An AI-assisted Approach for Checking the Completeness of Privacy Policies Against GDPR

Damiano Torre[a], Sallam Abualhaija[a],
Mehrdad Sabetzadeh[b,a], and Lionel Briand[a,b]
[a]SnT, University of Luxembourg, Luxembourg
[b]School of EECS, University of Ottawa, Canada
damiano.torre@uni.lu, sallam.abualhaija@uni.lu,
m.sabetzadeh@uottawa.ca, lbriand@uottawa.ca

Katrien Baetens, Peter Goes,
and Sylvie Forastier
Linklaters LLP, Luxembourg
katrien.baetens@linklaters.com, peter.goes@linklaters.com,
sylvie.forastier@linklaters.com

***Abstract*—Privacy policies are critical for helping individuals make informed decisions about their personal data. In Europe, privacy policies are subject to compliance with the General Data Protection Regulation (GDPR). If done entirely manually, checking whether a given privacy policy complies with GDPR is both time-consuming and error-prone. Automated support for this task is thus advantageous. At the moment, there is an evident lack of such support on the market. In this paper, we tackle an important dimension of GDPR compliance checking for privacy policies. Specifically, we provide automated support for checking whether the content of a given privacy policy is complete according to the provisions stipulated by GDPR. To do so, we present: (1) a conceptual model to characterize the information content envisaged by GDPR for privacy policies, (2) an AI-assisted approach for classifying the information content in GDPR privacy policies and subsequently checking how well the classified content meets the completeness criteria of interest; and (3) an evaluation of our approach through a case study over 24 unseen privacy policies. For classification, we leverage a combination of Natural Language Processing and supervised Machine Learning. Our experimental material is comprised of 234 real privacy policies from the fund industry. Our empirical results indicate that our approach detected 45 of the total of 47 incompleteness issues in the 24 privacy policies it was applied to. Over these policies, the approach had eight false positives. The approach thus has a precision of 85% and recall of 96% over our case study.**

***Index Terms*—Legal Compliance, Privacy Policies, The General Data Protection Regulation (GDPR), Natural Language Processing (NLP), Machine Learning (ML), Case Study Research.**

I. Introduction

In Europe and indeed worldwide, the General Data Protection Regulation (GDPR) [1] is widely viewed as a benchmark for data protection and privacy regulations. GDPR harmonizes data privacy laws across Europe, providing further protection to individuals for controlling their personal data in the face of new technological developments [2].

While undoubtedly beneficial to individuals in many ways, the reality is that organizations are having severe difficulties in understanding what compliance with GDPR means [3]. There is thus a pressing need for cost-effective methods that can help different business sectors deal with privacy issues. This need has not gone unnoticed by the research community. For example, Perrera et al. [4] propose systematic guidance to help software engineers develop privacy-aware applications; Torre et al. [5] propose the use of Model-driven Engineering as a platform for GDPR compliance automation; and Ayala-Rivera and Pasquale [6] present a stepwise approach for eliciting requirements related to GDPR compliance.

To comply with GDPR, organizations need to consider the principles of personal data processing set out in the regulation, and to regularly review their measures, practices and processes related to the collection, use and protection of personal data. Every organization, whether EU-based or not, which is collecting, processing or in some way handling the personal data of EU citizens and residents must comply with GDPR.

In this paper, we concern ourselves with GDPR *privacy policies.* A privacy policy can be viewed as a technical document stating the multiple privacy-related requirements that a system should satisfy in order to help users make informed decisions about the data an organization collects and uses. In other words, a privacy policy explains how an organization handles personal data and how it applies the GDPR principles. Privacy policies are usually defined through natural-language statements. Natural language is an ideal medium for expressing privacy policies since it is flexible and universal [7]. Despite these positive characteristics, natural language does not lend itself easily to automated analysis, and further, leaves ample room for quality issues such as incompleteness, inconsistency and ambiguity to occur [8].

This paper tackles an important dimension of GDPR compliance checking for privacy policies. Specifically, in collaboration with legal experts from Linklaters (a multinational law firm), we develop an AI-assisted approach for checking whether a given privacy policy is “complete” according to the provisions of GDPR. We use the term “complete” rather than “compliant” to signify the fact that our current approach can only detect the presence (or absence) of the information content types that GDPR envisages for privacy policies; we do not yet perform a deep semantic analysis of the detected content for verifying compliance.

The contributions of this paper are two-fold:

(1) We develop a conceptual model for characterizing the content of privacy policies, as per the provisions of GDPR. This conceptual model (a) provides an abstract and yet precise set of information elements that one can expect to find in GDPR privacy policies, and (b) serves as an enabler for automated completeness checking – the second contribution of this paper, described next.

(2) We devise an approach based on Natural Language Pro-

2332-6441/20/$31.00 ©2020 IEEE
DOI 10.1109/RE48521.2020.00025

cessing (NLP) and Machine Learning (ML) for automatically classifying the content of a given privacy policy. To do so, we use the information elements in the conceptual model developed in (1) as classification types, i.e., metadata, for the content of privacy policies. Subsequently, we use the automatically generated metadata to check whether a given policy meets the information requirements stipulated by GDPR.

As we discuss in more detail in the next sections, the metadata types relevant to GDPR privacy policies are numerous. Examples include: PROCESSING_PURPOSES to mark the purposes of the processing for which personal data is being collected, LEGAL_BASIS to mark the legal basis for the processing of personal data, and the data subject right ACCESS to mark the clause(s) giving an individual the right to request from the controller access to their personal data.

The paper investigates four Research Questions (RQs):

RQ1: What are the metadata types required for checking the completeness of a privacy policy according to GDPR? We answer RQ1 by building a conceptual model that specifies GDPR's information requirements for privacy policies. Our conceptual model was developed in close collaboration with subject-matter experts. The concepts in this model are described in a glossary and are further traceable to the articles of GDPR. Drawing on our conceptual model, we define a set of criteria for specifying how a privacy policy should be checked for completeness against GDPR.

RQ2: How can the metadata required for completeness checking of a privacy policy be extracted automatically? We answer RQ2 by (1) analyzing the text of privacy policies through multiple NLP techniques including parsing, similarity measures and word embeddings, and (2) extracting metadata types based on the NLP analysis performed alongside ML classification over word embeddings. Our metadata extraction approach is applicable to all the metadata types of the conceptual model resulting from RQ1.

RQ3: How accurately can we extract metadata from a given privacy policy? RQ3 examines the accuracy of our metadata extraction approach. For RQ3 and the subsequent RQ4, we scope ourselves to DATA_SUBJECT_RIGHT, LEGAL_BASIS, and the specializations of these two metadata types (see Fig. 1). Based on feedback from legal experts, these two metadata types and their descendants are the most critical and immediate information that experts need to check while verifying the completeness of a given privacy policy. Our results indicate that our approach achieves an average precision of 99% and 95% with an average recall of 93% and 89% for DATA_SUBJECT_RIGHT and LEGAL_BASIS, respectively.

RQ4: How accurately can we check the completeness of a given privacy policy? In RQ4, we investigate how well we can identify incompleteness in privacy policies (as per the provisions of GDPR). We answer RQ4 by applying the relevant completeness criteria identified in RQ1 to the metadata extracted in RQ3. On a test set made up of 24 privacy policies, our approach automatically detected 45 criteria violations out of a total of 47 in the gold standard, while also detecting eight false positives. Our completeness checking approach thus has a precision of 85% and recall of 96% over our test set.

