PolicyChecker: Analyzing the GDPR Completeness of Mobile Apps' Privacy Policies

Anhao Xiang
Department of Computer Science
Colorado School of Mines
Golden, Colorado, USA
xianganhao@mines.edu

Weiping Pei
School of Cyber Studies
The University of Tulsa
Tulsa, Oklahoma, USA
weiping-pei@utulsa.edu

Chuan Yue
Department of Computer Science
Colorado School of Mines
Golden, Colorado, USA
chuanyue@mines.edu

ABSTRACT

The European General Data Protection Regulation (GDPR) mandates a data controller (e.g., an app developer) to provide all information specified in Articles (Arts.) 13 and 14 to data subjects (e.g., app users) regarding how their data are being processed and what are their rights. While some studies have started to detect the fulfillment of GDPR requirements in a privacy policy, their exploration only focused on a subset of mandatory GDPR requirements. In this paper, our goal is to explore the state of GDPR-completeness violations in mobile apps' privacy policies. To achieve our goal, we design the *PolicyChecker* framework by taking a *rule and semantic role based* approach. PolicyChecker automatically detects completeness violations in privacy policies based not only on all mandatory GDPR requirements but also on all if-applicable GDPR requirements that will become mandatory under specific conditions. Using *PolicyChecker*, we conduct the first large-scale GDPR-completeness violation study on 205,973 privacy policies of Android apps in the UK Google Play store. PolicyChecker identified 163,068 (79.2%) privacy policies containing data collection statements; therefore, such policies are regulated by GDPR requirements. However, the majority (99.3%) of them failed to achieve the GDPR-completeness with at least one unsatisfied requirement; 98.1% of them had at least one unsatisfied mandatory requirement, while 73.0% of them had at least one unsatisfied if-applicable requirement logic chain. We conjecture that controllers' lack of understanding of some GDPR requirements and their poor practices in composing a privacy policy can be the potential major causes behind the GDPR-completeness violations. We further discuss recommendations for app developers to improve the completeness of their apps' privacy policies to provide a more transparent personal data processing environment to users.

CCS CONCEPTS

• **General and reference** → **Measurement**; • **Security and privacy** → **Privacy protections**.

KEYWORDS

Mobile App, Privacy Policy, GDPR, Completeness

CCS '23, November 26–30, 2023, Copenhagen, Denmark

ACM ISBN 979-8-4007-0050-7/23/11.
https://doi.org/10.1145/3576915.3623067

ACM Reference Format:
Anhao Xiang, Weiping Pei, and Chuan Yue. 2023. PolicyChecker: Analyzing the GDPR Completeness of Mobile Apps' Privacy Policies. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23), November 26–30, 2023, Copenhagen, Denmark.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3576915.3623067

1 INTRODUCTION

The ability of mobile apps to collect sensitive personal data has raised concerns about the privacy of app users. Studies have revealed that apps can collect excessive personal data [48] or gain access to protected data without user consent [41]. Such privacy threats imposed by mobile apps have stimulated advancements not only in technical solutions to defend app users from privacy invasion [8, 11], but also in understanding how users perceive privacy risk in using an app [12, 13]. However, in recent years, the most significant advancement in protecting the privacy of app users has come from the privacy law legislations such as the California Consumer Privacy Act (CCPA) and GDPR.

Since its adoption in 2018, GDPR has promoted a reformation of app user privacy protection and data processing transparency. GDPR not only mandates developers to process personal data lawfully (Art. 6) but also imposes transparency obligations on developers (Art. 12). As a result, developers add user consent prompts that adhere to the GDPR requirements on data processing lawfulness and provide online privacy notices and policies that inform users about how their data are being processed and what are their rights. Although many mobile apps already had a privacy policy before GDPR, GDPR has stimulated the most widespread privacy policy content updates [3, 10]. Most importantly, a privacy policy is no longer a "nice-to-have" document provided by the app developers to demonstrate a good gesture in the pre-GDPR era. Instead, it is now a legally binding document between a developer and a user [15], and providing an incomplete privacy policy to users will be considered a violation of GDPR which may result in large fines. For example, in July 2021, an unprecedented fine of $247 million was imposed on WhatsApp by the Irish Data Protection Commission [9] due to WhatsApp's breaching of transparency obligations under Arts. 12, 13, and 14 [15–17] of GDPR.

As mandated by GDPR Art. 12, a developer must *"provide the information referred to in Arts. 13 and 14 to a user in a concise, transparent, intelligible, and easily accessible form, using clear and plain language"* [15]. According to the UK Information Commission Office's (ICO) guideline [29], requirements in Arts. 13 and 14 are categorized as "always required" and "required if applicable", referred to as the "mandatory" and "if-applicable" requirements in

this paper. In particular, mandatory requirements must be fulfilled without exception, for example, developers must provide the user rights statements such as *right to access* in privacy policies regardless of how they process data. However, if-applicable requirements are more flexible and depend on whether a certain data practice is performed by a developer, for example, data transfer intention should be disclosed if there is an international data transfer practice.

The fulfillment of GDPR transparency obligations requires organizations to assess the ***completeness*** of privacy policies in terms of not only all mandatory requirements but also all if-applicable requirements that will become mandatory under the conditions specified in Arts. 13 and 14. Completeness is one of the six principles that contribute to a GDPR-compliant privacy policy (Section 3). A GDPR-compliant privacy policy should be ***GDPR-complete***, i.e., all information pertinent to the mandatory and if-applicable requirements specified in Arts. 13 and 14 must be presented in a privacy policy. Some studies have started to detect the information completeness of mobile apps' privacy policies in recent years [26, 34]. However, they mainly focused on analyzing a subset of the mandatory requirements; meanwhile, they neglected the implications of if-applicable requirements. Besides, most prior studies were on small-scale datasets which may not comprehensively reveal the state of GDPR-completeness violations in the wild.

In this work, ***our goal*** is to explore the state of GDPR-completeness violations in mobile apps' privacy policies. We achieve our goal by answering the following two major research questions:

RQ1*: How complete are mobile apps' privacy policies in terms of providing all the information to fulfill both mandatory and if-applicable requirements specified in GDPR Arts. 13 and 14?* To answer RQ1, we design the ***PolicyChecker*** framework and conduct the first large-scale GDPR-completeness study on 205,973 privacy policies of Android apps in the UK Google Play store. PolicyChecker checks all mandatory requirements in Arts. 13 and 14, and further utilizes *requirement logic chains* to model the conditions when each of the if-applicable requirements will become mandatory.

Among 163,068 analyzable privacy policies that declared data collection practices, *PolicyChecker* detected that 161,952 (99.3%) of them had at least one unsatisfied requirement. The most common completeness violations pertain to the lack of information disclosure regarding users' rights related to data processing. For example, the *right to restrict processing* (Arts. 13.2.(b) and 14.2.(c)) and *right to lodge a complaint* (Arts. 13.2.(d) and 14.2.(e)) statements are missing in 84.5% and 81.0% of privacy policies, respectively. Meanwhile, by applying requirement logic chains, *PolicyChecker* detected that among 77,522 privacy policies that indicated consent-based data processing, 66.1% of them failed to further indicate whether users have the right to withdraw their consents (Arts. 13.2.(c) and 14.2.(d)).

RQ2*: What are the potential causes for mobile apps' privacy policies to fail to provide all information required by GDPR Arts. 13 and 14, and correspondingly what could be done by app developers to improve the completeness of their apps' privacy policies?* We answer RQ2 by both automatically and manually analyzing the results of our large-scale study. We conjecture that developers likely lack a good understanding of some GDPR requirements and the data collection practices of their own apps, reflected in the poor transparency on disclosing indirect data collection in their privacy policies. Our findings and tentative interpretations are consistent with what researchers (e.g., Alomar et al. [2]) in the Human-Computer Interaction (HCI) domain found about the organizations' lack of a good understanding of their obligations under GDPR requirements. Such a lack of understanding can be a potential major cause behind the high number of unsatisfied requirements; meanwhile, developers' misuse of policy generators and their prominent copy-and-paste practices led to the high-level content similarity and common unsatisfied requirements among a large portion (33.1%) of privacy policies.

We further discuss recommendations for app developers to (1) review their internal compliance processes and establish appropriate channels for users to exercise rights, (2) exercise caution when generating privacy policies using tools, and (3) perform due diligence to make app-specific modifications to ensure that privacy policies accurately reflect apps' actual data practices and legal stances. Overall, we make three major contributions in this paper:

- We review GDPR articles on information transparency and extract 26 requirements on what need to be included in a privacy policy. Meanwhile, we derive six requirement logic chains to model the conditions under which if-applicable requirements will become mandatory. These results provide a foundation for this study, and could also help app developers establish a better understanding of their obligations and support future privacy policy compliance analysis research.
- We design the *PolicyChecker* framework to automatically detect GDPR-completeness violations in mobile apps' privacy policies. While existing studies detect violations based on a subset (10 out of 26) of GDPR requirements, *PolicyChecker* operates on a set of 20 requirements (three are excluded in our current implementation (Section 3.1) and three are excluded in our analysis due to the lack of data samples (Appendix B.2.1 in [39]) with five requirement logic chains to comprehensively detect violations.
- By using *PolicyChecker*, we conduct the first large-scale study on 205,973 mobile apps' privacy policies collected from the UK Google Play store to analyze the state of their completeness against the GDPR requirements (note that the UK GDPR is currently identical to the EU GDPR but could be subject to changes in the future). We provide corresponding recommendations to app developers.

The rest of this paper is organized as follows. Section 2 reviews related studies. Section 3 presents our analysis of GDPR requirements on privacy policies. Section 4 describes the design of *PolicyChecker*. Section 5 analyzes the GDPR-completeness of our collected privacy policies. Section 6 discusses the implications of this study, the recommendations to app developers, the limitations of this study, and the potential future work. Section 7 concludes the paper. Due to the page limitation, we provide some details in the long version of this paper [39].

2 RELATED WORK

Privacy policy completeness analysis. Prior studies [26, 34] have started to detect the information completeness in privacy policies. Both studies trained a machine learning model on a human-annotated privacy policy dataset to classify policy sentences into

predefined categories. In particular, Liu et al. [34] constructed a dataset with 36,610 sentences (from 304 privacy policies) annotated based on ten categories of the mandatory information required by GDPR Art. 13. The sentence classifiers in [26] were trained based on the OPP-115 dataset [46] with mandatory requirements from GDPR Arts. 13 and 14 encoded using the OPP-115 taxonomy.

Liu et al. [34] found 1,180 completeness violations in a dataset of 304 privacy policies. The most common violations are related to the unfulfilled requirements *right to access*, and the *right to restrict processing*. However, the authors did not report results on other mandatory requirements. In addition, the analysis results in [34] cannot represent the actual state of GDPR-completeness violations since the analysis was performed on a dataset constructed with intentionally selected high-quality policies. Hamdani et al. [26] did not include a discussion on the completeness violations in their privacy policy dataset. Their dataset of 30 annotated privacy policies solely served the purpose of evaluating their sentence classifiers.

