Assessment of Privacy Risks in Mobile and Web Applications/Services

by

Hoang Duc Bui

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2022

Doctoral Committee:

Professor Kang G. Shin, Chair
Professor J. Alex Halderman
Professor Rada Mihalcea
Assistant Professor Florian Schaub

Hoang Duc Bui

ducbui@umich.edu

ORCID iD: 0000-0002-9529-0797

To

My mother, Lê Thị Hà, and my father, Bùi Ngọc Thái, for their unwavering love.

And my brother, Bùi Ngọc Minh, for taking care of our family for me to chase my dreams.

ACKNOWLEDGEMENTS

First and foremost, I would like to sincerely thank my advisor, Kang G. Shin, for advising me throughout my PhD program. During the hard time and slow progress in my first two years, he patiently helped me to surpass the hardship. He gave me great freedom in my work while handling all time-consuming logistics, so I only needed to focus on my research. I really appreciate his timely responses whenever I needed help even though he was very busy. I enjoy every moment of my long journey under the guidance of my advisor.

I would like to thank my dissertation committee members, Professor J. Alex Halderman, Professor Rada Mihalcea and Professor Florian Schaub, for all of their valuable feedback and insightful thoughts about my dissertation. I also want to thank Professor Peter Honeyman and Professor Daniel Genkin for joining my prelim exam committee and providing me with constructive comments.

I am fortunate to collaborate with excellent researchers, Jong-Min Choi and Junbum Shin at Samsung Research, as well as Michigan CSE students, Yuan Yao and Brian Tang. Without their help, I would have never finished some of my papers.

I also would like to thank all RTCL members who overlapped with me during my time at Michigan, especially, Chun-Yu Chen, Mert Pesé, Juncheng Gu, Yu-Chih Tung, Kyong Tak Cho, Youngmoon Lee, Arun Ganesan, Dongyao Chen, Taeju Park, Eugene Kim, Youssef Tobah, Hsun-Wei Cho, Noah Curran, Wei-Lun Huang, Mingke Wang, Haichuan Ding, Jinkyu Lee, Hoon Sung Chwa, Suining He, Xiufeng Xie and Hamed Yousefi.

The research reported in this dissertation was supported in part by the US National Science Foundation under Grant No. CNS-1646130 and the Army Research Office under

Grant No. W911NF-21-1-0057. Additionally, the work in Chapter IV was supported in part by Samsung Research.

Finally, I want to thank my family for their unconditional love and continuous encouragement during my long study. Without them, I will not be here today.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

LIST OF ABBREVIATIONS

CCPA	California Consumer Privacy Act	130
E2E	End-to-End	3
FIPP	Fair Information Practice Principles	1
FTC	Federal Trade Commission	1
GDPR	General Data Protection Regulation	130
ML	machine learning	10
NER	named entity recognition	14
NLP	natural language processing	10
PII	personally identifiable information	90
IoT	Internet of Things	210

ABSTRACT

There are multiple issues in the privacy notices and choices of ubiquitous mobile apps and online services, which collect data in all corners of users' daily lives and increase the risks to users' privacy. While most, if not all, applications utilize privacy policies to inform users of their data practices, it is difficult and time-consuming for users to comprehend the policies due to their great length and use of legal language. Furthermore, the actual implementation of data practices and opt-out choices are not always consistent with the stated privacy policies and users' privacy preferences.

This dissertation systematically and rigorously assesses privacy risks in the user interface, purposes of data collection and use, and opt-out choices of mobile apps and web services. First, it addresses the issues in the user interface of privacy policies with `PI-Extract`, an automated framework that extracts and presents data practices stated in privacy policies to help users read and understand them easily and fast. Second, the dissertation analyzes the consistency between privacy policies and actual data practices. Specifically, it develops `PurPliance`, an automated system that detects the inconsistencies between the data-usage purposes stated in the privacy policy and those of the actual execution behavior of an Android app. Furthermore, it creates `ExtPrivA` to check the discrepancies between browser extensions' data collection and their privacy disclosures. Finally, the dissertation examines the (in)consistencies of opt-out choices of online services by developing two automated frameworks: `ConsentChk` and `OptOutCheck`. The former detects cookie consent violations by checking the (in)consistencies between the cookie consent/rejection and the actual usage of each cookie on a website. The latter analyzes (in)consistencies between trackers' data practices and the opt-out choice statements in their privacy policies. These consistency-analysis systems uncovered a large number of privacy violations of apps and services whose actual behavior was not consistent with their disclosed policies. The detected inconsistencies are potential breaches of consumer-protection laws, such as the General Data Protection Regulation (GDPR), California Consumer Privacy Act (CCPA) and Federal Trade Commission Act (FTC Act). These automatic analysis techniques offer a practical and scalable privacy assessment that benefits all stakeholders of the Web and mobile ecosystems, including users, developers and regulators.