Structure. Sec. II provides background. Sec. III presents the qualitative study we conducted for building our privacy-policy conceptual model. Sec. IV describes our approach for extracting privacy-policy metadata. Sec. V examines through a case study the accuracy of our metadata extraction and our completeness checking approach. Sec. VI discusses threats to validity. Sec. VII compares our contributions with related work. Sec. VIII concludes the paper.

II. BACKGROUND

GDPR. GDPR [1] is a complex regulation comprised of 173 recitals and 99 articles divided into 11 chapters. GDPR applies primarily to organizations within the EU. However, the regulation may also apply to organizations outside the EU, e.g., when these organizations offer goods or services to, or monitor individuals in the EU. If an organization is subject to GDPR, it has to identify itself as either a data controller or data processor. A controller determines the purpose and means of the processing, whereas a processor acts on the instructions of the controller. The responsibilities of a given organization under GDPR vary depending on whether it is a processor or a controller. Processors notably have to: (1) implement adequate technical and organizational measures to keep personal data safe and secure, and, in cases of data breaches, notify the controllers; (2) appoint a statutory data protection officer (if needed) and conduct a formal impact assessment for certain types of high-risk processing; (3) keep records about their data processing; and (4) comply to GDPR restrictions when transferring personal data outside the EU. In comparison to processors, controllers are subject to more provisions. In particular, in addition to having to meet the obligations mentioned above, controllers have to: (1) adhere to six core personal data processing principles, namely, fair and lawful processing, purpose limitation, data minimization, data accuracy, storage limitation, and data security; (2) keep identifiable individuals informed about how their personal data will be used; and (3) preserve the individual rights envisaged by GDPR, e.g., the right to be forgotten and the right to lodge a complaint. GDPR includes some specific provisions in relation to privacy policies. We discuss these provisions in Sec. III.

NLP and ML. Our proposed approach heavily relies on Natural Language Processing (NLP) and Machine Learning (ML). For the basic NLP pipeline, we use the DKPro toolkit [9]; this toolkit has already been used in the context of RE, e.g., see [10]. As part of our metadata identification approach, we transform words into a mathematical representation, called word embeddings [11], [12]. There are two well-known techniques for learning word embeddigs: Word2Vec [13] and GloVe [14]. We use GloVe's pre-trained models. Noting that our implementation is Java-based, we perform operations on word embeddings using Deeplearing4j [15]. For computing similarity between two textual entities, we use Cosine Similarity [16]. Our metadata identification approach further uses ML-based classification. For classification and handling imbalance in our dataset, we employ

WEKA [17], [18]. Due to space, we cannot provide a detailed introduction to all the ML machinery underlying our work. For this, we refer the reader to ML textbooks, e.g., [19].

III. A Conceptual Model of Privacy-Policy Metadata (RQ1)

In this section, we present the following three artifacts to answer RQ1: (1) a conceptual model specifying, in a comprehensive manner, the metadata types pertinent to GDPR privacy policies; (2) a glossary defining these metadata types with traceability to the articles of GDPR; (3) a set of completeness criteria for privacy policy as per the provisions of GDPR. Our conceptual model (artifact 1) is shown in Fig. 1. Our glossary and completeness criteria (artifacts 2 and 3) are provided as online annexes [20]. The above-mentioned artifacts were built using an interactive and incremental method in three main steps: (1) reading the articles of GDPR that address privacy policies, (2) creating and refining the artifacts introduced above, and (3) validating these artifacts with legal experts. Building the artifacts took four iterations with each iteration requiring, on average, one month. We had several face-to-face and off-line validation sessions with legal experts. The sessions, which lasted between two and three hours each, collectively added up to approximately 30 hours.

Initially, as suggested by our collaborating legal experts from Linklaters, we analyzed Art(icles) 13 and 14 of GDPR – the main GDPR articles targeting privacy policies – and extracted important concepts to create the metadata and the dependencies between them. Art. 13 focuses on personal data collected directly from a data subject (e.g., filling an online form or an interview), whereas Art. 14 focuses on personal data obtained indirectly from a data subject (e.g., obtained from a public website or public list). In particular, we observe that Art. 13.2(e) *(whether the provision of personal data is a statutory or contractual requirement, or a requirement necessary to enter into a contract, as well as whether the data subject is obliged to provide the personal data and of the possible consequences of failure to provide such data)* is related to the direct collection of personal data, while Art. 14.2(f) *(from which source the personal data originate, and if applicable, whether it came from publicly accessible sources)* deals with indirect collection. These observations were taken into consideration while building the three artifacts discussed above. In addition to Art. 13 and 14, and as per the recommendation of legal experts, we examined Art. 6, 9, 21, 37, 46, 47, 49, 55, and 56. Figure 2 illustrates an excerpt of Art. 13 from which we have inferred the hierarchical representation of four metadata types: Controller, Controller_Representative, and their descendants Identity and Contact. These metadata types refer to four distinct concepts: (1) the identity of the data controller (Controller.Identity), (2) the contact details of the data controller (Controller.Contact), (3) the identity of the data controller's representative (Controller_Representative.Identity), and (4) the contact details of the data controller's representative (Controller_Representative.Contact). The metadata types Identity and Contact were ultimately specialized with the inclusion of other sub-metadata types.

Our conceptual model in Fig. 1 is organized into three hierarchical levels: **level-1**, shaded yellow, **level-2**, shaded grey, and **level-3**, shaded white. The colors were introduced to make the model more readable to annotators and legal experts. The methodology we used for identifying the metadata types from GDPR and building the conceptual model is *hypothesis coding* [21]. Briefly, hypothesis coding refers to the application of a predetermined set of codes to qualitative data in order to assess researcher-generated hypothesis. The codes were developed from a prediction – in our case, based on a detailed reading of the relevant articles of GDPR – about what one would find in the actual data – in our case, actual privacy policies – before the data was collected and analyzed. In particular, based on GDPR, we created for privacy policies a model comprised of a set of codes. Usually, the application of this coding methodology can range from simple frequency counts to more complex multivariate analyses. In our context, we are interested in the presence or absence of metadata types in a given privacy policy in order to check its completeness according to GDPR. We created, with the help of legal experts, our conceptual model to be as close as possible to the terminology in GDPR.

Each metadata type we identified represents a code, which is a short phrase that symbolically assigns a summative, salient, essence-capturing, and/or evocative attribute to a portion of a given privacy policy [21]. For example, the metadata type Recipients refers to the text of the privacy policy where the recipients or categories of recipients of the personal data are specified (see Art. 13.1(e) of GDPR). In addition, we use *sub-coding* [21], which refers to sub-codes as a second-order tag assigned after a primary code, in order to enrich our metadata types in terms of specificity. For example, the metadata type PD_Origin (in yellow) is specialized into two sub-metadata types: Direct and Indirect (in gray). Then, Indirect is further specialized into: Third-Party, Publicity and Cookie (in white).

