is actually disclosed to individuals [215], e.g., about the purpose of data processing (an essential requirement in GDPR) [216]. Several approaches are proposed to check the compliance of websites by classifying the content of their disclosed PPs according to the provisions of GDPR. Our work addresses three main challenges introduced in the above-listed work. First, we derive the compliance requirements from GDPR in collaboration with legal experts, which alleviates the risk of misinterpreting GDPR. We further make these requirements publicly available. Second, our work addresses the compliance of DPAs, another essential source for requirements for data processing activities in software systems. Our proposed automation is applicable to any DPA and does not assume conformance with any predefined template.

4.8 Threats to Validity

Below, we discuss threats to the validity of our empirical results and what we did to mitigate these threats.

Internal Validity. The main concern with respect to internal validity is bias. To mitigate this threat, we curated our evaluation dataset exclusively through third-party annotators. The annotators were never exposed to the implementation details of our approach. Another potential threat is subjectivity in our interpretation of the GDPR text when extracting the compliance requirements related to DPA. To mitigate this threat, requirements extraction was done in close collaboration with three legal experts, who have expertise in European and international laws. Our extracted requirements are further made publicly available and thus open to scrutiny. Another threat to internal validity is our reliance on most-frequent word sense (MFS) disambiguation for identifying the semantically related words to enrich the texts of DPAs. Using more advanced disambiguation methods might improve the results of our approach. However, the wide use of MFS in the NLP literature and its reported performance provide reasonable confidence about the results of our approach. That said, further experimentation can help mitigate this threat.

External Validity. We evaluated our approach on 30 real DPAs from different sectors. Our approach performed well when identifying violations related to the GDPR requirements. This provides us with reasonable confidence that our solution is generalizable. That said, experimenting with additional DPAs can help further mitigate support external validity.

4.9 Extending our Methodology beyond GDPR

Our suggested procedure for assessing the compliance of DPAs can be applied on regulations beyond GDPR or document types other than DPAs. To illustrate this point, assume that a regulation is denoted as $\mathcal{G}$, and a legal document type is denoted as $\mathcal{T}$. In our context, GDPR is an instance of $\mathcal{G}$, and DPAs are instances of $\mathcal{T}$. To reuse the same methodology for other instances of $\mathcal{G}$ and $\mathcal{T}$, based on our experience, we anticipate the following steps and effort needed for each step.

1. Requirements elicitation from $\mathcal{G}$ with respect to $\mathcal{T}$, consuming about 30% of the effort. This step depends mainly on the availability of legal experts and the size and complexity of the provisions in $\mathcal{G}$.

2. Define semantic frames (SFs) over the compliance requirements derived from $\mathcal{G}$, consuming about 10% of the effort. Prior to defining the SFs, one has to check whether the same semantic roles (SRs) defined in our work can be re-used as-is in the new context. If not, SRs must be refined accordingly. Defining the SFs should preferably be performed in collaboration with legal experts. Therefore, this step depends as well on their availability, though their degree of involvement would typically be much lower than in the previous step.

3. Curate an annotated dataset for $\mathcal{T}$, consuming about 10% of the effort. The first thing to consider is whether (unlabeled) data for $\mathcal{T}$ are available. Curating a labeled dataset of examples from $\mathcal{T}$ is a prerequisite for developing any automation. The annotation task involves manually checking the compliance of the textual content of $\mathcal{T}$ against the requirements derived from $\mathcal{G}$. This step largely depends on data availability as well as the availability of annotators who have the right expertise. One should anticipate for providing training material about compliance with respect to $\mathcal{G}$.

4. Develop an automation strategy for checking the compliance of $\mathcal{T}$ against $\mathcal{G}$, consuming about 50% of the effort. The main things to adjust in our approach include the extraction rules of SRs, and the similarity thresholds applied for matching the predicates and arguments between the text of $\mathcal{T}$ and the requirements of $\mathcal{G}$.

4.10 Expert Interview Survey

To get feedback from legal experts at our collaborating partner about the usefulness of DERECHA in practice, we organized an interview survey. To this end, the survey was conducted with the same settings as the one conducted for *CompAι*, described in Section 3.9 in the previous chapter. The survey material consists of two data processing agreements (DPA1 – DPA2), automatically analyzed by DERECHA. Table 4.11 lists the details of the analyzed DPAs. For each DPA, the table reports the total number of pages, the number of pages marked by DERECHA as containing text that satisfy GDPR as well as the total number of GDPR requirements identified. Using Likert scales [119], our survey aimed at collecting feedback from the experts on the four questions and two follow-up questions listed in the DPAs-related questionnaire.

Table 4.11: Survey material details.

Data processing agreement	Pages	Pages with text satisfying GDPR	Requirements found by DERECHA
DPA1	5	5	42
DPA2	11	10	61
Summary	16	15	103

The DPAs-related questionnaire includes the following questions:

Q1- On this page, indicate all the requirements that have not been identified by DERECHA? *Highlight all.*

a. The requirements found by DERECHA helped me easily spot the missed requirements. (Asked for each missed requirement)

○ *Strongly agree* ○ *Agree* ○ *Neutral* ○ *Disagree* ○ *Strongly disagree* ○ *Not relevant*

Q2 On this page, indicate all requirements found by DERECHA that are not requirements? *Highlight all.*

a. The requirement found by DERECHA is not an requirement, but it provides useful information that would trigger further discussion. (Asked for each requirement marked as wrong by experts)

○ *Strongly agree* ○ *Agree* ○ *Neutral* ○ *Disagree* ○ *Strongly disagree* ○ *Not relevant*

Q3 On this page, I would perform the compliance analysis faster and more efficiently with the help of DERECHA than without it.

○ *Strongly agree* ○ *Agree* ○ *Neutral* ○ *Disagree* ○ *Strongly disagree* ○ *Not relevant*

Q4 On this page, given my time budget in daily practice, it is likely that I would have missed some important information if I had done the compliance analysis fully manually.

○ *Strongly agree* ○ *Agree* ○ *Neutral* ○ *Disagree* ○ *Strongly disagree* ○ *Not relevant*

4.10.1 Survey Results

Table 4.12 summarizes the results from the expert interview survey, which we conducted according to the procedure described in chapter 3 (Section 3.9). The table provides overall statistics from the survey, showing for each DPA the number of requirements found by DERECHA, the number of requirements marked as correct by experts (true positives or TPs), the number of requirements marked as wrong by experts (false positives or FPs),

Table 4.12: Interview survey results.

Data processing agreement	Requirements found by DERECHA	TPs	FPs	FNs	P(%)	R(%)
DPA1	42	39	3	2	92.9	95.1
DPA2	61	54	7	5	88.5	91.5
Summary	103	93	10	7	90.3	93.0

the number of requirements missed by DERECHA (false negatives or FNs), and the corresponding precision and recall metrics.

With regard to Question 1, the experts identified a total of seven (2+5) FNs. Further, the experts marked as correct 93 out of 103 requirements found by DERECHA. Thus, the average recall of DERECHA is 93.0%. For each of the seven FNs, the experts answered the follow-up Question 1a from the DPAs-related questionnaire. Notice that in occasions, these FNs were found in the sourrunding text of FPs. For instance, when an requirement is present in a given sentence but DERECHA finds the requirement in a sentence close to the former one. Both experts provided positive answers to Question 1-a for all FNs, that is the experts “Strongly Agree” that the findings of DERECHA did help in identifying the FNs.

With regard to Question 2, the experts marked as wrong a total of nine requirements found by DERECHA (i.e., FPs). Thus, the average precision of DERECHA is 90.3%. In each case, for the follow-up Question 2a, the answers chosen by both experts were “Agree” in eight cases and “Neutral” in the remaining two. This indicates that the sentences that are falsely classified as satisfying some GDPR requirements often contain useful information for the compliance checking process.

With regard to Question 3, we had 15 (5 + 10) responses, one response per each page containing requirements. The 66.6% of the experts responses used the answer “Strongly Agree”. The remaining 33.4% used the category "Agree". The experts agreed that DERECHA helps them check the compliance of DPAs more efficiently. With regard to Question 4, similar to Question 3, we had 15 responses in total. The 86.6% of the experts responses used the category "Strongly Agree". The remaining 13.4% used the category "Agree". The experts agreed that DERECHA helps them locate requirements that they might otherwise overlook in a daily basis, within the given time budget.

Overall, for Questions 3 and 4, the 76.7% of the responses used the category "Strongly Agree", the remaining 23.3% used the category "Agree". These results show that automated support is highly beneficial for assisting human experts in efficiently checking the compliance of DPAs.

4.11 Summary

In this chapter, we proposed an automated approach for checking the compliance of DPAs against GDPR, since DPAs have a significant impact on the requirements of systems processing personal data. In close collaboration with legal experts, we extracted and documented DPA-related requirements as “shall” requirements. Such documentation provides a shared understanding between requirements engineers and legal experts. In our approach, we first manually created a representation of the GDPR requirements based on SFs. Our approach then automatically generates SFs-based representations for the textual content of a DPA, and subsequently compares this representation against GDPR to check whether the DPA is compliant. Our evaluation is based on 30 real DPAs, curated by trained third-party annotators with a strong background in law. Over this evaluation dataset, DERECHA achieves an average accuracy of 84.6%, a precision of 89.1% and a recall of 82.4%. We also show how higher accuracy can be achieved by focused reviews on a small percentage of DPA statements. We compared DERECHA against a baseline that relies on an off-the-shelf NLP pipeline. DERECHA outperforms this baseline with an average gain of ≈20 pp in accuracy.

As future work, we envision the use of a larger number of DPAs and further investigate the applicability of machine learning methods for enabling automated compliance checking. Another direction that we would like to explore is to run a large-scale experiment to check the compliance of the DPAs available from major service providers against GDPR.

Chapter 5

ML-enabled Automation for Compliance Checking of Data Processing Agreements against GDPR

Most current software systems involve processing personal data, an activity that is regulated in the EU by the general data protection regulation (GDPR) through data processing agreements (DPAs). Developing compliant software requires adhering to DPA-related requirements in GDPR. Checking the compliance of DPAs entirely manually is however time-consuming and error-prone. Automated support is undoubtedly beneficial. However, devising the best automation in an application context is challenging given the various available methods and enabling technologies for representing the legal knowledge and automating the compliance checking process.

In this chapter, we propose an automation strategy that is primarily based on machine learning (ML) for checking GDPR compliance in DPAs. Specifically, we devise an approach that leverages a combination of conceptual modeling and ML. We create a comprehensive conceptual model encapsulating all information types pertinent to DPAs in GDPR, and further define a set of compliance criteria. Our approach uses ML to categorize the textual content of a given DPA according to the conceptual model and concludes whether the DPA complies with GDPR.

Structure. The remainder of this chapter is structured as follows. Section 5.1 presents the motivation and contributions of this chapter. Section 5.2 presents the research questions. Section 5.3 presents the background related to the chapter. Section 5.4 positions our work against related work. Section 5.5 describes our conceptual model. Section 5.6 presents our approach ($D\iota\kappa AIo$). Section 5.7 reports on our empirical evaluation. Section 5.8 discusses the practical considerations when devising an automation for regulatory compliance. Section 5.9 addresses validity considerations. Finally, Section 5.10 concludes the chapter.

5.1 Motivation and Contributions

The exponential growth of AI has significantly impacted modern software systems. Integrating AI technologies in software systems has enabled developing new features that better capture users' needs [217]. Intelligent automation has led to remarkable improvements in diverse application domains such as healthcare [218], transport [219], manufacturing [220], and finance [221]. Much of the progress of AI can be attributed, among other factors, to the increasing availability of large datasets which are paramount to drive the application of ML, including the training of complex neural networks [42]. Such data can in many cases be personal, sensitive, or confidential. Handling massive amounts of personal data in adherence to applicable laws has added an additional burden on engineers to properly address legal requirements as part of RE practice.

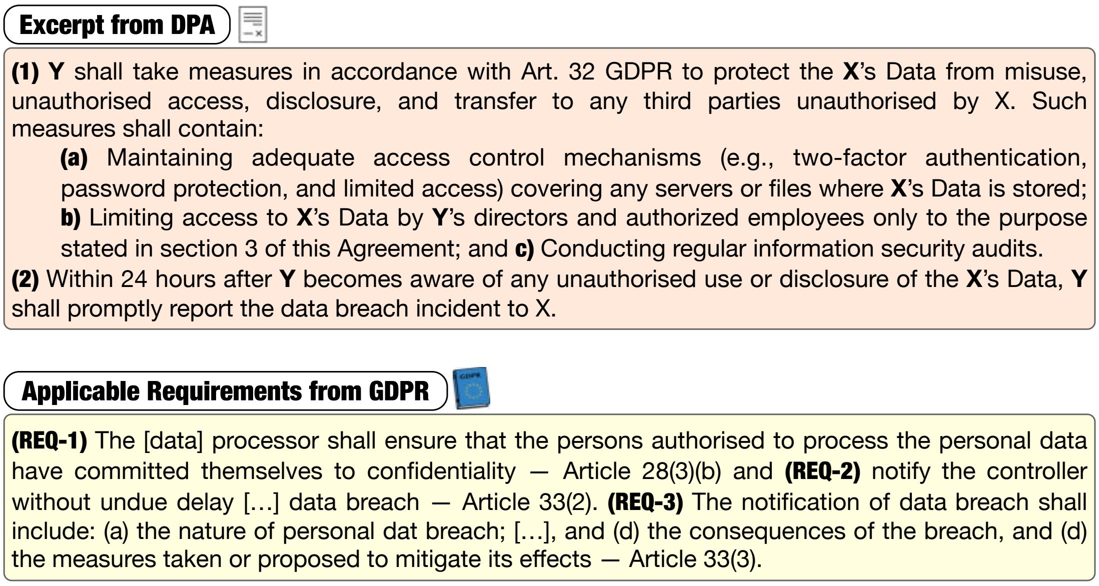
Developing legally compliant software requires specifying explicit legal requirements. This task involves interpreting and transforming the legislative text into such requirements. To extract relevant information from regulations, requirements engineers who understand the functions and properties of the system-to-be should ideally collaborate with legal analysts who understand the law. Even with legal expertise, the task is still challenging, time-consuming and error-prone. First, regulations typically use legal language which can be ambiguous and is normally targeted at governing an entire industry, not specific to a software system [13]. Second, regulations are often composed of long articles and use complex NL structures, e.g., cross-references [12].

New regulations are being continuously enforced to address concerns about privacy and data protection. This impacts software systems processing personal data. GDPR defines *data processing* (the focus of this chapter) as any operation performed on personal data such as collecting, recording, storing, using, or disclosing by transmission or dissemination (GDPR, Article 4(2)). Sanctions for violating GDPR can be substantial. Statistics show that about 337 fines reaching up to 1 billion euros have been enforced due to non-compliance to GDPR [222]. As stated in Chapter 3, according to GDPR, individuals are informed through privacy policies (PPs) about their rights and the terms of personal data handling. Recall from Chapter 4 that to further ensure that personal data remains protected, a DPA must exist between data controller and data processor. Such legal agreements are essential sources for eliciting legal requirements concerning data processing that are different from those requirements elicited from PPs. While the former can be important for software systems used directly by individuals, the latter regulates systems that involve personal data processing.