CHAPTER I

Introduction

1.1 Motivation

While service providers and advertisers have been increasingly tracking and collecting the personal information of users via ubiquitous mobile apps and web services for various purposes, the intrusive data practices raise considerable privacy risks to users. Advertisers use the collected data to build accurate users' profiles to provide targeted advertisements to increase their revenue, while businesses utilize users' information for marketing analytics and product development [168]. On the other hand, consumers have become more concerned about their privacy and are demanding greater transparency from apps/services about how their information is collected, used and shared [56].

The Notice and Choice principles, which are part of the Fair Information Practice Principles (FIPP) set forth by the Federal Trade Commission (FTC) [60], have been widely adopted by online services and mobile apps in the form of privacy policies and opt-out choices. The privacy policies serve as a specification of the privacy practices that a website or an app must follow as well as a contract between apps/services and users [58, 62]. The Notice and Choice framework is the basis for protecting users from unfair data practices where users are fully notified about the data practices of service providers and can make informed decisions on whether to choose or opt out of the apps/services.

The complexity of privacy policies and the discrepancies between the policies and actual data practices pose significant privacy risks to users. First, privacy policies are often lengthy and written in vague/legal language, making it hard for users to understand and thus forcing them to blindly accept the policy terms [212]. Second, the purposes of data usage are an important factor for users to decide whether or not to agree on the collection and sharing of their data; this, despite its importance, has not yet been studied well. For example, users will agree to the collection of their sensitive data (such as credit card numbers) so long as the data-collection purposes are appropriate for and acceptable to them (such as for making payments). Third, the actual data practices of online services after user opt-out have not been thoroughly investigated, either. For example, a website may still track a user via advertisers' cookies even after the user already rejected the website's cookies for advertising purposes. The inconsistencies between apps'/services' data policy statements and practices not only lose users' trust in them and pose privacy risks to users but also can be deemed by the regulators as a deceptive data practice [112].

Fully automated systems will be useful for all stakeholders of mobile and web ecosystems. First, businesses, especially small companies, which do not have enough financial resources/incentives for thoroughly testing the privacy features of their products/services, can leverage the automatic assessment of privacy risks and auditing third-party providers integrated with their services. Second, regulators can readily audit and detect unfair and deceptive data practices of a large number of companies very fast and economically. Finally, an automated system can inform users to avoid privacy risks due to misleading statements in the privacy policies of online services.

1.2 Research Challenges & Thesis Statement

Given the significant gap between inherently vague privacy policies and highly sophisticated software systems, it is very challenging to ensure the consistency of mobile/web apps with the statements in their privacy policies while keeping them up to date throughout the life of the services. Privacy policies are typically lengthy and complicated as they are required to disclose complex data collection, usage and sharing while app/service providers prefer general yet vague terms to allow future product/service development and expansion. On the other hand, apps do not specify the data types they collect from users and the purposes of data collection, making it challenging to determine the actual data practices of an app or service.

Thesis Statement:

In this dissertation, we develop End-to-End (E2E) systems to automatically assess the users' privacy risks of mobile and web apps via the analysis of privacy policies, app execution and user interfaces.

1.3 Dissertation Contributions

The main objective of this dissertation is to develop a **framework that assesses privacy risks of user-data collection/use and opt-out choices of mobile apps and web services**. It aims to holistically assess the privacy notices and opt-out choices of apps and services: from the user interface to the actual collection of user data. The dissertation is composed of three parts as follows.

1. Analysis of the presentation of privacy policies for helping users beware of the practices performed by websites on their data.

2. Analysis of the flow-to-policy consistency between the data-collection statements in privacy policies and the data flows of mobile/web apps.

