

NLP-Based Automated Compliance Checking of Data Processing Agreements Against GDPR

Orlando Amaral Cejas[✉], Member, IEEE, Muhammad Ilyas Azeem[✉], Member, IEEE,
Sallam Abualhaija[✉], Member, IEEE, and Lionel C. Briand[✉], Fellow, IEEE

Abstract—When the entity processing personal data (the processor) differs from the one collecting personal data (the controller), processing personal data is regulated in Europe by the General Data Protection Regulation (GDPR) through *data processing agreements* (DPAs). Checking the compliance of DPAs contributes to the compliance verification of software systems as DPAs are an important source of requirements for software development involving the processing of personal data. However, manually checking whether a given 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 verifying these requirements in the DPA. Legal texts introduce additional complexity due to convoluted language and inherent ambiguity leading to potential misunderstandings. In this paper, we propose an automated solution to check the compliance of a given DPA against GDPR. In close interaction with legal experts, we first built two artifacts: (i) the “shall” requirements extracted from the GDPR provisions relevant to DPA compliance and (ii) a glossary table defining the legal concepts in the requirements. Then, we developed an automated solution that leverages natural language processing (NLP) technologies to check the compliance of a given DPA against these “shall” requirements. Specifically, our approach 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 it further provides recommendations about missing information in the DPA. Over a dataset of 30 actual DPAs, the approach correctly finds 618 out of 750 genuine violations while raising 76 false violations, and further correctly identifies 524 satisfied requirements. The approach has thus an average precision of 89.1%, a recall of 82.4%, and an accuracy of 84.6%. Compared to a baseline that relies on off-the-shelf NLP tools, our approach provides an average accuracy gain of ≈20 percentage points. The accuracy of our approach can be improved to ≈94% with limited manual verification effort.

Manuscript received 20 September 2022; revised 18 April 2023; accepted 16 June 2023. Date of publication 27 June 2023; date of current version 19 September 2023. This work was supported in part by Linklaters, Luxembourg’s National Research Fund (FNR) under Grant BRIDGES/19/IS/13759068/ARTAGO, and in part by the NSERC of Canada under the Discovery and CRC programs. Recommended for acceptance by P. Spoletini. (*Corresponding author: Orlando Amaral Cejas*)

Orlando Amaral Cejas, Muhammad Ilyas Azeem, and Sallam Abualhaija are with the SnT Centre for Security, Reliability, and Trust, University of Luxembourg, 4365 Esch-sur-Alzette, Luxembourg (e-mail: orlando.amaral.cejas@uni.lu; ilyas.azeem@uni.lu; sallam.abualhaija@uni.lu).

Lionel C. Briand is with the SnT Centre for Security, Reliability, and Trust, University of Luxembourg, 4365 Esch-sur-Alzette, Luxembourg, and also with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: lionel.briand@uni.lu).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TSE.2023.3288901>, provided by the authors.

Digital Object Identifier 10.1109/TSE.2023.3288901

Index Terms—Requirements engineering (RE), the general data protection regulation (GDPR), regulatory compliance, natural language processing (NLP), data processing agreement (DPA), privacy.

I. INTRODUCTION

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 [1]. 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 [2]. In Europe and even worldwide, the general data protection regulation (GDPR) was put into effect in 2018 to oblige organizations to provide such data protection guarantees when handling personal data of European residents. Violating GDPR can levy penalty fines reaching up to tens of millions of euros [3]. 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 privacy policies between organizations and individuals (often signed by individuals) as well as *data processing agreements* (DPAs) between two organizations involved in collecting and processing personal data. Individuals are not necessarily aware of DPAs. We focus in our work on verifying the compliance of DPAs against the provisions of GDPR. From a requirements engineering (RE) standpoint, this compliance verification 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) [4].

DPA are legally binding agreements that regulate the data processing activities according to GDPR. To be deemed GDPR-compliant, the 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. To illustrate, 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 requires Sefer university to sign a DPA with the accounting office.

A. Running Example

Fig. 1 shows on the left-hand side an excerpt from a DPA that addresses the scenario above, and on the right-hand side a set of DPA-related requirements extracted from GDPR, ordered by their occurrence in the DPA. In Section III, we elaborate on the complete list of GDPR requirements. In the remainder of the paper, 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. 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 I. 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 [5]. 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 verify 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. 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 impedes compliance verification. 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*. Verifying all key elements

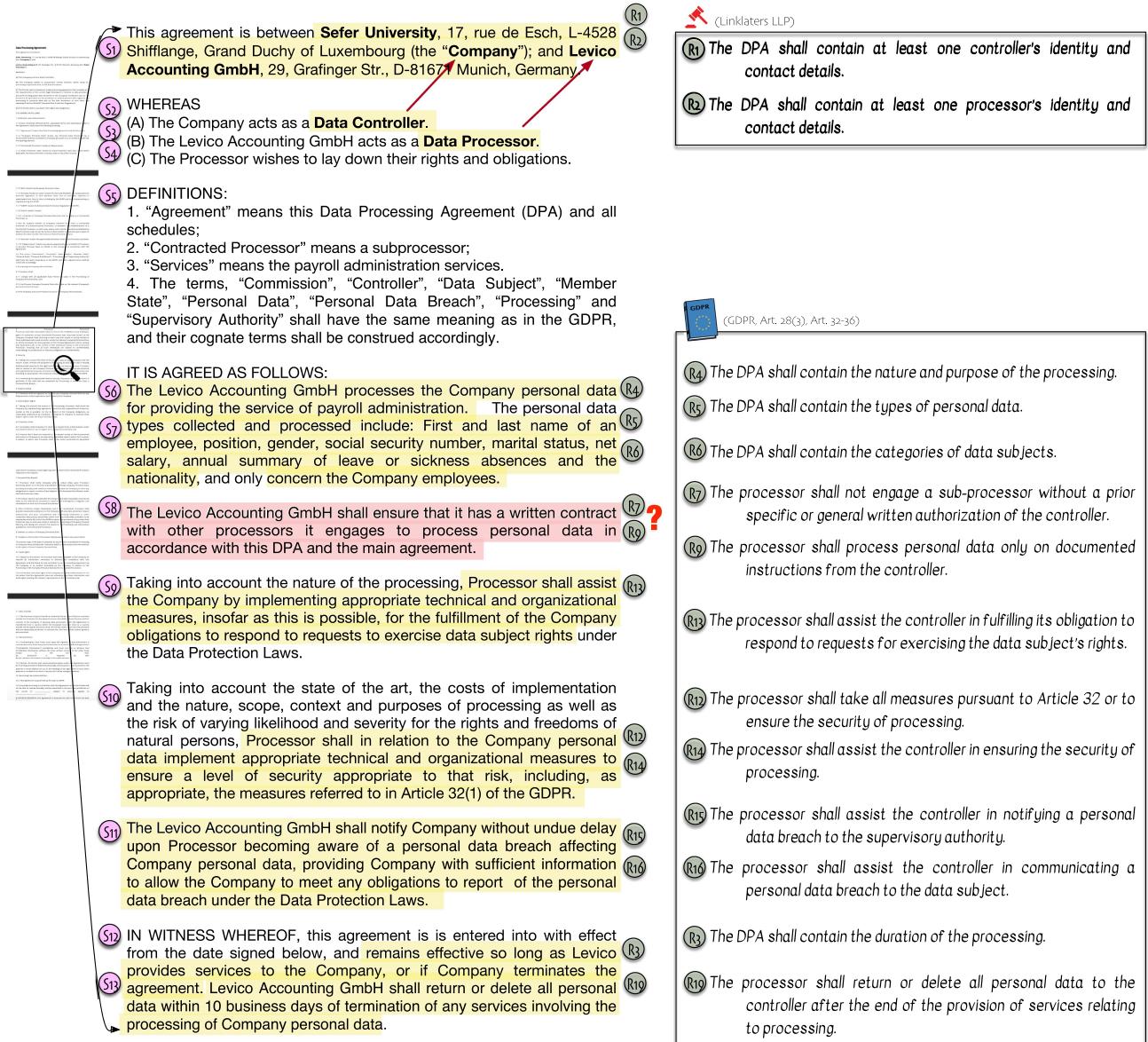


Fig. 1. Excerpt from a DPA verified against GDPR requirements.