Based on our interpretation and understanding of GDPR articles, we created an initial version of the metadata conceptual model that contained 20 metadata types along with their definitions. We kept track of GDPR articles to ensure traceability in our glossary (artifact 2). Table I presents an excerpt of our glossary. These (interim) artifacts were then presented to legal experts for feedback. In addition to pointing out issues and omission, our collaborating legal experts were encouraged to bring to our attention any GDPR article or external documentation/information that needed to be considered in the context of privacy policies. The feedback obtained from legal experts was, by and large, concerned with information that was not explicitly included in GDPR (e.g., the European Working Party [22]). For example, Art. 13.1(f) states that "*the controller intends to transfer personal data to a third country or international organization and the existence or absence of an adequacy decision by the Commission, or [...] appropriate or suitable safeguards [...]*".

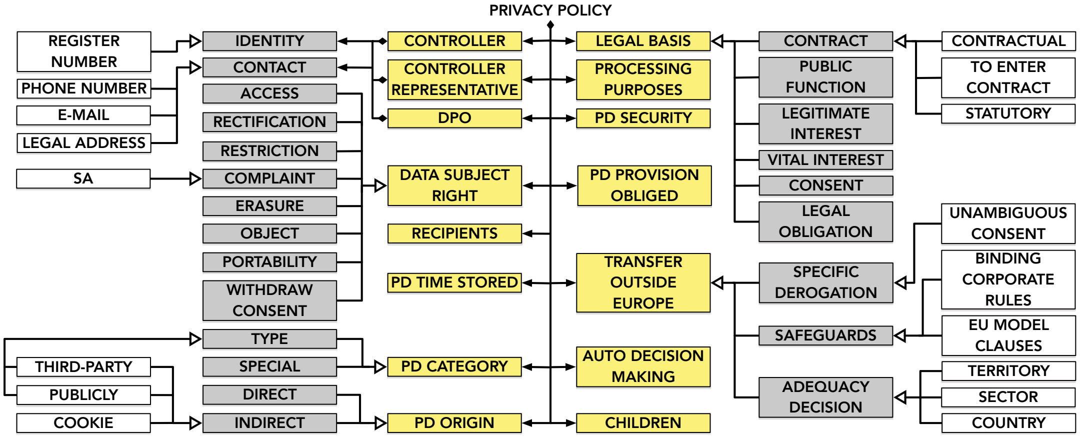
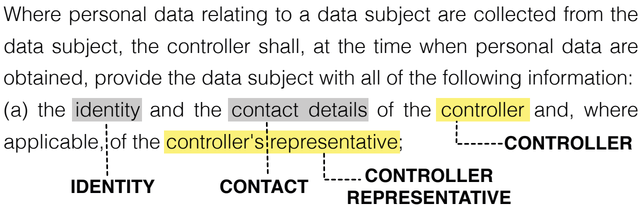


Fig. 1: Conceptual Model of Privacy-Policy Metadata.



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

Fig. 2: Example of Coding in the Context of GDPR.

This article is addressed in Fig. 1 by the metadata type TRANSFER_OUTSIDE_EUROPE.ADEQUACY_DECISION. In response to the legal experts' feedback, we created the sub-metadata types of ADEQUACY_DECISION that are not discussed in GDPR. Similarly, we created another level-3 sub-metadata type: EU_MODEL_CLAUSES.

Once the conceptual model started to stabilize, we created the proper dependencies between the metadata types in order to enable privacy-policy completeness checking. Legal experts asked us to create a questionnaire that would help them specify the exact content of a given privacy policy for completeness checking. This questionnaire contains a set of critical questions whose answers depend on context and are often left tacit in privacy policies. Nevertheless, the answers to these questions have important implications on what needs to be explicitly covered in privacy policies, and hence on completeness checking. The questionnaire is made of the following five questions: **Q1:** Do you plan to transfer the collected personal data outside Europe? *Yes/No*. **Q2:** (Q2.1): Is the processing of personal data carried out by a public authority or body (except for courts acting in their judicial capacity)? *Yes/No*; (Q2.2): Does the core of your activities consist of processing operations which, by nature, scope and/or purposes, require regular and systematic monitoring of data subjects on a large scale? *Yes/No*; (Q2.3): Does the core of your activities consist of processing on a large scale personal data relating to special categories (e.g., racial or ethnic origin, political opinions, or religious or philosophical beliefs) or to criminal convictions and offenses? *Yes/No*. **Q3:** Will there be other recipients of the collected personal data besides you? *Yes/No*. **Q4:** Where will the activities carried out by your organization take place? *Select country*. **Q5:** In addition to directly collecting personal data from the data subject, will this privacy policy also be used to indirectly collect personal data? *Yes/No*.

We then created the necessary dependencies between the metadata types of Fig. 1 and the possible answers to the questions presented above. Each of the five questions activates the checking of one or more metadata types. We have documented these dependencies for completeness checking in a set of completeness criteria. To facilitate the validation of these criteria with legal experts, we capture them as activity diagrams, following the observation by Soltana et al. [23] that legal experts can understand activity diagrams with relative ease given some basic training. For example, Art. 14.2(f) (discussed earlier) is addressed by PD_ORIGIN.INDIRECT (and its descendants) as described in Fig. 3.

The completeness criterion (represented by an activity diagram) in Fig. 3 uses three shapes to represent different types of actions or steps in a process: (1) a circle represents the start and endpoint, (2) a diamond indicates a decision, and (3) a rectangle stands for an action representing that (3.1) a metadata type was correctly found in a privacy policy, or that a metadata type was not needed in a privacy policy according to GDPR (in green), (3.2) a warning that corresponds to a metadata type that was partially found, in other words, a metadata type is found but some mandatory related information is missing (in orange), and (3.3) an error indicating that a mandatory metadata type was not found at all in a privacy policy (in red).

Figure 3 shows the criterion to check the completeness of a privacy policy with respect to the metadata type PD_ORIGIN.INDIRECT. This criterion (derived from Art. 13 and 14 of GDPR) does the following: **(1)** If the answer to Q5 is No, then checking PD_ORIGIN.INDIRECT is *not needed*; otherwise 2; **(2)** If the indirect origin of the personal data is not mentioned at all, then PD_ORIGIN.INDIRECT is *not found*; otherwise 3; **(3)** If the personal data is collected indirectly from third-parties, then PD_ORIGIN.INDIRECT.THIRD-PARTY is *found*; otherwise 4; **(4)** If the personal data is collected indirectly from public sources, then PD_ORIGIN.INDIRECT.PUBLICLY

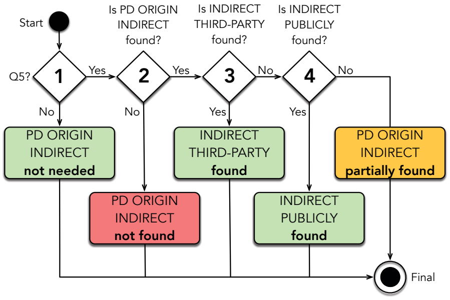

Start
Is PD ORIGIN INDIRECT found?
Is INDIRECT THIRD-PARTY found?
Is INDIRECT PUBLICLY found?
Q5?
1
Yes
2
Yes
3
No
4
No
No
No
Yes
Yes
PD ORIGIN INDIRECT not needed
INDIRECT THIRD-PARTY found
PD ORIGIN INDIRECT partially found
PD ORIGIN INDIRECT not found
INDIRECT PUBLICLY found
Final

Fig. 3: Completeness Criterion for PD_Origin.Indirect.

is *found*; otherwise PD_Origin.Indirect is *partially found* because the specific indirect source of the personal data (i.e., third-party or public) is missing.