3. Analysis of the consistency between the consent/opt-out settings and their enforcement of websites and online trackers.

Fig. 1.1 depicts the proposed systems with respect to the relationships among privacy policies, end users and applications. Table 1.1 summarizes the issues each proposed system addresses. Next, we outline each proposed system.

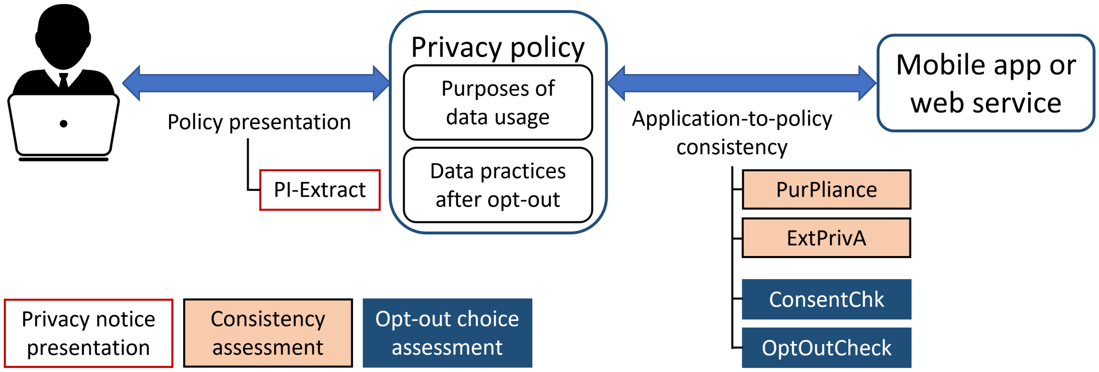

Privacy policy
Purposes of data usage
Data practices after opt-out
Mobile app or web service
Policy presentation
PI-Extract
Application-to-policy consistency
PurPliance
ExtPrivA
ConsentChk
OptOutCheck
Privacy notice presentation
Consistency assessment
Opt-out choice assessment

Figure 1.1: The proposed systems in the relationships between privacy policies, end users and applications.

System	Principle (Implementation)	Assessment	Target Ecosystem
`PI-Extract` [45]	Notice (Privacy policies)	Policy presentation	Websites
`PurPliance` [46]	Notice (Privacy policies)	Flow-to-policy consistency	Mobile apps
`ExtPrivA`	Notice (Privacy policies)	Flow-to-policy consistency	Browser extensions
`ConsentChk`	Choice (Consent settings)	Cookie consent enforcement	Websites
`OptOutCheck`	Choice (Opt-out settings)	Opt-out choice enforcement	Online trackers

Table 1.1: Summary of the proposed systems.

1.3.1 Presentation of Privacy Policies

Apps and websites need to inform users of their privacy policies. Without being aware of the data practices performed by a service, users may face high privacy risks due to their uninformed decisions when sharing their personal data with the service. Therefore, in the first part of the dissertation, we assess and improve the presentation of privacy policies to help users understand data practices of services better and faster.

Privacy policies are the documents required by law and regulations that notify users of the collection, use, and sharing of their personal information on services or applications. While the extraction of personal data objects and their usage thereon is a fundamental step in their automated analysis, it remains challenging due to the complex policy statements written in legal (vague) language. Prior work is limited by its small/generated datasets and manually created rules. We formulate the extraction of fine-grained personal data phrases and the corresponding data collection or sharing practices as a sequence-labeling problem that can be solved by an entity-recognition model. We create a large dataset with 4.1k sentences (97k tokens) and 2.6k annotated fine-grained data practices from 30 real-world privacy policies to train and evaluate neural networks. We present a fully automated system, called `PI-Extract`, which accurately extracts privacy practices by a neural model and outperforms, by a large margin, strong rule-based baselines. We conduct a user study on the effects of data practice annotation which highlights and describes the data practices extracted by `PI-Extract` to help users better understand privacy-policy documents. Our experimental evaluation results show that the annotation significantly improves the users' reading comprehension of policy texts, as indicated by a 26.6% increase in the average total reading score.

1.3.2 Flow-to-Policy Consistency Analysis

While the first part helps users understand the data practices of online services better, the data usage in an application's behavior is not always consistent with the purposes stated in its privacy policy. Therefore, we propose ways to detect the inconsistencies in mobile and web apps.