5.1.1 Practical Scenario

To illustrate, let institution X (the data controller) be an online shopping firm that collects personal information from customers including name, birth date, postal address, social security and bank account numbers. X shares some customer information (e.g., address) with another institution Y (a logistics firm) to manage the delivery of purchased items to the customers. According to the DPA between X and Y, only authorized employees in Y must have access to such information. Fig. 5.1 shows an excerpt from the DPA between X and Y alongside the applicable provisions from GDPR. If Y restricts access to authorized employees, but does not prevent downloading customer files to a shared space accessible by all its employees, such data is vulnerable to being leaked accidentally or maliciously.

To avoid violating GDPR, the requirements engineers in Y can leverage the DPA to specify explicit requirements about the technical measures needed to secure a system's data flow. Thus, ensuring that the DPA provides complete processor's obligations is paramount to developing compliant software. Fig. 5.1 shows at the bottom the legal actions for handling data breach in GDPR, e.g., notifying the controller without undue



Excerpt from DPA

(1) Y shall take measures in accordance with Art. 32 GDPR to protect the **X**'s Data from misuse, unauthorised access, disclosure, and transfer to any third parties unauthorised by X. Such measures shall contain:

(a) Maintaining adequate access control mechanisms (e.g., two-factor authentication, password protection, and limited access) covering any servers or files where **X**'s Data is stored; **b)** Limiting access to **X**'s Data by **Y**'s directors and authorized employees only to the purpose stated in section 3 of this Agreement; and **c)** Conducting regular information security audits.

(2) Within 24 hours after **Y** becomes aware of any unauthorised use or disclosure of the **X**'s Data, **Y** shall promptly report the data breach incident to X.

Applicable Requirements from GDPR

(REQ-1) The [data] processor shall ensure that the persons authorised to process the personal data have committed themselves to confidentiality — Article 28(3)(b) and **(REQ-2)** notify the controller without undue delay […] data breach — Article 33(2). **(REQ-3)** The notification of data breach shall include: (a) the nature of personal dat breach; […], and (d) the consequences of the breach, and (d) the measures taken or proposed to mitigate its effects — Article 33(3).

Figure 5.1: Example of an excerpt DPA between institution X (controller) and Y (processor).

delay (**REQ-2**). The figure further shows on top an excerpt of a DPA where the procedure is more detailed, e.g., reporting the breach within 24 hours (sentence (**2**)). The requirements engineers can translate the different DPA statements into concrete legal requirements, e.g., encrypting data, disabling downloads, and notification alerts.

Regulatory compliance has long been studied in RE [186,189,223,224]. Existing work relies mostly on model-driven engineering [225, 226], restricted NL and predefined templates [190, 192]. Automated approaches for enabling compliance checking have also been investigated. Applied technologies include semantic parsing [21], rule-based [115], NLP [145], ML [20], or a combination of the latter two [23, 90]. The majority of existing work focuses on the completeness and compliance of PPs against several regulations including GDPR [22, 227–229]. While a DPA is yet another legal document imposed by GDPR, it contains legal requirements that impact software systems throughout the data processing activities beyond to what is exposed in PPs. For instance, software must include stronger authentication mechanisms to ensure data protection.

In Chapter 4, we propose eliciting DPA-related requirements from GDPR and documenting them in NL as "shall" requirements. We further implement a rule-based automation (referred to as DERECHA) that verifies whether DPAs satisfy GDPR requirements based on the semantics of the DPAs' textual content. This approach suffers from two drawbacks. First, using NL to represent GDPR requirements makes these requirements prone to various quality issues such as ambiguity and inconsistency. While NL facilitates communication between legal experts and requirements engineers during the elicitation phase, the latter might need to handle emerging quality issues when managing the overall requirements throughout the software development lifecycle. Second, adapting rules to regulation changes would entail major changes in the GDPR requirements and rule-based system, making the significant involvement of legal experts inevitable. It has been acknowledged that regulation changes and understanding the impact of this change on the compliance process is challenging [230–235].

To alleviate these drawbacks, in this chapter, we propose leveraging conceptual modeling and a combination of ML and NLP. In the regulatory compliance context, modeling helps define structured domain knowledge from the regulation [236]. In our work, we create a conceptual model that captures the semantics of DPAs, including rights and obligations. We then utilize NLP and ML to automatically classify the textual content of DPAs according to the information types in our conceptual model. Such classification is a prerequisite for detecting GDPR breaches. Incorporating ML in the automated solution has various practical benefits. Most notably, adapting the overall solution to future regulation changes does not require the intensive involvement of legal experts, in contrast to changing rules. Information types that are no longer required can simply be dropped from both the conceptual model and solution. Introducing new information types, however, requires modifying the conceptual model with the help of legal experts by adding concepts and relationships. While the task is still not trivial, it is less likely to lead to inconsistencies in comparison to changing multiple NL requirements and the corresponding rules. Then, for each new type, one must build a respective ML classifier to be plugged into the solution. Compared to the rule-based solution, however, ML requires creating annotated datasets. Developing an ML-based solution is nonetheless advantageous since examples of DPAs are available online and new annotated datasets are required only occasionally (e.g., upon a regulation change). In other words, such an ML-based solution would be practically advantageous in many contexts, provided that it fares as accurately as its alternatives.

5.1.2 Contributions

The chapter makes the following contributions:

(1) We create, building on our previous work in RE [24, 26], a holistic representation of DPA-related GDPR requirements in the form of a conceptual model that contains a total of 63 information types capturing any content to be expected in a GDPR-compliant DPA. We describe our model in Section 5.5.

(2) We devise an AI-enabled automation strategy ($D\iota\kappa AIo$) for checking the textual content of DPAs against the conceptual model. Since compliance against regulations is often checked late in the software development process [237], we aim to prevent unnecessary costs by ensuring that the complete set of legal requirements concerning data processing is captured at an early stage in RE. We describe the details of $D\iota\kappa AIo$ in Section 5.6, and further discuss in Section 5.8 the benefits of applying ML compared with the alternative presented in Chapter 4. Our replication package provides tool support and empirical data, including our non-proprietary, annotated DPAs [238].

(3) We empirically evaluate $D\iota\kappa AIo$ on 180 real DPAs including a total of $\approx$ 50,000 sentences. As we elaborate in Section 5.7.2, this dataset was curated as part of our work using third-party annotators who have a strong background in law. On an evaluation set of 30 DPAs, $D\iota\kappa AIo$ detects 483 out of 582 actual genuine violations, while introducing 93 false violations, thus yielding a precision of 83.9% and a recall of 83.0%. Though the overall performance of $D\iota\kappa AIo$ is comparable to that of `DERECHA` (see Chapter 4), the practical benefits of ML, as highlighted above, makes it a more practical solution in the long-term when regulation changes and access to legal expertise is restricted.

Significance. The significance of our work is two-fold: (1) Regulatory compliance regarding data processing is a major concern for requirements engineers, particularly with the new challenges arising from the widespread application of artificial intelligence. We provide a novel, accurate approach to assist both legal experts and requirements engineers in assessing the compliance of DPAs against GDPR; (2) Existing work does not investigate alternative approaches to address this problem. We provide insights, based on a large case study, about how the two main approaches compare.

5.2 Research Questions

RQ1. Which ML classification algorithm yields the most accurate results for identifying GDPR-relevant information types in DPAs? Step 4 of $D\iota\kappa AIo$, which builds an ML classifier for identifying the information types present in a given DPA, can be implemented using several alternative classification algorithms and learning features. RQ1 investigates the accuracy of these alternative classifiers. The most accurate classifier is used to answer the subsequent RQs.

RQ2. How accurate is $D\iota\kappa AIo$ ***in identifying information types in practice?*** Considering the best-performing ML classifier from RQ1, RQ2 assesses the accuracy of $D\iota\kappa AIo$ on unseen DPAs. To draw conclusions about the usefulness of devising a hybrid classification method (i.e., combining ML with semantic similarity), RQ2 further compares $D\iota\kappa AIo$ against a baseline that uses only ML.

RQ3. How accurate is $D\iota\kappa AIo$ ***in detecting GDPR violations in DPAs?*** In RQ3, we evaluate how well our approach detects GDPR violations in DPAs. We further compare $D\iota\kappa AIo$ against `DERECHA`.

5.3 Background

We analyze different supervised machine learning algorithms in this chapter. Specifically, we examine six widely-applied ML classification algorithms in the context of RE [239], namely decision tree (DT), feed-forward neural network (FNN), Linear Discriminant Analysis (LDA), logistic regression (LR), random forest (RF), and support vector machine (SVM). The definitions of these algorithms are presented below.

- LR is a statistical model used in machine learning for binary classification tasks. It is a supervised learning algorithm that predicts the probability (a value between 0 and 1) of an instance belonging to a particular class based on input features [40].

- LDA is a dimensionality reduction technique commonly used for classification tasks. Its main goal is to find a linear combination of features that maximizes the separation between different data classes. It assumes that the input data can be represented as a mixture of Gaussian distributions and projects the input data onto a lower-dimensional space while maximizing the class separability [41].

- DT is a supervised learning method that is used for classification and regression tasks. DTs are flowchart-like structure where internal nodes represent feature tests or attributes, branches represent the outcomes of those tests, and leaf nodes represent the final decision or prediction [37].

- RF is an ensemble supervised learning method that combines multiple decision trees to make predictions. Each tree is trained on a different subset of the training data and the final output is predicted based on which prediction has the majority of the votes [38].

- SVM is a popular supervised machine learning algorithm widely used for solving both linearly separable and non-linearly separable problems. SVM aims at finding an optimal hyperplane (decision boundary) in an n-dimensional feature space that separates the data points of different classes. This hyperplane is chosen based on extreme points which are defined as support vectors [39].

- FNN are structured in a series of interconnected layers of artificial nodes (neurons). The information flows in one direction, from the input layer through the hidden layers to the output layer. These artificial networks can model complex relationships in data and learn from large datasets [42].

5.4 Related Work

In this section, we discuss related work on legal requirements representation and automated compliance checking.

Extracting and representing legal requirements from regulations and regulated documents (e.g., PPs) are extensively studied in the RE literature. In an early work, Breaux et al. [190, 240, 241] apply semantic parameterization for extracting rights and obligations from privacy regulations. Semantic parameterization enables expressing NL domain descriptions of goals as specifications in description logic. A similar method is used by Binsbergen et al. [192] to formalize norms. Hassan et al. [242] extract governance requirements from the law and enterprise regulations and transform them to formal specifications through logic models. Zeni et al. [189] develop GaiusT tool that supports extracting legal requirements from regulations. The tool is based on textual semantic annotation techniques where legal text is annotated based on concepts defined in an ontology. A similar method has been applied by Governatori et al. [243] to represent legal documents. Many approaches rely on model-driven engineering methods [26, 107, 186, 225, 226, 244, 245]. Usman et al. [237] provide insights into common practices and challenges when checking and analysing regulatory compliance. They provide an empirical evidence on the challenges experienced during regulatory compliance. Other approaches propose extracting descriptions of data practices from PPs through manual means such as crowd-sourcing or the involvement domain experts [22, 227, 246, 247]. More recently, Abualhaija et al. [12] propose using question-answering to assist engineers with retrieving compliance-relevant information requirements from a regulation. Zasada et al. [248] evaluate the expressiveness and lexical complexity of compliance rule languages.

Automated means for checking the completeness and compliance of legal requirements have been also investigated in RE. Hamdani et al. [249] present an automated GDPR compliance checking approach that relies on natural language processing (NLP) to extract data practices from PPs and encodes GDPR rules to check the presence of mandatory information. NLP technologies have been utilized also for solving other problems, e.g., Bhatia et al. [21] identify incompleteness in PPs, Lippi et al. [115] automatically detect potentially unfair clauses in online terms of service, and Sleimi et al. [145] extract semantic metadata from legal requirements. Elluri et al. [250] automatically analyze the compliance of PPs against GDPR using BiLSTM multi-class classification and BERT. Torre et al. [90] describe an automated solution which combines NLP and ML for the compliance checking of PPs according to GDPR. More recently, Rahman et al. [215] and Aborujilah et al. [251] presented ML-based techniques to monitor users' compliance with mobile applications. Tesfay et al. [20] utilize ML for summarizing privacy concerns in privacy notices to make such notices more readable and comprehensible for non-experts. Harkous et al. [228] introduce an automated framework based on neural networks for analyzing PPs.

We distinguish our work from the above as follows:

(1) We concentrate our work on eliciting from GDPR the legal requirements pertinent to DPA compliance. Compared to the strand of research on PPs, we focus on aspects of data protection regulations that must be addressed when personal data is being subsequently shared between the organizations which collect and process data. The legal requirements imposed on data controllers (through PPs) are different from those imposed on data processors (through DPAs). Concretely, we use conceptual modeling, as commonly done in RE in other contexts, to represent the DPA-related requirements in GDPR. We extend the conceptual model presented in [26] with additional information types derived from the requirements outlined in Chapter 4.

(2) With regard to automating the compliance checking, many approaches in RE apply NLP technologies or ML. The closest to our work in this chapter is `DERECHA`, introduced in Chapter 4. Recall that `DERECHA` is composed of executable rules developed on top of requirements that are represented in natural language (NL) form. Driven by our motivation to address the limitations of using NL representation and rules as discussed in Section 5.1, our approach leverages a combination of NLP and ML. We discuss the advantages of our approach over `DERECHA` in Section 5.8.

5.5 A Conceptual Model of Information Pertinent to DPA Compliance in GDPR

Our work draws on two existing artifacts for checking DPA compliance against GDPR. The first artifact (thereafter referred to as A) is a list of 45 compliance requirements written in NL concerning data processing in GDPR presented in Chapter 4. The second artifact (thereafter referred to as B) [26] is a conceptual model describing 45 information types that can be found in any DPA. In our work, we aim to build a comprehensive conceptual model that acts as an enabler for devising an automated compliance checking approach using primarily ML. To do so, we create a conceptual model by merging the two artifacts.

Fig. 5.2 shows the resulting conceptual model composed of 63 information types organized into four hierarchical levels: **level-1** (shaded grey), **level-2** (shaded black), **level-3** (shaded yellow), and **level-4** (shaded white). Similar to the original artifacts (A and B), we differentiate between mandatory and optional information types. Mandatory information types originate directly from GDPR provisions, whereas optional types are based on best practices. Missing a mandatory information type leads to a non-compliant DPA, while missing an optional information type raises a warning that the DPA does not follow common practices. In our conceptual model, 33 of the information types are mandatory (font in black) and 30 are optional (font in blue). In the remainder of this chapter, referring to an information type implicitly implies the hierarchical label (e.g., referring to DATA BREACH implies PROCESSOR OBLIGATION;INFORM CONTROLLER;DATA BREACH).