In this work, we propose the *PolicyChecker* framework and conduct the first large-scale GDPR-completeness study on 205,973 privacy policies in the wild. Our policy completeness analysis covers the most comprehensive set of requirements, including both the mandatory and if-applicable GDPR requirements.

Privacy policy data processing and purpose statement analysis. Another strand of research focuses on analyzing the data processing and processing purpose statements in privacy policies [4–7, 44, 48, 49]. PurPliance [7] analyzes the predicate-argument structure of privacy policy sentences to identify processing purpose clauses. The extracted purposes were compared with the actual purposes derived from the apps' network traffic, and the result indicates that 70% of apps (n=23.1k) have inconsistencies. MAPS [49] combines the policy analysis with code analysis to show that privacy policies often under-report the practices performed by apps. PolicyLint [4] studies the contradictions in data processing statements and reports 14% (n=11.4k) of app policies with internal contradictions that may be indicative of misleading statements. Most recently, POLICYCOMP [48] analyzes the data collection statements in an app's privacy policy using semantic roles and compares them with the app's counterparts to detect excessive data processing practices. Their result showed that the data collection statements in 48.3% of apps' privacy policies (n=10k) are overbroad. Bhatia et al. [5] created a taxonomy of processing purpose statements from five shopping sites' privacy policies. Bhatia et al. [6] further investigated the lack of processing purpose in data processing statements which are represented as semantic frames. Shvartzshnaider et al. [44] used the framework of contextual integrity to detect vague language and missing contextual details in data processing statements.

Development of privacy policies. The development of privacy policies has been studied intensively in the literature [3, 10, 33]. Martin et al. [10] investigated the impact of GDPR on the privacy policies of popular websites in European Union in 2018 at the time when GDPR was enacted. Their result indicated that 72.6% of websites with existing privacy policies updated their contents, and 15.7% of websites adopted a privacy policy for the first time. Similarly, Linden et al. [33] studied the privacy policy landscape after the GDPR enactment and found that privacy policies in the post-GDPR era covered more data practice disclosures. Amos et al. [3] conducted a longitudinal analysis on a dataset of one million websites' privacy policies spanning over two decades. Their result suggested that privacy policies have become even more difficult to read, with the concerning lack of transparency in tracking and cookie information disclosure. They also pointed out that GDPR has stimulated the largest privacy policy updates in the decade.

3 PRIVACY POLICY UNDER GDPR

As the pioneer in data protection and privacy regulation, GDPR [19] regulates the processing of personal data by a company or an organization known as a *data controller* from EU individuals who are referred to as *data subjects*. According to Art. 4, *data processing* refers to any operation performed on personal data, such as collection, recording, organization, structuring, and storage, etc. GDPR consists of 99 articles organized into 11 chapters. A controller shall be responsible for and be able to demonstrate compliance with the general principles of data processing. We summarize the following six principles that contribute to a GDPR-compliant privacy policy based on the definition of the *principle of processing* laid out in GDPR Art. 5 [18] and GDPR legal analysis from existing literature: (1) **completeness** [18, 26, 34]; (2) lawfulness [18]; (3) fairness [18]; (4) accessibility & readability [18, 26]; (5) purpose limitation & data minimization [18, 48]; (6) accuracy [4, 7, 18]. A detailed discussion of these six principles is provided in Appendix A.1 in [39].

In this work, we focus on checking the completeness of a privacy policy according to Arts. 13 and 14. These two articles describe regulations on what and how information related to data processing must be provided to data subjects, and those are the most important regulations to be reflected in privacy policies [26]. In the following subsections, we first introduce mandatory and if-applicable information provision requirements specified in Arts. 13 and 14. We then describe logic chains to be used to deduce the conditions under which if-applicable requirements would become mandatory.

3.1 Requirements in GDPR Arts. 13 and 14

To obtain an accurate and comprehensive understanding of the specified requirements, we carefully reviewed Arts. 13 and 14 as well as other relevant articles (e.g., Arts. 6, 9, 22, 46, 47, 49, 89) and referred to recitals and commentaries made by the UK Information Commissioner's Office (ICO) [27]. Finally, we constructed a list of mandatory and if-applicable requirements as shown in Table 1 for us to use in this study. Although Arts. 13 and 14 both specify what information must be provided to data subjects by data controllers, they are used in different scenarios. Specifically, Art. 13 is required when personal data are directly obtained from a data subject (direct collection), while Art. 14 is used in situations where personal data are obtained from other sources than a data subject (indirect collection).

Prior studies only focused on a subset of mandatory requirements, making their analyses incomplete. Authors in [34] only examined requirements R2, R3, R5, and R16 to R22. Authors in [26] focused on requirements R1 to R5, R9, R16, R17, R18, R21, and R25; they also miss-categorized *recipient of personal data* (R9) as a mandatory requirement. In addition, prior studies treated requirements *right to portability (R21)*, *right to erasure (R17)*, and *right to object (R20)* as mandatory without considering that a data controller might be exempted from providing such rights based on the complicated

Table 1: GDPR Arts. 13 and 14 requirements (if-applicable requirements are denoted as $R\#^*$, excluded requirements are denoted as $R\#^-$, and the rest are mandatory requirements)

ID	Article Reference	Required Information
$R1$	13.1.(a) & 14.1.(a)	Controller identity
$R2$	13.1.(a) & 14.1.(a)	Contact Information
$R3$	13.2.(a) & 14.2.(a)	Data retention time limit
$R4$	13.2.(a) & 14.2.(a)	Data retention criteria
$R5$	13.1.(c) & 14.1.(c)	Data processing purpose
$R6$	13.1.(c) & 14.1.(c)	Legal basis
$R7^*$	13.1.(d) & 14.2.(b)	Interest pursued
$R8^*$	13.2.(c) & 14.2.(d)	Right to withdraw consent
$R9^*$	13.1.(e) & 14.1.(e)	Recipients of the personal data
$R10^*$	13.1.(f) & 14.1.(f)	International data transfer intention
$R11^*$	13.1.(f) & 14.1.(f)	Adequacy decision
$R12^*$	13.1.(f) & 14.1.(f)	Transfer safeguards
$R13^*$	13.2.(f) & 14.2.(g)	Automated decision system in use
$R14^*$	13.2.(f) & 14.2.(g)	Decision system logic
$R15^*$	13.2.(f) & 14.2.(g)	System significance and impact
$R16$	13.2.(b) & 14.2.(c)	Right to data access
$R17^-$	13.2.(b) & 14.2.(c)	Right to data erasure
$R18$	13.2.(b) & 14.2.(c)	Right to data rectification
$R19$	13.2.(b) & 14.2.(c)	Right to restrict processing
$R20^-$	13.2.(b) & 14.2.(c)	Right to object processing
$R21^-$	13.2.(b) & 14.2.(c)	Right to data portability
$R22$	13.2.(d) & 14.2.(e)	Right to lodge complaints
$R23^*$	13.2.(e)	Data collection necessity
$R24^*$	13.2.(e)	User obligation and consequences
$R25$	14.1.(d)	Categories of personal data
$R26$	14.2.(f)	Source of the personal data

conditions defined in Arts 17, 20, and 21. In this work, we exclude the requirements R17, R20, and R21 from the implementation of our GDPR-completeness analysis. We provide a detailed analysis of Arts. 13 and 14 requirements in Appendix A.2 in [39].

3.2 Logic Chains in If-Applicable Requirements

Arts. 13 and 14 contain if-applicable requirements which could become mandatory under specific conditions. However, all prior studies (e.g., [26, 34]) neglected if-applicable requirements, making their analyses incomplete. The major challenge to taking into account if-applicable requirements in GDPR-completeness analysis is to determine the conditions that make if-applicable requirements mandatory. To address this challenge, we constructed the following set L of six logic chains (from L_1 to L_6) that model how a specific condition turns an if-applicable requirement into a "mandatory" requirement. We use "*Legal_basis*" to refer to the set of six legal bases {*user_consent*, *legal_obligation*, *contract_performance*, *vital_interest*, *public_interest*, *legitimate_interest*}; more details about the six legal bases are discussed in Appendix A.2 in [39]. We use "$\rightarrow$" to denote the logic implication relation in a chain.

- L_1 = *user_consent* $\in$ *Legal_basis* $\rightarrow$ $R8$: *"**Where** the processing is based on point (a) of Article 6(1) or point (a) of Article 9(2), the existence of the right to withdraw consent at any time ..." (Arts. 13.2.(c) and 14.2.(d))*. The existence of information on using user consent as the legal basis turns the if-applicable requirement *right to withdraw consent* (R8) into a mandatory requirement.
- L_2 = *legitimate_interest* $\in$ *Legal_basis* $\rightarrow$ $R7$: *"**Where** the processing is based on point (f) of Article 6.(1), the legitimate interests pursued by the controller or by a third party ..." (Arts. 13.1.(d) and 14.2.(b))*. If a controller declares its legitimate interests as the data processing legal basis, the controller is obliged to provide the details of what interests are pursued (R7). For example, the statement *"we have a legitimate business interest to process your data"* indicates the use of legitimate interest as the legal basis and thus turns the if-applicable requirement R7 into a mandatory requirement. The existence of another statement, e.g., *"our legitimate business interest in protecting app security"*, can fulfill R7.
- L_3 = *data_sharing_practice declared* $\rightarrow$ $R9$: *"The recipients or categories of recipients of the personal data, if any ..." (Arts. 13.1.(e) and 14.1.(e))*. When a controller declares a data-sharing practice, GDPR requires the controller to provide the data receiver's identity and thus turns the if-applicable requirement R9 into a mandatory requirement.
- L_4 = $R10 \rightarrow [R11 \vee R12]$: *"Where applicable, the controller intends to transfer personal data to a third country ... **and** the existence or absence of an adequacy decision by the Commission, **or** ... reference to the appropriate or suitable safeguards, **and** the means to obtain a copy of them ..." (Arts. 13.1.(f) and 14.1.(f))*. When a controller discloses an intention to transfer personal data to an international location, the controller must also disclose the existence or absence of an adequacy decision (R11). Moreover, the absence of an adequacy decision will make R12 mandatory, i.e., the controller must provide information about transfer safeguards and inform users about how to obtain a copy of the transfer safeguards.
- L_5 = $R13 \rightarrow [R14 \wedge R15]$: *"The existence of automated decision-making, including profiling, referred to in Article 22(1) and (4) **and**, at least in those cases, meaningful information about the logic involved, **as well as** the significance **and** the envisaged consequences of such processing for the data subject ..." (Art. 13.2.(f) and 14.2.(g))*. When a controller specifies the use of an automated decision system (R13), the controller must elaborate on the decision logic (R14) and any possible impact on data subjects (R15).
- L_6 = [*legal_obligation* $\in$ *Legal_basis* $\vee$ *contract_performance* $\in$ *Legal_basis*] $\rightarrow$ $R24$: *"...whether the data subject is obliged to provide the personal data **and** of the possible consequences of failure to provide such data ..." (Art. 13.2.(e))*. When a controller specifies that users have an obligation to provide personal data under the law or contract, the controller needs to inform users about the possible consequences when refusing to provide the data.