While privacy laws and regulations require apps and services to disclose the purposes of their data collection to the users (i.e., *why do they collect my data?*), the data usage in an app's actual behavior does not always comply with the purposes stated in its privacy policy. Automated techniques have been proposed to analyze apps' privacy policies and their execution behavior, but they often overlooked the *purposes* of the apps' data collection, use and sharing. To mitigate this oversight, we propose `PurPliance`, an automated system that detects the inconsistencies between the data-usage purposes stated in a natural language privacy policy and those of the actual execution behavior of an Android app. `PurPliance` analyzes the predicate-argument structure of policy sentences and classifies the extracted purpose clauses into a taxonomy of data purposes. Purposes of actual data usage are inferred from network data traffic. We propose a formal model to represent and verify the data-usage purposes in the extracted privacy statements and data flows to detect *policy contradictions* in a privacy policy and *flow-to-policy inconsistencies* between network data flows and privacy statements. Our evaluation results of end-to-end contradiction detection have shown PurPliance to improve detection precision from 19% to 95% and recall from 10% to 50% compared to a state-of-the-art method. Our analysis of 23.1k Android apps has also shown PurPliance to detect contradictions in 18.14% of privacy policies and flow-to-policy inconsistencies in 69.66% of apps, indicating the prevalence of inconsistencies of data practices in mobile apps.

Besides the mobile environments, the Web is another major platform with billions of users while web browsers extend their functionality through the use of third-party extensions. However, running with privileged permissions, these browser extensions pose privacy risks to users as they can collect and share users' sensitive information with third parties. Therefore, we propose ExtPrivA to assess the privacy risks of web browser extensions.

All major web browsers support extensions to provide additional functionalities and enhance users' browsing experience while the extensions can access and collect users' data during their web browsing. Although web extensions inform users of their data practices via multiple forms of notices, prior work has overlooked the gap between the actual data practices and the published privacy notices of browser extensions. To fill this gap, we propose ExtPrivA that automatically detects the inconsistencies between browser extensions' data collection and their privacy disclosures. From the privacy policies and Dashboard disclosures, ExtPrivA extracts privacy statements to have a clear interpretation of the privacy practices of an extension. The system emulates user interactions to trigger the extension's functionalities and analyzes the initiators of network requests to accurately extract the users' data transferred by the extension from the browser to external servers. Our end-to-end evaluation has shown ExtPrivA to detect inconsistencies between the privacy disclosures and data-collection behavior with an 85% precision. In a large-scale study of

47.2k extensions on the Chrome Web Store, we have found 820 extensions with 1,290 flows that are inconsistent with their privacy statements. Even worse, we have found 525 pairs of contradictory privacy statements in the Dashboard disclosures and privacy policies of 360 extensions. These discrepancies between the privacy disclosures and the actual data-collection behavior of an extension are deemed as serious violations of the Store's policies. Our findings highlight the critical issues in the privacy disclosures of browser extensions that potentially mislead, and pose high privacy risks to, end-users.

1.3.3 Consistency Analysis of Opt-out Choices

After addressing the issues associated with the implementation of the Notice principle in the first and second parts of the dissertation, the third part presents solutions to meet the challenges under the Choice principle. In addition to the privacy policies that inform users of the data practices, online services provide users with actionable choices to opt out of their tracking and data collection. However, the choices given to users to either opt in or out of the data practices are largely left unchecked while there is no guarantee that the consent and opt-out settings follow the stated policies by the services. Therefore, we propose the following systems/solutions to assess the privacy risks of the cookie consent preferences of websites and the opt-out choices of online trackers.