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 paper, we propose a natural language processing (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 - a multinational law firm). 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 GDPR advisory and compliance services.

Ensuring that the DPA excerpt in Fig. 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 VI, much work has been done in the RE literature on automating regulatory compliance. Despite being an essential source for compliance requirements of data

TABLE I
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. (<i>Linklaters LLP</i>)
R2 (MD2)	The DPA shall contain at least one processor's identity and contact details. (<i>Linklaters LLP</i>)
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 are under 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))
R25 (PO19)	When assessing the level of security, the processor shall take into account the risk of accidental or unlawful destruction, loss, alteration, 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.

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 [5] 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 II, 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. 2 illustrates SF elements describing a *processing* event in a requirement example extracted from GDPR Art. 28(3)(a) –

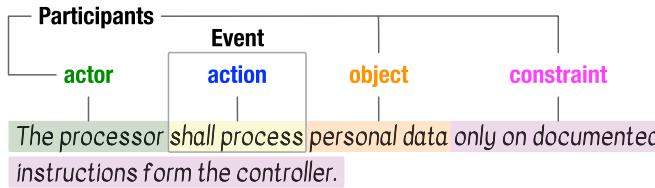


Fig. 2. Illustration of SF-based representation.

corresponding to R9 in Fig. 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*).

B. Research Questions

This paper 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. Verifying 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 III.

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 IV.

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 V.

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.

C. 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, including the remaining 30 DPAs, was used exclusively for evaluation. The evaluation set collectively contains a total of 7048 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 IV. 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.

D. Structure

Section II explains the necessary background for our approach. Section III presents how we extracted the DPA-related requirements from GDPR. Section IV describes our automated approach. Section V evaluates our approach. Section VI reviews related work. Section VII discusses threats to validity. Section VIII describes how our methodology can be adapted beyond GDPR or DPAs. Finally, Section IX concludes the paper.

II. BACKGROUND

In this section, we briefly introduce GDPR, and explain the NLP background on semantic frames.

A. GDPR

GDPR [6] is the benchmark for data privacy regulations in Europe with 173 recitals and 99 articles divided into 11 chapters. GDPR applies primarily to organizations in Europe, but it can also impose obligations on organizations outside Europe when they offer goods or services, or monitor individuals in Europe. According to GDPR, there are often two main actors that should be identified in agreements related to processing personal data, namely a data controller and data processor. A data controller determines the purpose and means of processing personal data, whereas a data processor processes personal data based on the instructions of the controller. GDPR imposes various obligations on a given organization when it is acting as a processor or controller, the latter being subject to more provisions than the former.

Data processors are obliged to: (1) implement adequate technical and organizational measures to ensure the security of personal data, and, in case of data breaches, notify the controllers without undue delay; (2) appoint a statutory data protection officer (if needed) and conduct a formal data protection impact assessment for certain types of high-risk processing; (3) keep records about their data processing; and (4) comply to GDPR restrictions when transferring personal data outside Europe. Data controllers have to meet the obligations listed above, and further: (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) ensure the individual rights envisaged by GDPR, e.g., the right to be forgotten and the right to lodge a complaint.

GDPR includes some specific provisions in relation to DPAs, which in turn can have an impact on software development. For example, GDPR requires data processors to return or delete personal data without undue delay after the processing service is

completed. For some cloud service providers, implementing this requirement, where immediate deletion is not possible, requires archiving personal data first until the next delete command is performed. We elaborate the GDPR requirements concerning DPAs in Section III.

B. Semantic Frames

A semantic frame (SF) describes a set of elements that participate in a particular event [5]. Each element in the frame is a text span that is labeled with a semantic role (SR), denoted as $[\text{span}]_{\text{role}}$. 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 [7] and PropBank [8] 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 [9]. 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*) [10], [11]. SRL can be employed to improve the solutions of many downstream NLP tasks such as machine translation [12], question answering [13], [14] and relation extraction [15], [16]. 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 [14], [17], [18], 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 IV.

III. 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 Table I as well as in the tables provided as supplementary material, available online.

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 completeness 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 I) 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, see supplementary material, available online) are derived from good practice in writing DPAs according to 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 I, 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 requirements according to the legal experts' interpretation, thereby alleviating any ambiguity in the GDPR. During these steps, we created the glossary terms (the second artifact provided as supplementary material, available online) with traceability links to GDPR articles to facilitate the understanding of the 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 I. We adopted the GDPR text as-is when it resembled a "shall" requirement such as R34

(see supplementary material, available online) 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 requirements whether it is satisfied by any statement in the DPA. As we elaborate in Section V-B, our manual analysis aimed at establishing in-depth understanding of the compliance requirements before outsourcing the annotation to external annotators.

IV. AUTOMATED APPROACH FOR DPA COMPLIANCE CHECKING AGAINST GDPR (*RQ2*)

To address *RQ2*, we devise an NLP-based approach for checking the compliance of DPAs against GDPR at a phrasal level (hereafter referred to as *DERECHA*, standing for **D**PA **S**emantic **f**RamE-based **C**ompliance **H**ecking **A**gainst **GDPR**). 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 [19] often yields accurate models for many downstream tasks [20], [21], [22]. To this end, we note that such language models are integrated into currently available off-the-shelf semantic role labeling 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 V-C against the performance of *DERECHA*. That said, we believe that attempting to extend our dataset and experimenting with machine learning or deep learning models would be important to refine our conclusions. This is left for future work.

Fig. 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 III. Note that this step is performed only once and can be reused across DPAs. In practice, from a user perspective, the SF-based representations identified in this paper can be adopted as-is. In step B, we apply an NLP pipeline 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". In the last step, 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 these steps next.

A. Step A: Defining SFs in Compliance Requirements

In this step, we define an SF-based representation to characterize the information content of each requirement in GDPR. This representation aims at decomposing the requirement and thus enabling compliance checking at a phrasal level.