Our *PolicyChecker* framework applies these logic chains to check the completeness of privacy policies with respect to if-applicable requirements. For example, the privacy policy of a popular mobile game app contains the sentence *"we process the following personal data when you **consent**"* which indicates the use of consent as a legal basis. PolicyChecker will first identify this sentence as the fulfillment of the mandatory requirement R6. Then the logic chain L_1 is triggered to turn the if-applicable requirement R8 into a mandatory requirement that must be considered in the analysis. Finally,

since no statement about the right to withdraw consent exists in the privacy policy, *PolicyChecker* will report the logic chain L_1 violation and GDPR-incompleteness for this app's privacy policy.

4 POLICYCHECKER

To achieve our goal in this work, we design the *PolicyChecker* framework which automatically checks the ***completeness*** of a privacy policy according to GDPR requirements. The completeness principle (Section 3) serves as the fundamental compliance guideline for us to answer the core question: Does a given privacy policy provide all the information to fulfill both mandatory and if-applicable requirements defined in GDPR Arts. 13 and 14?

4.1 Design Overview

Design approach. To design *PolicyChecker*, we investigate a *rule and semantic role based* approach that automatically detects the completeness of a privacy policy against GDPR requirements. Privacy policies are legal documents describing a set of predefined topics with limited variations in word choices and styles of expressions, which lead to traceable commonalities (e.g., verb choices, sentence structures) under each topic. This observation motivated us to explore a "rule-based" design in detecting topics in privacy policies, as one could derive a set of traits for each topic and apply rule-based filtering to identify potential sentences belonging to it.

In more details, our *rule and semantic role based* approach prefilters potential topic-related sentences based on a verb-topic mapping, and models the predicate-argument structure of a target sentence. The sentence's predicate-argument structure is then analyzed based on the rules predefined for each topic. Eventually, this approach outputs the topic(s) that a sentence belongs to. After identifying a set of topics from a privacy policy, this approach again adopts the rule-based design to perform the completeness analysis.

High-level architecture of *PolicyChecker*. Figure 1 depicts the high-level architecture of the *PolicyChecker* framework, which processes each sentence from a privacy policy using an NLP pipeline comprising five main steps.

In Step (1), *PolicyChecker* determines the practice described in a sentence by matching the verb extracted from the sentence against the predefined practice-to-verbs mapping (detailed in Section 4.2). In Step (2), if a practice is identified in a sentence, *PolicyChecker* constructs the predicate-argument structure of the sentence, including semantic role labels and corresponding semantic arguments. In Step (3), *PolicyChecker* matches the semantic roles against predefined roles for the practice. Each practice must have two basic roles: the actor responsible for performing the action and the object related to the action, represented by ***Arg0*** and ***Arg1*** labels, respectively; this constraint is relaxed under special cases, such as when a sentence is in the passive tense (details are provided in Appendix B.3 in [39]). In addition to the two basic roles, *PolicyChecker* uses predefined auxiliary semantic roles for certain practices to narrow down the arguments in the predicate that can potentially satisfy GDPR requirements. In Step (4), *PolicyChecker* validates the meaning of each semantic argument by identifying the entity in the sentence using a Named Entity Recognition (NER) model or performing n-gram matching and textual similarity comparison against a predefined template. If each semantic argument has the correct meaning, *PolicyChecker* adds the ID of the requirement that the sentence fulfilled to a set. When one of the six legal bases (details of legal bases are provided in Appendix A.2.1 in [39]) is detected in the sentence, *PolicyChecker* adds the name of the legal basis to a set.

Whenever Step (1) identifies no practice in a sentence or the output generated in Step (3) or Step (4) is false, *PolicyChecker* halts the analysis of the current predicate or sentence and moves to analyze the next predicate or sentence. Otherwise, after completing Steps (1) to (4), *PolicyChecker* merges the fulfilled-requirement set and the legal-basis set with the corresponding policy-level sets. In Step (5), *PolicyChecker* generates the analysis report for each privacy policy based on the policy-level fulfilled-requirement set, the legal basis set, and three artifacts: a set of mandatory requirements, a set of if-applicable requirements, and the requirement logic chains.

We provide an example to illustrate some steps using the sentence *"we store your geolocation data for 6 months"*. Based on the predefined verb-to-topic mapping, *PolicyChecker* identifies this sentence with the verb "store" to be a potential sentence that describes the data retention policy. Then, the semantic role labeling result indicates that in this sentence the subject is *"we"* and the predicate is *"store your geolocation data for 6 months"*. Based on a set of predefined rules that specify the must-have components in a sentence in order for it to be considered as a data retention policy statement, *PolicyChecker* further verifies if the sentence has the correct subject ("we" represents the data controller), if the predicate conveys the correct action ("store" means retaining data), if the action is performed on an intended object entity ("geolocation data" is a *DATA* entity recognized by the NER model), and if the action is modified by a time constraint argument ("6 months"). When all requirements are satisfied, *PolicyChecker* adds *Data_retention_time_limit* to a set of fulfilled requirements.

NLP models in use. To analyze a sentence's semantic structure, *PolicyChecker* utilizes the Semantic Role Labeling (SRL) model. SRL constructs the predicate-argument structure of a sentence and answers the questions of "*who* did *what* to *whom*". SRL assigns semantic role labels to each argument phrase, indicating its relationship with the predicate. In general, semantic roles ***Arg0*** (Argument0), ***Arg1*** (Argument1), and ***Arg2*** (Argument2) identify the agent (the entity responsible for performing the action), object, and instrument (indirect object), respectively. ***ArgM-tmp*** (Argument Modifier-temporality), ***ArgM-loc*** (Argument Modifier-location), and ***ArgM-mnr*** (Argument Modifier-manner) indicate when, where, how the action took place. ***ArgM-prp*** (Argument Modifier-purpose), ***ArgM-pnc*** (Argument Modifier purpose-not-cause), and ***ArgM-cau*** (Argument Modifier-cause) indicate the purpose and cause of the action. PolicyChecker uses the *AllenNLP* [14] implementation of the SRL model which is for the English PropBank SRL task and is trained on the Ontonotes 5.0 dataset [45]. In recent years, there is an increasing adoption of the SRL techniques in analyzing privacy policies [7, 48], from which SRL has demonstrated the effectiveness of extracting the data collection flow and the collection purposes (e.g., who collected what data from whom for what purposes).

Additionally, *PolicyChecker* utilizes the NER model to verify that the intended entity is indeed carrying out the action and that the object is a valid item. An NER model classifies named entities (e.g., people, organizations, locations, etc.) from a text into predefined

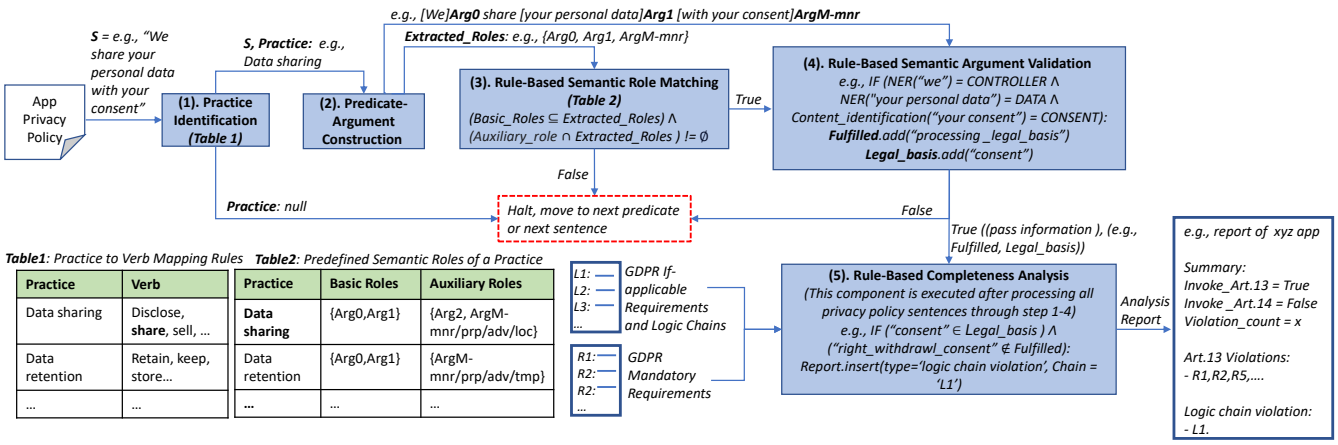

Figure 1: The High-level Architecture of the *PolicyChecker* Framework

categories. PolicyChecker uses the domain-adapted NER model published by PurPliance [7] to identify personal data (e.g., IP address, email, phone number, etc.) as the *DATA* entity. Moreover, *PolicyChecker* labels pronouns "we," "I", and "me" as the *CONTROLLER* entity, and "you", "user", and "data subject" as the *USER* entity.

The NER model can only verify the DATA entity object. Whereas other entity objects (e.g., consent entity) are validated using the textual semantic similarity comparison against a predefined template. Semantic similarity between two texts measures the closeness of their meanings instead of their syntactic structures. In this work, we estimate the closeness of text meanings by calculating the cosine angle between the two texts' vector representations generated by the sentence-BERT language model (SBERT all-MiniLM-L6-v2) [42].

Design choice justification. There are at least two alternative approaches for checking the completeness of privacy policies. First, one could train a supervised sentence-level topic classification model. Using the topic labels predicted for each sentence and a list of rules derived from the GDPR requirements, a conclusion could be made about whether a given privacy policy contains all required topics. Although this is a major approach adopted by prior studies [26, 34, 36], there is no existing dataset with a complete set of annotations covering all topics or requirements listed in Table 1. Poplavska et al. analyzed the connections and gaps between the OPP-115 annotation categories and the GDPR principles [40]; they considered that NLP researchers can continue to use the OPP-115 taxonomy for privacy policy analyses. However, neither the OPP-115 [46] nor the OPP-350 [49] privacy policy dataset can be adapted for us to achieve our goal because their annotations only focused on data collection and data usage related topics (e.g., an app's access of GPS location information by a first or third party) and missed annotations for most of the topics listed in Table 1. Findings in [26] showed that OPP-115 could be utilized to perform compliance checking against 10 GDPR requirements, but errors often occur on detecting certain types of violations due to the difficulty of aligning OPP-115 vs. GDPR concepts (given that OPP-115 was created pre-GDPR). All these reasons also prevented us from experimentally comparing our approach with the prior studies (e.g., [26]) that built classifiers from OPP-115. Similarly, the lack of a comprehensive and large human-annotated privacy policy Question Answering (QA) dataset stalls the adoption of modern QA models in achieving our goal.