Online services increasingly provide users with *cookie consent settings* to accept/reject the cookies placed on their web browsers. Unlike other GDPR-specific requirements, using cookies without user consent would violate consumer protection laws anywhere in the world. However, little has been done to understand the violations of users' cookie consent/rejection from a global standpoint. To remedy this important oversight, we propose an end-to-end automated system, called `ConsentChk`, that analyzes and detects inconsistencies between a website's cookie usage and users' cookie consent preferences. `ConsentChk` detects and analyzes the cookie usage and consent preferences even on the websites that do not display cookie banners for new visitors. We design a machine-learning-based classifier to detect/locate cookie preference buttons to activate cookie setting menus with an 85.96% top-3 score. We build a formal model to systematically categorize the types of cookie consent violations. Our in-depth evaluation demonstrates a high precision of $> 91\%$ of ConsentChk's end-to-end violation detection performance. In a large-scale study on 101,703 top global websites, we find 82.20% and 81.86% of the websites with detected cookie settings to use user-rejected cookies when accessing them from inside and outside of the EU, respectively. Our measurement of rejected cookie usage violations covers cookie management platforms with a 3X more market share than state-of-the-art studies. Our findings indicate the prevalence of misleading, or even deceptive, cookie consent management, raising their awareness among all stakeholders — end-users, website owners and developers as well as regulators.

While ConsentChk verifies the cookie consent settings on publisher websites, its techniques do not apply to the opt-out mechanisms used by *online trackers* that run on publisher websites as a third party. Therefore, we propose the following system to analyze the opt-out choices of the online trackers.

Online trackers, such as advertising and analytics service companies, have provided users with choices to opt out of their tracking and data collection to mitigate the users' concerns of increasing privacy risks. While opt-out choices of online services for the cookies placed on their own websites have been examined before, the choices provided by trackers for their third-party tracking services on publisher websites have been largely overlooked. There is no guarantee that a tracker's opt-out option would faithfully follow the statements in its privacy policy. To address this concern, we develop an automated framework, called OptOutCheck, that analyzes (in)consistencies between trackers' data practices and the opt-out choice statements in their privacy policies. We create sentence-level classifiers, which achieve ≥84.6% precision on previously-unseen statements, to extract the opt-out policies that state neither tracking nor data collection for opted-out users from trackers' privacy-policy documents. OptOutCheck analyzes both tracker and publisher websites to detect opt-out buttons, perform the opt-out, and extract the data flows to the tracker servers after the user opts out. Finally, we formalize the opt-out policies and data flows to derive logical conditions to detect the inconsistencies. In a large-scale study of 2.9k popular trackers, OptOutCheck detected opt-out choices on 165 trackers and found 11 trackers who exhibited data practices inconsistent with their stated opt-out policies. Since inconsistencies are violations of the trackers' privacy policies and demonstrate data collection without user consent, they are likely to cause a loss of users' trust in the online trackers and trigger the necessity of an automatic auditing process.

1.4 Road Map

The remainder of this dissertation is structured as follows. Chapter II presents the common technical background of the proposed systems. Chapter III introduces PI-Extract to assess and improve the presentation of privacy policies. Chapter IV discusses PurPliance that checks the (in)consistencies in data usage purposes of mobile apps. Chapter V presents ExtPrivA that detects the discrepancies between the actual behavior and the disclosed privacy practices of web browser extensions. Chapters VI and VII introduce ConsentChk and OptOutCheck to address the issues in the opt-out choices of websites and online trackers, respectively. Lastly, Chapter VIII concludes the dissertation and discusses future directions.

CHAPTER II

Background

2.1 Legal Frameworks

The FTC set forth five general FIPP in its report to Congress in 1998: notice/awareness, choice/consent, access/participation, integrity/security, and enforcement/redress [62]. Rooted in the tenets of the Privacy Act of 1974 [173], these principles were proposed to assure fairness and privacy protection of data practices, i.e., the collection and use of personal information. Since then, the Notice and Choice principles have been widely adopted by companies and underlie the mechanisms for companies to disclose their data practices [212].

Privacy policies have been the *de facto* form of the disclosure of privacy practices of companies. The data practices comprise data types, actions performed on the data, and data usage purposes. Although privacy policies are long and hard to understand, they are still valuable for their important accountability function as they inform consumer advocates, regulators, the media, and other interested parties about the companies' data practices [60].

2.2 Privacy Policy Analysis

Researchers have widely leveraged natural language processing (NLP) and machine learning (ML) for analyzing natural-language privacy policies. Privee [314] and Polisis [141] analyze privacy policies at the document- and paragraph-level to answer users' questions. However, both are limited by their coarse-grained analyses while our sentence- and phrase-level analyses provide more detailed and comprehensive results. PolicyLint [20] uses dependency parsing to extract privacy statements from policy documents but does not analyze purposes of data collection.