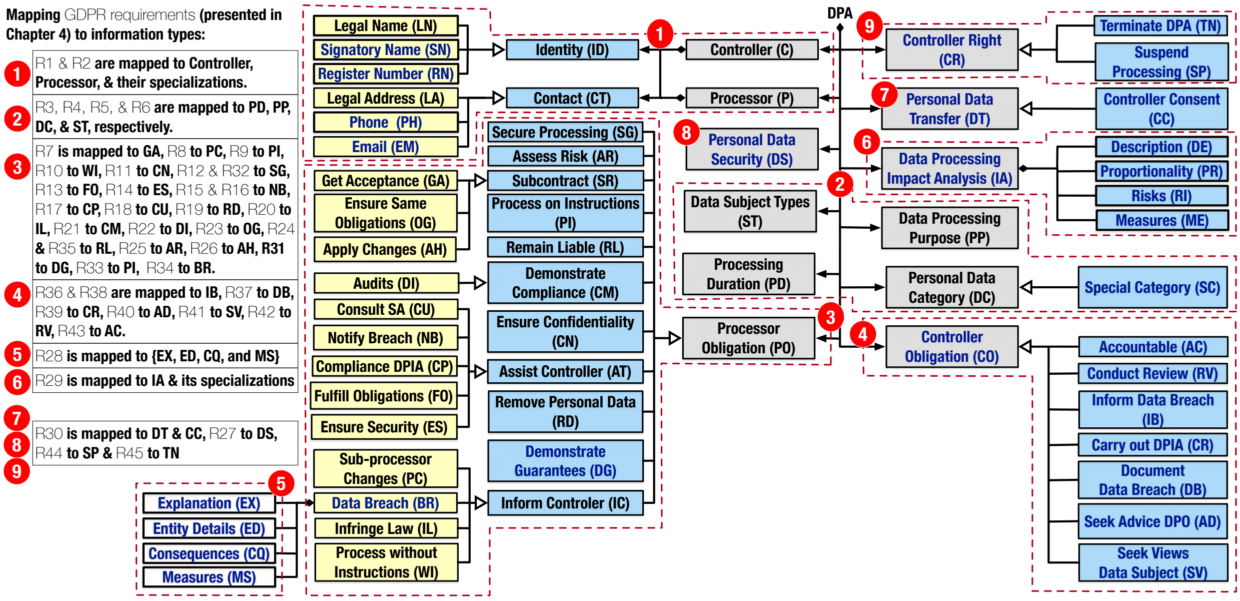

Mapping GDPR requirements (presented in Chapter 4) to information types:
1 R1 & R2 are mapped to Controller, Processor, & their specializations.
2 R3, R4, R5, & R6 are mapped to PD, PP, DC, & ST, respectively.
3 R7 is mapped to GA, R8 to PC, R9 to PI, R10 to WI, R11 to CN, R12 & R32 to SG, R13 to FO, R14 to ES, R15 & R16 to NB, R17 to CP, R18 to CU, R19 to RD, R20 to IL, R21 to CM, R22 to DI, R23 to OG, R24 & R35 to RL, R25 to AR, R26 to AH, R31 to DG, R33 to PI, R34 to BR.
4 R36 & R38 are mapped to IB, R37 to DB, R39 to CR, R40 to AD, R41 to SV, R42 to RV, R43 to AC.
5 R28 is mapped to {EX, ED, CQ, and MS}
6 R29 is mapped to IA & its specializations
7
8 R30 is mapped to DT & CC, R27 to DS, R44 to SP & R45 to TN
9
DPA
Legal Name (LN)
Signatory Name (SN)
Register Number (RN)
Legal Address (LA)
Phone (PH)
Email (EM)
Identity (ID)
Contact (CT)
Controller (C)
Processor (P)
Controller Right (CR)
Terminate DPA (TN)
Suspend Processing (SP)
Personal Data Transfer (DT)
Controller Consent (CC)
Secure Processing (SG)
Assess Risk (AR)
Subcontract (SR)
Process on Instructions (PI)
Remain Liable (RL)
Demonstrate Compliance (CM)
Ensure Confidentiality (CN)
Assist Controller (AT)
Remove Personal Data (RD)
Demonstrate Guarantees (DG)
Inform Controler (IC)
Get Acceptance (GA)
Ensure Same Obligations (OG)
Apply Changes (AH)
Audits (DI)
Consult SA (CU)
Notify Breach (NB)
Compliance DPIA (CP)
Fulfill Obligations (FO)
Ensure Security (ES)
Sub-processor Changes (PC)
Data Breach (BR)
Infringe Law (IL)
Process without Instructions (WI)
Explanation (EX)
Entity Details (ED)
Consequences (CQ)
Measures (MS)
Personal Data Security (DS)
Data Subject Types (ST)
Processing Duration (PD)
Processor Obligation (PO)
Data Processing Impact Analysis (IA)
Description (DE)
Proportionality (PR)
Risks (RI)
Measures (ME)
Data Processing Purpose (PP)
Personal Data Category (DC)
Special Category (SC)
Controller Obligation (CO)
Accountable (AC)
Conduct Review (RV)
Inform Data Breach (IB)
Carry out DPIA (CR)
Document Data Breach (DB)
Seek Advice DPO (AD)
Seek Views Data Subject (SV)

Figure 5.2: Conceptual model of DPA compliance-related information in GDPR.

The figure further decomposes the conceptual model into regions to highlight the source requirements from A used to derive the information types in the model. Our method for creating this conceptual model is as follows. We first identified the information types in A using a qualitative method similar to the one described in Chapter 3. For example, we identify from **REQ-2** (equivalent to R34 in A) and **REQ-3** (equivalent to R28 in A) in Fig. 5.1 the hierarchical information types: PROCESSOR OBLIGATION, INFORM CONTROLLER, DATA BREACH, as well as the different compositions of DATA BREACH. We then rely on trace links to the actual GDPR provisions associated with A and B to map each requirement in A to an information type in B. Since we consider additional information from A, we adjust the specializations in B to extend the model. For instance, in relation with the information types listed above, we added the specialization INFORM CONTROLLER to *B* to encapsulate other related requirements, e.g., PROCESS WITHOUT INSTRUCTIONS derived from R10. We applied the following modes for deriving the final information types: (i) one-to-one representation, e.g., R3 in A is represented as the information type PROCESSING DURATION (region 2); (ii) one-to-many representation, e.g., R28 in A is represented as the four specializations of DATA BREACH (region 5); and (iii) many-to-one representation, e.g., R12 and R32 in A are represented as the information type SECURE PROCESSING (region 3).

To check the compliance of a DPA according to GDPR provisions, we define a set of 37 criteria, of which 22 criteria are concerned with mandatory information types and 15 are concerned with optional ones. The criteria are defined on the specializations since they can lead to violations in the more generic information types. For instance, we define 18 criteria concerned with mandatory information types under PO, including six coming from level-2 and 12 from level-3. Each criterion verifies whether the DPA satisfies or violates GDPR with respect to a particular information type. In our context, a violation refers to an absent information type, whether mandatory or optional. We denote a criterion using the negation sign (¬) to refer to the absence of an information type, e.g., the criterion **¬PD** checks whether the DPA contains any sentence related to PROCESSING DURATION (PD). If no sentence is found, then the DPA violates GDPR. The list of compliance criteria formulated based on the absence of information types are listed in Table 5.5 in Section 5.7. The conceptual model and the compliance criteria are the basis for developing $D\iota\kappa AIo$, as we explain next.

5.6 Approach

Fig. 5.3 shows an overview of $D\iota\kappa AIo$, which is composed of six steps, labeled 1 – 6. $D\iota\kappa AIo$ takes as input a DPA (see the excerpt in Fig. 5.1) and returns as output a report with recommendations about the DPA compliance. In Step 1, we preprocess the DPA using an NLP pipeline. In Step 2, we transform the text into embeddings. In Step 3, we train a ML classifier. In Step 4, we use ML to classify the text in the input DPA according to the information types in Fig. 5.2. In Step 5, we use cosine similarity to classify the text according to GDPR requirements presented in Chapter 4. Finally, in Step 6, we combine the classification results from Steps 4 and 5 to generate recommendations about whether the input DPA is compliant. We elaborate these steps next.

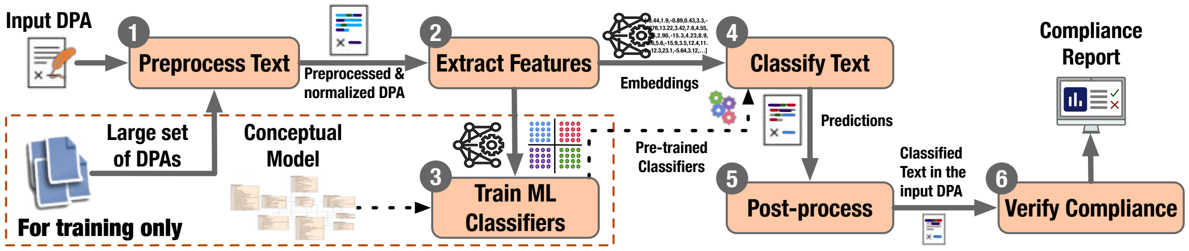


Figure 5.3: Overview of $D\iota\kappa AIo$.

5.6.1 Step 1: Preprocess Text.

In this step, we parse the input DPA using an NLP pipeline conformed of the categories A and C from Figure 2.1 form Section 2.2. The first category obtains the sentences, whereas the second is used to normalize the text since it identifies the canonical forms of the words and removes stopwords, e.g., “the applied” becomes “apply”. The normalized sentences are then passed on to the next step.

5.6.2 Step 2: Extract Features.

Step 2 transforms the sentences from Step 1 into embeddings. Embeddings represent the learning features which are a prerequisite for using ML-based and similarity-based classification. As we discuss in Section 5.7, we experiment with four alternative methods for generating the embeddings. These alternatives include the pre-trained models from word2vec [59], GloVe [60], fasttext [61] which generate 300-dimensional vectors for each word, and SBERT [77] which generates 768-dimensional vectors. SBERT produces sentence embeddings directly, accounting for the context in which the words occur. To compute the sentence embeddings corresponding to the three remaining alternatives, we take the average of the words embeddings in the sentence following common practices [23, 252]. The sentence embeddings in the input DPA are used in Step 4 for predicting the information types in each sentence and in Step 5 for computing the semantic similarity between the sentences and the average embedding of all the training data under each information type. Steps 1 and 2 are also performed on the training data to generate the embeddings needed to train the ML classifiers in Step 3.

5.6.3 Step 3: Train ML Classifier.

We use feature embeddings from Step 2 to train an ML classifier on a large set of manually-annotated DPAs. We discuss the annotation process in Section 5.7.2. In this step, we train a binary classifier for each information type in Fig. 5.2 to achieve multi-label classification since the same text in a DPA can be about multiple information types. To train each binary classifier, we use as positive examples all sentences annotated with a particular information type (e.g., PERSONAL DATA SECURITY) and as negative examples all other sentences excluding the ones ascribed to that information type (e.g., all but PERSONAL DATA SECURITY). The resulting training dataset for each information type was, as expected, highly skewed with significantly more negative examples than positive examples. We restrict ML only for those information types that have at least 20 positive examples, noting that our preliminary experiments showed poor performance for information types with less than 20 positive examples. Inspired by existing work [79], we oversample the positive examples to match the maximum number of positive examples among information types. We then undersample the negative examples to obtain a balanced dataset. The classifiers are trained on a feature matrix in which each row corresponds to a sentence and the columns are the sentence embeddings extracted in Step 2. The binary classes that a classifier predicts indicate the presence or absence of an information type in a sentence. The trained ML classifiers are fed into Step 4.

5.6.4 Step 4: Classify using ML.

In Step 4, we first loop over the sentence embeddings from Step 2 generated for the input DPA. We then apply each binary classifier created in Step 3 to predict in each sentence whether a particular information type is present or not. For example, one ML classifier predicts that the information type PERSONAL DATA SECURITY is present in sentence **(1)** in Fig. 5.1. The labels predicted by the ML classifiers for each sentence are then passed on to Step 6.

5.6.5 Step 5: Classify using Similarity.

In this step, we use semantic similarity to predict the presence of an information type in a sentence. In particular, we compute the cosine similarity [56] between the embeddings of each sentence in the input DPA and the embeddings of the GDPR requirements related to DPA compliance (Chapter 4). The corresponding embeddings for each GDPR requirement is generated via the same process described earlier in Steps 1 and 2. The motivation behind Step 5 is to predict the information types with too few positive examples to effectively train ML classifiers. To increase the confidence of the prediction in Step 6, we apply similarity-based classification on all information types.

We predict an information type using semantic similarity as follows. We first loop over the sentences in the input DPA. We then loop over each GDPR requirement. For each sentence and GDPR requirement, we compute the cosine similarity of their respective embeddings. If the similarity value is above a certain threshold, the sentence is deemed similar to the requirement. We assign an information type to the sentence based on the mapping to GDPR requirements, as illustrated in Fig. 5.2. For example, sentence **(1) in Fig. 5.1** has a similarity value of 0.52 with the requirement R27 (from Chapter 4), which states that "*[...] measures to ensure a level of security can include: (a) pseudonymization and encryption [...]*". The similarity-based classifier predicts the presence of PERSONAL DATA SECURITY (DS) in the sentence since R27 is mapped to DS.

An alternative for realizing this step is to group the training data under one information type and measure the similarity of the sentence in the input DPA against the average embedding of all sentences in that group. Again, if the similarity is greater than a threshold, the sentence will be assigned the same information type as the group in the training data. We also implemented this alternative and experimented with it.

According to results from preliminary experiments, we select 0.5 as the threshold value as it produced, on average, the best results across all requirements. Note that it might be beneficial to define different thresholds for different requirements since the sentences corresponding to the requirements in the DPA can contain more content than is needed to comply with the requirement. Such additional content can indeed reduce the similarity between the compliant sentence and its relevant requirement. However, we find the threshold we selected in our work to be sufficient since the purpose of $D\iota\kappa AIo$ is to identify the presence of information types in at least one sentence in the DPA. The predictions made in this step are passed on to Step 6.

5.6.6 Step 6: Check Compliance.

Step 6 combines the labels predicted in Steps 4 and 5 to conclude a final prediction about the presence of information types in the input DPA, thus determining its compliance. Step 6 concludes that an information type ($\mathcal{I}$) is present in a given sentence (s) only if both ML and similarity-based classifiers predict $\mathcal{I}$ in s. For example, the final prediction for the sentence mentioned in Step 4 will be PERSONAL DATA SECURITY since this information type is predicted by both the ML classifier in Step 4 and the similarity-based classifier in Step 5.

The output of $D\iota\kappa AIo$ is generated at a DPA level as follows: If $\mathcal{I}$ is predicted in at least one sentence, then $\mathcal{I}$ is present in the DPA. Otherwise, $\mathcal{I}$ is absent. $D\iota\kappa AIo$ produces as output a set of violations corresponding to mandatory or optional information types found to be absent in the input DPA.