Note that the criterion in Fig. 3 does not refer to Cookie although Cookie is a subtype of PD_Origin.Indirect in the conceptual model of Fig. 1. The above-shown criterion strictly follows GDPR, which does not regulate cookies. However, our collaborating legal experts suggested that coverage of cookies is almost always expected in privacy policies, hence the inclusion of Cookie in our conceptual model.

IV. Metadata Identification in Privacy Policies (RQ2)

In this section, we address RQ2 by developing a solution for automatically extracting metadata from privacy policies. Our proposed solution uses a combination of NLP and ML to perform sentence classification. In our context, *sentence* is the unit of analysis, and refers to the textual entity that results from applying an NLP sentence splitting module, irrespective of whether the sentences identified by such a module correspond to grammatical sentences. The rationale behind using sentences rather than phrases as units of analysis is that sentences are more likely to contain the context necessary for understanding their meaning [24] and thus lead to more accurate classification results.

To facilitate discussion throughout this section, we refer to three levels of metadata types, *level-1*, *level-2* and *level-3*, as introduced in Sec. III. Metadata extraction can be formulated as a hierarchical, multi-label and multi-class classification problem. The hierarchical, multi-class classification nature of the problem can be seen in the conceptual model of Fig. 1, where most level-1 metadata types are further classified into sub-metadata types (level-2 and level-3). Multi-label classification reflects the fact that a sentence can refer to one or more metadata types. Therefore, our solution can predict one or more potential labels (metadata types) for each sentence in a given privacy policy. In the following, we first explain how we collected the privacy policies from which we created our training and test sets. We then describe the details of our approach.

A. Collection of Privacy Policies

We have collected a total of 234 privacy policies from the fund domain. This domain is one of the main domains where our industry partner, Linklaters, is active in. Fund management companies aim at attracting national and foreign investors who are seeking a financial market where they can set up their investments. The impact of focusing on the fund domain, as we explain in Sec. VI, is that we cannot yet comment on the accuracy of our proposed automation outside this domain. At the same time, we must emphasize that the conceptual model described in Sec. III is domain-agnostic, noting that it was derived from GDPR and the (domain-independent) knowledge of legal experts about privacy policies. Out of the 234 policies, almost 60% were provided to us by Linklaters. For the remaining 40%, we downloaded privacy policies from companies in the fund registry of Luxembourg , which has a substantial footprint in fund management.

TABLE I: Glossary Excerpt.

Metadata	Reference	Intuitive Description
PD Origin	Art. 14.2(f)	From which source the personal data originates (i.e., direct or indirect), and if applicable, whether it came from a publicly and/or third-party and/or cookie sources.
Indirect	Art. 14	When the personal data are not obtained from the data subject.
Third-party	Art. 14	When the personal data are obtained from organisations external to the data controller.
Publicly	Art. 14	When the personal data are obtained from public sources (i.e., from a public website).

All 234 privacy policies were annotated according to the conceptual model of Fig. 1 [25], a subset of which was validated by experts. We partitioned the privacy policies into two batches. The first batch (30 policies) was annotated by one of the authors of this paper who has acquired domain expertise through close interaction with the collaborating law firm. During the annotation of the first batch, we kept track of the keywords that are frequently used to express certain metadata types. The second batch (209 policies) was annotated by four third-party individuals. Three of these individuals are graduate students in social sciences; they are native English speakers with considerable prior exposure to legal documents. The fourth annotator is a computer-science graduate student with an excellent command of English and six months of prior internship experience on a legal informatics project. All four annotators attended two four-hour training sessions, focused on GDPR concepts and the definitions of our metadata types. The annotators were further provided with detailed guidelines on how the metadata types should be annotated with examples from the first batch.

The annotators were asked to annotate each sentence in the privacy policies with the metadata types that they deemed to be present in the sentence. For example, if a sentence discusses the metadata type Access, then this sentence would be annotated (once) with Data_Subject_Right.Access. If the sentence happens to refer to metadata types other than Access, then the sentence would be annotated with *all* the applicable metadata types. For example, a sentence already annotated with Data_Subject_Right.Access can be further annotated with Data_Subject_Right.Erasure, PD_Time_Stored, and other metadata types. Following best practice, the entire document collection (234 privacy policies) is split randomly into two subsets containing about 90%

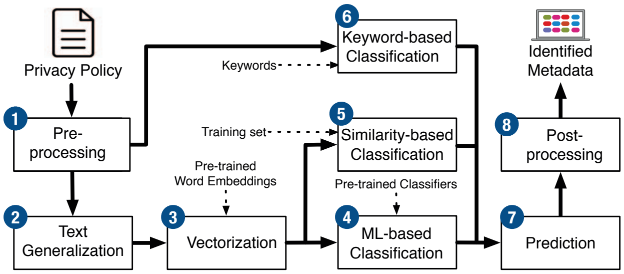


Fig. 4: Overview of Metadata Identification Approach.

and 10% of the policies, respectively used for training and development (210 policies) and for evaluation (24 policies).

B. *Metadata Identification Approach*

Figure 4 provides an overview of our approach. The approach takes as input a privacy policy and returns as output metadata annotations for each sentence in the input privacy policy. In the first three steps of the approach, the text of the input privacy policy is pre-processed and transformed into a mathematical representation. In Step 4, we classify the sentences of the input privacy policy using ML-based classifiers. Each classifier in this step is a predictor for the presence of a certain level-1 or level-2 metadata type in sentences. In Step 5, we classify the sentences by analyzing how similar each sentence in the input privacy policy is to the groups of sentences annotated with level-2 metadata types. In Step 6, we classify the sentences based on looking up pre-defined keywords and focusing on the metadata types at level-2 and level-3. In Step 7, we combine the results of Steps 4, 5, and 6 in order to predict metadata types for each sentence in the input privacy policy. Finally, in Step 8, we refine through some post-processing the results produced in Step 7. Below, we elaborate the steps of our approach.

1) Pre-processing: Pre-processing is the first step, in which the input privacy policy is parsed, and its text extracted and split into sentences. In this step, on top of the standard NLP pipeline – consisting of tokenization, sentence splitting, stopword removal, and lemmatization – we apply a named entity recognizer to obtain annotations for *organizations* and *locations* in the text.

2) Text Generalization: Generalization is the task of replacing the specific textual entities with more generic ones. Specifically, we replace named entities (as identified by the named entity recognizer module of Step 1) with their types. For example, the country name *United Kingdom* and the company name *Google LLC* will be replaced by *location* and *organization*, respectively. Similarly, we generalize emails, postal addresses, telephone numbers, and websites. The intuition behind this step is to normalize the text such that, despite the diversity in the privacy policies we use for training (e.g., the mentions of different locations), the approach can still learn common patterns and predict the metadata types with high accuracy.