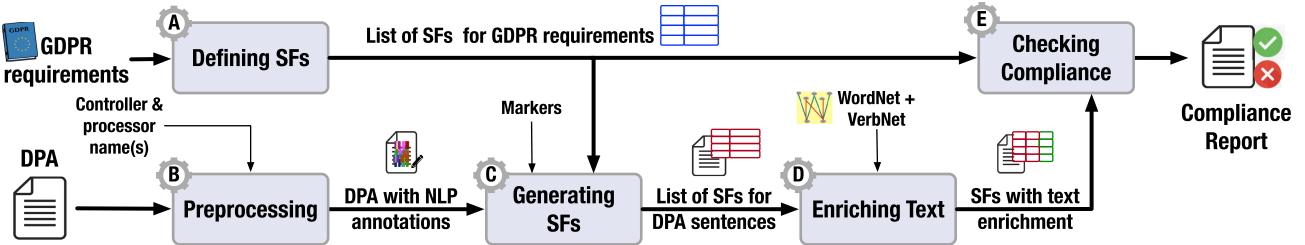


Fig. 3. Overview of our DPA compliance checking approach (DERECHA).

As can be seen in Table I, 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. 2. 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 IV-E.

To define the SF-based representation, we first identify the SRs in each requirement. Inspired by existing literature [9], [23], [24], we collate a list of 10 SRs pertinent to DPA compliance requirements in GDPR. Four SRs are illustrated in Fig. 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 I, 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 II 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 (ii) SF-based representations for the requirements (R7 – R25 and R30 – R45) passed on to step E. These representations enable automated compliance verification of the statements at a phrasal level. An example of such representation for *PO6* (R12 in Table I) contains the predicate *<take>*, and a set of four arguments $\mathcal{A} = \{a_i : 0 \leq i \leq 3\}$ such that $a_0 = [\text{the processor}]_{\text{actor}}$, $a_1 = [\text{all measures}]_{\text{object}}$, $a_2 = [\text{Article 32}]_{\text{reference}}$, $a_3 = [\text{to ensure the security of processing}]_{\text{reason}}$. The SF-based representations for all 45 requirements are provided as supplementary material, available online.

B. Step B: Preprocessing

Given a DPA as input, this step applies an NLP pipeline composed of seven modules, including: (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. For semantic parsing, we apply two widely-known lexical resources, namely WordNet [25], [26] and VerbNet [27], [28]. WordNet is a lexical database for English. It groups words into sets of synonyms called *synsets* and further connects the

TABLE II
EXTRACTION RULES OF SEMANTIC ROLES

SR	Description (D) [†] and Example (E) from Fig. 1
Action	(D) The <i>action</i> 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 <i>action</i> in S6 is “processes”.
Actor	(D) The <i>actor</i> is the NP containing the subject of the root verb in the statement (i.e., the <i>action</i>). (E) The <i>actor</i> in S6 is “Levico Accounting GmbH”
Object	(D) The <i>object</i> is either the NP that contains the object of the root verb AND the <i>action</i> does not contain any <i>beneficiary</i> 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 <i>object</i> (e.g., “process”) from other actions that evoke <i>beneficiary</i> (e.g., “inform”), we define markers. (E) The <i>object</i> in S6 is “Company personal data”
Beneficiary	(D) Similar to <i>object</i> , the <i>beneficiary</i> is either the NP that contains a preposition linked to the root verb AND the <i>action</i> contains a <i>beneficiary</i> marker, OR the NP that contains a subject of a verb that is different from the root verb. Since the <i>beneficiary</i> is an entity that benefits from the <i>action</i> , it can be associated directly with the action given that the <i>action</i> has a marker that pinpoints the necessity of a <i>beneficiary</i> . Alternatively, the <i>beneficiary</i> can be an entity associated with another action in the statement. (E) The <i>beneficiary</i> in S11 is “Company”
Condition [‡]	(D) The <i>condition</i> is the PP, OR ADVP, OR any phrase annotated as subordinate clause by the dependency parser given that it contains a <i>condition</i> marker. (E) The <i>condition</i> in S12 is “if Company terminates the agreement”.
Constraint [‡]	(D) The <i>constraint</i> is identified in similar manner as <i>condition</i> except that the phrase should contain a <i>constraint</i> marker. The <i>constraint</i> in S11 is “without undue delay”
Time [‡]	(D) The <i>time</i> is the NP, OR PP, OR ADVP, OR any phrase annotated as conjunctive preposition by the dependency parser given that it contains a <i>time</i> marker. (E) The <i>time</i> in S12 is “remains effective so long as Levico provides services”
Reason [‡]	(D) The <i>reason</i> 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 <i>reason</i> in S6 is “for providing the service of payroll administration”.
Situation [‡]	(D) The <i>situation</i> is the NP OR VP that contains a <i>situation</i> marker. (E) The <i>situation</i> in S13 is “termination of any services”.
Reference [‡]	(D) The <i>reference</i> is the NP that contains a <i>reference</i> marker. (E) The <i>reference</i> 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 [23], [24].

synsets via semantic relations. For example, *is-a* connects a hyponym (more specific synset) to a hypernym (more general synset) [29]. VerbNet groups verbs into classes according to their structure, e.g., the verbs under the *motion* class include run, escape, and leave [30].

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.

C. 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 I 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 semantic role labeling. 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 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 V-D. Further, this would lead to multiple answers, most of them being irrelevant, as explained in Section V-C. 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 V-C.

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 IV-B). In particular, we apply a set of rules (listed in Table II) 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.

Our rules incorporate a set of markers that help identify the SRs. Examples of markers are shown in Table III. Except for *situation* and *reference*, the markers we use in our work originate

TABLE III
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

from the RE literature [23], [24]. We improved 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 IV. 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 $[\text{span}_1 | \dots | \text{span}_n]_{\text{role}}$. For example, the argument $[\text{after} | \text{end}]_{\text{time}}$ 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.

D. 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 IV-B. 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 our paper. We note that MFS has been widely used as a baseline in the NLP literature and has often shown sufficient performance [31].

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. 1 is replaced with “the processor”. As explained in Section I, 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.

E. Step E: Compliance Checking

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

Metadata Requirements. To verify 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 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 the controller and processor.

For verifying the remaining requirements (R3 – R6 and R27 – R29), we apply a method inspired by Lesk algorithm [32], 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 III), 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.

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 IV shows a simplified example of checking compliance between a DPA statement and requirement R7. Our compliance

TABLE IV
EXAMPLE OF COMPLIANCE CHECKING USING SF-BASED REPRESENTATIONS AGAINST R7

R7. The processor shall not engage a sub-processor without a prior specific or general written authorization of the controller.						
GDPR	actor	(p)	object	NOT REQUIRED	constraint	
	the processor	not engage	a sub-processor		without	a prior
Levico Accounting GmbH can employ sub-contractors for the performance of the service of Levico's obligations.						
DPA*	actor	(p)	object	reason	constraint	
	Levico Accounting GmbH processor	can employ hire engage	sub-contractors sub-processor	for the performance of Levico's obligations. processor duty	time	MISSING

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

checking method using SFs comprises four steps, as elaborated in Algorithm 1. 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 indicating the matching degree. Finally, we conclude the compliance decision for the DPA. We explain each step next.

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 1).

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 [33] and widely-applied in NLP applications [34], [35], [36]. 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 [37]. 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.

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 1). 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 IV).

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 IV 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 [38], [39], [40]. 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 IV, 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.

Algorithm 1: SF-Based Compliance Checking of DPA Statements Against a Requirement r .

Require: \mathcal{S} : Statements in the input DPA.