5.7 Empirical Evaluation

In this section, we empirically evaluate $D\iota\kappa AIo$.

5.7.1 Implementation and Availability

We implemented $D\iota\kappa AIo$ using both Java 8.0 and Python 3.8. Different steps rely on different technologies (displayed in Fig. 5.3), as described next. In Step 1, we use the DKPro 1.10 toolkit [99] for operationalizing the NLP pipeline. In Step 2, we extract the sentence embeddings from SBERT [77] using the paraphrase-MPNet-base-v2 model [253]. This model is available in the Transformers 4.6.1 library [254] provided by Hugging

Face (`https://huggingface.co/`) and operated in PyTorch [255]. We employ WEKA 3.9.6 [101, 102] in Steps 3 and 4 and implement cosine similarity [56] in Step 5. All non-proprietary material related to this tool is available online [238].

5.7.2 Data Collection and Preparation

Our data collection focused on procuring a large set of DPAs and manually annotate them with the information types described in the conceptual model of Fig. 5.2. We collected a total of 180 DPAs, of which 50 were obtained from online sources and 130 were provided by our industry collaborator (Linklaters LLP). Data collection was performed by three third-party annotators. All annotators are law students and went through a half-day training on GDPR compliance. The annotators produced their annotations over a four-month period, during which they declared an average of 165 hours. To mitigate fatigue, the annotation was organized in three batches, and the annotators were encouraged to restrict their work to two hours at a time.

For better understandability, we also shared with the annotators the original DPAs and the DPA-relevant compliance requirements in GDPR, presented in Chapter 4. Given their background in law, the annotators found it more efficient to read through textual requirements derived from GDPR instead of learning the definitions of the information types in our conceptual model. The four hierarchical levels of the information types in our conceptual model increase the learning duration expected by the annotators. As another measure of fatigue mitigation, we shared CSV files containing automatically generated sentences for each DPA where the annotators could select from a drop list next to each sentence up to three requirements that the sentence satisfies. We also included an additional column of free text where the annotators could add remarks, e.g., when a sentence satisfies more than three requirements.

The annotators were instructed to examine each sentence in the DPA and select all requirements that are satisfied by the sentence. As a quality measure, we compute the interrater agreement using Cohen's Kappa (κ) [105] on a subset of $\approx$10%, consisting of DPAs that were independently analyzed by two annotators. The interrater agreement is computed for level-1 information types only. The average κ value across pair-wise agreements of the annotations is 0.78, indicating "substantial agreement" [171]. The disagreements were discussed and resolved by the annotators.

Table 5.1: Data collection results.

$\mathcal{I}$ (mandatory types are in **bold**)	$\mathcal{T}$	$\mathcal{E}$	
	Sentences	Sentences	DPAs
CONTROLLER	62	20	16
PROCESSOR	97	40	27
DATA SUBJECT TYPES	67	56	26
PROCESSING DURATION	115	22	17
DATA PROCESSING PURPOSE	136	34	27
PERSONAL DATA CATEGORY	75	37	27
PROCESSOR OBLIGATION	2,680	725	29
CONTROLLER RIGHT	17	3	3
CONTROLLER OBLIGATION	27	5	5
PERSONAL DATA SECURITY	1,085	462	24
PERSONAL DATA TRANSFER	66	20	20
DATA PROCESSING IMPACT ANALYSIS	4	3	1

The overall document collection resulted in analyzing $\approx$50,000 sentences, out of which about 4,000 ($\approx$8%)

are ascribed to at least one information type. We randomly partitioned the analyzed DPAs into 150 DPAs ($\approx$85%) used for training the ML classifiers in our approach, and 30 DPAs ($\approx$15%) used for evaluation. Hereafter, we refer to the training dataset as $\mathcal{T}$, and the evaluation set as $\mathcal{E}$. Table 5.1 provides overall statistics about our data collection. For each level-1 information type ($\mathcal{I}$), the table lists the number of sentences ascribed with $\mathcal{I}$ in $\mathcal{T}$ and $\mathcal{E}$. Note that the sentences are not mutually exclusive since one sentence can simultaneously satisfy multiple information types. The table further shows the number of DPAs in $\mathcal{E}$ where $\mathcal{I}$ is present. For example, four sentences are available in $\mathcal{T}$ about DATA PROCESSING IMPACT ANALYSIS, three sentences are available in $\mathcal{E}$, and this information type is present only in one DPA in $\mathcal{E}$.

5.7.3 Evaluation Procedure

To answer our RQs, we conduct the experiments below.

EXPI. This experiment addresses RQ1. In EXPI, we evaluate the different alternative ML classifiers trained over different learning features (LFs). EXPI examines the ML classification algorithms mentioned in Section 5.3. We train each ML classifier over different LFs. The first three LFs are based on the pre-trained embeddings from word2vec (LF1), GloVe (LF2), fasttext (LF3), whereas the last one (LF4) is based on the sentence embeddings extracted from SBERT. More details on the ML classifiers and LFs can be found in Chapter 2. Over the training set $\mathcal{T}$, we tune the hyperparameters [256] to optimize the accuracy of each alternative and further evaluate the alternative classifiers using ten-fold cross-validation.

To compare the classifiers, we define, for each information type $\mathcal{I}$, true positives (TPs) as the sentences correctly classified as $\mathcal{I}$, false positives (FPs) as the sentences that are falsely predicted as $\mathcal{I}$, and false negatives (FNs) as the sentences that should be predicted as $\mathcal{I}$ but are missed by the classifier. For each alternative classifier and LF, we aggregate the resulting TPs, FPs, and FNs for all information types and then compute the overall precision (P), recall (R) and F1 measure as: $P = |TP|/(|TP| + |FP|)$, $R = |TP|/(|TP| + |FN|)$, and $F1 = 2 * P * R/(P + R)$.

EXPII. This experiment answers RQ2. Given the best-performing alternative from EXPI, EXPII assesses how well $D\iota\kappa AIo$ can identify information types on the unseen DPAs in our evaluation set, $\mathcal{E}$. To evaluate $D\iota\kappa AIo$, we redefine TP, FP, and FN to better fit the context of compliance checking, as follows: TPs are DPAs where $D\iota\kappa AIo$ correctly predicts the presence of $\mathcal{I}$, i.e., at least one sentence is about $\mathcal{I}$. FPs are DPAs where $D\iota\kappa AIo$ falsely assumes the presence of $\mathcal{I}$, i.e., $D\iota\kappa AIo$ identifies at least one sentence for an absent $\mathcal{I}$ in the DPA. Finally, FNs are DPAs where $D\iota\kappa AIo$ falsely predicts the absence of $\mathcal{I}$, i.e., $D\iota\kappa AIo$ does not find any sentence about $\mathcal{I}$. We then report P, R, and F1, computed as in EXPI. We further compare $D\iota\kappa AIo$ against a baseline that uses ML only. The baseline is built according to Step 3 in $D\iota\kappa AIo$ (see Fig. 5.3) and used as in Step 4.

EXPIII. To address RQ3, we report in EXPIII the results of $D\iota\kappa AIo$ in detecting GDPR violations in the DPAs in $\mathcal{E}$. Recall from Section 5.6 that a violation corresponds to an absent information type. To evaluate $D\iota\kappa AIo$ in EXPIII, we define TPs, FPs, and FNs in converse with EXPII. Concretely, a TP is a genuine violation that is correctly detected by $D\iota\kappa AIo$. Similarly, an FP is a violation that is falsely introduced by $D\iota\kappa AIo$, and an FN is a violation that is missed by $D\iota\kappa AIo$. We then compute P and R as in EXPI. EXPIII compares the performance of $D\iota\kappa AIo$ on the same evaluation set against DERECHA.

5.7.4 Answers to the RQs

RQ1. Table 5.2 reports the accuracy of 24 ML-based alternatives considered in our study for identifying information types in DPAs. Recall from Chapter 2 that LF1, LF2, and LF3 are context-independent embeddings, in contrast to LF4 which provides contextual embeddings. The table shows that LF1, LF2, and LF3 yield similar accuracy across the different ML alternatives, with an average F1 of 72.8%, 71.6%, and 71.4%, respectively. Since such embeddings learn similar information from text without any consideration of context, varying the source from which these embeddings are extracted has little impact on the accuracy of the ML classifiers in our study. Training over LF4, in contrast, significantly improves the accuracy, reaching an average F1 of 81.0%. LF4 yields an average gain in F1 of 8.2 percentage points (pp), 9.4 pp, and 9.6 pp compared to LF1, LF2, and LF3, respectively. The 768-dimensional embeddings in LF4 enable obtaining more knowledge about syntax and semantics in comparison with the 300-dimensional LF1 – LF3 embeddings.

Table 5.2: Accuracy of alternative ML classifiers for identifying information types (**RQ1**).

	DT			FNN			LDA			LR			RF			SVM		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
LF1	67.4	73.5	69.0	71.8	74.1	72.5	69.5	76.3	71.4	69.6	74.8	71.1	73.8	74.9	74.1	76.6	82.8	78.7
LF2	67.0	73.6	68.6	70.5	72.9	71.2	67.5	73.1	69.0	68.3	72.8	69.5	72.9	73.9	73.2	75.7	82.2	77.9
LF3	66.7	69.5	67.4	70.6	72.3	71.1	70.3	75.0	71.6	67.8	71.9	68.9	71.2	72.6	72.6	74.8	80.9	76.8
LF4	**75.2**	**80.0**	**76.8**	**81.7**	**80.2**	**81.1**	**82.4**	**82.9**	**82.6**	**79.5**	**79.5**	**79.5**	**80.7**	**82.0**	**81.2**	**83.8**	**86.2**	**84.7**

LF1 – LF4: embeddings from word2vec, GloVe, fasttext, and SBERT, respectively.

Focusing on LF4, Table 5.2 shows that SVM outperforms alternative classifiers across all three metrics. Pairwise analysis using a paired t-test [257] shows statistical significance in favor of SVM ($p - value < 0.05$). Our results are not surprising considering the robust performance of SVM reported in the RE literature [252,258,259]. While RF is often reported to perform on par with SVM, our results rather show that LDA is comparable to SVM, with an average loss of 2.1 pp in F1. LDA has been investigated in diverse RE contexts and achieved promising results [260–262].

The answer to RQ1 is that SVM trained over LF4 is the best-performing alternative for identifying information types in DPAs, with an average precision of 83.8% and recall of 86.2%. We answer the subsequent RQs using this alternative.

RQ2. Table 5.3 lists the results of $D\iota\kappa AIo$ compared with a baseline that uses ML only (discussed in Section 5.7.3) for identifying the information types in DPAs. Note that the baseline is only applicable to information types with more than 20 positive examples (see Step 3 in Section 5.6) and thus it has zero precision and recall for other information types. In contrast, our approach is applicable to all information types. Overall, the table shows that $D\iota\kappa AIo$ extracts mandatory and optional information types with an average F1 of 82.0% and 76.9%, respectively. $D\iota\kappa AIo$ significantly outperforms the baseline with a gain of ≈ 7 pp in F1 for identifying mandatory types and ≈ 11 pp for identifying optional ones. The results indicate the necessity of devising a hybrid classification method to overcome the complexity of the hierarchical classification problem which can be clearly seen in our conceptual model in Fig. 5.2. This conclusion is in line with the work presented in Chapter 4. Note that for the semantic similarity we employ the first reported alternative. We compute the cosine similarity between each sentence in the input DPA and the GDPR requirements related to DPA compliance (recall Section 5.6). We use this alternative due to its slightly better results (≈ 1.2 pp in F1), for the chosen ML-based alternative and embeddings (SVM and LF4).

In Table 5.4, we provide a breakdown of the results for each information type ($\mathcal{I}$). We note that the automated analysis in this chapter excludes the identity and contact details of CONTROLLER and PROCESSOR. The reason

Table 5.3: Results of information type identification (**RQ2**).

Solution	$\mathcal{I}$	TPs	FPs	FNs	TNs	P	R	F1
Hybrid†	Mandatory	368	93	69	130	79.8	84.2	82.0
	Optional	60	11	25	354	84.5	70.6	76.9
	Summary	428	104	94	484	**80.5**	**82.0**	**81.2**
ML	Mandatory	332	118	105	105	73.8	76.0	74.9
	Optional	52	21	33	344	71.2	61.2	65.8
	Summary	384	139	138	449	73.4	73.6	73.5

† Our approach combines ML with semantic similarity.

is that such details are often mentioned in the same sentence at the beginning of a DPA. Automated means such as named entity recognition or regular expressions are thus not adequate for differentiating CONTROLLER from PROCESSOR. Such details, which are often known to the human analyst, can be provided as input to *DικAIo* in order to enable their automatic detection.

Table 5.4: Accuracy per information type (**RQ2**)

$\mathcal{I}$	P	R	$\mathcal{I}$	P	R	$\mathcal{I}$	P	R
PD	61.1	64.7	**CN**	88.9	92.3	**RL**	81.8	81.8
PP	91.7	81.5	**SG**	88.9	66.7	**AR**	66.7	88.9
DC	92.0	85.2	**FO**	91.7	95.7	**AH**	68.2	93.8
ST	92.0	88.5	**CP**	69.0	73.1	DS	84.2	66.7
GA	75.0	88.2	**RD**	90.5	94.4	IB	95.8	85.2
DC	73.9	89.5	**IL**	63.0	100	CC	75.0	95.5
PI	96.3	89.7	**CM**	86.7	100			
WI	63.3	100	**OG**	76.0	86.4			

[1] Mandatory information types are in bold; See Fig. 5.2 for the full names.
[2] Information types with zero precision and recall are omitted.

We obtained zero precision and recall for 15 information types, three of which are mandatory, the remaining ones being optional. For better readability, we omit these information types from Table 5.4. The reason for the low performance on these information types is the very few training examples in our dataset to develop accurate ML classifiers. As shown earlier in Table 5.1, optional information types have substantially fewer examples compared to mandatory ones. The only exception is PERSONAL DATA SECURITY which captures the technical measures to ensure the level of security that a DPA must include, often expressed in a list spanning multiple sentences. Consequently, our approach tends to systematically predict these information types as absent. We elaborate in RQ3 the impact of these results on the compliance checking.

The answer to RQ2 is that *DικAIo* identifies information types in DPAs with an average precision, recall, and F1 of 80.5%, 82.0%, and 81.2%, respectively.

RQ3. Table 5.5 shows the results of *DικAIo* in detecting violations according to the list of compliance criteria. Recall from Section 5.5 that a violation is related to missing mandatory or optional information types. Overall, *DικAIo* identifies GDPR violations in DPAs with an average precision of ≈84% and recall of 83%. Following the above discussion in RQ2, the information types wrongly predicted as absent by *DικAIo* resulted in a perfect recall for 15 criteria. These criteria correspond to three missing mandatory information types and 12 missing optional ones as can be seen from Table 5.5. However, the average precision achieved by *DικAIo* for the exact same criteria is ≈94.0%. The precision for detecting absent mandatory information types only is ≈84.5%.