3) Vectorization: The third step is to transform sentences into embeddings. For this, we utilize the pre-trained word-vector model from GloVe [14] (introduced Section II). This model represents 100-dimensional vectors generated by training on extensive text corpora from Wikipedia and the web. We use this word-vector model for creating sentence embeddings. First, we retrieve the corresponding embeddings for each word in a sentence as given by the pre-trained model, and then we average over all the word embeddings in the sentence to get a single vector representing the sentence embeddings [26].

4) ML-based Classification: In this step, we build ML classifiers for predicting level-1 and level-2 metadata types in each sentence of the input privacy policy. Currently, the number of positive examples we have in our training set is not sufficient for building accurate ML classifiers for level-3 metadata types. This is the reason why our current solution restricts ML classification to level-1 and level-2.

Our (level-1 and level-2) classifiers are trained on a feature matrix in which each row corresponds to a sentence and the columns are the 100-dimensional sentence embeddings from Step 3. The prediction class for each classifier is a level-1 or level-2 metadata type. This means that, Step 4 creates one classifier for each level-1 and level-2 metadata type in the model of Fig. 1. Inspired by Wang and Manning [27], we use SVM with its default hyper-parameters for sentence classification. We train the SVM classifiers with positive examples representing the sentences that have been annotated with a particular metadata type (e.g., DATA_SUBJECT_RIGHT) and negative examples annotated with any other metadata type at the same level (e.g., all but DATA_SUBJECT_RIGHT). In most of the cases, we had an imbalanced dataset with positive examples being under-represented; we thus had to perform under-sampling (over negative examples) [19].

5) Similarity-based Classification: In this step, we classify the sentences of the input privacy policy based on how similar each sentence in the policy is to the group of sentences that have a certain level-2 metadata annotation (e.g., DATA_SUBJECT_RIGHT.ACCESS) as per the training set discussed in Sec. IV-A. Restricting similarity-based classification to level-2 was prompted by this type of classification not being conclusive enough for level-1 and level-3 metadata types, as observed in our experiments.

Similar to Step 4, Step 5 characterizes each sentence using the vector representation built in Step 3. Since an individual sentence can have multiple metadata type annotations, the same sentence embeddings can be part of several groups. Step 5 creates one group for each level-2 metadata type. Each group is represented by a single vector, computed by averaging the embeddings of all the sentences in that group. To predict whether a sentence S should be annotated with a certain level-2 metadata type M, we compute the *cosine similarity* between S and the averaged vector of the group of sentences annotated by M (in the training set). If the cosine similarity is above a pre-specified threshold, we predict M to be a metadata type for S. We set the value of this threshold to 0.9. This threshold was arrived at empirically by evaluating the accuracy of the prediction using a range of similarity threshold values between 0.6 and 0.9 with a step of 0.1 on a subset of the privacy policies in the training set.

6) Keyword-based Classification: As explained in Sec. IV-A, while annotating the privacy policies, we developed a list of keywords for level-2 and level-3 metadata types. We observed that the sentences referring to level-1 metadata types have less in common than those from level-2 and level-3 and, therefore, keeping keywords does not help at that level. In this step, we conduct a keyword search over the sentences in the input privacy policy. If a sentence S contains one or more of the keywords associated with metadata type M, then we predict that S should be annotated with M. For example, if a given sentence contains the keyword *oppose*, this sentence will be predicted as having DATA_SUBJECT_RIGHT.OBJECT as an annotation. Our keyword lists contain a total of 183 entries across level-2 and level-3 metadata types. The keywords will be made publicly available if the paper is accepted (see footnote 1 on page 1).

7) Prediction: This step combines the ML-based (Step 4), similarity-based (Step 5) and keyword-based (Step 6) classifications to produce a final recommendation about which metadata types should be ascribed to a given sentence. Our strategy for combining the above three classifications is elaborated in Algorithm 1. The algorithm predicts the metadata types relevant to a given sentence at all levels (level-1, level-2 and level-3). We explain the details of this algorithm, next.

The algorithm starts with an initially empty set of annotations, $\mathcal{M}$ (Line 1). It then predicts level-1 and level-2 metadata types by considering two cases, Case 1 (Lines 2–13) and Case 2 (Lines 14–21), as described below. Case 1 applies when some level-1 metadata type has been predicted for a sentence through ML-based classification. Case 2 applies otherwise. Note that Case 1 and Case 2 may attempt to add to $\mathcal{M}$ the same level-1 metadata annotation multiple times. In such cases, only one copy of the annotation is retained (since $\mathcal{M}$ is a set).

Case 1: For each level-1 metadata type ℓ_i predicted via ML-based classification (cf_1), the algorithm checks (Line 4) if ℓ_i is specialized by any level-2 metadata type in the conceptual model of Fig. 1. If ℓ_i has no specializations, ℓ_i is predicted as a metadata type for the given sentence (Line 5). Otherwise, a more elaborate analysis is performed based on level-2 metadata types (Lines 7–11). Specifically, the algorithm checks the similarity-based and keywords-based predictions for every level-2 metadata type, ℓ_j, that specializes ℓ_i. If the similarity between the sentence vector, $\vec{S}$, and the average vector of the associated group, $\vec{av}(\ell_j)$, is above the threshold of 0.9, and further, ℓ_j is predicted by keyword search as well, then both ℓ_i and ℓ_j are added to $\mathcal{M}$.

Case 2: For any level-1 metadata type ℓ'_i that has not been predicted by ML-based classification, we check whether any specialization ℓ_j of ℓ'_i has been predicted by ML-based classification (cf_2). If this prediction is confirmed by either the similarity-based or keyword-based classification, then both ℓ'_i and ℓ_j are added to $\mathcal{M}$.

Finally, the algorithm attempts to predict level-3 metadata types (Lines 22–28) based on any already predicted level-2 metadata types. Specifically, the algorithm considers all level-3 metadata types that specialize some level-2 metadata type already in $\mathcal{M}$. Any considered level-3 metadata type that is also predicted by keyword search is added to $\mathcal{M}$.

Algorithm 1 Metadata Prediction for a Sentence S

```
Require: $\vec{S}$: vector representation of $S$; $cf_1$, $cf_2$: binary
    classifiers trained on level-1 and level-2 metadata types,
    respectively; $\vec{av}(t)$: average vector for the group of sen-
    tences annotated with metadata type $t$; $\mathcal{K}$: set of metadata
    types predicted based on keyword search in $S$.
Output: A set, $\mathcal{M}$, of metadata types predicted for $S$
 1: $\mathcal{M} \leftarrow \emptyset$
 2: Let $\mathcal{L}_1$ be the set of (level-1) metadata types predicted for
    $\vec{S}$ by $cf_1$                                   // Case 1 begins
 3: for $\ell_i \in \mathcal{L}_1$ do
 4:   if $\ell_i$ has no level-2 specializations then
 5:     Add $\ell_i$ to $\mathcal{M}$
 6:   else
 7:     for $\ell_j$ s.t. $\ell_j$ is a (level-2) specialization of $\ell_i$ do
 8:       if sim$(\vec{S}, \vec{av}(\ell_j)) \geq 0.9$ and $\ell_j \in \mathcal{K}$ then
 9:         Add both $\ell_i$ and $\ell_j$ to $\mathcal{M}$
10:       end if
11:     end for
12:   end if
13: end for                                              // Case 1 ends
14: Let $\mathcal{L}'_1$ be the set of (level-1) metadata types not predicted
    for $\vec{S}$ by $cf_1$                               // Case 2 begins
15: for $\ell'_i \in \mathcal{L}'_1$ do
16:   for $\ell_j$ s.t. $\ell_j$ is a (level-2) specialization of $\ell'_i$ do
17:     if $\ell_j$ is predicted for $\vec{S}$ by $cf_2$ and
        (sim$(\vec{S}, \vec{av}(\ell_j)) \geq 0.9$ or $\ell_j \in \mathcal{K}$) then
18:       Add both $\ell'_i$ and $\ell_j$ to $\mathcal{M}$
19:     end if
20:   end for
21: end for                                              // Case 2 ends
22: for level-2 metadata type $\ell_j \in \mathcal{M}$ do   // Predict level-3
23:   for level-3 metadata type $\ell_q$ specializing $\ell_j$ do
24:     if $\ell_q \in \mathcal{K}$ then
25:       Add $\ell_q$ to $\mathcal{M}$
26:     end if
27:   end for
28: end for
```