- 1: $\text{score} \leftarrow 0$ //Set matching degree score of r to zero
- 2: $\text{degree}_i \leftarrow 0$
- 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** $\langle p_r \rangle \cap \langle p_i \rangle \neq \emptyset$ **or** $\text{sim}(\langle p_r \rangle, \langle p_i \rangle) > \theta_p$ **then** // Check compliance only if the two predicates match
- 7: **for** each argument a_{rj} in \mathcal{A}_r **do**
- 8: $\text{found} \leftarrow 0$ //Look for matching arguments
- 9: let ℓ_{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 ℓ_{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** (ℓ_i equals ℓ_r) $\&$ (($t_i \cap t_r \neq \emptyset$) **or** $\text{sim}(t_i, t_r) > \theta_a$) **then**
- 14: $\text{found} \leftarrow \text{found} + 1$
- 15: **end if**
- 16: **end for**
- 17: $\text{degree}_i \leftarrow (\text{found}_i + 1) / (\text{len}(\mathcal{A}_r) + 1)$ //Compute matching degree of s_i
- 18: **end for**
- 19: **end if**
- 20: **if** $\text{score} < \text{degree}_i$ **then**
- 21: $\text{score} \leftarrow \text{degree}_i$ //Compute the maximum matching degree score for r
- 22: **end if**
- 23: **end for**
- 24: **if** $\text{score} \neq 0$ **then** //Compliance decision
- 25: r is satisfied
- 26: **else**
- 27: r is violated
- 28: **end if**

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 23 in Algorithm 1). 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 IV 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 verifying and possibly correcting decisions when the confidence level is low. We elaborate on this in Section V-D.

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 shows the detailed report corresponding to the DPA example in Fig. 1 as generated by *DERECHA*.

V. EVALUATION

This section presents the empirical evaluation for our compliance checking approach.

A. Implementation

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 [41]. For operationalizing the NLP pipeline applied in step B of Fig. 3, we use the DKPro 1.10 toolkit [42]. In particular, our operationalization employs the OpenNLP tokenizer, POS-tagger and text chunker [43], Mate lemmatizer [44] and Stanford dependency parser [45]. 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 [46] for accessing WordNet, and JVerbNet 1.2 [47] for accessing VerbNet. To compute semantic similarity needed in step E in our approach, we use the Jaro-Winkler distance [48] and the *wup* metric [33] as implemented by the WS4J 1.0 library [49].

B. 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

Compliance checking report generated by DERECHA (DPA sEmantic fRamE-based Compliance cHecking Against 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, Graffinger 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.

Fig. 4. Example report generated by *DERECHA*.

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 verify 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 paper 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 III 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) [50] 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” [51], 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.

The overall document collection resulted in analyzing a total of 7048 statements in both phases, out of which 1742 ($\approx 25\%$) 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 V 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 V confirms that the small number of statements we have, for each requirement, prevents us from developing *DERECHA* based on machine learning.

C. Evaluation Procedure

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

TABLE V
DOCUMENT COLLECTION RESULTS

Cat [†]	ID	Total		Dev Data		Test Data (Ev)	
		D	S	D	S	D	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**.

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 $(1+\beta^2)*(P*R)/(\beta^2*P+R)$. 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 semantic role labeling module provided by AllenNLP [16] which is based on the BERT language model [19]. 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 pre-defined 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 I), *B1* generates two SF-based representations. The first one contains *⟨engage⟩*, $\mathcal{A} = \{\text{[the processor]}_{\text{argument0}}, \text{[a sub-processor]}_{\text{argument1}}\}$, and the second one is *⟨written⟩*, $\mathcal{A} = \{\text{[authorization]}_{\text{argument1}}\}$. *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 IV-A, 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 32 GB of memory. Our goal is to assess whether execution times suggest that *DERECHA* is applicable in practice.

D. Results and Discussion

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

Table VI 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 V-C. As explained in Section V-B, 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 verify the compliance of our 45 requirements. *DERECHA* correctly finds 618 requirement

TABLE VI
RESULTS OF COMPLIANCE CHECKING

ID	Cat [†]	DERECHA							B1								
		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	-	-	-	-	-	-	-	-	
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	PO10	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**.

violations (out of 750) and 524 satisfied requirements, but also leads to 76 false violations. As pinpointed in Section V-C, *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 $F_{0.5}$.

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.

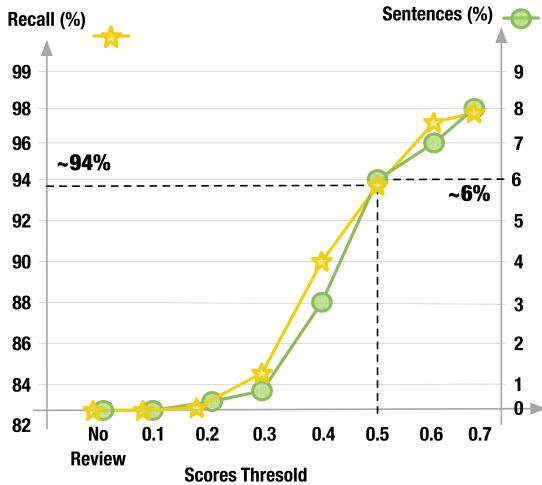


Fig. 5. Validation-accuracy trade-off.

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 IV-E) 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 below the selected threshold. Assuming we apply the above heuristic, Fig. 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 $\approx 6\%$ of the statements. For an average-sized DPA with 200 statements, this percentage corresponds 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 $\approx 90.6\%$ precision and $\approx 93.1\%$ 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. 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 I). 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 VI 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 semantic role labeling (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. 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.

To conclude, **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 verify its compliance against all 45 requirements.

VI. RELATED WORK

Regulatory compliance is a long-standing research topic in the RE literature [52], [53], [54], [55], [56], [57], [58], [59]. 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.

A. GDPR Compliance of Data Processing Activities

Several studies analyze the impact of GDPR on data sharing platforms [60], [61]. Urban et al. [62] study the behavior of online advertising companies in data tracking and information sharing with respect to the GDPR. Alhazmi and Arachchilage [63] 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 [64] propose a game design framework to help software developers better understand and implement privacy-related requirements in GDPR. Shastri et al. [65] 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 I 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. 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 paper, provides automated assistance to the developers and requirements engineers to verify the compliance of a DPA prior to relying on it for specifying compliance requirements for developing GDPR-compliant systems.

B. Elicitation of Compliance Requirements From Regulation

In the RE literature, representing requirements in regulations (including GDPR) relies heavily on model-driven engineering [66], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76]. Other methods for representing compliance requirements have however been explored. Ayala-Rivera and Pasquale [77] 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. [78] 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 [79] propose a frame-based requirements analysis method that extracts semi-formal representations of requirements from regulations. More recently, formal languages [80], [81], [82], [83] and predefined templates [84] 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. [75]. 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 verifying 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 [85] and EARS [86]) 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 [87].

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 [52], [67], [88]. It further facilitates transforming legal text to formal specifications [89]. 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. [23]. In our work, we refine and adapt a set of 10 SRs from literature [72], [90], [91], [92], [93], [94], [95].