Such results show that these information types are often left out from the DPAs and are thus actually absent. Pinpointing the absence of mandatory types is essential in our context. Warning the user about missing optional types can also be informative as they capture best practices. For the remaining criteria, we analyze the root causes of errors (both FPs and FNs) made by *DικAIo* and provide our conclusions below.

Table 5.5: Accuracy of our compliance checking approach (**RQ3**).

ID	TPs	FPs	FNs	P(%)	R(%)	ID	TPs	FPs	FNs	P(%)	R(%)
		Mandatory						**Optional**			
¬**PD**	6	6	7	50.0	46.2	¬TN	27	3	0	90.0	100
¬**PP**	1	5	2	16.7	33.3	¬DS	3	8	3	27.3	50.0
¬**DC**	1	4	2	20.0	33.3	¬BR	2	4	1	33.3	66.7
¬**ST**	2	3	2	40.0	50.0	¬IA	29	1	0	96.7	100
¬**GA**	8	2	5	80.0	61.5	¬CC	0	0	2	0.0	0.0
¬**PC**	5	2	6	71.4	45.5	¬IB	29	1	0	96.7	100
¬**PI**	0	3	1	0.0	0.0	¬DP	28	2	0	93.3	100
¬**WI**	0	0	11	0.0	0.0	¬AC	26	4	0	86.7	100
¬**CN**	1	2	3	33.3	25.0	¬SP	29	1	0	96.7	100
¬**AH**	7	1	7	87.5	50.0	¬SC	30	0	0	100	100
¬**SG**	4	8	2	33.3	66.7	¬SV	30	0	0	100	100
¬**FO**	5	1	2	83.3	71.4	¬DG	30	0	0	100	100
¬**ES**	26	4	0	86.7	100	¬DB	30	0	0	100	100
¬**NB**	23	7	0	76.7	100	¬RV	30	0	0	100	100
¬**CP**	0	1	9	0.0	0.0	¬AD	30	0	0	100	100
¬**CU**	27	3	0	90.0	100						
¬**RD**	2	7	2	22.2	50.0						
¬**IL**	2	1	10	66.7	16.7						
¬**CM**	0	0	4	0.0	0.0						
¬**OG**	2	3	6	40.0	25.0						
¬**RL**	4	4	4	50.0	50.0						
¬**AR**	4	2	8	66.7	33.3						
Summary		**TPs** = 483		**FPs** = 93			**FNs** = 99		**P** = 83.9%		**R** = 83.0%

We use the negation sign (¬) to refer to the absence of an information type; criteria concerning mandatory information types are in bold.

* *Complex text:* Some information types are expressed in complex sentences which can be formulated in a different way than the corresponding GDPR requirement. Though mandatory, this information type is often a part of even a longer sentence. For example, the sentence *"The provider shall process client data only on the written instructions of the client as specified in the services agreement and this addendum includes with regard to transfer of personal data to third country or an international organization as set forth in Article 6(1)..."* describes a complex text containing multiple information types, among which is Process Without Instructions. Thus, considering the entire sentence as the unit of analysis negatively affects both the ML training and the measurement of semantic similarity. Examples on this error include the 11 FNs in ¬**WI**.

* *Similar information types:* The automated classification does not properly distinguish information types that are very similar, leading to false violations. Examples include the FPs and FNs in ¬**CP** and ¬**CM**.

DERECHA (Chapter 4), which solves the same problem, has a precision of 89.1% and recall of 82.4%. *DικAIo* achieves a slightly higher recall with a gain of 0.6 pp at the cost of a drop in precision of ≈5 pp. Our analysis shows that DERECHA performs slightly better in checking most of the criteria concerned with information types that have <100 positive examples in our training set. Conversely, *DικAIo* performs better when there is a sufficient number of examples. ML is not expected to fare well in such situations, thus making the two approaches complementary. Both approaches trigger FPs and FNs which entail the need for manual work by a human analyst to correct them. While too many FPs might require the analyst to review the entire DPA, filtering out FNs can also be effort-intensive if many sentences are found to be about a particular information type.

Though neither approaches provide a perfect precision or recall, such an automation is meant to assist the human analyst in compliance checking. *DικAIo* can be used, for example, to categorize the content of the DPA into a set of pre-defined information types, helping thereby the analyst to quickly browse through the DPA text and decide about its compliance. The time needed by *DικAIo* to analyze the longest DPA in our collection (with 600 sentences) is ≈12 minutes, which is practical since the approach is typically applied offline. In Section 5.8, we reflect on the benefits of using ML in contrast with defining rules in our context.

The answer to RQ3 is that *DικAIo* correctly detects 483 out of 582 genuine violations, while introducing 93 false violations. This corresponds to a precision of 83.9% and a recall of 83.0%. These results are comparable with the ones achieved by DERECHA.

5.8 Discussion

Below, we provide insights regarding the advantages of using conceptual modeling and ML in *DικAIo*.

(1) Representing legal requirements as a conceptual model: Manual work is always needed to reach a precise interpretation of the regulation to represent in an analyzable form. Creating such a representation should typically involve both legal experts and requirements engineers. Legal experts might not be familiar with conceptual modeling but our observation is that they can learn with relative ease. Modeling the GDPR regulation entails defining, in a structured way, the different actors (e.g., controller and processor) as well as their obligations and rights. Creating such conceptual models is a common task in RE since these models can be used by engineers at a later stage in software development [263, 264]. Changes in regulations entail adding or removing classes and relationships in the conceptual model, thus reducing the chances of introducing inconsistencies compared to solutions requiring multiple changes in NL requirements.

(2) Automating compliance checking using ML: In this chapter, we show that *DικAIo* performs on par with DERECHA. We conclude, therefore, that accuracy is not a differentiating factor for selecting the best enabling technology. When automating compliance checking, several other factors must be considered a priori:

- The availability of legal experts: *DικAIo* requires less of their involvement than DERECHA.
- The availability of both qualified annotators and data (DPAs): Any ML approach has such prerequisites but DPAs are readily available and we provide such annotated DPAs in our replication package.
- The budget constraints for developing the automated solution: ML-based solutions require less implementation work since they are based on learning.

5.9 Threats to Validity

Below, we discuss possible threats to the validity of our results and the steps we followed to palliate these threats.

Internal Validity. Bias is the main concern for internal validity. To mitigate this threat, we curated the manual annotation through third-party annotators. The annotators were not exposed to our implementation details at any time. Another potential threat is related to the creation of our conceptual model (presented in Fig. 5.2). To this end, we note that we based our work on substantial experience gained from close collaboration with legal experts. Future improvements may lead to accuracy improvements. Both our model and the basis artifacts are publicly available and thus open to scrutiny.

External Validity. Our evaluation was based on a relatively large dataset with real DPAs from different sources. The results obtained by $D\iota\kappa AIo$ over the evaluation set are reflective of a real scenario since the approach had no exposure to any of the DPAs in the evaluation set during training. This provides some confidence about the generalizability of $D\iota\kappa AIo$. Further examination through user studies would nonetheless be beneficial to improve external validity. Another threat is related to the overfitting of ML classifiers for some information types that had very few learning examples. To reduce the effect of overfitting, we improve the distribution of the minority classes using oversampling techniques and we further combine ML with semantic similarity-based classification.

5.10 Summary

In this chapter, we proposed an automated approach ($D\iota\kappa AIo$) for compliance checking of DPAs against GDPR. $D\iota\kappa AIo$ relies on a comprehensive conceptual model that we created based on two different representations introduced in our previous work [24, 26]. By leveraging a combination of NLP and ML, $D\iota\kappa AIo$ then automatically classifies the content in DPAs according to information types in the conceptual model. The resulting classification is used to provide recommendations about the compliance of the DPAs. We curated a total of 180 real DPAs, with the majority of the annotation work performed by third-party annotators. Over an evaluation set of 30 real DPAs, $D\iota\kappa AIo$ can detect GDPR violations in DPAs with a precision of 83.9% and a recall of 83.0%. We empirically compare the performance of $D\iota\kappa AIo$ with `DERECHA`, introduced in Chapter 4. `DERECHA` exclusively relies on rules over semantic analysis of DPAs' textual content and further represents GDPR requirements in NL. Our empirical evaluation demonstrates that $D\iota\kappa AIo$ yields comparable performance compared with rule-based approach `DERECHA`. While accuracy does not seem to be a differentiating factor, we discussed other factors to consider when selecting an automated solution in the context of regulatory compliance. These factors notably include the availability of (annotated) data to develop accurate ML classifiers, the relative ease of adapting the solution to future changes in the regulation, and the involvement of legal experts in creating the representation of the legal requirements and updating them.

In the future, we plan to conduct user studies to assess the usefulness of $D\iota\kappa AIo$ in practice, particularly the time and effort that such an automation saves.

Chapter 6

Conclusion

This final chapter wraps up the dissertation by reviewing the work presented in the preceding chapters, highlighting the key contributions and exploring further directions.

6.1 Summary

In this dissertation, we presented three solutions aimed at providing automated support for compliance checking of privacy policies (PPs) and data processing agreements (DPAs) against the General Data Protection Regulation (GDPR). Our solutions leverage machine learning (ML) and natural language processing (NLP), two major sub-fields of artificial intelligence (AI). We empirically evaluated the solutions on real PPs and DPAs obtained from our industrial partners (Linklaters LLP) and from online sources. The raw content of the PPs and DPAs in our collection was analyzed and labeled by third-party annotators with a strong background in law. We released our solutions under open-source licenses. Below, we briefly outline the main contributions made in this dissertation.

In Chapter 3, we proposed an AI-enabled approach ($CompA\iota$) for checking the compliance of PPs according to GDPR. We first developed a conceptual model aimed at providing a thorough characterization of the content of PPs. Based on this conceptual model, we devised criteria describing how a privacy policy should be checked for compliance against GDPR. Using NLP and ML, we then developed automated support for classifying the content of a given PP and thus identifying the information types necessary for checking its compliance. To develop and evaluate $CompA\iota$, we curated a total of 234 real PPs. On an evaluation set of 48 PPs, our compliance checking approach achieved an average precision of 93% and a recall of 90%. Compared to an intuitive automated solution that uses keyword search, our AI-based approach leads to a significant improvement in precision and recall of 24.5 and 38 percentage points, respectively. We further assessed the usefulness of $CompA\iota$ in practice through an interview survey with two legal experts from our collaborating partner, Linklaters LLP. The overall feedback of the legal experts shows that $CompA\iota$ efficiently generates useful information that can be leveraged by the human analyst during the compliance checking process to optimize their time.

In Chapter 4, we proposed an NLP-based solution (`DERECHA`) for checking the compliance of DPAs against GDPR. In close collaboration with legal experts, we extracted and documented DPA-related requirements as "shall" requirements. On top of these requirements, we defined an intermediate representation using semantic frames (SFs) to describe the semantics of key phrases in these requirements. Using the same semantic analysis, we developed DERECHA to automatically identify SF-based representations for the text in a given DPA. By comparing the two representations, DERECHA concludes whether the DPA is compliant with GDPR. Over an evaluation set of 30 real DPAs, DERECHA achieves an average accuracy of 84.6%, a precision of 89.1% and a recall of 82.4%. Compared to a baseline that relies on an off-the-shelf NLP toolkit, our solution provides a significant gain with an average of ≈20 percentage points in accuracy. We also conducted an interview survey with two legal experts from Linklaters LLP to assess the usefulness of DERECHA in practice. We received positive feedback from the legal experts similar to the one we received for $CompA\iota$.

In Chapter 5, we proposed an automated approach ($D\iota\kappa AIo$) for checking the compliance of DPAs against GDPR. $D\iota\kappa AIo$ relies on a comprehensive conceptual model that we created based on two different representations previously introduced in the RE literature [24,26]. By leveraging a combination of NLP and ML, $D\iota\kappa AIo$ then automatically classifies the content in a given DPA according to information types in the conceptual model. The resulting classification is used to provide recommendations about the compliance of the DPA. To develop and evaluate $D\iota\kappa AIo$, we curated a total of 180 real DPAs. Over an evaluation set of 30 real DPAs, $D\iota\kappa AIo$ can detect GDPR violations in DPAs with a precision of 83.9% and a recall of 83.0%. We empirically compare the performance of $D\iota\kappa AIo$ with DERECHA. Our evaluation demonstrates that the two approaches yield comparable performance. While accuracy does not seem to be a differentiating factor, we discussed other factors to consider when selecting an automated solution in the context of regulatory compliance. These factors notably include availability of (annotated) data to develop accurate ML classifiers, the adaptability of the solution to future changes in the regulation, and the level of involvement of legal experts in creating the representation of the legal requirements and updating them.

6.2 Future work

In this dissertation, we have presented three solutions aimed at providing automated support for GDPR compliance checking of two types of legal documents, namely PPs and DPAs. The immediate future direction to the work presented in this dissertation is to integrate these solutions into a single toolkit that can be customized towards the needs of the human analyst. For instance, whether the analysis is needed at phrasal level or sentence level. Other features can be as well built into the toolkit for better usability. We highlight two example features, namely the explainability of the output and the adaptation towards regulation change. Explainability in our context is significant, since the decision and consequences of breaching GDPR highly depend on the understandability of the outcome by the human analyst. To this end, we can extend the tool to provide decisions with probabilities instead of binary decisions about the compliance status of a legal document. Explainability helps analysts effectively review the output of the tool and take decisions. Human oversight is a key requirement indicated in the Ethics Guidelines for Trustworthy AI which is released by the European Commission [265]. As for the adaptation of the regulatory change, we discussed with the legal experts an additional feature that allows human analysts to create the compliance criteria in the toolkit. While the back-end automation would be still dependent on the information types in the conceptual model, creating compliance criteria dynamically address the cases when some information types are no longer required or the way they are combined has changed in the regulation. This feature provides a step towards addressing the regulatory change.

Bibliography

[1] Paul N. Otto and Annie I. Anton. Addressing legal requirements in requirements engineering. In *15th IEEE International Requirements Engineering Conference*, pages 5–14, 2007.

[2] Alexandros Bousdekis, Katerina Lepenioti, Dimitris Apostolou, and Gregoris Mentzas. A review of data-driven decision-making methods for industry 4.0 maintenance applications. *Electronics*, 10(7), 2021.

[3] Santiago del Rey, Silverio Martínez-Fernández, and Antonio Salmerón. Bayesian network analysis of software logs for data-driven software maintenance. *IET Software*, 2023.

[4] Chen Yang, Shulin Lan, Lihui Wang, Weiming Shen, and George GQ Huang. Big data driven edge-cloud collaboration architecture for cloud manufacturing: a software defined perspective. *IEEE access*, 8, 2020.

[5] Fatih Gurcan and Nergiz Ercil Cagiltay. Big data software engineering: Analysis of knowledge domains and skill sets using lda-based topic modeling. *IEEE access*, 7, 2019.