Bhatia *et al.* [35] extract common patterns of purposive statements from privacy policies and use semantic frames to analyze the incompleteness of privacy goals, which include the purposes of data practices [36]. Shvartzshnaider *et al.* [276] analyze information flows in a limited set of privacy policies following the contextual integrity framework [231]. However, these semi-automated methods require laborious manual efforts of experts or crowd workers.

2.3 Cookie Consent Management

Cookie consent management can be classified into 3 types: *local*, *decentralized*, and *centralized* [79]. In the local type, websites collect users' cookie preferences and block/unblock cookies *locally* based on the consent. Cookie consent libraries create and maintain special *consent cookies* to record users' consent preferences on websites. For example, OneTrust and Cookiebot store the consent preferences in cookies named *OptanonConsent* [237] and *CookieConsent* [77], respectively. Furthermore, the libraries provide websites with integrated scripts to (un)block other third-party cookies on the websites [68, 69, 74].

The decentralized type uses opt-out mechanisms of third-party advertisers to block the advertisers' third-party cookies. For example, the tools provided by advertising organizations can set a special cookie called *opt-out cookie* [13, 149]. Prior work [72, 183, 191] studied the compliance of this type of opt-out choices.

The centralized type uses services provided by a third party that records users' cookie preferences and notifies registered advertisers whether to collect users' data or not. The Transparent Consent Framework (TCF) falls into this category. It is orchestrated by Interactive Advertising Bureau (IAB) Europe that manages the registered consent management platforms (CMPs) and advertisers called *vendors*. The compliance and characteristics of TCF implementation have been studied recently [145, 208]. Unless stated otherwise, we will henceforth use the version 2 when mentioning TCF because websites must have this version since August 2020 [95].

CHAPTER III

PI-Extract

3.1 Introduction

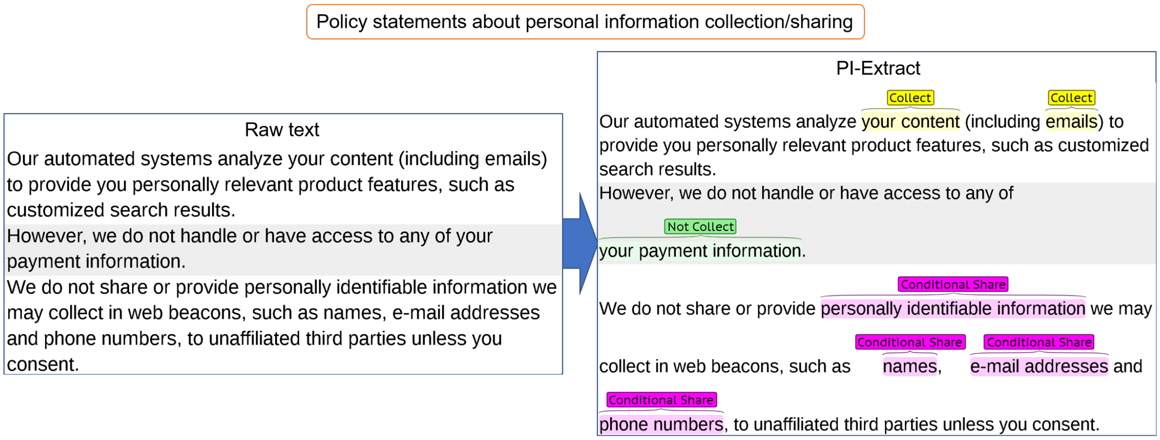
Under the FTC framework of *Notice and Choice* [62], privacy policies are a binding contract that services, offered through websites or mobile apps, must adhere to. While this framework is accepted in the US and EU [233], it is up to users to read, and give consent to, the privacy policies. Thus, law and regulations, such as GDPR require services, to provide users with transparent and easy-to-read privacy policies [243].

It is desirable to help users understand the terms used in the privacy notices to raise their awareness of privacy. Despite their growing concerns about data collection and sharing [213, 256], users rarely read them due mainly to their legal sophistication and difficulty to understand [50, 268, 269]. Hard-to-understand privacy policies can also lead end-users to blind consent or click-through agreements, risking their privacy since clicking an agreement icon on a website is considered as giving consent to the service provider to lawfully collect and process both general and sensitive personal data [97]. Users are more likely to take necessary steps to protect their privacy if they (especially non-technical users) can understand, and are made aware of privacy at stake [204].