C. Automated Support for Compliance Checking

Automated and semi-automated approaches using machine learning (in combination with NLP) [76], [96], [97], [98], [99], [100], [101], as well as manual approaches through crowdsourcing [88], [102], [103], [104], 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 [105], [106], [107], [108], [109]. Several approaches rely on Semantic Web as an enabler [110], [111]. Li et al. [112] 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. [113] 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. [23] 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. [24] propose using semantic frames to detect incompleteness in privacy policies. Specifically, the authors manually analyze 15 privacy policies 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. [23] and Bhatia et al. [24] 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., privacy policies) which are not informative for individuals [114], [115]. 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 [116], e.g., about the purpose of data processing (an essential requirement in GDPR) [117]. Several approaches are proposed to check the compliance of websites by classifying the content of their disclosed privacy policies 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.

VII. 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-parties (non-authors). 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.

VIII. 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 verify 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} .

IX. CONCLUSION

In this paper, we propose an automated approach for checking the compliance of data processing agreements (DPAs) against the general data protection regulation (GDPR). Such DPAs have a significant impact on the requirements of systems processing personal data. In close collaboration with legal experts, we extract and document DPA-related requirements as “shall” requirements. Such documentation provides a shared understanding between requirements engineers and legal experts. In our approach, we first manually create a representation of the GDPR requirements based on semantic frames (SFs). Our approach then automatically generates SF-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 compare *DERECHA* against a baseline that relies on an off-the-shelf NLP pipeline. *DERECHA* outperforms this baseline with an average gain of ≈ 20 percentage points in accuracy. In the future, we plan to curate more 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 verify the compliance of the DPAs available from major service providers against GDPR.

ACKNOWLEDGMENTS

The authors would like to specially thank Katrien Baetens, Sylvie Forastier, Peter Goes and Catherine Freichel from Lincklers LLP for enriching this work through their expertise. This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

REFERENCES

- [1] N. Zhan, S. Sarkadi, N. C. Pacheco, and J. Such, “A model for governing information sharing in smart assistants,” in *Proc. AAAI/ACM Conf. AI Ethics Soc.*, AAAI Press, 2022, pp. 845–855.
- [2] Á. Feal et al., “Don’t accept candies from strangers: An analysis of third-party SDKs,” in *Proc. Comput. Privacy Data Protection Conf.*, 2020, pp. 1–27.
- [3] D. G. for Justice & Consumers (European Commission), “The GDPR: New opportunities, new obligations,” *Justice Consum.*, 2018.
- [4] N. Pantlin, C. Wiseman, and M. Everett, “Supply chain arrangements: The ABC to GDPR compliance—A spotlight on emerging market practice in supplier contracts in light of the GDPR,” *Comput. Law Secur. Rev.*, vol. 34, no. 4, pp. 881–885, 2018.
- [5] D. Jurafsky and J. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice Hall, 2009.
- [6] European Union, “General data protection regulation,” *Official J. Eur. Union*, 2018.
- [7] C. F. Baker, C. J. Fillmore, and J. B. Lowe, “The Berkeley FrameNet project,” in *Proc. 36th Annu. Meeting Assoc. Comput. Linguistics 17th Int. Conf. Comput. Linguistics*, 1998, pp. 86–90.
- [8] P. R. Kingsbury and M. Palmer, “From treebank to propbank,” in *Proc. Int. Conf. Lang. Resour. Eval.*, 2002, pp. 1989–1993.
- [9] D. Gildea and D. Jurafsky, “Automatic labeling of semantic roles,” *Comput. Linguistics*, vol. 28, no. 3, pp. 245–288, 2002.
- [10] Z. Li, H. Zhao, S. He, and J. Cai, “Syntax role for neural semantic role labeling,” *Comput. Linguistics*, vol. 47, no. 3, pp. 529–574, 2021.
- [11] L. Márquez, X. Carreras, K. C. Litkowski, and S. Stevenson, “Semantic role labeling: An introduction to the special issue,” *Comput. Linguistics*, vol. 34, no. 2, pp. 145–159, 2008.
- [12] C. Shi et al., “Knowledge-based semantic embedding for machine translation,” in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 2245–2254.
- [13] J. Berant, A. Chou, R. Frostig, and P. Liang, “Semantic parsing on freebase from question-answer pairs,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1533–1544.
- [14] D. Shen and M. Lapata, “Using semantic roles to improve question answering,” in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, 2007, pp. 12–21.
- [15] Y. Lin, Z. Liu, and M. Sun, “Neural relation extraction with multi-lingual attention,” in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 34–43.
- [16] P. Shi and J. Lin, “Simple BERT models for relation extraction and semantic role labeling,” 2019, *arXiv: 1904.05255*.
- [17] S. Wu and M. Palmer, “Semantic mapping using automatic word alignment and semantic role labeling,” in *Proc. 5th Workshop Syntax Semantics Struct. Statist. Transl.*, 2011, pp. 21–30.
- [18] K. Woodsend and M. Lapata, “Text rewriting improves semantic role labeling,” *J. Artif. Intell. Res.*, vol. 51, pp. 133–164, 2014.
- [19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, *arXiv: 1810.04805*.
- [20] J. Howard and S. Ruder, “Fine-tuned language models for text classification,” 2018, *arXiv: 1801.06146*.
- [21] N. Jiang and M.-C. de Marneffe, “Evaluating BERT for natural language inference: A case study on the CommitmentBank,” in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 6086–6091.
- [22] A. Yates, R. Nogueira, and J. Lin, “Pretrained transformers for text ranking: BERT and beyond,” in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, 2021, pp. 1154–1156.
- [23] A. Sleimi, N. Sannier, M. Sabetzadeh, L. C. Briand, and J. Dann, “Automated extraction of semantic legal metadata using natural language processing,” in *Proc. IEEE 26th Int. Requirements Eng. Conf.*, 2018, pp. 124–135.
- [24] J. Bhatia, M. C. Evans, and T. D. Breaux, “Identifying incompleteness in privacy policy goals using semantic frames,” *Requirements Eng.*, vol. 24, no. 3, pp. 291–313, 2019.
- [25] G. A. Miller, “WordNet: A lexical database for English,” *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [26] G. A. Miller, *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: MIT Press, 1998.
- [27] A. Korhonen and T. Briscoe, “Extended lexical-semantic classification of English verbs,” in *Proc. Comput. Lexical Semantics Workshop Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol.*, 2004, pp. 38–45.
- [28] K. Kipper, A. Korhonen, N. Ryant, and M. Palmer, “Extending VerbNet with novel verb classes,” in *Proc. 5th Int. Conf. Lang. Resour. Eval.*, 2006, pp. 1027–1032.
- [29] E. Agirre and P. Edmonds, *Word Sense Disambiguation: Algorithms and Applications*, vol. 33. Berlin, Germany: Springer, 2007.
- [30] K. K. Schuler, *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Philadelphia, PA, USA: Univ. Pennsylvania, 2005.
- [31] R. Navigli, K. C. Litkowski, and O. Hargraves, “SemEval-2007 task 07: Coarse-grained english all-words task,” in *Proc. 4th Int. Workshop Semantic Eval.*, 2007, pp. 30–35.
- [32] M. Lesk, “Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone,” in *Proc. 5th Annu. Int. Conf. Syst. Documentation*, 1986, pp. 24–26.
- [33] Z. Wu and M. Palmer, “Verbs semantics and lexical selection,” in *Proc. 32nd Annu. Meeting Assoc. Comput. Linguistics*, 1994, pp. 133–138.
- [34] C. D. Corley and R. Mihalcea, “Measuring the semantic similarity of texts,” in *Proc. Assoc. Comput. Linguistics Workshop Empirical Model. Semantic Equivalence Entailment*, 2005, pp. 13–18.
- [35] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, “Automatic evaluation of topic coherence,” in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2010, pp. 100–108.
- [36] V. Shwartz and I. Dagan, “Still a pain in the neck: Evaluating text representations on lexical composition,” *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 403–419, 2019.
- [37] T. Pedersen et al., “WordNet:: Similarity-measuring the relatedness of concepts,” in *Proc. 19th Nat. Conf. Artif. Intell. 16th Conf. Innov. Appl. Artif. Intell.*, 2004, pp. 1024–1025.
- [38] S. Cahyono, “Comparison of document similarity measurements in scientific writing using Jaro-Winkler distance method and paragraph vector method,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 662, no. 5, IOP Publishing, 2019, Art. no. 052016.
- [39] M. A. Yulianto and N. Nurhasanah, “The hybrid of Jaro-Winkler and Rabin-Karp algorithm in detecting Indonesian text similarity,” *Jurnal Online Informatika*, vol. 6, no. 1, pp. 88–95, 2021.
- [40] R. A. Sulaiman, D. N. Jawawi, and S. A. Halim, “A dissimilarity with Dice-Jaro-Winkler test case prioritization approach for model-based testing in software product line,” *KSII Trans. Internet Inf. Syst.*, vol. 15, no. 3, pp. 932–951, 2021.
- [41] Apache-POI, “The Java API for microsoft documents.” 2022. Accessed: Feb. 2022. [Online]. Available: <https://github.com/apache/poi>
- [42] R. Eckart de Castilho and I. Gurevych, “A broad-coverage collection of portable NLP components for building shareable analysis pipelines,” in *Proc. Workshop Open Infrastructures Anal. Frameworks Hum. Lang. Technol.*, 2014, pp. 1–11.
- [43] Apache OpenNLP, 2022. Accessed: Feb. 2022. [Online]. Available: <https://opennlp.apache.org>
- [44] Mate tools, 2022. Accessed: Feb. 2022. [Online]. Available: <https://www.ims.uni-stuttgart.de/en/research/resources/tools/matetools/>
- [45] M. C. De Marneffe and C. D. Manning, “Stanford typed dependencies manual,” Stanford Univ., Tech. Rep., 2008.
- [46] extJWNL, “Extended Java WordNet library.” 2022. Accessed: Feb. 2022. [Online]. Available: <http://extjwnl.sourceforge.net/#>