[6] Mohsen Azimi, Armin Dadras Eslamlou, and Gokhan Pekcan. Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. *Sensors*, 20(10), 2020.

[7] ElMouatez Billah Karbab and Mourad Debbabi. Maldy: Portable, data-driven malware detection using natural language processing and machine learning techniques on behavioral analysis reports. *Digital Investigation*, 28, 2019.

[8] European Union. General data protection regulation. Accessed Nov. 07, 2021 [Online].

[9] European Data Protection Board. 1.2 billion euro fine for facebook as a result of edpb binding decision. Accessed Jul. 01, 2023 [Online].

[10] Klaus Pohl. *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated, 2010.

[11] Mitra Bokaei Hosseini, John Heaps, Rocky Slavin, Jianwei Niu, and Travis Breaux. Ambiguity and generality in natural language privacy policies. In *2021 IEEE 29th International Requirements Engineering Conference (RE)*, 2021.

[12] Sallam Abualhaija, Chetan Arora, Amin Sleimi, and Lionel Briand. Automated question answering for improved understanding of compliance requirements: A multi-document study. In *In Proceedings of the 30th IEEE International Requirements Engineering Conference, Melbourne, Australia 15-19 August 2022*, 2022.

[13] Travis Breaux and Thomas Norton. Legal accountability as software quality: A u.s. data processing perspective. In *In Proceedings of the 30th IEEE International Requirements Engineering Conference, Melbourne, Australia 15-19 August 2022*, 2022.

[14] Eunju Park, Sungjun Han, Hogon Bae, Raekyung Kim, Seungjae Lee, Daejune Lim, and Hankyu Lim. Development of automatic evaluation tool for mobile accessibility for android application. In *2019 International Conference on Systems of Collaboration Big Data, Internet of Things & Security (SysCoBIoTS)*, pages 1–6, 2019.

[15] Jaime Benjumea, Enrique Dorronzoro, Jorge Ropero, Octavio Rivera-Romero, and Alejandro Carrasco. Privacy in mobile health applications for breast cancer patients. In *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, pages 634–639, 2019.

[16] Luca Verderame, Davide Caputo, Andrea Romdhana, and Alessio Merlo. On the (un)reliability of privacy policies in android apps. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2020.

[17] Danny S. Guaman, Jose M. Del Alamo, and Julio C. Caiza. Gdpr compliance assessment for cross-border personal data transfers in android apps. *IEEE Access*, 9:15961–15982, 2021.

[18] Miguel Cozar, David Rodriguez, Jose M. Del Alamo, and Danny Guaman. Reliability of ip geolocation services for assessing the compliance of international data transfers. In *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 181–185, 2022.

[19] Jaspreet Bhatia, Travis D. Breaux, and Florian Schaub. Mining privacy goals from privacy policies using hybridized task recomposition. *ACM Transaction on Software Engineering and Methodology*, 25(3), 2016.

[20] Welderufael B. Tesfay, Peter Hofmann, Toru Nakamura, Shinsaku Kiyomoto, and Jetzabel Serna. Privacyguide: Towards an implementation of the EU GDPR on internet privacy policy evaluation. In *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics, IWSPA@CODASPY 2018, Tempe, AZ, USA, March 19-21, 2018*, 2018.

[21] Jaspreet Bhatia, Morgan C. Evans, and Travis D. Breaux. Identifying incompleteness in privacy policy goals using semantic frames. *Requirements Engineering*, 24(3), 2019.

[22] Thomas Linden, Rishabh Khandelwal, Hamza Harkous, and Kassem Fawaz. The privacy policy landscape after the gdpr. *Proceedings on Privacy Enhancing Technologies*, 2020(1):47–64, 2020.

[23] Orlando Amaral, Sallam Abualhaija, Damiano Torre, Mehrdad Sabetzadeh, and Lionel Briand. AI-enabled automation for completeness checking of privacy policies. *IEEE Transactions on Software Engineering*, 2021.

[24] Orlando Amaral, Muhammad Ilyas Azeem, Sallam Abualhaija, and Lionel C Briand. Nlp-based automated compliance checking of data processing agreements against gdpr. *IEEE Transactions on Software Engineering*, pages 1–23, 2023.

[25] Orlando Amaral, Sallam Abualhaija, and Lionel C. Briand. Ml-based compliance verification of data processing agreements against gdpr. In *2023 IEEE 31th International Requirements Engineering Conference (RE)*, 2023.

[26] Orlando Amaral, Sallam Abualhaija, Mehrdad Sabetzadeh, and Lionel Briand. A model-based conceptualization of requirements for compliance checking of data processing against GDPR. In *29th IEEE International Requirements Engineering Conference Workshops*, 2021.

[27] Julia Hirschberg and Christopher D Manning. Advances in natural language processing. *Science*, 349(6245), 2015.

[28] D. Jurafsky and J. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall, 2nd edition, 2009.

[29] Mário Rodrigues, António Teixeira, et al. *Advanced applications of natural language processing for performing information extraction.* Springer, 2015.

[30] Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics, 1999.

[31] Jeffrey EF Friedl. *Mastering regular expressions.* " O'Reilly Media, Inc.", 2006.

[32] Martha Palmer, Daniel Gildea, and Nianwen Xue. *Semantic role labeling.* Morgan & Claypool Publishers, 2011.

[33] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245), 2015.

[34] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, 4th edition, 2016.

[35] Amanpreet Singh, Narina Thakur, and Aakanksha Sharma. A review of supervised machine learning algorithms. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016.

[36] Charu C. Aggarwal. *Machine Learning for Text.* Springer Publishing Company, Incorporated, 1st edition, 2018.

[37] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1986.

[38] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1. IEEE, 1995.

[39] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3), 1995.

[40] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2), 1958.

[41] Suresh Balakrishnama and Aravind Ganapathiraju. Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, 18(1998):1–8, 1998.

[42] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61, 2015.

[43] Monisha Kanakaraj and Ram Mohana Reddy Guddeti. Performance analysis of ensemble methods on twitter sentiment analysis using nlp techniques. In *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, 2015.

[44] Md. Rakibul Hasan, Maisha Maliha, and M. Arifuzzaman. Sentiment analysis with nlp on twitter data. In *2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*, 2019.

[45] Poonam Gupta and Vishal Gupta. A survey of text question answering techniques. *International Journal of Computer Applications*, 53(4), 2012.

[46] Anna Rogers, Matt Gardner, and Isabelle Augenstein. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension. *ACM Computing Surveys*, 55(10):1–45, 2023.

[47] Nikita Munot and Sharvari S Govilkar. Comparative study of text summarization methods. *International Journal of Computer Applications*, 102(12), 2014.

[48] Rahul, Surabhi Adhikari, and Monika. Nlp based machine learning approaches for text summarization. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 2020.

[49] Zhaorong Zong and Changchun Hong. On application of natural language processing in machine translation. In *2018 3rd International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, 2018.

[50] Kai Jiang and Xi Lu. Natural language processing and its applications in machine translation: A diachronic review. In *2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*, 2020.

[51] Huashan Pan, Xin Yan, Zhengtao Yu, and Jianyi Guo. A khmer named entity recognition method by fusing language characteristics. In *The 26th Chinese Control and Decision Conference (2014 CCDC)*, 2014.

[52] Peng Sun, Xuezhen Yang, Xiaobing Zhao, and Zhijuan Wang. An overview of named entity recognition. In *2018 International Conference on Asian Language Processing (IALP)*, 2018.

[53] Jingjing Cai, Jianping Li, Wei Li, and Ji Wang. Deeplearning model used in text classification. In *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2018.

[54] Zhen Li, Xiting Wang, Weikai Yang, Jing Wu, Zhengyan Zhang, Zhiyuan Liu, Maosong Sun, Hui Zhang, and Shixia Liu. A unified understanding of deep nlp models for text classification. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4980–4994, 2022.

[55] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008.

[56] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge, 2008.

[57] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[58] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, 2014.

[59] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv:1301.3781*, 2013.

[60] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[61] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[62] Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark Carman. Are word embedding-based features useful for sarcasm detection? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[63] Jiaqi Mu and Pramod Viswanath. *All-but-the-Top: Simple and Effective Postprocessing for Word Representations*. International Conference on Learning Representations, 2018. Accessed on: Nov. 07, 2021.

[64] Liang-Chih Yu, Jin Wang, K Robert Lai, and Xuejie Zhang. Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017.

[65] Debanjan Ghosh, Weiwei Guo, and Smaranda Muresan. Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *proceedings of the 2015 conference on empirical methods in natural language processing*, 2015.

[66] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 2015.

[67] Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL*, pages 529–535, 2018.

[68] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics, 2014.

[69] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, 2014.

[70] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, 2010.

[71] William Blacoe and Mirella Lapata. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, 2012.

[72] Xunjie Zhu, Tingfeng Li, and Gerard De Melo. Exploring semantic properties of sentence embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018.

[73] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. *Towards Universal Paraphrastic Sentence Embeddings*. International Conference on Learning Representations, 2016. Accessed on: Nov. 07, 2021.

[74] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 2227–2237. Association for Computational Linguistics, 2018.

[75] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. *Improving language understanding with unsupervised learning*. OpenAI Blog, 2018. Accessed on: Nov. 07, 2021.

[76] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics, 2019.

[77] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[78] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1), 2019.

[79] Randell Rasiman, Fabiano Dalpiaz, and Sergio España. How effective is automated trace link recovery in model-driven development? In *Requirements Engineering: Foundation for Software Quality*, pages 35–51. Springer International Publishing, 2022.

[80] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 2002.

[81] Nitesh V Chawla. Data mining for imbalanced datasets: An overview. *Data mining and knowledge discovery handbook*, 2009.

[82] Charith Perera, Mahmoud Barhamgi, Arosha K. Bandara, Muhammad Ajmal, Blaine A. Price, and Bashar Nuseibeh. Designing privacy-aware internet of things applications. *Inf. Sci.*, 512(1), 2020.

[83] Damiano Torre, Ghanem Soltana, Mehrdad Sabetzadeh, Lionel C. Briand, Yuri Auffinger, and Peter Goes. Using models to enable compliance checking against the GDPR: an experience report. In *22nd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS 2019, Munich, Germany, September 15-20, 2019*, 2019.

[84] Damiano Torre, Mauricio Alferez, Ghanem Soltana, Mehrdad Sabetzadeh, and Lionel C. Briand. Modeling data protection and privacy: Application and experience with GDPR. *Software and Systems Modeling*, 2021 (In Press).

[85] V. Ayala-Rivera and L. Pasquale. The grace period has ended: An approach to operationalize GDPR requirements. In *Proceedings of 31st IEEE International Conference on Requirements Engineering (RE'18)*, 2018.

[86] João Caramujo, Alberto Rodrigues da Silva, Shaghayegh Monfared, André Ribeiro, Pável Calado, and Travis Breaux. RSL-IL4Privacy: A domain-specific language for the rigorous specification of privacy policies. *Requirements Engineering*, 24(1):1–26, 2019.

[87] Jaspreet Bhatia and Travis D. Breaux. Semantic incompleteness in privacy policy goals. In *26th IEEE International Requirements Engineering Conference, RE 2018, Banff, AB, Canada, August 20-24, 2018*, 2018.

[88] Rocky Slavin, Xiaoyin Wang, Mitra Bokaei Hosseini, James Hester, Ram Krishnan, Jaspreet Bhatia, Travis D. Breaux, and Jianwei Niu. Toward a framework for detecting privacy policy violations in android application code. In *Proceedings of the 38th International Conference on Software Engineering, ICSE 2016, Austin, TX, USA, May 14-22, 2016*, 2016.

[89] M. Fan, L. Yu, S. Chen, H. Zhou, X. Luo, S. Li, Y. Liu, J. Liu, and T. Liu. An empirical evaluation of gdpr compliance violations in android mhealth apps. In *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*, 2020.

[90] Damiano Torre, Sallam Abualhaija, Mehrdad Sabetzadeh, Lionel C. Briand, Katrien Baetens, Peter Goes, and Sylvie Forastier. An ai-assisted approach for checking the completeness of privacy policies against GDPR. In *28th IEEE International Requirements Engineering Conference, RE 2020, Zurich, Switzerland, August 31 - September 4, 2020*, 2020.

[91] Orlando Amaral, Damiano Torre, Sallam Abualhaija, Mehrdad Sabetzadeh, and Lionel C. Briand. *Glossary and Completeness Criteria traceability to the GDPR articles*. Available at `https://figshare.com/s/2e2a0777202164c4d712`, July 2023.

[92] J. Saldana. *The Coding Manual for Qualitative Researchers*. SAGE Publishing, 2016.

[93] European Commission. Article 29 working party - guidelines on data protection officers (dpos). Accessed Nov. 07, 2021 [Online].

[94] Ghanem Soltana, Nicolas Sannier, Mehrdad Sabetzadeh, and Lionel C. Briand. Model-based simulation of legal policies: Framework, tool support, and validation. *Software & Systems Modeling*, 17(3):851–883, 2018.

[95] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[96] Laura Michaelis. Word meaning, sentence meaning, and syntactic meaning. *Cognitive approaches to lexical semantics*, 23:163–210, 2003.

[97] Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2*, 2012.

[98] Alexandros Komninos and Suresh Manandhar. Dependency based embeddings for sentence classification tasks. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016.

[99] Richard Eckart de Castilho and Iryna Gurevych. A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT'14)*, 2014.

[100] Eclipse Deeplearning4j Development Team. Deeplearning4j: Open-source distributed deep learning for the jvm, apache software foundation license 2.0, 2020. last accessed: January 2020.

[101] Jane Huffman Hayes, Wenbin Li, and Mona Rahimi. Weka meets tracelab: Toward convenient classification: Machine learning for requirements engineering problems: A position paper. In *Artificial Intelligence for Requirements Engineering (AIRE), 2014 IEEE 1st International Workshop on*, 2014.

[102] Frank Eibe, MA Hall, and IH Witten. The weka workbench. online appendix for data mining: Practical machine learning tools and techniques. *Morgan Kaufmann*, 2016.

[103] Orlando Amaral, Sallam Abualhaija, Damiano Torre, Mehrdad Sabetzadeh, and Lionel C. Briand. “compliance checking for privacy policies using artificial intelligence (compai)”, 2023. Available at `https://figshare.com/articles/online_resource/CompAI/23676069`, July 2023.

[104] ALFI. "association of the luxembourg fund industry - 946 member funds", 2019. last accessed: March 2019.

[105] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement (EPM)*, 20(1):37–46, 1960.

[106] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia Medica (BM)*, 22(3):276–282, 2012.

[107] Evangelia Vanezi, Georgia M. Kapitsaki, Dimitrios Kouzapas, Anna Philippou, and George A. Papadopoulos. Diálogop - A language and a graphical tool for formally defining GDPR purposes. In *Proceedings of the 2020 Research Challenges in Information Science - 14th International Conference, RCIS*, volume 385, pages 569–575. Springer, 2020.