Second, one can take a topic modeling approach based on text clustering [47], although this approach has not been adopted to perform privacy policy analysis in the literature. Intuitively, sentences describing the same topic share similar semantics. Therefore, each cluster should represent a unique topic. Topic words can then be derived using techniques such as Term Frequency–Inverse Document Frequency (TF-IDF) from each cluster. This approach has shortcomings in its incapability to differentiate topics at a fine-grained level; therefore, sentences that should belong to different clusters can end up in the same cluster, while sentences that should belong to the same cluster can end up in different clusters.

Our *rule and semantic role based* approach is different from and more appropriate than these two alternative approaches from three aspects: (1) Compared with the first approach, our approach does not rely on a human-annotated dataset to train a supervised model for topic detection and completeness analysis; As we noted, we could not find such annotated datasets in our analysis context; (2) Compared with the clustering-based topic modeling approach, our approach has the ability to differentiate topics at a fine-grained level, which is the key to accurately detecting completeness violations in a policy; (3) In the case when regulations are updated to include new requirements, our approach can be quickly adapted to perform new analyses by adding new rules and leveraging SRL.

4.2 Practice Identification

PolicyChecker identifies sentences that contain GDPR requirement-related information by applying verb filtering to locate predicates that are likely relevant. For example, a predicate with the verb "retain" for expressing the data retention practice has a high chance of containing information regarding the data retention period in its modifier arguments.

PolicyChecker focuses on identifying the common verbs used by the following types of practices in privacy policies: (1) data

Table 2: List of verbs used by *PolicyChecker*

Practice	Verb
Data collection	collect, gather, obtain, receive, record, store, solicit
Data sharing	disclose, distribute, exchange, give, lease, provide, rent, release, report, sell, send, share, trade, transfer, transmit
Data using/handling	access, analyze, check, combine, connect, know, process, use, utilize, deploy
Data retention	retain, hold, keep, possess, save, withhold, store
Data transfer	transfer, move, relocate, transmit
Transfer safeguarding	rely on, base on, count on, depend on
Consent giving	give, grant, consent, provide, accord
Consent solicit	ask, seek, require, request, demand, obtain, gain, acquire, get, receive, retain
Consent withdrawal	withdraw, revoke, retract
Rights entitle	have, entitle, designate
Rights request	request, ask, demand

collection, sharing, handling, using, retention, and transfer practices; (2) consent giving, soliciting, and withdrawal practices; (3) data transfer safeguarding practices; (4) rights entitle and request practices. Table 2 lists the verbs used by *PolicyChecker* for each practice. The detailed process of deriving the verb lists is provided in Appendix B.1 in [39]. For a sentence in a privacy policy, *PolicyChecker* extracts all verbs from the sentence's Part of Speech (POS) tags to match the predefined verb list. The extracted verbs are converted to their lemma form (e.g., *"keeping"* is converted to *"keep"*) before the matching. Although identifying practices based on the list of predefined verbs is a rule-based approach, it is powerful and commonly used in analyzing privacy policies [4, 7]. The list of verbs constructed in our work is more extensive than those in existing studies as it includes all practices to perform completeness checking against requirements in GDPR Arts. 13 and 14.

4.3 Predicate Argument Construction and Semantic Role Matching

When *PolicyChecker* finds from the sentence a verb associated with a predefined practice, it constructs the predicate-argument structure surrounding the predicate that the verb belongs to. The predicate-argument structure involves semantic arguments and their role labels. For example, given the predicate-argument structure *"[we]Arg0 share [your personal data]Arg1"*, the phrases *"we"* and *"your personal data"* are called semantic arguments. And a label (e.g., *Arg0, Arg1*) is assigned to each semantic argument, indicating its relationship (e.g., the object of the predicate, the modifier of the predicate, etc.) to the predicate.

Given the predicate-argument structure of the sentence, *PolicyChecker* first checks the existence of two fundamental semantic roles: the subject and the object, represented by *Arg0* and *Arg1* labels, respectively. Sentences that lack the subject and object roles are considered grammatically incorrect. Note that although the lack of an object is acceptable in some instances (e.g., with intransitive verbs), the verbs considered in this paper are all transitive verbs that require an object. A sentence with grammar errors cannot be further analyzed because the SRL prediction becomes unreliable.

For sentences describing data collection, sharing, use, handling, retention, and transfer practices with the presence of the two fundamental roles, *PolicyChecker* further seeks to find GDPR requirement related information from their auxiliary roles (e.g., modifiers, instruments, etc.). For example, given a sentence describing data collection practice: $[We]_{Arg0}$ *collect* $[your\ personal\ data]_{Arg1}$ $[for\ marketing\ purpose]_{ArgM\text{-}prp}$, the sentence has the two fundamental roles *Arg0* ("we") and *Arg1* ("your personal data") and a modifier role *ArgM-prp* that describes the purpose of the action. Therefore the sentence can potentially satisfy the requirement R5 (data processing purpose). We provide the list of auxiliary roles for each of the aforementioned practices in Table 6 of Appendix B.2 in [39].

For sentences describing consent solicitation, giving, withdrawal, and transfer safeguarding practice, *PolicyChecker* finds GDPR requirement related information directly from the two fundamental roles (e.g., $[we]_{Arg0}$ *ask* $[your\ consent]_{Arg1}$). Therefore no auxiliary roles are considered in such a case.

By applying SRL [14] to model the predicate-argument structure of a sentence and then analyzing semantic roles for each data practice, *PolicyChecker* intensively identifies the sentences that contain GDPR-related information. Meanwhile, in our work, the set of semantic role matching rules is comprehensive as it covers all data processing practices as described above.

4.4 Semantic Argument Validation

PolicyChecker further verifies the meanings of the identified semantic arguments. For example, given the following sentence with its predicate-argument: $[we]_{Arg0}$ *ask* $[your\ consent]_{Arg1}$, while *PolicyChecker* can infer from the label *Arg0* that the entity referred to as "we" is responsible for the action "ask" and the phrase "your consent" is the object, it is important to verify that the intended entity is indeed carrying out the action and the object is a valid consent-related item. Only after a successful verification, *PolicyChecker* will conclude that the sentence satisfies the requirement R6 (legal basis). To do so, *PolicyChecker* uses NER entity enforcement and n-gram matching techniques. Whenever the n-gram matching fails to identify the argument, *PolicyChecker* invokes the semantic similarity comparison. The templates for n-gram matching and semantic similarity comparison are provided in Tables 4 and 5 of Appendix B.2 in [39]. PolicyChecker's argument validation design not only utilizes the NER model [7] to recognize data entities in a sentence but also incorporates the n-gram matching as well as textual similarity comparison. These design considerations allow *PolicyChecker* to perform a wide range of argument content validations.

In the rest of this subsection, we discuss the validation procedures for data collection and retention practices as well as consent soliciting, giving, and withdrawal practices. The validation procedures for other types of practices are discussed in Appendix B.2 in [39]. For the sake of presentation simplicity, we refer to the n-gram matching and semantic similarity comparison process as *content identification*.

4.4.1 Data collection practice. For a sentence describing the data collection practice, *PolicyChecker* has the following validations: (1) the *Arg0* argument must have a *CONTROLLER* entity (i.e., the

controller must be the receiver of the personal data); (2) the *Arg1* argument must be a *DATA* entity. Additionally, *PolicyChecker* seeks to find evidence of direct or indirect data collection by searching for the auxiliary role *Arg2* in the collection practice. If evidence exists, the *Arg2* argument with an *organization/company (ORG/COM)* entity indicates that the personal data originated from a third-party source (fulfillment of the requirement R26: source of personal data). Therefore, the controller is conducting indirect data collection (Art. 14). Similarly, the *Arg2* argument with a *USER* entity indicates that the personal data originated directly from the user and should be identified as direct data collection (Art. 13).

When indirect data collection practice is identified from the sentence, *PolicyChecker* re-examines the content in *Arg1* argument to determine if the sentence satisfies the requirement R25 (categories of personal data received from other sources). To satisfy the requirement, the *Arg1* argument cannot be a general or vague description of data (e.g., *"we collect personal data"*).

A sentence describing data collection practice could also include the purpose of collection and the use of user consent. PolicyChecker identifies the purpose of collection from arguments with the auxiliary label *ArgM-cau*, *ArgM-prp*, or *ArgM-pnc*, which are typical semantic roles indicating the purpose of the action. The extracted argument describes a purpose if it begins with a common prefix ("to", "for", "in order to", "so as to", or "so that"). In addition, as pointed out in PurPliance [7], the purposes extracted from policy statements can be categorized as production, marketing, and legality purposes. In many cases, the processing purpose constitutes the legal basis of the processing. We discuss the legal basis and legitimate interest identification from purpose clauses in Appendix B.3.1 in [39].

The use of user consent is identified from arguments with an auxiliary label *ArgM-mnr* (e.g., *"with your consent, ..."*), *ArgM-tmp* (e.g., *"when your consent, ..."*), or *ArgM-adv* (e.g., *"if your consent, ..."*). PolicyChecker validates that the argument is consent related by applying the *content identification* process. In general, a sentence describing data collection practices may include information that meets the requirements of R5, R6, R25, and R26.

4.4.2 Data retention practice. For a sentence describing data retention practice, the*Arg0* argument must contain the *CONTROLLER* entity. This means that the data controller is the entity responsible for retaining personal data. The *Arg1* argument must contain the *DATA* entity. PolicyChecker further looks for the existence of argument with the *ArgM-tmp* label (time constraint modifier). The *Argm-tmp* argument fulfills the requirement R3 if it contains a *TIME* entity (e.g., month, year) indicating the specific data retention time. Otherwise, the argument is a general description of the retention time (e.g., *"for as long as we are under contract"*) that fulfills the requirement R4. PolicyChecker uses a similar procedure in data collection practice to identify any data retention purpose and use of user consent. A sentence describing data retention practices may include information that meets requirements R3, R4, R5, and R6.

4.4.3 Consent giving, soliciting, and withdrawal practice. For a sentence expressing consent-giving and consent-withdrawal practices, the *Arg0* argument must contain a *USER* entity, and the *Arg1* argument is validated to be a consent-related item using the content identification process. The only difference with consent-soliciting practice is that the *Arg0* argument must contain a *CONTROLLER* entity instead of a *USER* entity.

Besides these validation procedures for general practices, some cases need to be additionally considered to further improve the effectiveness of *PolicyChecker*. We discuss the design for sentences with negative and passive sentiments in Appendices B.3.2 and B.3.3 in [39].