- [47] JVerbnet, "Java library for VerbNet," 2022. Accessed: Feb. 2022. [Online]. Available: <http://projects.csail.mit.edu/jverbnet/>
- [48] M. A. Jaro, "Advances in record-linkage methodology as applied to matching the 1985 Census of Tampa, Florida," *J. Amer. Statist. Assoc.*, vol. 84, no. 406, pp. 414–420, 1989.
- [49] WS4J, "WordNet similarity for Java." 2022. Accessed: Feb. 2022. [Online]. Available: <https://github.com/Sciss/ws4j>
- [50] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960.
- [51] A. J. Viera et al., "Understanding interobserver agreement: The Kappa statistic," *Fam. Med.*, vol. 37, no. 5, pp. 360–363, 2005.
- [52] T. D. Breaux, A. I. Antón, K. Boucher, and M. Dorfman, "Legal requirements, compliance and practice: An industry case study in accessibility," in *Proc. IEEE 16th Int. Requirements Eng. Conf.*, 2008, pp. 43–52.
- [53] S. Ghanavati, D. Amyot, and L. Peyton, "A systematic review of goal-oriented requirements management frameworks for business process compliance," in *Proc. IEEE 4th Int. Workshop Requirements Eng. Law*, 2011, pp. 25–34.
- [54] A. K. Massey, R. L. Rutledge, A. I. Antón, and P. P. Swire, "Identifying and classifying ambiguity for regulatory requirements," in *Proc. IEEE 22nd Int. Requirements Eng. Conf.*, 2014, pp. 83–92.
- [55] S. Ghanavati, A. Rifaut, E. Dubois, and D. Amyot, "Goal-oriented compliance with multiple regulations," in *Proc. IEEE 22nd Int. Conf. Requirements Eng.*, 2014, pp. 73–82.
- [56] M. C. Evans, J. Bhatia, S. Wadkar, and T. D. Breaux, "An evaluation of constituency-based hyponymy extraction from privacy policies," in *Proc. IEEE 25th Int. Requirements Eng. Conf.*, 2017, pp. 312–321.
- [57] O. Akhigbe, D. Amyot, G. Richards, and L. Lessard, "GoRIM: A model-driven method for enhancing regulatory intelligence," *Softw. Syst. Model.*, vol. 21, pp. 1613–1641, 2022.
- [58] S. Abualhaija, C. Arora, A. Sleimi, and L. Briand, "Automated question answering for improved understanding of compliance requirements: A multi-document study," in *Proc. IEEE 30th Int. Requirements Eng. Conf.*, Melbourne, Australia, Aug. 15–19, 2022, pp. 39–50.
- [59] T. Breaux and T. Norton, "Legal accountability as software quality: A U.S. data processing perspective," in *Proc. IEEE 30th Int. Requirements Eng. Conf.*, Melbourne, Australia, Aug. 15–19, 2022, pp. 101–113.
- [60] M. Horák, V. Stupka, and M. Husák, "GDPR compliance in cybersecurity software: A case study of DPPIA in information sharing platform," in *Proc. 14th Int. Conf. Availability Rel. Secur.*, 2019, pp. 1–8.
- [61] N. N. Loideain, "A port in the data-sharing storm: The GDPR and the Internet of Things," *J. Cyber Policy*, vol. 4, no. 2, pp. 178–196, 2019.
- [62] T. Urban, D. Tatang, M. Degeling, T. Holz, and N. Pohlmann, "Measuring the impact of the GDPR on data sharing in ad networks," in *Proc. 15th ACM Asia Conf. Comput. Commun. Secur.*, 2020, pp. 222–235.
- [63] A. Alhazmi and N. A. G. Arachchilage, "I'm all ears! Listening to software developers on putting GDPR principles into software development practice," *Pers. Ubiquitous Comput.*, vol. 25, no. 5, pp. 879–892, 2021.
- [64] A. Alhazmi and N. A. G. Arachchilage, "A serious game design framework for software developers to put GDPR into practice," in *Proc. 16th Int. Conf. Availability Rel. Secur.*, 2021, pp. 1–6.
- [65] S. Shastri, M. Wasserman, and V. Chidambaram, "GDPR anti-patterns," *Commun. ACM*, vol. 64, no. 2, pp. 59–65, 2021.
- [66] P. N. Otto and A. I. Anton, "Addressing legal requirements in requirements engineering," in *Proc. IEEE 15th Int. Requirements Eng. Conf.*, 2007, pp. 5–14.
- [67] A. Siena, A. Perini, A. Susi, and J. Mylopoulos, "A meta-model for modelling law-compliant requirements," in *Proc. IEEE 2nd Int. Workshop Requirements Eng. Law*, 2009, pp. 45–51.
- [68] J. C. Maxwell and A. I. Anton, "Developing production rule models to aid in acquiring requirements from legal texts," in *Proc. IEEE 17th Int. Requirements Eng. Conf.*, 2009, pp. 101–110.
- [69] S. Ghanavati, D. Amyot, and L. Peyton, "Compliance analysis based on a goal-oriented requirement language evaluation methodology," in *Proc. IEEE 17th Int. Requirements Eng. Conf.*, 2009, pp. 133–142.
- [70] S. Ingolfo, A. Siena, and J. Mylopoulos, "Nōmos 3: Reasoning about regulatory compliance of requirements," in *Proc. IEEE 22nd Int. Requirements Eng. Conf.*, 2014, pp. 313–314.
- [71] G. Soltana, E. Fournet, M. Adedjouma, M. Sabetzadeh, and L. Briand, "Using UML for modeling procedural legal rules: Approach and a study of Luxembourg's tax law," in *Proc. Int. Conf. Model Driven Eng. Lang. Syst.*, Springer, 2014, pp. 450–466.
- [72] N. Zeni, N. Kiyavitskaya, L. Mich, J. R. Cordy, and J. Mylopoulos, "GaiusT: Supporting the extraction of rights and obligations for regulatory compliance," *Requirements Eng.*, vol. 20, no. 1, pp. 1–22, 2015.
- [73] P. Pullonen, J. Tom, R. Matulevicius, and A. Toots, "Privacy-enhanced BPMN: Enabling data privacy analysis in business processes models," *Softw. Syst. Model.*, vol. 18, no. 6, pp. 3235–3264, 2019.
- [74] E. Vanezi, G. M. Kapitsaki, D. Kouzapas, A. Philippou, and G. A. Papadopoulos, "DiálogoP - A language and a graphical tool for formally defining GDPR purposes," in *Proc. 14th Int. Conf. Res. Challenges Inf. Sci.*, Springer, 2020, pp. 569–575.
- [75] O. Amaral, S. Abualhaija, M. Sabetzadeh, and L. Briand, "A model-based conceptualization of requirements for compliance checking of data processing against GDPR," in *Proc. IEEE 29th Int. Requirements Eng. Conf. Workshops*, 2021, pp. 16–20.
- [76] O. Amaral, S. Abualhaija, D. Torre, M. Sabetzadeh, and L. Briand, "AI-enabled automation for completeness checking of privacy policies," *IEEE Trans. Softw. Eng.*, vol. 48, no. 11, pp. 4647–4674, Nov. 2022.
- [77] V. Ayala-Rivera and L. Pasquale, "The grace period has ended: An approach to operationalize GDPR requirements," in *Proc. IEEE 31st Int. Conf. Requirements Eng.*, 2018, pp. 136–146.
- [78] T. D. Breaux, M. W. Vail, and A. I. Anton, "Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations," in *Proc. IEEE 14th Int. Requirements Eng. Conf.*, 2006, pp. 49–58.
- [79] T. D. Breaux and A. I. Antón, "A systematic method for acquiring regulatory requirements: A frame-based approach," in *Proc. 6th Int. Workshop Requirements High Assurance Syst.*, 2007.
- [80] L. T. van Binsbergen, L.-C. Liu, R. van Doesburg, and T. van Engers, "eFLINT: A domain-specific language for executable norm specifications," in *Proc. 19th ACM SIGPLAN Int. Conf. Generative Program.: Concepts Experiences*, 2020, pp. 124–136.
- [81] X. Xu and H. Cai, "Ontology and rule-based natural language processing approach for interpreting textual regulations on underground utility infrastructure," *Adv. Eng. Inform.*, vol. 48, 2021, Art. no. 101288.
- [82] A. Mashkoor, M. Leuschel, and A. Egyed, "Validation obligations: A novel approach to check compliance between requirements and their formal specification," in *Proc. IEEE/ACM 43rd Int. Conf. Softw. Eng.: New Ideas Emerg. Results*, 2021, pp. 1–5.
- [83] S. Tokas, O. Owe, and T. Ramezanifarkhani, "Static checking of GDPR-related privacy compliance for object-oriented distributed systems," *J. Log. Algebr. Methods Program.*, vol. 125, 2022, Art. no. 100733.
- [84] P. Sangaroonsilp, H. K. Dam, M. Choetkertikul, C. Rakghitwetsagul, and A. Ghose, "A taxonomy for mining and classifying privacy requirements in issue reports," 2021, *arXiv:2101.01298*.
- [85] P. Klaus and R. Chris, *Requirements Engineering Fundamentals*, 1st ed. San Rafael, CA, USA: Rocky Nook, 2011.
- [86] A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, "Easy approach to requirements syntax (EARS)," in *Proc. IEEE 17th Int. Requirements Eng. Conf.*, 2009, pp. 317–322.
- [87] O. A. Krapivkina, "Semantics of the verb shall in legal discourse," *Jezikoslovje*, vol. 18, no. 2, pp. 305–317, 2017.
- [88] J. Bhatia, T. D. Breaux, and F. Schaub, "Mining privacy goals from privacy policies using hybridized task decomposition," *ACM Trans. Softw. Eng. Methodol.*, vol. 25, no. 3, 2016, Art. no. 22.
- [89] J. C. Maxwell and A. I. Antón, "The production rule framework: Developing a canonical set of software requirements for compliance with law," in *Proc. 1st ACM Int. Health Inform. Symp.*, New York, NY, USA, 2010, pp. 629–636.
- [90] G. Boella, L. D. Caro, L. Humphreys, L. Robaldo, P. Rossi, and L. 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," *Artif. Intell. Law*, vol. 24, no. 3, pp. 245–283, 2016.
- [91] S. Ingolfo, I. Jureta, A. Siena, A. Perini, and A. Susi, "Nomos 3: Legal compliance of roles and requirements," in *Proc. Int. Conf. Conceptual Model.*, Springer, 2014, pp. 275–288.
- [92] N. Zeni, E. A. Seid, P. Engiel, S. Ingolfo, and J. Mylopoulos, "Building large models of law with NōmosT," in *Proc. Int. Conf. Conceptual Model.*, Springer, 2016, pp. 233–247.
- [93] S. Ghanavati, A. Rifaut, E. Dubois, and D. Amyot, "Goal-oriented compliance with multiple regulations," in *Proc. IEEE 22nd Int. Requirements Eng. Conf.*, 2014, pp. 73–82.
- [94] A. K. Massey, P. N. Otto, and A. I. Antón, "Prioritizing legal requirements," in *Proc. IEEE 2nd Int. Workshop Requirements Eng. Law*, 2009, pp. 27–32.
- [95] J. C. Maxwell and A. I. Antón, "The production rule framework: Developing a canonical set of software requirements for compliance with law," in *Proc. 1st ACM Int. Health Inform. Symp.*, 2010, pp. 629–636.