8) Post-processing: In the eighth and final step of our approach, we refine the results of Step 7 by considering the metadata types predicted for the sentences surrounding a given sentence. The intuition behind this step is the observation that certain metadata types, e.g., DATA_SUBJECT_RIGHT (DSR), are discussed in consecutive sentences of privacy policies. Based on this observation, when a sentence S is predicted as having a specific metadata type M as an annotation, the surrounding context, specifically, the preceding and succeeding sentences, can provide a confirmatory measure as to whether M is a reliable prediction for S.

The number of sentences that we consider as potentially being in the same context depends on the number of metadata types in level-2. For example, eight sentences (before and after) are considered to be potentially in context for DSR because there are eight DSR level-2 metadata types in total.

We employ several such context-based heuristics for post-processing (not listed here due to space). For example, in relation to DSR, we apply the following heuristic: if a sentence S is predicted as having DSR_i (e.g., Data_Subject_Right.Access) as an annotation, then we look at the eight preceding and eight succeeding sentences. If none of these surrounding sentences discuss some DSR_j (e.g., Data_Subject_Right.Erasure), then we remove DSR_i from the annotations for S; this is because the context around S lends no support for DSR_i being a correct annotation for S.

V. Case Study

In this section, we describe the case study through which we answer RQ3 and RQ4. RQ3 aims to assess how accurate our automated metadata identification method is, whereas RQ4 aims to examine the accuracy of our overall approach for checking the completeness of privacy policies according to GDPR. We present our case study in terms of: (1) objectives and design (Sec. V-A); (2) defining the completeness criteria of interest for the case study (Sec. V-B); (3) identifying metadata in privacy policies (Sec. V-C); and (4) checking the completeness of privacy policies (Sec. V-D).

A. Objectives and Design

We evaluate our approach on the test set (as explained in Sec. IV-A) with a total of 24 privacy policies that are not used during training or development. We focus our evaluation on two metadata types: Data_Subject_Rights (DSR) and Legal_Basis (LB) and their descendants (see Fig. 1). We scope our empirical evaluation to these metadata types for two main reasons: (1) **importance**: legal experts considered these types to be the first to check when verifying the completeness of a privacy policy, and (2) **interdependence**: the selected metadata types are interdependent. By interdependence, we mean that the metadata types should appear together in a privacy policy due to the conceptual relationship that exists between the metadata types. For example, if the LB sub-metadata type Contract is present, then DSR sub-metadata type Portability needs to be present as well.

B. Completeness Criteria

In our case study, an incompleteness issue is raised when at least one of the following completeness criteria is violated:

C1: The following DSR sub-metadata types must always be present: Access, Complaint, Rectification, and Restriction. The absence of any one of these would render a privacy policy incomplete.

C2: If the LB sub-metadata type Contract is present, then the DSR sub-metadata type Portability should be present as well.

C3: If either of the LB sub-metadata types Legitimate_Interest or Public_Function are present, then the DSR sub-metadata type Object should be present as well.

C4: If the LB sub-metadata type Consent is present, then the DSR sub-metadata types Erasure, Object, Portability, and Withdraw_Consent should be present as well.

TABLE II: Accuracy Results for Metadata Identification.

DSR	P%	R%	LB	P%	R%
Access	100	91	Consent	95	100
Complaint	100	100	Contract	90	95
Complaint.SA	100	100	Contract.To Enter Contract	100	87
Erasure	100	89	Contract. Contractual	94	100
Object	94	94	Contract. Statutory	83	100
Portability	100	100	Legal Obligation	100	96
Rectification	100	95	Legitimate Interest	100	81
Restriction	100	94	Public Function	75	60
Withdraw Consent	100	94	Vital Interest	100	80

We note that, for a given privacy policy, C2 and C3 check for one incompleteness issue each, whereas C1 and C4 check for four different incompleteness issues each.

C. Identifying Metadata in Privacy Policies (RQ3)

In this section, we assess the accuracy of our approach in identifying the metadata types DSR and LB in privacy policies. To measure accuracy, we use as gold standard the manually annotated versions of the 24 privacy policies in our test set. Our approach works at the sentence level, as explained in Sec. IV. However, to answer RQ3, we need to evaluate the presence or absence of the metadata types of interest irrespectively of the number of sentences containing these metadata types. For example, if five sentences happen to discuss the metadata type Data_Subject_Right.Access, and at least one of them is correctly identified by our approach, then we consider this metadata type to be correctly identified for the underlying privacy policy. This way of measuring accuracy is motivated by our objective, which is checking completeness; as can be seen from the criteria presented in Sec. V-B, what one needs to ascertain to be able to verify the criteria is the presence or absence of the metadata types within an entire privacy policy.

To evaluate our metadata identification approach, we define: (1) *True Positives (TPs)* to be the cases where at least one sentence is correctly identified for an expected metadata type according to the gold standard, (2) *False Positives (FPs)* to be the cases where an incorrect metadata type is detected, and (3) *False Negatives (FNs)* to be the cases where we miss a metadata type that exists in the gold standard. Following this definitions, we compute precision ($TP/(TP+FP)$) and recall ($TP/(TP+FN)$) across the test privacy policies for the different sub-metadata types of DSR and LB.

Table II reports the accuracy of metadata identification for DSR and LB. As shown by the table, our approach achieves a precision of 100% and a recall greater than 90% for seven out of the eight DSR level-2 types, and perfect precision and recall for the only DSR level-3 metadata type (Complaint.SA). Achieving high precision implies that our approach is able to correctly identify the metadata types considered with zero or very few FPs. In the case of Object, the approach yields one FP out of the total of 18

TABLE III: Accuracy Results for Incompleteness Detection.

#	TPs	FPs	FNs	P%	R%
C1	16	4	0	80	100
C2	5	1	0	83	100
C3	3	0	2	100	60
C4	21	3	0	88	100
Summary	**45**	**8**	**2**	**85**	**96**

instances where OBJECT is predicted. For LB, our approach yields a precision greater than 90% for all of the level-2 and level-3 metadata types except PUBLIC_FUNCTION and CONTRACT.STATUTORY, and a recall greater than 90% except for LEGITIMATE_INTEREST, PUBLIC_FUNCTION, VITAL_INTEREST and CONTRACT.TO_ENTER_CONTRACT. With regard to PUBLIC_FUNCTION and VITAL_INTEREST, we note that these metadata types are rare in the fund domain. We, therefore, have few examples in our experimental material (both for training and development and for testing). As expected, the accuracy of our approach is notably lower for these metadata types. Our approach leads to four FNs over the metadata type LEGITIMATE_INTEREST. Sentences related to this metadata type tend to be rather generic and vague, thus making it difficult to conclusively identify LEGITIMATE_INTEREST. This explains the lower recall.