[108] Pille Pullonen, Jake Tom, Raimundas Matulevicius, and Aivo Toots. Privacy-enhanced BPMN: Enabling data privacy analysis in business processes models. *Software & Systems Modeling*, 18(6), 2019.

[109] N. V. Narendra Kumar and R. K. Shyamasundar. Realizing purpose-based privacy policies succinctly via information-flow labels. In *2014 IEEE Fourth International Conference on Big Data and Cloud Computing, BDCloud 2014, Sydney, Australia, December 3-5, 2014*, 2014.

[110] David Sánchez, Alexandre Viejo, and Montserrat Batet. Automatic assessment of privacy policies under the gdpr. *Applied Sciences*, 11(4), 2021.

[111] Najmeh Mousavi Nejad, Pablo Jabat, Rostislav Nedelchev, Simon Scerri, and Damien Graux. Establishing a strong baseline for privacy policy classification. In Marko Hölbl, Kai Rannenberg, and Tatjana Welzer, editors, *ICT Systems Security and Privacy Protection*, 2020.

[112] Fei Liu, Rohan Ramanath, Norman M. Sadeh, and Noah A. Smith. A step towards usable privacy policy: Automatic alignment of privacy statements. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, 2014.

[113] Shomir Wilson, Florian Schaub, Rohan Ramanath, Norman M. Sadeh, Fei Liu, Noah A. Smith, and Frederick Liu. Crowdsourcing annotations for websites' privacy policies: Can it really work? In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, 2016.

[114] Michele Guerriero, Damian Andrew Tamburri, and Elisabetta Di Nitto. Defining, enforcing and checking privacy policies in data-intensive applications. In *Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems, SEAMS@ICSE 2018, Gothenburg, Sweden, May 28-29, 2018*, 2018.

[115] Marco Lippi, Przemysław Pałka, Giuseppe Contissa, Francesca Lagioia, Hans-Wolfgang Micklitz, Giovanni Sartor, and Paolo Torroni. Claudette: an automated detector of potentially unfair clauses in online terms of service. *Artificial Intelligence and Law*, 27(2), 2019.

[116] Muneera Bano, Didar Zowghi, and Chetan Arora. Requirements, politics, or individualism: What drives the success of covid-19 contact-tracing apps? *IEEE Software*, 38(1):7–12, 2021.

[117] Majid Hatamian, Samuel Wairimu, Nurul Momen, and Lothar Fritsch. A privacy and security analysis of early-deployed COVID-19 contact tracing android apps. *Empir. Softw. Eng.*, 26(3), 2021.

[118] Sophia Kununka, Nikolay Mehandjiev, and Pedro Sampaio. A comparative study of android and ios mobile applications' data handling practices versus compliance to privacy policy. In *Privacy and Identity Management. The Smart Revolution - 12th IFIP WG 9.2, 9.5, 9.6/11.7, 11.6/SIG 9.2.2 International Summer School, Ispra, Italy, September 4-8, 2017, Revised Selected Papers*, 2017.

[119] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[120] Nicole Zhan, Stefan Sarkadi, Natalia Criado Pacheco, and Jose Such. A model for governing information sharing in smart assistants. In *AAAI/ACM Conference on AI, Ethics, and Society*. AAAI Press, 2022.

[121] Álvaro Feal, Julien Gamba, Narseo Vallina-Rodriguez, Primal Wijesekera, Joel Reardon, Serge Egelman, Juan Tapiador, et al. Don't accept candies from strangers: An analysis of third-party sdks. In *Computers, Privacy and Data Protection Conference*, 2020.

[122] European Union. The GDPR: New opportunities, new obligations. *Justice and Consumers*, 2018.

[123] Nick Pantlin, Claire Wiseman, and Miriam Everett. Supply chain arrangements: The abc to gdpr compliance—a spotlight on emerging market practice in supplier contracts in light of the gdpr. *Computer law & Security review*, 34(4):881–885, 2018.

[124] Collin F Baker. Framenet: A knowledge base for natural language processing. In *Proceedings of Frame Semantics in NLP: A workshop in honor of Chuck Fillmore (1929-2014)*, 2014.

[125] Yinglin Wang. Semantic information extraction for software requirements using semantic role labeling. In *2015 IEEE International Conference on Progress in Informatics and Computing (PIC)*, 2015.

[126] Charles J Fillmore and Collin F Baker. Frame semantics for text understanding. In *Proceedings of WordNet and Other Lexical Resources Workshop, NAACL*, volume 6, 2001.

[127] Charles J Fillmore et al. Frame semantics. *Cognitive linguistics: Basic readings*, 34:373–400, 2006.

[128] Yuk Wah Wong and Raymond Mooney. Learning for semantic parsing with statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, 2006.

[129] Ding Liu and Daniel Gildea. Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, 2010.

[130] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90, 1998.

[131] Paul R Kingsbury and Martha Palmer. From treebank to propbank. In *LREC*, pages 1989–1993, 2002.

[132] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288, 2002.

[133] Zuchao Li, Hai Zhao, Shexia He, and Jiaxun Cai. Syntax role for neural semantic role labeling. *Computational Linguistics*, 47(3):529–574, 2021.

[134] Lluís Màrquez, Xavier Carreras, Kenneth C Litkowski, and Suzanne Stevenson. Semantic role labeling: an introduction to the special issue. *Computational linguistics*, 34(2):145–159, 2008.

[135] Chen Shi, Shujie Liu, Shuo Ren, Shi Feng, Mu Li, Ming Zhou, Xu Sun, and Houfeng Wang. Knowledge-based semantic embedding for machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2245–2254, 2016.

[136] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544, 2013.

[137] Dan Shen and Mirella Lapata. Using semantic roles to improve question answering. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 12–21, 2007.

[138] Yankai Lin, Zhiyuan Liu, and Maosong Sun. Neural relation extraction with multi-lingual attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 34–43, 2017.

[139] Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255, 2019.

[140] Shumin Wu and Martha Palmer. Semantic mapping using automatic word alignment and semantic role labeling. In *Proceedings of Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 21–30, 2011.

[141] Kristian Woodsend and Mirella Lapata. Text rewriting improves semantic role labeling. *Journal of Artificial Intelligence Research*, 51:133–164, 2014.

[142] Jeremy Howard and Sebastian Ruder. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146, 2018.

[143] Nanjiang Jiang and Marie-Catherine de Marneffe. Evaluating bert for natural language inference: A case study on the commitmentbank. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing*, pages 6086–6091, 2019.

[144] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. Pretrained transformers for text ranking: Bert and beyond. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1154–1156, 2021.

[145] Amin Sleimi, Nicolas Sannier, Mehrdad Sabetzadeh, Lionel C. Briand, and John Dann. Automated extraction of semantic legal metadata using natural language processing. In *Proceedings of 26th IEEE International Requirements Engineering Conference*, 2018.

[146] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[147] George A Miller. *WordNet: An electronic lexical database*. MIT Press, 1998.

[148] Anna Korhonen and Ted Briscoe. Extended lexical-semantic classification of english verbs. In *Proceedings of the Computational Lexical Semantics Workshop at Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies*, pages 38–45, 2004.

[149] Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. Extending verbnet with novel verb classes. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, 2006.

[150] Eneko Agirre and Philip Edmonds. *Word sense disambiguation: Algorithms and applications*, volume 33. Springer Science & Business Media, 2007.

[151] Karin Kipper Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon.* University of Pennsylvania, 2005.

[152] Roberto Navigli, Kenneth C Litkowski, and Orin Hargraves. Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 30–35, 2007.

[153] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26, 1986.

[154] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, page 133–138. Association for Computational Linguistics, 1994.

[155] Courtney D Corley and Rada Mihalcea. Measuring the semantic similarity of texts. In *Proceedings of the Association for Computational Linguistics workshop on empirical modeling of semantic equivalence and entailment*, pages 13–18, 2005.

[156] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, pages 100–108, 2010.

[157] Vered Shwartz and Ido Dagan. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419, 2019.

[158] Ted Pedersen, Siddharth Patwardhan, Jason Michelizzi, et al. Wordnet:: Similarity-measuring the relatedness of concepts. In *AAAI*, volume 4, pages 25–29, 2004.

[159] SC Cahyono. Comparison of document similarity measurements in scientific writing using jaro-winkler distance method and paragraph vector method. In *IOP Conference Series: Materials Science and Engineering*, volume 662, page 052016. IOP Publishing, 2019.

[160] Muhamad Arief Yulianto and Nurhasanah Nurhasanah. The hybrid of jaro-winkler and rabin-karp algorithm in detecting indonesian text similarity. *Jurnal Online Informatika*, 6(1):88–95, 2021.

[161] R Aduni Sulaiman, Dayang NA Jawawi, and Shahliza Abdul Halim. A dissimilarity with dice-jaro-winkler test case prioritization approach for model-based testing in software product line. *KSII Transactions on Internet and Information Systems (TIIS)*, 15(3):932–951, 2021.

[162] Apache POI. The java api for microsoft documents.

[163] Apache opennlp, 2022. last accessed: February 2022.

[164] Mate tools, 2022. last accessed: February 2022.

[165] Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Stanford University, 2008.

[166] extJWNL. Extended java wordnet library, 2022. last accessed: February 2022.

[167] JVerbnet. Java library for verbnet, 2022. last accessed: February 2022.

[168] Matthew A Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420, 1989.

[169] WS4J. Wordnet similarity for java, 2022. last accessed: February 2022.

[170] Orlando Amaral, Muhammad Ilyas Azeem, Sallam Abualhaija, and Lionel C. Briand. "dpa semantic frame-based compliance checking against gdpr (derecha)", 2023. Available at `https://figshare.com/articles/online_resource/DERECHA/23676216`, July 2023.

[171] Anthony J Viera, Joanne M Garrett, et al. Understanding interobserver agreement: the kappa statistic. *Fam med*, 37(5):360–363, 2005.

[172] Travis D. Breaux, Annie I. Antón, Kent Boucher, and Merlin Dorfman. Legal requirements, compliance and practice: An industry case study in accessibility. In *2008 16th IEEE International Requirements Engineering Conference*, pages 43–52, 2008.

[173] Sepideh Ghanavati, Daniel Amyot, and Liam Peyton. A systematic review of goal-oriented requirements management frameworks for business process compliance. In *2011 Fourth International Workshop on Requirements Engineering and Law*, pages 25–34, 2011.

[174] Aaron K. Massey, Richard L. Rutledge, Annie I. Antón, and Peter P. Swire. Identifying and classifying ambiguity for regulatory requirements. In *2014 IEEE 22nd International Requirements Engineering Conference*, pages 83–92, 2014.

[175] S. Ghanavati, A. Rifaut, E. Dubois, and D. Amyot. Goal-oriented compliance with multiple regulations. In *Proceedings of 22nd IEEE International Conference on Requirements Engineering (RE'14)*, 2014.

[176] Morgan C. Evans, Jaspreet Bhatia, Sudarshan Wadkar, and Travis D. Breaux. An evaluation of constituency-based hyponymy extraction from privacy policies. In *2017 IEEE 25th International Requirements Engineering Conference*, pages 312–321, 2017.

[177] Okhaide Akhigbe, Daniel Amyot, Gregory Richards, and Lysanne Lessard. Gorim: a model-driven method for enhancing regulatory intelligence. *Software and Systems Modeling*, pages 1–29, 2021.

[178] Martin Horák, Václav Stupka, and Martin Husák. Gdpr compliance in cybersecurity software: a case study of dpia in information sharing platform. In *Proceedings of the 14th international conference on availability, reliability and security*, pages 1–8, 2019.

[179] Nora Ni Loideain. A port in the data-sharing storm: the gdpr and the internet of things. *Journal of Cyber Policy*, 4(2):178–196, 2019.

[180] Tobias Urban, Dennis Tatang, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. Measuring the impact of the gdpr on data sharing in ad networks. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pages 222–235, 2020.

[181] Abdulrahman Alhazmi and Nalin Asanka Gamagedara Arachchilage. I'm all ears! listening to software developers on putting gdpr principles into software development practice. *Personal and Ubiquitous Computing*, 25(5):879–892, 2021.

[182] Abdulrahman Alhazmi and Nalin AG Arachchilage. A serious game design framework for software developers to put gdpr into practice. In *The 16th International Conference on Availability, Reliability and Security*, pages 1–6, 2021.

[183] Supreeth Shastri, Melissa Wasserman, and Vijay Chidambaram. Gdpr anti-patterns. *Communications of the ACM*, 64(2):59–65, 2021.

[184] Alberto Siena, Anna Perini, Angelo Susi, and John Mylopoulos. A meta-model for modelling law-compliant requirements. In *2009 Second International Workshop on Requirements Engineering and Law*, pages 45–51, 2009.

[185] Jeremy C. Maxwell and Annie I. Anton. Developing production rule models to aid in acquiring requirements from legal texts. In *2009 17th IEEE International Requirements Engineering Conference*, pages 101–110, 2009.

[186] Sepideh Ghanavati, Daniel Amyot, and Liam Peyton. Compliance analysis based on a goal-oriented requirement language evaluation methodology. In *2009 17th IEEE International Requirements Engineering Conference*, pages 133–142. IEEE, 2009.

[187] S. Ingolfo, A. Siena, and J. Mylopoulos. Nòmos 3: Reasoning about regulatory compliance of requirements. In *Proceedings of 22nd IEEE International Requirements Engineering Conference*, 2014.

[188] Ghanem Soltana, Elizabeta Fourneret, Morayo Adedjouma, Mehrdad Sabetzadeh, and Lionel Briand. Using UML for modeling procedural legal rules: Approach and a study of luxembourg's tax law. In *International Conference on Model Driven Engineering Languages and Systems*, pages 450–466. Springer, 2014.

[189] Nicola Zeni, Nadzeya Kiyavitskaya, Luisa Mich, James R. Cordy, and John Mylopoulos. GaiusT: Supporting the extraction of rights and obligations for regulatory compliance. *Requirements Engineering*, 20(1), 2015.

[190] Travis D. Breaux, Matthew W. Vail, and Annie I. Anton. Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In *14th IEEE International Requirements Engineering Conference*, pages 49–58, 2006.

[191] Travis D Breaux and Annie I Antón. A systematic method for acquiring regulatory requirements: A frame-based approach. *6th International Workshop on Requirements for High Assurance Systems*, 2007.

[192] L. Thomas van Binsbergen, Lu-Chi Liu, Robert van Doesburg, and Tom van Engers. Eflint: A domain-specific language for executable norm specifications. In *Proceedings of the 19th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences*. Association for Computing Machinery, 2020.

[193] Xin Xu and Hubo Cai. Ontology and rule-based natural language processing approach for interpreting textual regulations on underground utility infrastructure. *Advanced Engineering Informatics*, 48:101288, 2021.

[194] Atif Mashkoor, Michael Leuschel, and Alexander Egyed. Validation obligations: a novel approach to check compliance between requirements and their formal specification. In *2021 IEEE/ACM 43rd International Conference on Software Engineering: New Ideas and Emerging Results*, pages 1–5. IEEE, 2021.