The main thesis of this section is that *automatic extraction and presentation of data practices help users understand privacy policies better*. The data practices comprise the data objects and privacy actions (collection or sharing) performed thereon. We focus on the users who want to understand the privacy practices, and help them comprehend the privacy notices faster and better. Motivating uninterested readers of privacy documents is orthogonal to the theme of this section.

Prior work on extracting information from privacy policies has several fundamental limitations. First, existing techniques like PolicyLint [20] use information extraction methods that have high precision but low recall to minimize false positives for their detection of policy contradictions. In contrast, our goal is to achieve both high precision and recall rates. Presenting the data practices to help users improve their reading comprehension requires not only high precision but also high recall because a high false positive or false negative rate (i.e., low precision or recall) will lower the users' confidence and even make them abandon the visualization tool altogether. Furthermore, prior work relied on limited datasets which were either generated from a small number of template sentence patterns [20] or created by non-expert *crowdsourced* workers [279, 302]. A template-generated dataset fails to capture complex and flexible grammatical structures and vocabulary of statements in privacy documents. Crowdworkers are not trained to interpret legal documents so their interpretation may deviate significantly from experts' [257]. Finally, prior information extraction methods are commonly based on a fixed set of manually crafted rules [20, 34, 71] or rely on manual analyses [33, 34, 98], which do not scale to the large number of privacy policies for online services, smartphones and IoT products.

To address the above limitations of prior work, we design and implement a fully automated system, called `PI-Extract`, which accurately extracts data objects and distinct data actions performed thereon (collection/not-collection or sharing/not-sharing). We formulate the information extraction problem as a sequence-labeling problem which can be solved by a named entity recognition (NER) model. We create a large dataset of data practices in real-world privacy policies to train a state-of-the-art neural NER model [202] with contextualized word embeddings [83].



Our automated systems analyze your content (including emails) to provide you personally relevant product features, such as customized search results.

However, we do not handle or have access to any of your payment information.

We do not share or provide personally identifiable information we may collect in web beacons, such as names, e-mail addresses and phone numbers, to unaffiliated third parties unless you consent.

Figure 3.1: PI-Extract extracts and presents collection and sharing practices of personal information in privacy-policy statements.

PI-Extract presents the extracted data objects and actions as data practice annotation (DPA) on privacy policy text to reduce users' burdens in reading and comprehending the policy documents. DPA highlights phrases to help users easily identify personal data types in the privacy-policy excerpt and provides a short description of data action to help users determine whether the data types are collected/shared or not. Fig. 3.1 shows an example of DPA created by PI-Extract. We have conducted an experiment to evaluate the effect of DPA on user comprehension, the impact of wrong predictions, and the effect of annotations on the reading effort. The results show a significant improvement in reading comprehension of DPA over the plain text version. Effects of wrong predictions on comprehension and effects of annotations on answering time are also evaluated.

This section makes the following contributions:

- Construction of a large fine-grained dataset of phrase-level regulated personal information types and the data actions performed on them. The resulting corpus (available on GitHub [87]) comprises 30 real-world privacy policies (4.1k sentences and 97k tokens) with 2.6k annotated data practices and achieves a 98.74% F1 inter-annotator agreement. To the best of our knowledge, this is the largest dataset of fine-grained data practices in real-world privacy policies known to date (Section 3.5).

- A fully automated system, called PI-Extract, which extracts data objects and privacy practices performed thereon. PI-Extract leverages a neural NER model, with contextualized word embeddings, trained on our large dataset and achieves an F1 score higher than a rule-based approach based on the method of PolicyLint [20] (Section 3.6).

- A user study of a presentation method called *data practice annotation* (DPA), which presents extracted data types and privacy actions as text highlights and annotations to help users understand privacy policies better. An experiment on 150 users showed that the DPA significantly improves the users' comprehension of the privacy texts as indicated by a significant improvement (26.6%) of the average total reading score over the plain text version. The majority of participants found our DPA very or extremely helpful in their qualitative feedback. To the best of our knowledge, this is the first application and study of the effects of text highlighting and annotation in reading comprehension of privacy-policy texts (Section 3.7).