4.5 GDPR-Completeness Checking Algorithm

Unlike prior studies [26, 34] that performed the GDPR-completeness checking only based on mandatory requirements, *PolicyChecker*'s checking algorithm more comprehensively detects unsatisfied mandatory requirements and also uniquely detects unsatisfied if-applicable requirements based on our identified requirement logic chains. We define the following notations used in *PolicyChecker*: (1) S: the set of satisfied requirements; (2) S_{legal}: the set of legal bases identified from a privacy policy; (3) $S_{interests}$: the set of legitimate interests identified from the policy; (4) P: the set of practices (Table 2) identified from the policy. Moreover, GDPR-completeness checking relies on the following requirement sets: (1) R_M: mandatory requirements shared by Arts 13 and 14; (2) $R_{M_{14}}$: mandatory requirements specific to Art 14; Algorithm 1 lays out the GDPR-completeness checking procedure. To detect unsatisfied mandatory requirements, the algorithm calculates the set difference between the mandatory requirements set and the satisfied requirements set. To detect unsatisfied if-applicable requirements, the algorithm operates based on the requirement logic chains described in Section 3.2. When the algorithm raises either a mandatory or a logic chain violation alert, it inserts the related information into a completeness analysis report. The process for detecting other types of violations (e.g., contact information, controller identity, etc.) is provided in Appendix B.4 in [39].

Algorithm 1 GDPR-Completeness Checking

Require: $R_M, R_{M_{14}}$ //Artifacts extracted from GDPR (Table 1)
Require: L //Set of requirement logic chains (Section 3.2)
Require: $S, S_{legal}, S_{interests}, P$ //Information identified from a policy
$V_M \leftarrow R_M - S$ //Set difference
$V_{M_{14}} \leftarrow R_{M14} - S$ //Set difference
if ("$invoke\_13$" $\in S \vee$ "$invoke\_14$" $\in S$) $\wedge$ ($V_M \neq \emptyset$) **then**
  for $v \in V_M$ **do**
    $raise\_mandatory\_violation(v)$
  end for
end if
if "$invoke\_14$" $\in S \wedge V_{M_{14}} \neq \emptyset$ **then**
  for $v \in V_{M_{14}}$ **do**
    $raise\_mandatory\_violation(v)$
  end for
end if
for $logic\_chain \in L$ **do**
  $Boolean\ pass \leftarrow logic\_chain.check(S, S_{legal}, S_{interests}, P)$
  if $not\ pass$ **then**
    $raise\_logic\_chain\_violation(logic\_chain)$
  end if
end for

5 RESULTS AND ANALYSIS

In this section, we presents our privacy policy collection process, the effectiveness of *PolicyChecker*, and our major findings.

5.1 App Privacy Policy Collection

Despite the fact that EU GDPR no longer applies to UK following Brexit, it is worth noting that UK has incorporated the full version of EU GDPR into its own legislation (Data Protection Act 2018) as UK GDPR [28]. As a result, apps in the UK Google Play store are still regulated by the GDPR standards. We choose UK Google Play store since the UK is the largest English-speaking country by population among the countries that have implemented GDPR requirements. Our crawling strategy is in line with previous studies which crawled privacy policies at a large scale for GDPR compliance analysis [30].

We first obtained the list of apps from AndroZoo [1] with a total of 6.5 million app names. We then crawled the metadata (such as developer name, user reviews, URL link to the privacy policy, etc.) of each Android apps available in the UK Google Play store from January 2023 to February 2023. Finally, we obtained apps that meet the following criteria for our study:

- Apps with at least one update after May 2018 (when GDPR was enacted).
- Apps with more than 10,000 downloads (i.e., apps with good popularity).
- Apps' metadata containing a privacy policy URL.

The first two criteria are identical to those used in [37] on studying GDPR violations in the Android app consent interfaces. The third criterion is for us to rule out apps that did not provide a privacy policy link on Google Play. Note that we do not consider the criterion used in [37] that requires apps to have sensitive permission request because our work focus on analyzing apps' privacy policies instead of their actual behaviors.

In total, we found 492,019 qualified apps with 205,159 unique developer names and 291,462 unique privacy policy URLs. Note that the number of unique URLs is less than the total number of apps due to an average of 2.4 apps belonging to the same developer sharing the same privacy policy. We then followed the procedure described in PolicyLint [4] to download the privacy policy HTML files using the Selenium web driver, convert them to plain text files, and remove non-English privacy policies. Among 291,462 privacy policy URLs, we excluded 29,367 URLs that are unable to access due to unresponsiveness or error codes and excluded 56,122 non-English privacy policies. In the end, our final dataset contains 205,973 privacy policies.

As shown in Table 3, *PolicyChecker* reported that no data collection is stated in 20.8% of the 205,973 privacy policies. We randomly selected 100 such privacy policies to manually analyze *PolicyChecker*'s detection accuracy on them. We found that the contents of 79 files are unrelated to the apps' privacy policies. Such unrelated files appear in our dataset due to the following main factors: (1) the privacy policy URL provided by a controller on Google Play points to an unrelated webpage, such as an Apache welcome page, a company's homepage, etc.; (2) the retrieved HTML file is an HTTP error message that was not handled by our HTML downloader. Although these 79 files are correctly detected by *PolicyChecker* as policies without data collection statements, incorrect detections may occur on a larger sample of policies and will be further discussed in Section 6.2. The contents of 17 files are related to the apps' privacy policies, but the policies do not mention any data collection practices or have negative sentiments in data collection statements (e.g., *"no data are collected or shared by us"*) and thus are detected by *PolicyChecker* as "no data collection" (detailed process is provided in Appendix B.3 in [39]). The contents of the remaining four files are related to the apps' privacy policies and contain data collection statements. However, *PolicyChecker* produced false detections due to the NER model's failure on detecting the correct entities in the data collection statements.

Table 3: The Distribution of Data Collection Practices Adopted in Mobile Apps' Privacy Policies

Data Collection Practice	Number of Policies
Direct collection only	133,298 (64.7%)
Indirect collection only	55 (0.0%)
Both direct and indirect collection	29,715 (14.4%)
# of analyzable privacy policies	**163,068 (79.2%)**
No data collection statement	42,905 (20.8%)

5.2 Overall Results

Table 3 shows that *PolicyChecker* detected data collection statements in 163,068 (79.2%) of the 205,973 privacy policies; therefore, those 163,068 analyzable privacy policies are mandated by GDPR Arts. 13 and 14 to provide all necessary information. The majority (64.7%) of app privacy policies state that personal data are directly collected from users; a smaller percentage (14.4%) of policies declare that personal data are collected from both users and third parties.

Overall, *PolicyChecker* reveals that only 1,116 (***0.7%***) of analyzable privacy policies provide all the necessary information. Conversely, ***99.3%*** of them are found to be incomplete as they had at least one completeness violation; 98.1% of them had at least one violation of mandatory requirement, while 73.0% of them had at least one violation of the if-applicable requirement logic chain. As shown in Figure 2, seven or more unique completeness violations are detected in over 50% of privacy policies; six or more unique mandatory requirements are unfulfilled in over 50% of privacy policies. Note that "completeness violations" are calculated by adding the number of unsatisfied mandatory requirements and if-applicable requirement logic chains in a privacy policy, with each type of unsatisfied requirement and logic chain counted only once in a privacy policy. For requirements R3 and R4, we condense them into one "retention policy" violation when both requirements are unsatisfied in a privacy policy.

To look deeper into the results, we report the density of the 16 different types of completeness violations in Figure 3. Note that these 16 types of violations correspond to the 23 requirements described in Table 1. As discussed in Section 3, violations and requirements are not one-to-one mappings because one violation can be triggered by multiple requirements (especially for if-applicable requirements). For example, when a controller intends to transfer personal data to a third country (R10), the existence or absence of an adequacy decision (R11) or information about transfer safeguards (R12) should be provided. These three requirements correspond to the "data transfer chain" violation. Our results suggest that mandatory requirements

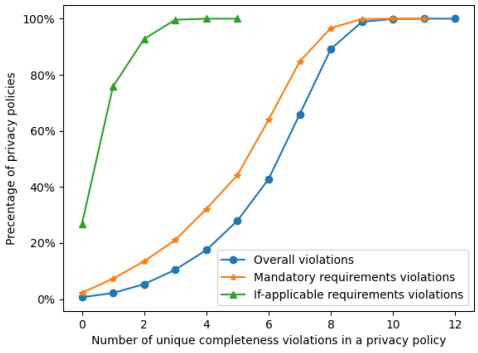

Figure 2: Cumulative Distributions of Completeness Violations in 163,068 Analyzable Privacy Policies

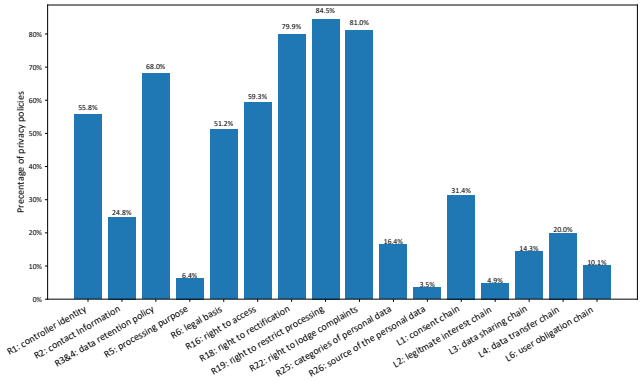

Figure 3: Density of Different Types of Completeness Violations in 163,068 Analyzable Privacy Policies

regarding data retention (R3 and R4) and users' rights related to data processing (R16, R18, R19, and R22) are frequently unfulfilled, while logic chains regarding the data transfer practice and users' right to withdraw consent under consent-based processing (L1 and L4) are frequently broken. Our results in Figures 2 and 3 reveal the prevalence of GDPR-incompleteness in mobile apps' privacy policies; such results were not derived by prior studies (e.g., [26, 34]).

We then selected 300 privacy policies from our dataset to annotate the ground-truth completeness violations. Our manual annotation and analysis of ground-truth completeness violations serve two purposes: (1) Assessing *PolicyChecker*'s effectiveness in identifying GDPR-completeness violations (Section 5.3); (2) Providing insights for an in-depth analysis of our findings (Section 5.4).

5.3 Manual Evaluation

To have a diverse set of ground-truth policies for evaluating the effectiveness of *PolicyChecker*, we selected the 300 privacy policies in two ways. We randomly selected the first 100 policies, but adjusted the composition of this first subset of 100 ground-truth policies based on the cumulative distributions of completeness violations (Figure 2) in 163,068 analyzable privacy policies. We did so by first detecting the number of completeness violations in each ground-truth policy using *PolicyChecker*, and then adding or discarding randomly-sampled policies so that 10% of the policies have zero to four (exclusively) violations, 55% of the policies have four to eight (exclusively) violations, and 35% of the policies have eight to 12 violations as shown in Figure 2. We randomly selected 200 other policies without any distribution adjustment and without filtering through *PolicyChecker*. This second subset of 200 randomly-sampled ground-truth policies can help us evaluate the effectiveness of *PolicyChecker* especially in terms of its recall ratio. Additionally, for all these 300 policies, we utilize the cosine similarity score computed between two documents' TF-IDF representation (TF-IDF cosine similarity [43]) to avoid selecting policies with similar content (e.g., policies generated by the same policy generator).