[195] Shukun Tokas, Olaf Owe, and Toktam Ramezanifarkhani. Static checking of gdpr-related privacy compliance for object-oriented distributed systems. *Journal of Logical and Algebraic Methods in Programming*, 125:100733, 2022.

[196] Pattaraporn Sangaroonsilp, Hoa Khanh Dam, Morakot Choetkiertikul, Chaiyong Ragkhitwetsagul, and Aditya Ghose. A taxonomy for mining and classifying privacy requirements in issue reports. *arXiv preprint arXiv:2101.01298*, 2021.

[197] Pohl Klaus and Rupp Chris. *Requirements Engineering Fundamentals*. Rocky Nook, 1st edition, 2011.

[198] Alistair Mavin, Philip Wilkinson, Adrian Harwood, and Mark Novak. Easy approach to requirements syntax (EARS). In *Proceedings of the 17th IEEE International Requirements Engineering Conference*, 2009.

[199] Olga A Krapivkina. Semantics of the verb shall in legal discourse. *Jezikoslovlje*, 18(2.):305–317, 2017.

[200] Jeremy C Maxwell and Annie I Antón. The production rule framework: developing a canonical set of software requirements for compliance with law. In *proceedings of the 1st ACM International Health Informatics Symposium*, pages 629–636, 2010.

[201] Guido Boella, Luigi Di Caro, Llio Humphreys, Livio Robaldo, Piercarlo Rossi, and Leendert van der Torre. Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law. *Artificial Intelligence and Law*, 24(3):245–283, 2016.

[202] Silvia Ingolfo, Ivan Jureta, Alberto Siena, Anna Perini, and Angelo Susi. Nomos 3: Legal compliance of roles and requirements. In *International Conference on Conceptual Modeling*, pages 275–288. Springer, 2014.

[203] Nicola Zeni, Elias A Seid, Priscila Engiel, Silvia Ingolfo, and John Mylopoulos. Building large models of law with nómost. In *International Conference on Conceptual Modeling*, pages 233–247. Springer, 2016.

[204] Aaron K Massey, Paul N Otto, and Annie I Antón. Prioritizing legal requirements. In *2009 Second International Workshop on Requirements Engineering and Law*, pages 27–32. IEEE, 2009.

[205] Najmeh Mousavi Nejad, Pablo Jabat, Rostislav Nedelchev, Simon Scerri, and Damien Graux. Establishing a strong baseline for privacy policy classification. In *Information and Communication Technology Systems Security and Privacy Protection*, volume 580, pages 370–383. Springer, 2020.

[206] David Sànchez, Alexandre Viejo, and Montserrat Batet. Automatic assessment of privacy policies under the gdpr. *Applied Sciences*, 11(4), 2021.

[207] Rajaa El Hamdani, Majd Mustapha, David Restrepo Amariles, Aurore Troussel, Sébastien Meeùs, and Katsiaryna Krasnashchok. A combined rule-based and machine learning approach for automated gdpr compliance checking. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pages 40–49, 2021.

[208] Abdel-Jaouad Aberkane, Geert Poels, and Seppe Vanden Broucke. Exploring automated gdpr-compliance in requirements engineering: A systematic mapping study. *Ieee Access*, 9:66542–66559, 2021.

[209] Monica Palmirani and Guido Governatori. Modelling legal knowledge for GDPR compliance checking. In Monica Palmirani, editor, *Legal Knowledge and Information Systems: The Thirty-first Annual Conference, Groningen, The Netherlands, 12-14 December 2018*, volume 313 of *Frontiers in Artificial Intelligence and Applications*, pages 101–110. IOS Press, 2018.

[210] Piero A Bonatti, Sabrina Kirrane, Iliana M Petrova, and Luigi Sauro. Machine understandable policies and gdpr compliance checking. *KI-Künstliche Intelligenz*, 34(3):303–315, 2020.

[211] Ze Shi Li, Colin Werner, Neil Ernst, and Daniela Damian. Gdpr compliance in the context of continuous integration. *arXiv preprint arXiv:2002.06830*, 2020.

[212] Adeline Nazarenko, François Lévy, and Adam Wyner. A pragmatic approach to semantic annotation for search of legal texts - an experiment on GDPR. In Schweighofer Erich, editor, *Legal Knowledge and Information Systems: The Thirty-fourth Annual Conference, Vilnius, Lithuania, 8-10 December 2021*, volume 346 of *Frontiers in Artificial Intelligence and Applications*, pages 23–32. IOS Press, 2021.

[213] Rex Chen, Fei Fang, Thomas Norton, Aleecia M McDonald, and Norman Sadeh. Fighting the fog: Evaluating the clarity of privacy disclosures in the age of ccpa. In *Proceedings of the 20th Workshop on Privacy in the Electronic Society*, pages 73–102, 2021.

[214] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. We value your privacy... now take some cookies: Measuring the gdpr's impact on web privacy. *Informatik Spektrum*, 42, 2019.

[215] Muhammad Sajidur Rahman, Pirouz Naghavi, Blas Kojusner, Sadia Afroz, Byron Williams, Sara Rampazzi, and Vincent Bindschaedler. Permpress: Machine learning-based pipeline to evaluate permissions in app privacy policies. *IEEE Access*, 10, 2022.

[216] Abdel-Jaouad Aberkane, Seppe Vanden Broucke, and Geert Poels. Investigating organizational factors associated with gdpr noncompliance using privacy policies: A machine learning approach. In *2022 IEEE 4th International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications (TPS-ISA)*, pages 107–113. IEEE, 2022.

[217] Robert Feldt, Francisco Gomes de Oliveira Neto, and Richard Torkar. Ways of applying artificial intelligence in software engineering. In *6th IEEE/ACM International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)*, 2018.

[218] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015.

[219] Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, Tsung-Han Lin, Thi Nguyen, Tzu-Kuo Huang, Jeff Schneider, and Nemanja Djuric. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *International Conference on Robotics and Automation (ICRA)*, 2019.

[220] Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X Gao, and Dazhong Wu. Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, 2018.

[221] Toan Luu Duc Huynh, Erik Hille, and Muhammad Ali Nasir. Diversification in the age of the 4th industrial revolution: The role of artificial intelligence, green bonds and cryptocurrencies. *Technological Forecasting and Social Change*, 159, 2020.

[222] Fines statistics: Highest fine by type of violation. `https://www.enforcementtracker.com/?insights`. Accessed: 2010-09-30.

[223] S. Ghanavati, A. Rifaut, E. Dubois, and D. Amyot. Goal-oriented compliance with multiple regulations. In *22nd IEEE International Requirements Engineering Conference*, 2014.

[224] Okhaide Akhigbe, Daniel Amyot, and Gregory Richards. A systematic literature mapping of goal and non-goal modelling methods for legal and regulatory compliance. *Requirements Engineering*, 24(4), 2019.

[225] Damiano Torre, Ghanem Soltana, Mehrdad Sabetzadeh, Lionel C. Briand, Yuri Auffinger, and Peter Goes. Using models to enable compliance checking against the GDPR: an experience report. In *22nd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems,*, 2019.

[226] Pille Pullonen, Jake Tom, Raimundas Matulevicius, and Aivo Toots. Privacy-enhanced BPMN: Enabling data privacy analysis in business processes models. *Software & Systems Modeling*, 18(6), 2019.

[227] Shomir Wilson, Florian Schaub, Rohan Ramanath, Norman M. Sadeh, Fei Liu, Noah A. Smith, and Frederick Liu. Crowdsourcing annotations for websites' privacy policies: Can it really work? In *Proceedings of the 25th International Conference on World Wide Web*, 2016.

[228] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G Shin, and Karl Aberer. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th {USENIX} security symposium ({USENIX} security 18)*, 2018.

[229] Ellen Poplavska, Thomas B. Norton, Shomir Wilson, and Norman M. Sadeh. From prescription to description: Mapping the GDPR to a privacy policy corpus annotation scheme. In Villata Serena, Jakub Harasta, and Petr Kremen, editors, *Legal Knowledge and Information Systems - JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9-11, 2020*, 2020.

[230] European Union. Commercial sector: launch of the adoption procedure for a draft adequacy decision on the eu-u.s. data privacy framework. Accessed Jun. 02, 2023 [Online].

[231] Paul Breitbarth. The impact of gdpr one year on. *Network Security*, 2019(7):11–13, 2019.

[232] Supreeth Shastri, Vinay Banakar, Melissa Wasserman, Arun Kumar, and Vijay Chidambaram. Understanding and benchmarking the impact of gdpr on database systems. *arXiv preprint arXiv:1910.00728*, 2019.

[233] Aashaka Shah, Vinay Banakar, Supreeth Shastri, Melissa F Wasserman, and Vijay Chidambaram. Analyzing the impact of gdpr on storage systems. In *USENIX Workshop on Hot Topics in Storage and File Systems*, 2019.

[234] He Li, Lu Yu, and Wu He. The impact of gdpr on global technology development, 2019.

[235] ELLINOR Johansson, KONSTA Sutinen, JULIUS Lassila, VALTER Lang, MINNA Martikainen, and OTHMAR M Lehner. Regtech-a necessary tool to keep up with compliance and regulatory changes. *ACRN Journal of Finance and Risk Perspectives, Special Issue Digital Accounting*, 8:71–85, 2019.

[236] Damiano Torre, Mauricio Alferez, Ghanem Soltana, Mehrdad Sabetzadeh, and Lionel Briand. Modeling data protection and privacy: application and experience with gdpr. *Software and Systems Modeling*, 20(6):2071–2087, 2021.

[237] Muhammad Usman, Michael Felderer, Michael Unterkalmsteiner, Eriks Klotins, Daniel Mendez, and Emil Alégroth. Compliance requirements in large-scale software development: An industrial case study. In *Product-Focused Software Process Improvement*, 2020.

[238] Orlando Amaral, Sallam Abualhaija, and Lionel C. Briand. “dpa compliance checking using ai technologies (dikaio)”, 2023. Available at `https://figshare.com/articles/online_resource/DIKAIO/22274761`, July 2023.

[239] Sallam Abualhaija, Chetan Arora, Mehrdad Sabetzadeh, Lionel Briand, and Eduardo Vaz. A machine learning-based approach for demarcating requirements in textual specifications. In *Proceedings of the 27th IEEE International Requirements Engineering Conference (RE'19)*, 2019.

[240] Travis D Breaux and Annie I Antón. Analyzing goal semantics for rights, permissions, and obligations. In *13th IEEE International Conference on Requirements Engineering (RE'05)*, 2005.

[241] Travis D Breaux, Annie I Antón, and Jon Doyle. Semantic parameterization: A process for modeling domain descriptions. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 18(2), 2008.

[242] Waël Hassan and Luigi Logrippo. Governance requirements extraction model for legal compliance validation. In *2009 Second International Workshop on Requirements Engineering and Law*, 2009.

[243] Guido Governatori, Mustafa Hashmi, Ho-Pun Lam, Serena Villata, and Monica Palmirani. Semantic business process regulatory compliance checking using legalruleml. In *20th International Conference, EKAW: Knowledge Engineering and Knowledge Management*, 2016.

[244] Ghanem Soltana, Mehrdad Sabetzadeh, and Lionel C Briand. Model-based simulation of legal requirements: Experience from tax policy simulation. In *24th IEEE International Requirements Engineering Conference*, 2016.

[245] Damiano Torre, Mauricio Alférez, Ghanem Soltana, Mehrdad Sabetzadeh, and Lionel C. Briand. Model driven engineering for data protection and privacy: Application and experience with GDPR. *CoRR*, abs/2007.12046, 2020.

[246] Travis D. Breaux and Florian Schaub. Scaling requirements extraction to the crowd: Experiments with privacy policies. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pages 163–172, 2014.

[247] Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N. Cameron Russell, Thomas B. Norton, Eduard Hovy, Joel Reidenberg, and Norman Sadeh. The creation and analysis of a website privacy policy corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1330–1340. Association for Computational Linguistics, 2016.

[248] Andrea Zasada, Mustafa Hashmi, Michael Fellmann, and David Knuplesch. Evaluation of compliance rule languages for modelling regulatory compliance requirements. *Software*, 2(1), 2023.

[249] Rajaa El Hamdani, Majd Mustapha, David Restrepo Amariles, Aurore Troussel, Sébastien Meeùs, and Katsiaryna Krasnashchok. A combined rule-based and machine learning approach for automated gdpr compliance checking. In *18th International Conference on Artificial Intelligence and Law*, 2021.

[250] Lavanya Elluri, Sai Sree Laya Chukkapalli, Karuna Pande Joshi, Tim Finin, and Anupam Joshi. A bert based approach to measure web services policies compliance with gdpr. *IEEE Access*, 9:148004–148016, 2021.

[251] Abdulaziz Aborujilah, Abdulaleem Z. Al-Othmani, Zalizah Awang Long, Nur Syahela Hussien, and Dahlan Abdul Ghani. Conceptual model for automating gdpr compliance verification using natural language approach. In *2022 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE)*, pages 1–6, 2022.

[252] Saad Ezzini, Sallam Abualhaija, Chetan Arora, and Mehrdad Sabetzadeh. Automated handling of anaphoric ambiguity in requirements: A multi-solution study. In *44th International Conference on Software Engineering*, 2022.

[253] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in Neural Information Processing Systems*, 33:16857–16867, 2020.

[254] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 2020.

[255] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *33rd Conference on Neural Information Processing Systems*, 2019.

[256] Ron Kohavi. *Wrappers for performance enhancement and oblivious decision graphs*. Stanford University, 1996.

[257] Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Machine Learning*, 2001.

[258] Abhishek Sainani, Preethu Rose Anish, Vivek Joshi, and Smita Ghaisas. Extracting and classifying requirements from software engineering contracts. In *28th IEEE International Requirements Engineering Conference*, 2020.

[259] Fahad ul Hassan and Tuyen Le. Automated requirements identification from construction contract documents using natural language processing. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, 12(2), 2020.

[260] Cheong Hee Park and Haesun Park. A comparison of generalized linear discriminant analysis algorithms. *Pattern Recognition*, 41(3), 2008.

[261] Joyoshree Ghosh and Shaon Bhatta Shuvo. Improving classification model's performance using linear discriminant analysis on linear data. In *10th International Conference on Computing, Communication and Networking Technologies*. IEEE, 2019.

[262] Hua Wang, Chris Ding, and Heng Huang. Multi-label linear discriminant analysis. In *Proceedings on Computer Vision, Greece, September 5-11*. Springer, 2010.

[263] Garm Lucassen, Marcel Robeer, Fabiano Dalpiaz, Jan Martijn EM Van Der Werf, and Sjaak Brinkkemper. Extracting conceptual models from user stories with visual narrator. *Requirements Engineering*, 22, 2017.

[264] Balasubramaniam Ramesh and Vasant Dhar. Supporting systems development by capturing deliberations during requirements engineering. *IEEE Transactions on Software Engineering*, 18(6), 1992.

[265] European Comission. Ethics guidelines for trustworthy ai. Accessed Jul. 01, 2023 [Online].