The answer to RQ3 is: Our approach achieves an average precision of 99% and 95%, and an average recall of 93% and 89% for DATA_SUBJECT_RIGHT and LEGAL_BASIS, respectively.

D. Completeness Checking of Privacy Policies (RQ4)

While the accuracy of metadata identification has a direct influence on the accuracy of completeness checking, RQ3 per se does not measure the accuracy of completeness checking; this is instead the subject of RQ4.

To answer RQ4, we need to redefine TP, FP, and FN in the context of completeness checking. Specifically, we define: (1) *TPs* as cases where the approach correctly identifies a genuine violation of some completeness criterion, (2) *FPs* as cases where some criterion that is actually satisfied by the policy is incorrectly found to be violated, and (3) *FNs* as cases where there is actually a violation of some criterion but this violation goes undetected by our approach.

In Tab. III, we show our accuracy results for completeness checking. We have organized the results in this table according to the four criteria, C1-C4, presented in Sec. V-B. Below, we discuss the results for each of the four criteria.

C1: All 16 actual incompleteness issues are detected alongside four FPs. These FPs are caused by the metadata identification phase having missed some DSR sub-metadata types: ACCESS (two instances), RECTIFICATION (one instance), and RESTRICTION (one instance).

C2: All five actual incompleteness issues are detected. The metadata identification phase incorrectly detected the presence of the LB sub-metadata type CONTRACT (one instance). This prompted checking the presence of the DSR sub-metadata type PORTABILITY, which is absent. This results in one FP.

C3: Three genuine incompleteness issues are detected and two are missed. The missed issues stem from the metadata identification phase missing the LB sub-metadata types: LEGITIMATE_INTEREST (one instance) and PUBLIC_FUCNTION (one instance). There are no FPs for C3.

C4: All 21 actual incompleteness issues are identified alongside three FPs. The FPs are due to the metadata identification phase missing the DSR sub-metadata types: ERASURE (two instances), and WITHDRAW_CONSENT (one instance).

When checking the completeness of privacy policies, recall is more important than precision. This is because a human analyst can dismiss the FPs (false alarms) with relative ease as long as there are not too many of them. On the other hand, a deficit in recall means that the automation is missing incompleteness issues. Unless recall is very high, an expert is thus unlikely to trust the automatically generated results and may opt for a fully manual analysis.

At a privacy-policy level, our approach is able to correctly identify all the privacy policies that have some incompleteness issue (10 out of the 24 privacy policies in the test set), while raising a false alarm on only two (out of the 24) policies. This means that no incomplete policy could pass through our approach undetected, although our approach did not have perfect recall on one of the criteria, namely C3.

The answer to RQ4 is: Our approach is able to detect 45 out of the total of 47 incompleteness issues in the 24 privacy policies in our test set. Over these policies, the approach had eight false positives. The approach thus has a precision of 85% and recall of 96% over our case study. At the level of policies, the approach could identify all the incomplete policies (i.e., a recall of 100%) while mistakenly identifying two complete policies as incomplete (i.e., a precision of 10 / (10+2) $\simeq$ 83%).

VI. THREATS TO VALIDITY

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

Internal Validity. Bias was an important concern in relation to internal validity. To mitigate bias, we curated most ($\sim$90%) of the manual annotations through third-parties (non-authors). Another potential threat to internal validity is that the authors interpreted the text of GDPR provisions in order to create the privacy policy conceptual model presented in Fig. 1. To minimize the threat posed by subjective interpretation, this phase was done in close collaboration with independent legal experts (lawyers specialized in data protection). In addition, our conceptual model is explicit and thus open to scrutiny.

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

VII. Related Work

In this section, we discuss related work on (1) identifying privacy-policy requirements, and (2) completeness checking of privacy policies. It is important to note that our discussion of related work is not meant at providing a detailed coverage of the already extensive literature on GDPR. For the purposes of this paper, when it comes to GDPR, we are interested in only those threads of work that address privacy policies.

Identifying privacy-policy requirements. Caramujo et al. [7] target privacy policies from the web and mobile applications, and propose a domain-specific language along with model transformations for specifying privacy-policy models. Similarly, Pullonen et al. [28] present a multi-level model to be used as an extension of the Business Process Model and Notation to enable the visualization, analysis, and communication of the privacy-policy characteristics of business processes. Finally, Kumar and Shyamasundar [29] explore the suitability of information-flow controls as a tool for specifying and enforcing privacy-policy requirements. These existing works address a subset of the privacy-policy metadata types discussed in this paper. In addition, all of them focus on providing guidelines that are not strictly based on GDPR. In contrast, we systematically identify the requirements that, according to GDPR, must be met by privacy policies for completeness.

Checking the completeness of privacy policies. Tesfay et al. [30] propose an ML-based method for classifying the content of privacy policies across multiple categories using predefined keywords. Bhatia et al. [31] develop a semi-automated framework for extracting privacy goals from privacy policies through crowdsourcing and NLP. Further, there are crowdsourcing initiatives like the ones presented by Liu et al. [32] and Wilson et al. [33], where privacy policies are manually annotated in order to match their text segments against privacy issues of interest. Guerriero et al. [34] proposed a framework for specifying, enforcing and checking privacy policies in data-intensive applications. Bhatia et al. [35] present a semantic frame-based representation for privacy statements that can be used to identify incompleteness in four categories of data action: collection, retention, usage, and transfer. Lippi et al. [36] present 33 metadata types for GDPR privacy policies and provide automatic support for vagueness detection based on manually crafted rules and (terminology-dependent) ML. Our work in this paper has a different analytical focus, namely completeness checking. In terms of metadata types, the set of 55 types that we propose cover all the ones identified by Lippi et al., except for one metatadata type, Policy Change, which is orthogonal to our purposes. These approaches rely to a large extent on the exact phrasing of the policies to be able to extract and classify information. They do not present a thorough conceptualization of the content expected in privacy policies. The scope of application of these approaches is thus limited and, where automation is provided, the accuracy is not high enough for industrial use. In this paper, we addressed the above limitations by considering a wider set of metadata types and using an advanced combination of NLP and ML for automated support.

VIII. Conclusion

In this paper, we proposed an AI-enabled approach for completeness checking of privacy policies according to the General Data Protection Regulation (GDPR). We first developed a conceptual model aimed at providing a thorough characterization of the content of privacy policies. Based on this conceptual model, we devised criteria describing how a privacy policy should be checked for completeness against GDPR. Second, using Natural Language Processing and Machine Learning, we developed automated support for classifying the content of privacy policies and thus providing the information elements necessary for checking privacy-policy completeness.