We evaluate the effectiveness of *PolicyChecker* on all the 16 types of violations based on the 300 policies. Our manual annotation process includes the following steps: (1) the first and second authors of this paper manually analyzed 300 privacy policies by carefully reading each statement in a given policy and recorded it with the requirement(s) that the statement fulfills; (2) After having a list of fulfilled requirements for a privacy policy, each of the first two authors follows the completeness analysis process described in Section 4.5 to determine any potential completeness violation and mark them accordingly; (3) The first two authors then discuss the annotations in several sessions spanning over three weeks to reach an agreement on all 300 privacy policies.

We measured the inter-rater agreement in our annotations using Krippendorff's alpha-reliability (α) [32]. The average α of all 16 types of violations in the 300 privacy policies is 0.78. The low (but acceptable for tentative conclusions [31]) inter-rater agreement is mainly due to the ambiguous sentences in privacy policies, which is well recognized in the literature (e.g., in [34]).

Overall, we identified 1,252 completeness violations with a median of six among the 300 policies. On average, over the 300 ground-truth policies, *PolicyChecker* achieves 88.5% on accuracy, 78.2% on precision, 83.0% on recall, and 80.0% on F1-score, with the details in Table 7 of Appendix C.2 in [39]. The recall ratio on the second subset of 200 randomly-sampled ground-truth policies is 81.0%. Note that a "positive" for a requirement in a policy represents that there is at least one detected violation of the requirement in the policy; a "negative" for a requirement in a policy represents that no violation of the requirement is detected in the policy. Our manual evaluation reports the effectiveness of *PolicyChecker* at the policy (instead of the sentence) level. This is because our overall design and measurement in this paper focus on whether certain information is absent in an entire policy. As in our manual evaluation, developers or analysts (if they want to use *PolicyChecker* or its analysis results) may need to spend some effort to check the detailed false positives or negatives even just for one policy.

We analyzed all 594 false detection results (out of all $4,800 = 16 \times 300$ detection results, i.e., with a 12.4% false detection ratio) including 380 false positives and 214 false negatives, with respect to the five major processing steps of the *PolicyChecker* framework (Figure 1). In Step (1), a practice identification error occurs when a sentence contains verbs that are not included in our verb list; such errors account for 7.0% of the false detection results. In Step (2), the SRL model fails to accurately predict the predicate-argument structure with overly complex sentence structures; such errors account for 11.4% of the false detection results. In Step (3), the semantic role matching rules do not incorporate certain edge cases; such

errors account for 5.7% of the false detection results. In Step (4), the NER model fails to recognize an entity in a sentence or the *content identification* process fails to identify the correct content in a sentence; such errors account for 20.0% of the false detection results. In Step (5), our design does not accommodate the situation where the detailed description of an entity (e.g., data and third-party entities) in a sentence is provided in a later part of the policy (i.e., cross-sentence references); such errors account for 24.3% of the false detection results. In addition, *PolicyChecker* determines if a noun represents the controller identity (R1) using external information (e.g., a developer's identity on Google Play); however, many developers provided different and questionable identities (e.g., the app names) in their apps' privacy policies; such errors account for 21.4% of the false detection results. Other minor factors, including text pre-processing errors and the information in privacy policies aimed at adhering to other regional regulations (e.g., CCPA), contribute to 10.2% of the false detection results.

We also briefly evaluated the effectiveness of *PolicyChecker* on identifying data practices (Table 2) at the sentence level. A sentence describes a data practice if it contains a valid verb (Step (1)), a correct predicate-argument structure (Steps (2), (3)), and valid meanings in its arguments (Step (4)). We randomly selected 300 ground-truth sentences from our dataset of 205,973 policies to perform the manual annotation (the inter-rater agreement using Krippendorff's alpha-reliability [32] is α=0.83). For this sentence level evaluation, PolicyChecker achieves 98.3% on accuracy, 88.9% on precision, 85.7% on recall, and 86.6% on F1-score.

5.4 Deeper Analysis

5.4.1 The state of GDPR-completeness violations. As shown in Figure 3, the most common completeness violations are caused by the lack of disclosure of users' rights related to data processing. For example, the *right to restrict processing (R19)* and *right to lodge a complaint (R22)* statements are missing in 84.5% and 81.0% of privacy policies, respectively. For if-applicable requirements, we found that 20% of policies failed to provide the *data transfer adequacy decision or transfer safeguard information (R11 and R12)*, and 31.4% of policies did not give users the *right to withdraw consent (R8)* information when conducting consent-based data processing.

User rights violations: Results from both the automatic and manual analyses indicate a severer absence of information regarding users' rights to access and rectify personal data, restrict data processing, and lodge a complaint with authority. Ensuring fair and transparent processing is one of the fundamental obligations of a data controller, which includes the disclosure of users' rights related to data processing. Nonetheless, our findings, along with those of previous research [34], indicate that the majority of data controllers fail to fulfill their obligations.

User consent violations: As per GDPR Art. 6, a controller may establish a legal basis for processing certain categories of personal data by obtaining user consent. In such cases, both Arts. 13 and 14 require the data controller to inform users about the *right to withdraw consent (R8)*. Unlike prior work that neglected the if-applicable requirement *R8* in their completeness violation detection, *PolicyChecker* is able to recognize the condition (e.g., a controller declares the consent-based processing in the policy) that will make the requirement *R8* applicable and further report corresponding completeness violations (Section 4.5).

Our results in Figure 3 show that only 48.6% of privacy policies provide a legal basis justification. In addition, 77,522 (98.0%) privacy policies state user consent as their legal basis for processing. However, 66.1% of them do not indicate if users have the right to withdraw their consent. Nguyen et al. [37] reported that 84 out of 100 examined app consent interaction interfaces do not provide any option for users to withdraw consent. Our results further indicate that the information regarding consent withdrawal is largely missing even in the basic privacy policy information channel that users rely on.

Dark patterns in soliciting user consents. Our manual analysis of the 300 ground-truth policies (as described in Section 5.3) suggests that controllers often describe the user consent acquirement as a default setting, and their consent related policy statements are sometimes expressed in an implicit or vague manner. In particular, we found that controllers declare in 27.0% of the 300 ground-truth policies that user consent is obtained automatically as soon as the users begin using the apps. For example, *"If you choose to use my Service, then you agree to the collection and use of information in relation to this policy." - Math Formula app; "Your Consent: If you agree with our Privacy Policy, there is nothing you need to do" - Positively Made app; "by utilizing the Program ... you consent to have your personal data transferred to ... " - Chicken Salad app*. Such practices violate GDPR's requirement that mandates user consent to be obtained through a statement or clear affirmative action by a user (GDPR Art.4.(11)). We also found that controllers have the implicit or vague consent solicitation statements in at least 1.7% of the 300 ground-truth policies. For example, *"We may use your Personal Information to ... unless you have not consented to allow" - Mobilix Solutions app*, is a double-negative sentence that can be very confusing for users. Both the *default setting* and *double-negative sentence* are manipulative dark pattern designs [25] well-studied in the HCI research domain [35, 38]. Our findings of the use of default setting and implicit language in consent-solicit statements align with the prior study [38] of dark patterns in the consent management platform design, in which Nouwens et al. reported that user consent is often obtained implicitly by the developers.

It should be noted that the purpose of *PolicyChecker* is to assess the completeness of a policy based on the requirements outlined in GDPR Arts. 13 and 14. It does not take into account misleading or ambiguous statements in the policy as violations. Nonetheless, it is important to point out that controllers have an obligation to provide a high-quality privacy policy based on the GDPR Art. 12: *"The controller shall take appropriate measures to provide any information ... to the data subject in a concise, transparent, intelligible and easily accessible form, using clear and plain language"*.

Takeaways: According to our analysis, the most common completeness violations in mobile app privacy policies are caused by the lack of disclosure of users' rights related to data processing. In particular, statements regarding the *right to restrict processing* (R19) and *right to lodge a complaint* (R22) are missing in 84.5% and 81.0% of privacy policies, respectively. Additionally, *PolicyChecker* finds that 51.4% of privacy policies failed to provide legal grounds for processing personal data. Among the policies that provided the legal basis justification, 98.0% of them stated that processing is based on

user consent. However, users are not given the right to withdraw their consent in 66.1% of the cases, which are direct violations of the GDPR principle on data processing *fairness*. In addition, our manual analysis result indicates that privacy policies use implicit and vague language to solicit user consent. Such practices violate the GDPR requirement on disclosing information using the *plain* and *simple* language (Art. 12).

5.4.2 Behind the violations. We now investigate the potential causes behind GDPR-completeness violations. Note that our investigation is solely based on the privacy policy analysis, and its results including our tentative interpretations should be further verified by future user studies.

Poor transparency on disclosing indirect data collection. PolicyChecker reports 32,349 completeness violations (due to unsatisfied Art.14-specific requirements R25 and R26) in 29,770 privacy policies that stated indirect data collection (e.g., collecting personal data from a third-party service). In particular, we found that 26,743 (89.8%) policies with indirect data collection statements failed to provide a detailed description of what types of personal data were obtained from third parties (R25). Furthermore, 5,707 (19.2%) policies with indirect data collection statements did not disclose the source (e.g., the identity of a third party) of the personal data (R26). However, during our manual analysis of the 300 ground-truth policies, we found that the majority (84%) of privacy policies are very transparent in disclosing the types of personal data obtained directly from users. Therefore, we observe an inconsistent pattern of information disclosure under different data collection scenarios.

We have two tentative interpretations regarding this observation. First, controllers might be unaware of the additional GDPR requirements (R25, R26) under indirect data collection situations; therefore, they do not see the need to disclose the types of data when the data are collected from third-party services. Second, controllers might lack the understanding of the complete data collection practice conducted by their own apps, especially when personal data are being collected by the third-party services used in the apps; therefore, even if they are aware of the GDPR requirements, they might not have the necessary knowledge to report what types of personal data are received from a third party. Although, to our knowledge, there is no existing user study specifically on developers' understanding of requirements in GDPR Arts. 13 and 14, studies such as [2] on developers' privacy compliance processes of child-directed mobile apps have indicated that developers lack a good understanding of their apps' data collection practices when third-party services are used. Additionally, Alomar et al. [2] pointed out that while 78% of 50 interviewed organizations (all subject to GDPR) are aware that their apps are required to comply with GDPR, they are not fully aware of their obligations under GDPR. Alomar et al. [2] also pointed out in their user study that "smaller developers" (i.e., developers in small teams or businesses) often follow some "best-effort" models and tend to ignore privacy compliance requirements. We do not have the accurate size information about the app development teams, but we observed that less popular (based on the number of downloads) apps tend to have more violations than more popular apps. For example, we found an average of 6.2 violations in the privacy policies of the 10,000 least downloaded apps and 4.8 violations in the privacy policies of the 10,000 most downloaded apps.