- [96] W. B. Tesfay, P. Hofmann, T. Nakamura, S. Kiyomoto, and J. Serna, "PrivacyGuide: Towards an implementation of the EU GDPR on internet privacy policy evaluation," in *Proc. 4th ACM Int. Workshop Secur. Privacy Analytics*, 2018, pp. 15–21.
- [97] M. Lippi et al., "CLAUDETTE: An automated detector of potentially unfair clauses in online terms of service," *Artif. Intell. Law*, vol. 27, no. 2, pp. 117–139, 2019.
- [98] N. M. Nejad, P. Jabat, R. Nedelchev, S. Scerri, and D. Graux, "Establishing a strong baseline for privacy policy classification," in *Proc. IFIP Int. Conf. Inf. Commun. Technol. Syst. Secur. Privacy Protection*, Springer, 2020, pp. 370–383.
- [99] D. Sánchez, A. Viejo, and M. Batet, "Automatic assessment of privacy policies under the GDPR," *Appl. Sci.*, vol. 11, no. 4, 2021, Art. no. 1762.
- [100] R. E. Hamdani, M. Mustapha, D. R. Amariles, A. Troussel, S. Meeùs, and K. Krasnashchok, "A combined rule-based and machine learning approach for automated GDPR compliance checking," in *Proc. 18th Int. Conf. Artif. Intell.*, 2021, pp. 40–49.
- [101] A.-J. Aberkane, G. Poels, and S. V. Broucke, "Exploring automated GDPR-compliance in requirements engineering: A systematic mapping study," *IEEE Access*, vol. 9, pp. 66 542–66 559, 2021.
- [102] F. Liu, R. Ramanath, N. M. Sadeh, and N. A. Smith, "A step towards usable privacy policy: Automatic alignment of privacy statements," in *Proc. 25th Int. Conf. Comput. Linguistics*, Dublin, Ireland, 2014, pp. 884–894.
- [103] S. Wilson et al., "Crowdsourcing annotations for websites' privacy policies: Can it really work?," in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 133–143.
- [104] T. Linden, R. Khandelwal, H. Harkous, and K. Fawaz, "The privacy policy landscape after the GDPR," in *Proc. Privacy Enhancing Technol.*, vol. 2020, no. 1, pp. 47–64, 2020.
- [105] E. Park et al., "Development of automatic evaluation tool for mobile accessibility for Android application," in *Proc. IEEE Int. Conf. Syst. Collaboration Big Data Internet Things Secur.*, 2019, pp. 1–6.
- [106] J. Benjumea, E. Dorronzoro, J. Ropero, O. Rivera-Romero, and A. Carrasco, "Privacy in mobile health applications for breast cancer patients," in *Proc. IEEE 32nd Int. Symp. Comput.-Based Med. Syst.*, 2019, pp. 634–639.
- [107] L. Verderame, D. Caputo, A. Romdhana, and A. Merlo, "On the (un)reliability of privacy policies in Android apps," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2020, pp. 1–9.
- [108] D. S. Guaman, J. M. Del Alamo, and J. C. Caiza, "GDPR compliance assessment for cross-border personal data transfers in Android apps," *IEEE Access*, vol. 9, pp. 15 961–15 982, 2021.
- [109] M. Cozar, D. Rodriguez, J. M. Del Alamo, and D. Guaman, "Reliability of IP geolocation services for assessing the compliance of international data transfers," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops*, 2022, pp. 181–185.
- [110] M. Palmirani and G. Governatori, "Modelling legal knowledge for GDPR compliance checking," in *Proc. 31st Annu. Conf. Legal Knowl. Inf. Syst.*, M. Palmirani, Ed., IOS Press, Groningen, The Netherlands, 2018, pp. 101–110.
- [111] P. A. Bonatti, S. Kirrane, I. M. Petrova, and L. Sauro, "Machine understandable policies and GDPR compliance checking," *KI-Künstliche Intelligenz*, vol. 34, no. 3, pp. 303–315, 2020.
- [112] Z. S. Li, C. Werner, N. Ernst, and D. Damian, "GDPR compliance in the context of continuous integration," 2020, *arXiv:2002.06830*.
- [113] A. Nazarenko, F. Lévy, and A. Wyner, "A pragmatic approach to semantic annotation for search of legal texts - An experiment on GDPR," in *Proc. 34th Annu. Conf. Legal Knowl. Inf. Syst.*, S. Erich, Ed., IOS Press, Vilnius, Lithuania, Dec. 8–10, 2021, pp. 23–32.
- [114] R. Chen, F. Fang, T. Norton, A. M. McDonald, and N. Sadeh, "Fighting the fog: Evaluating the clarity of privacy disclosures in the age of CCPA," in *Proc. 20th Workshop Privacy Electron. Soc.*, 2021, pp. 73–102.
- [115] M. Degeling, C. Utz, C. Lentzsch, H. Hosseini, F. Schaub, and T. Holz, "We value your privacy... now take some cookies: Measuring the GDPR's impact on web privacy," *Informatik Spektrum*, vol. 42, pp. 345–346, 2019.
- [116] M. S. Rahman et al., "PermPress: Machine learning-based pipeline to evaluate permissions in app privacy policies," *IEEE Access*, vol. 10, pp. 89248–89269, 2022.
- [117] A.-J. Aberkane, S. V. Broucke, and G. Poels, "Investigating organizational factors associated with GDPR noncompliance using privacy policies: A machine learning approach," in *Proc. IEEE 4th Int. Conf. Trust Privacy Secur. Intell. Syst. Appl.*, 2022, pp. 107–113.