We curated a considerable number of annotated privacy policies (234 policies in total), with the majority of the annotation work performed by third-parties. We evaluated our approach through a case study designed around two critical classes of metadata types, having to do with data subject rights and the legal basis for the processing of personal data. Our metadata identification approach achieved an average precision of 99% and 95% with an average recall of 93% and 89% for identifying these two classes of metadata types, respectively. We ran the relevant completeness criteria over the identified metadata. Our completeness checking approach was able to detect 45 out of the total of 47 incompleteness issues in the real-world privacy policies we used for validation. The approach generated eight false positives. Our completeness checking approach thus had a precision of 85% and a recall of 95% over our case study.

In the future, we plan to expand our metadata identification approach to cover a broader set of metadata types. Moreover, we would like to enhance our completeness criteria so that they consider not only the presence/absence of metadata but also the meaning of the sentences containing the metadata. Another important direction for future work is to go beyond our current case-study domain (funds) in order to assess the generalizability of our approach.

Acknowledgments

This paper was supported by Linklaters, Luxembourg's National Research Fund (FNR) under grant BRIDGES/19/IS/13759068/ARTAGO, and NSERC of Canada under the Discovery, Discovery Accelerator and CRC programs.

References

[1] European Union, "General data protection regulation," *Official Journal of the European Union*, 2018. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679

[2] EU-GDPR. (2019) EU GDPR portal. [Online]. Available: https://eugdpr.org

[3] C. Tankard, "What the GDPR means for businesses," *Network Security*, vol. 6, pp. 5–8, 2016.

[4] C. Perera, M. Barhamgi, A. K. Bandara, M. Ajmal, B. A. Price, and B. Nuseibeh, "Designing privacy-aware internet of things applications," *Inf. Sci.*, vol. 512, pp. 238–257, 2020.

[5] D. Torre, G. Soltana, M. Sabetzadeh, L. C. Briand, Y. Auffinger, and P. Goes, "Using models to enable compliance checking against the GDPR: an experience report," in *22nd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS 2019, Munich, Germany, September 15-20, 2019*, 2019, pp. 1–11.

[6] V. Ayala-Rivera and L. Pasquale, "The grace period has ended: An approach to operationalize GDPR requirements," in *Proceedings of 31st IEEE International Conference on Requirements Engineering (RE'18)*, 2018, pp. 136–146.

[7] J. Caramujo, A. Rodrigues da Silva, S. Monfared, A. Ribeiro, P. Calado, and T. Breaux, "RSL-IL4Privacy: A domain-specific language for the rigorous specification of privacy policies," *Requirements Engineering*, vol. 24, no. 1, pp. 1–26, 2019.

[8] J. Bhatia and T. D. Breaux, "Semantic incompleteness in privacy policy goals," in *26th IEEE International Requirements Engineering Conference, RE 2018, Banff, AB, Canada, August 20-24, 2018*, 2018, pp. 159–169.

[9] R. Eckart de Castilho and I. Gurevych, "A broad-coverage collection of portable NLP components for building shareable analysis pipelines," in *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT (OIAF4HLT'14)*, 2014, pp. 1–11.

[10] S. Abualhaija, C. Arora, M. Sabetzadeh, L. Briand, and E. Vaz, "A machine learning-based approach for demarcating requirements in textual specifications," in *Proceedings of the 27th IEEE International Requirements Engineering Conference (RE'19)*, 2019.

[11] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 746–751.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[14] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[15] E. D. D. Team, "Deeplearning4j: Open-source distributed deep learning for the jvm, apache software foundation license 2.0," 2020, last accessed: January 2020. [Online]. Available: http://deeplearning4j.org

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

[17] J. H. Hayes, W. Li, and M. Rahimi, "Weka meets tracelab: Toward convenient classification: Machine learning for requirements engineering problems: A position paper," in *Artificial Intelligence for Requirements Engineering (AIRE), 2014 IEEE 1st International Workshop on*. IEEE, 2014, pp. 9–12.

[18] F. Eibe, M. Hall, and I. Witten, "The weka workbench. online appendix for" data mining: Practical machine learning tools and techniques," *Morgan Kaufmann*, 2016.

[19] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. Morgan Kaufmann, 2016.

[20] D. Torre, S. Abualhaija, M. Sabetzadeh, and L. C. Briand, *Glossary and completeness criteria*, available at http://tiny.cc/x3gkqz,http://tiny.cc/5il9jz, June 2020.

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

[22] European Union, "Article 29 working party - guidelines on data protection officers (dpos)," *Justice and Consumers*, 2018.

[23] G. Soltana, N. Sannier, M. Sabetzadeh, and L. C. Briand, "Model-based simulation of legal policies: Framework, tool support, and validation," *Software & Systems Modeling*, vol. 17, no. 3, pp. 851–883, 2018.

[24] L. Michaelis, "Word meaning, sentence meaning, and syntactic meaning," *Cognitive approaches to lexical semantics*, vol. 23, pp. 163–209, 2003.

[25] D. Torre, S. Abualhaija, M. Sabetzadeh, and L. C. Briand, *Dataset of privacy policies annotated*, available at http://tiny.cc/n33xqz, June 2020.

[26] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[27] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2*. Association for Computational Linguistics, 2012, pp. 90–94.

[28] P. Pullonen, J. Tom, R. Matulevicius, and A. Toots, "Privacy-enhanced BPMN: Enabling data privacy analysis in business processes models," *Software & Systems Modeling*, pp. 1–30, 2019.

[29] N. V. N. Kumar and R. K. Shyamasundar, "Realizing purpose-based privacy policies succinctly via information-flow labels," in *2014 IEEE Fourth International Conference on Big Data and Cloud Computing, BDCloud 2014, Sydney, Australia, December 3-5, 2014*, 2014, pp. 753–760.

[30] 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 *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics, IWSPA@CODASPY 2018, Tempe, AZ, USA, March 19-21, 2018*, 2018, pp. 15–21.

[31] J. Bhatia, T. D. Breaux, and F. Schaub, "Mining privacy goals from privacy policies using hybridized task recomposition," *ACM Trans. Softw. Eng. Methodol.*, vol. 25, no. 3, pp. 22:1–22:24, 2016.

[32] F. Liu, R. Ramanath, N. M. Sadeh, and N. A. Smith, "A step towards usable privacy policy: Automatic alignment of privacy statements," in *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, 2014, pp. 884–894.

[33] S. Wilson, F. Schaub, R. Ramanath, N. M. Sadeh, F. Liu, N. A. Smith, and F. Liu, "Crowdsourcing annotations for websites' privacy policies: Can it really work?" in *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, 2016, pp. 133–143.

[34] M. Guerriero, D. A. Tamburri, and E. D. Nitto, "Defining, enforcing and checking privacy policies in data-intensive applications," in *Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems, SEAMS@ICSE 2018, Gothenburg, Sweden, May 28-29, 2018*, 2018, pp. 172–182.

[35] J. Bhatia, M. C. Evans, and T. D. Breaux, "Identifying incompleteness in privacy policy goals using semantic frames," *Requir. Eng.*, vol. 24, no. 3, pp. 291–313, 2019.

[36] M. Lippi, P. Pałka, G. Contissa, F. Lagioia, H.-W. Micklitz, G. Sartor, and P. Torroni, "Claudette: an automated detector of potentially unfair clauses in online terms of service," *Artificial Intelligence and Law*, vol. 27, no. 2, pp. 117–139, 2019.