Poor practices in using tools to create a privacy policy. As mentioned in [2], small-to-medium-size organizations often find it challenging to comply with regulatory standards, and they are in urgent need of usable tools to help them identify and fix mobile app privacy issues. In recent years, privacy policy auto-generating tools have gained significant popularity. However, the reliability of using automatic tools to generate privacy policies might be questionable since the tools might not provide GDPR-specific accommodations. And due to the lack of a comprehensive understanding of GDPR requirements, data controllers might not be able to verify the completeness of a policy generated by a non-compliant tool.

We conducted a preliminary experiment to investigate the state of automatically generated privacy policies and their potential correlations with the widespread GDPR-completeness violations. To do so, we first searched "mobile app privacy policy generator" on Google and selected the five top-recommended generators from the first page of the search results (excluding any sponsored ads and duplicated results): *Generator by Firebaseapp* [21], *Generator by Termly* [24], *Generator by AppPrivacyPolicy* [20], *Generator by PrivacyPolicies* [22], and *Generator by PrivacyPolicyOnline* [23]. Note that we performed the search in the Incognito mode to minimize any potential influence on the results.

Next, we created five *baseline privacy policies* from these five generators, respectively, by following their instructions and providing all required and optional information. We then manually analyzed the GDPR-completeness of these five baseline privacy policies. The detailed analysis results are listed in Table 8 of Appendix D.1 in [39]. We can see that four out of five baseline privacy policies contain six to eight completeness violations, indicating their inability to generate GDPR-complete privacy policies. Although the privacy policy generated from the *Generator by Termly* [24] is GDPR-complete, a considerable amount of effort is needed for using this generator.

We further measured the prevalence of using these five generators by performing TF-IDF cosine similarity comparisons between the privacy policies in our dataset and the five baseline privacy policies. We refer to the privacy policies with more than 60% and less than 90% textual similarity (compared with a baseline privacy policy) as *baseline-similar* policies, and those with more than 90% textual similarity as *baseline-identical* policies. The detailed analysis results are listed in Table 8 of Appendix D.1 in [39]. In summary, 25,010 (12.1%) privacy policies in our dataset are *baseline-similar* and 43,351 (21.0%) are *baseline-identical* when comparing to the baseline privacy policy generated from the *Generator by Firebaseapp* [21]; only a small number of policies are either *baseline-similar* or *baseline-identical* when comparing to the baseline privacy policies generated from the other four generators. These results imply that *Generator by Firebaseapp* [21] is popularly used and is the dominant privacy policy generator based on our dataset. Unfortunately, this generator does not produce GDPR-complete privacy policies as we discussed above.

Looking deeper into the results from the *Generator by Firebaseapp* [21], we found an average of 99.4% overlap between the completeness violations in its baseline privacy policy and the completeness violations detected from those *baseline-identical* privacy

policies, and found an average of 80.5% overlap between the completeness violations in its baseline privacy policy and the completeness violations detected from those *baseline-similar* policies. We then randomly sampled ten *baseline-similar* policies and ten *baseline-identical* policies for further analysis. Out of the ten sampled *baseline-similar* files, four contain unrelated HTML residues from converting HTML to text files. After removing the unrelated contents, these files also have more than 90% similarities with the baseline. The remaining six policies all contain custom information that can be categorized as follows: app permission policy, paid services, detailed descriptions of the data collected, security, and user rights. We found that the additional information provided in five out of six policies does not reduce the number of completeness violations, and only one policy added statements regarding users' rights to data access and rectification. For the ten sampled *baseline-identical* files, all of them have the same contents as the baseline privacy policy except for the basic information such as the developer name, third-party service, and contact information.

Takeaways: We observed that controllers exhibit an inconsistent pattern of information disclosure under different data collection scenarios, which may reflect their lack of understanding of GDPR requirements and their own apps' data collection practices.

Our analysis indicates that the *Generator by Firebaseapp* [21] only produces a bare minimum privacy policy but closely resembles 21.0% of the mobile app privacy policies in our dataset. Such policies in our dataset might be generated by developers directly using this generator or copied by developers from the policies of other apps. Our result suggests that at least in the case of the 21.0% of privacy policies that are *baseline-identical* in terms of this dominant generator, app developers failed to perform their due diligence, such as verifying the GDPR-completeness of their apps' privacy policies and making local adaptations. As a result, a set of unsatisfied requirements are shared by developers in their apps' privacy policies. It is worth noting that this result is just a conservative estimation of developers' poor privacy policy composition practices, as we only experimented with five out of many policy generators.

Additionally, we observed from the *baseline-similar* privacy policies that controllers tend to add additional information to the auto-generated policies instead of modifying the generated statements to match their apps' actual practices. This raises a concern about whether those statements (e.g., *"Your data are not collected by me in any way"*) generated by tools by default and shared among a large number of privacy policies can be trusted by users.

6 DISCUSSION

In this section, we first discuss the implications of our findings along with our recommendations to app developers. We then discuss the limitations of our work and the potential future work.

6.1 Implications and Recommendations

Fundamental user information obligations must be fulfilled. Our key findings summarized in Section 5.4 show that a majority of developers fail to fulfill their fundamental transparency obligations. Our findings on the severe absence of information regarding the core GDPR requirements on processing transparency are consistent with what other researchers (e.g., Liu et al. [34]) found about the significant lack of disclosure on users' *right to access* and *right to restrict processing*. Thus, we recommend mobile app developers to review their internal compliance processes and establish appropriate channels for users to exercise their rights under GDPR Arts. 13 and 14. Furthermore, developers should disclose such channels to users (e.g., through privacy policy updates) after their establishment. Meanwhile, developers should identify the most appropriate legal basis for justifying their apps' processing of personal data. Our analysis of information provision requirements in Appendix A.2 in [39] provides information for app developers to implement our recommendations.

In addition, recall that in Section 5.4.2 we looked into controllers' inconsistent patterns of information disclosure under different data collection scenarios, which may reflect their lack of understanding of GDPR requirements and their own apps' data collection practices. Our findings and tentative interpretations align with what other researchers (e.g., Alomar et al. [2]) found about organizations' lack of a good understanding of their obligations under GDPR and their apps' data practices when third-party services are used. Thus, we recommend mobile app developers to review the privacy policies of third-party services before integrating them into the apps.

If-applicable requirements should be met. Besides the completeness analysis of mandatory requirements, in this work, we take a first look into how well if-applicable requirements in GDPR are met by mobile apps' privacy policies. Our analysis results indicate that 66.1% of privacy policies that rely on user consent to process personal data do not entitle users to withdraw their consent. In addition, developers also failed to report the identities of the personal data recipients in 14.3% of privacy policies. Although no existing study has revealed the state of the GDPR-completeness violations in terms of the if-applicable requirements, our findings on the lack of disclosing users' right to withdraw consent are consistent with what was reported in [37] that many apps' consent interfaces do not provide users the option to withdraw consent. Our results further indicate that consent withdrawal information is missing even in the basic privacy policy information channel that users rely on. Therefore, we recommend developers to pay more attention to the additional requirements that might apply when they include certain information in privacy policies. Our analysis and discussion in Section 3.2 can help developers review and amend the lack of information concerning if-applicable requirements.

Automatic policy generation needs significant improvement. In Section 5.4.2, we introduced our preliminary experiment on the impact of policy generation tools on the state of GDPR-completeness violations. Our findings suggest that 12.1% and 20.0% of privacy policies in our dataset are similar and identical, respectively, to the policy generated by the *Generator by Firebaseapp [21]*. Such automatic policy generation practices are concerning, particularly when developers lack a good understanding of the GDPR requirements, which makes them unable to verify the completeness of the generated or copied policies and perform necessary local adaptations. Recall that in our manual analysis (Section 5.4.2) of *baseline identical* policies, we found that only the basic contents such as the developer name, contact information, and list of third-party services were modified. For *baseline similar* policies, we found that developers tend to add new information to the generated policy

and generally do not modify the existing contents. Our manual analysis results confirm that the aforementioned concerns do, in fact, exist in many cases. Such poor practices in creating privacy policies undermine the trustworthiness of the claims made by developers in these policies. We recommend app developers to exercise caution when generating privacy policies using such tools or copying them from other apps, and perform due diligence to make app-specific modifications to ensure that privacy policies accurately reflect apps' actual data processing practices and legal stances.

6.2 Limitations and Potential Future Work

The framework design aspect. We recognize three limitations of our *PolicyChecker* framework design in identifying GDPR requirement-related information from a privacy policy sentence. First, we rely on an SRL model trained on a generic English semantic dataset. Therefore *PolicyChecker*'s effectiveness is naturally bounded by the effectiveness of the SRL model. For example, the precision and F1 scores on the violations of some requirements are low as shown in Table 7 of Appendix C.2 in [39]. As SRL is still a challenging task in the NLP domain, the state-of-art model only achieves 86.49% F1-score on the Ontonotes 5.0 dataset [45]. In addition, training a domain-adapted SRL model is a challenging task commonly recognized in the literature [7, 48]. Thus, we expect that some future work would implement domain-adapted SRL models.

Second, currently in the early steps of PolicyChecker, we do not comprehensively extract all contextual information among sentences. PolicyChecker analyzes one sentence at a time from Steps (1) to (4), although it takes all sentences into account for completeness analysis in Step (5) and also leverages contextual information in detecting logic chain violations. We reported cross-sentence references as one potential factor leading to false detection results (Section 5.3). Capturing more contextual information in early steps can be an improvement to *PolicyChecker* in the future.

Third, in this work, we only consider privacy policies written in English. As shown in Section 5.1, 27.2% of privacy policies we retrieved from the UK Google Play store were written in other languages and excluded from our study. This limitation has hindered our ability to scale up our findings to encompass privacy policies from a wider range of app developers. However, it is challenging to include semantic pattern designs for multiple languages in one work since different languages have unique grammatical rules. It will be an interesting future work to explore a multiple-language version of the *PolicyChecker* framework.

The data collection aspect. Our work neither considers privacy policies provided in pdf format nor retrieves policies that are hosted using document viewers such as Google Docs. We rely on future work to implement a pdf-to-text conversion module and an image recognition module to convert privacy policy screenshots taken from document viewers' web pages to text. In addition, we recognize the potential shortage in our data collection strategy as we only focus on mobile apps from the UK Google Play store. We cannot derive findings to reveal the differences and similarities in the state of GDPR-completeness violations in mobile apps' privacy policies across different members of the European Nations. Expanding the data collection to include more regions' Google Play stores is a potential future work.

The result analysis aspect. As presented in Section 5.4, our interpretations regarding developers' lack of an understanding of GDPR requirements were solely derived from the privacy policy analysis. We believe that future user studies are necessary to fully understand developers' perceptions of GDPR and their practices in creating privacy policies. In addition, our analysis of the auto-generated privacy policies is limited to only five popular app privacy policy generators. Future work is needed to systematically analyze the impact of privacy policy generators at a larger scale, especially regarding how developers (mis)use such generators.