Orlando Amaral Cejas (Member, IEEE) received the BSc degree in automation engineering and the MSc degree in digital systems from the Technological University of Havana, Cuba, in 2012 and 2017. He is currently working toward the PhD degree as part of an industrial project on compliance checking of legal text according to the GDPR. During that time, he participated with the Improvement of the Cuban Intelligent Transport System. Currently, he is working as a doctoral researcher with SnT Centre for Security, Reliability and Trust, University of Luxembourg. His main research interests include: Regulatory compliance, model-driven engineering, applied natural language processing, text mining, and machine learning.



Muhammad Ilyas Azeem (Member, IEEE) received the PhD degree in software engineering from the Institute of Software, University of Chinese Academy of Sciences, Beijing, China, in 2020. He is working as a research associate with the SnT Centre for Security, Reliability, and Trust, University of Luxembourg. He is currently working on an industrial project on compliance checking of legal text according to the GDPR. His main research interests include requirement engineering, mining software repositories, natural language processing, machine learning, language models, and regulatory compliance.



Sallam Abualhaija (Member, IEEE) received the PhD degree in computer science from the Hamburg University of Technology, Germany, in 2016. She is a research scientist with the Interdisciplinary Centre for Security, Reliability, and Trust (SnT), University of Luxembourg. Her main research interests combine AI with software engineering, with an emphasis on: Requirements engineering, regulatory compliance, natural language processing, and machine learning. She believes in fostering stronger ties between academia and industry and has conducted the majority of her research in close collaboration with industry partners in the past years.



Lionel C. Briand (Fellow, IEEE) is a professor of software engineering and has shared appointments between (1) School of Electrical Engineering and Computer Science, University of Ottawa, Canada and (2) The SnT centre for Security, Reliability, and Trust, University of Luxembourg. He is the head of the SVV Department, SnT Centre and a Canada research chair in Intelligent Software Dependability and Compliance (Tier 1). He has conducted applied research in collaboration with industry for more than 25 years, including projects in the automotive, aerospace, manufacturing, financial, and energy domains. He is a fellow of ACM. He was also granted the IEEE Computer Society Harlan Mills award (2012), the IEEE Reliability Society Engineer-of-the-year award (2013), and the ACM SIGSOFT Outstanding Research Award (2022) for his work on software testing and verification. More details can be found on: <http://www.lbriand.info>.