7 CONCLUSION

In this paper, we investigated the fulfillment of GDPR Arts. 13 and 14 requirements in mobile apps' privacy policies. We designed the *PolicyChecker* framework by taking a *rule and semantic role based* approach. Using *PolicyChecker*, we conducted the first large-scale analysis on 205,973 mobile app privacy policies. Our results indicated that many app developers failed to fulfill their fundamental transparency obligations; meanwhile, if-applicable requirements in GDPR are overlooked by most app developers. We conjectured that controllers' lack of understanding of some GDPR requirements and their poor practices in composing a privacy policy can be the potential major causes behind the GDPR-completeness violations. We further made recommendations for app developers to improve the completeness of their apps' privacy policies. Note that *PolicyChecker*'s use is not limited to detecting GDPR-completeness violations in mobile apps' privacy policies; it can also be used in other application contexts (e.g., websites' privacy policies). We share our long version of the paper, our code, and our dataset [39] for researchers to conduct future privacy policy compliance studies.

ACKNOWLEDGMENTS

We thank anonymous reviewers and our shepherd for their valuable suggestions. Anhao and Chuan were partially supported by funding from Meta; Weiping was partially supported by the NSF grant CNS-2246143.

REFERENCES

[1] Kevin Allix, Tegawendé F Bissyandé, Jacques Klein, and Yves Le Traon. 2016. Androzoo: Collecting millions of android apps for the research community. In Proceedings of the International Conference on Mining Software Repositories (MSR).
[2] Noura Alomar and Serge Egelman. 2022. Developers Say the Darnedest Things: Privacy Compliance Processes Followed by Developers of Child-Directed Apps. In Proceedings of the Privacy Enhancing Technologies Symposium (PETS).
[3] Ryan Amos, Gunes Acar, Eli Lucherini, Mihir Kshirsagar, Arvind Narayanan, and Jonathan Mayer. 2021. Privacy policies over time: Curation and analysis of a million-document dataset. In Proceedings of The Web Conference.
[4] Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. 2019. PolicyLint: Investigating Internal Privacy Policy Contradictions on Google Play.. In Proceedings of the USENIX Security Symposium (USENIX Security).
[5] Jaspreet Bhatia and Travis D Breaux. 2017. A data purpose case study of privacy policies. In Proceedings of the IEEE International Requirements Engineering Conference (RE).
[6] Jaspreet Bhatia and Travis D Breaux. 2018. Semantic incompleteness in privacy policy goals. In Proceedings of the IEEE International Requirements Engineering Conference (RE).
[7] Duc Bui, Yuan Yao, Kang G Shin, Jong-Min Choi, and Junbum Shin. 2021. Consistency analysis of data-usage purposes in mobile apps. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS).

[8] Anupam Das, Nikita Borisov, and Matthew Caesar. 2014. Do you hear what I hear? Fingerprinting smart devices through embedded acoustic components. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS).

[9] dataprotection.ie 2021. Data Protection Commission announces decision in WhatsApp inquiry. Retrieved January 20, 2023 from https://www.dataprotection.ie/en/news-media/press-releases/data-protection-commission-announces-decision-whatsapp-inquiry

[10] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. 2019. We Value Your Privacy... Now Take Some Cookies: Measuring the GDPR's Impact on Web Privacy. Informatik Spektrum (2019).

[11] Sanorita Dey, Nirupam Roy, Wenyuan Xu, Romit Roy Choudhury, and Srihari Nelakuditi. 2014. AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable.. In Proceedings of the Network and Distributed System Security Symposium (NDSS).

[12] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. 2012. Android permissions: User attention, comprehension, and behavior. In Proceedings of the Symposium on Usable Privacy and Security (SOUPS).

[13] Sandra Gabriele and Sonia Chiasson. 2020. Understanding fitness tracker users' security and privacy knowledge, attitudes and behaviours. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI).

[14] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. [n. d.]. AllenNLP: A Deep Semantic Natural Language Processing Platform. In Proceedings of Workshop for NLP Open Source Software (NLP-OSS).

[15] GDPR. 2023. Art. 12 Transparent information, communication and modalities for the exercise of the rights of the data subject. Retrieved January 20, 2023 from https://gdpr.eu/article-12-how-controllers-should-provide-personal-data-to-the-subject/

[16] GDPR. 2023. Art. 13 Information to be provided where personal data are collected from the data subject. Retrieved January 20, 2023 from https://gdpr.eu/article-13-personal-data-collected/

[17] GDPR. 2023. Art. 14 Information to be provided where personal data have not been obtained from the data subject. Retrieved January 20, 2023 from https://gdpr.eu/article-14-personal-data-not-obtained-from-data-subject/

[18] GDPR. 2023. Art.5 Principles relating to processing of personal data. Retrieved January 20, 2023 from https://gdpr.eu/article-5-how-to-process-personal-data/

[19] GDPR. 2023. Complete guide to GDPR compliance. Retrieved January 20, 2023 from https://gdpr.eu/

[20] Generator by AppPrivacyPolicy 2023. "Generator by AppPrivacyPolicy". Retrieved August 10, 2023 from https://www.app-privacy-policy.com/

[21] Generator by Firebaseapp 2023. "Generator by Firebaseapp". Retrieved January 20, 2023 from https://app-privacy-policy-generator.firebaseapp.com/

[22] Generator by PrivacyPolicie 2023. "Generator by PrivacyPolicies", url = "https://privacypolicies.com/", lastaccessed = "August 10, 2023",.

[23] Generator by PrivacyPolicyOnline 2023. "Generator by PrivacyPolicyOnline". Retrieved August 10, 2023 from https://privacypolicyonline.com/

[24] Generator by Termly 2023. "Generator by Termly". Retrieved August 10, 2023 from https://termly.io/products/privacy-policy-generator/

[25] Colin M Gray, Yubo Kou, Bryan Battles, Joseph Hoggatt, and Austin L Toombs. 2018. The dark (patterns) side of UX design. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI).

[26] Rajaa El Hamdani, Majd Mustapha, David Restrepo Amariles, Aurore Troussel, Sébastien Meeùs, and Katsiaryna Krasnashchok. 2021. A combined rule-based and machine learning approach for automated GDPR compliance checking. In Proceedings of the International Conference on Artificial Intelligence and Law (ICAIL).

[27] ICO. 2023. Information Commissioner's Office. Retrieved January 20, 2023 from https://ico.org.uk/

[28] ICO. 2023. Overview – Data Protection and the EU. Retrieved January 20, 2023 from https://ico.org.uk/for-organisations/dp-at-the-end-of-the-transition-period/overview-data-protection-and-the-eu/

[29] ICO. 2023. What privacy information should we provide? Retrieved January 20, 2023 from https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/the-right-to-be-informed/what-privacy-information-should-we-provide/

[30] Georgios Kampanos and Siamak F Shahandashti. 2021. Accept all: The landscape of cookie banners in Greece and the UK. In Proceedings of the International Conference on ICT Systems Security and Privacy Protection (SEC).

[31] Klaus Krippendorff. 2004. Reliability in Content Analysis: Some Common Misconceptions and Recommendations. Human Communication Research 30, 3 (2004).

[32] Klaus Krippendorff. 2011. Computing Krippendorff's alpha-reliability. (2011).

[33] Thomas Linden, Rishabh Khandelwal, Hamza Harkous, and Kassem Fawaz. 2020. The privacy policy landscape after the GDPR. In Proceedings on Privacy Enhancing Technologies (PETS).

[34] Shuang Liu, Baiyang Zhao, Renjie Guo, Guozhu Meng, Fan Zhang, and Meishan Zhang. 2021. Have you been properly notified? automatic compliance analysis of privacy policy text with GDPR article 13. In Proceedings of The Web Conference.

[35] Arunesh Mathur, Gunes Acar, Michael J Friedman, Eli Lucherini, Jonathan Mayer, Marshini Chetty, and Arvind Narayanan. 2019. Dark patterns at scale: Findings from a crawl of 11K shopping websites. Proceedings of the ACM on Human-Computer Interaction (CSCW) (2019).

[36] Keika Mori, Tatsuya Nagai, Yuta Takata, and Masaki Kamizono. 2022. Analysis of Privacy Compliance by Classifying Multiple Policies on the Web. In Proceedings of the IEEE Annual Computers, Software, and Applications Conference (COMPSAC).

[37] Trung Tin Nguyen, Michael Backes, and Ben Stock. 2022. Freely Given Consent? Studying Consent Notice of Third-Party Tracking and Its Violations of GDPR in Android Apps. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS).

[38] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. 2020. Dark patterns after the GDPR: Scraping consent pop-ups and demonstrating their influence. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI).

[39] PolicyChecker Codebase 2023. "The long version, code, and dataset for PolicyChecker". Retrieved September 4, 2023 from https://github.com/AndyXiang945/PolicyChecker

[40] Ellen Poplavska, Thomas B. Norton, Shomir Wilson, and Norman M. Sadeh. 2020. From Prescription to Description: Mapping the GDPR to a Privacy Policy Corpus Annotation Scheme. In International Conference on Legal Knowledge and Information Systems.

[41] Joel Reardon, Álvaro Feal, Primal Wijesekera, Amit Elazari Bar On, Narseo Vallina-Rodriguez, and Serge Egelman. 2019. 50 ways to leak your data: An exploration of apps' circumvention of the android permissions system. In Proceedings of the USENIX Security Symposium (USENIX Security).

[42] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).

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

[44] Yan Shvartzshnaider, Noah Apthorpe, Nick Feamster, and Helen Nissenbaum. 2019. Going against the (appropriate) flow: A contextual integrity approach to privacy policy analysis. In Proceedings of the AAAI Conference on Human Computation and Crowdsourcing (HCOMP).

[45] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. Linguistic Data Consortium, Philadelphia, PA (2013).

[46] Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N Cameron Russell, et al. 2016. The creation and analysis of a website privacy policy corpus. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL).

[47] Zihan Zhang, Meng Fang, Ling Chen, and Mohammad Reza Namazi-Rad. 2022. Is Neural Topic Modelling Better than Clustering? An Empirical Study on Clustering with Contextual Embeddings for Topics. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).

[48] Lu Zhou, Chengyongxiao Wei, Tong Zhu, Guoxing Chen, Xiaokuan Zhang, Suguo Du, Hui Cao, and Haojin Zhu. 2023. POLICYCOMP: Counterpart Comparison of Privacy Policies Uncovers Overbroad Personal Data Collection Practices. (2023).

[49] Sebastian Zimmeck, Peter Story, Daniel Smullen, Abhilasha Ravichander, Ziqi Wang, Joel Reidenberg, N Cameron Russell, and Norman Sadeh. 2019. Maps: Scaling privacy compliance analysis to a million apps. In Proceedings on Privacy Enhancing Technologies (PETS).