3.2 What is PI-Extract for?

Personal data types and data practice extraction are critical steps in privacy policy analysis. Prior work on privacy policy analysis [20, 279, 302] includes these extractions in their pipelines. PI-Extract's extraction improvements will facilitate the development and performance enhancement of privacy policy analysis pipelines.

Presentation of extracted personal data objects and data practices as text annotations in privacy policies can be used in two ways. First, it can be **used after an information retrieval (IR) system** to highlight the data practices in short paragraphs which were previously extracted by the IR system. Highlighting search terms in the snippets of search result pages has been widely used by search engines to help users find the relevant results faster [144, 311]. Prior IR-based approaches, such as Polisis [141], present to the users relevant paragraphs from a privacy policy document, but large chunks of raw text are still daunting for users to read through and comprehend. Our visualization helps users search for information of interest in the text snippets and read the contextual statements surrounding the phrases of interest.

Second, the presentation can be **used with full privacy policies to facilitate the analysis of non-standardized policies** for researchers, organizations and individuals (such as journalists). For example, `PI-Extract` can be leveraged to assist scientists in recent systematic studies of privacy policies of menstrual apps [274] and mobile money services [43].

3.3 Related Work

Data Type Extraction. There has been prior work on extracting data types from privacy policies. Costante *et al.* [71] use pattern matching on tokens and named entities to extract personal information types collected by a website. Bhatia *et al.* [33] extract a lexicon of personal information types by identifying noun phrase chunking patterns from 15 human-annotated privacy policies. Bhatia *et al.* [34] and Evans *et al.* [98] use hyponymy patterns to extract personal data types from privacy policies. All of these methods rely on manually-specified rules and lack patterns for extracting data-sharing practices.

PolicyLint [20] extracts the data practices (collection/sharing) on data types to detect contradictions in privacy policies. Its NER model is trained on a small number of samples: only 600 sentences mainly generated from 9 subsumptive patterns, so its dataset and extraction capability are limited in terms of grammar and vocabulary. In contrast, our models are trained on a much larger and more comprehensive dataset — 4.1k sentences (97k tokens) from 30 real-world privacy documents — and thus covers a wider range of grammar and vocabulary. Furthermore, PolicyLint focused on extraction precision (similar to a linter tool), and hence did not evaluate the recall while `PI-Extract` balances between precision and recall to provide users with both correct and complete recognized data types in a document. Therefore, it is not designed to use for helping users understand the text because a low recall rate will provide users with incomplete information and will even reduce the user's confidence in the extraction tool.

GUILeak [302] extracts the data types collected by the services either via user inputs or automatic tools to detect violations in the data collection practices of Android apps. Salvin *et al.* [279] extract from privacy policies the platform information types collected by Android apps and map them to the corresponding Android API functions to detect violations in the implementation of the apps. They only consider data-collection practices, i.e., they do not distinguish data collection from 1st and 3rd parties. PoliCheck [21], built upon PolicyLint [20], can distinguish the receiving entities (1st or 3rd party) when detecting dataflow-to-policy inconsistencies, but suffers from the same limitations of PolicyLint.

Privacy Policy Datasets. Recently, researchers have devised labeled datasets to facilitate the development of machine learning algorithms for automated analysis of privacy policies. OPP-115 [303] is a corpus of annotated paragraphs of 115 website privacy policies. The annotation scheme consists of ten data practice categories, such as 1st-party collection or use, and each data practice has a list of attributes such as data type and purpose. Opt-out Choice dataset [229] includes opt-out choices, such as opt-outs from behavioral advertising. Polisis Twitter QA [141] is a collection of 120 tweets containing questions about privacy policies, alongside the annotated answers obtained from the corresponding privacy policies. APP-350 dataset [315] provides annotated sentences and paragraphs of 350 Android privacy policies, while `PI-Extract` has finer-grained annotations at the phrase level. Prior datasets are coarser-grained and less diverse than ours, or created by non-expert annotators. They comprise long text spans [229, 303, 315], large text segments [141], rigid examples generated from a small set of only 16 patterns [20], or annotations created by non-expert *crowdsourced* workers [279, 302].

User Interfaces for Privacy Policies. Numerous approaches have been proposed to make privacy policies more accessible to users. Polisis [141] retrieves and presents policy paragraphs relevant to a user's question in a chatbot. Since Polisis is based on coarse-grained annotations in OPP-115 dataset [303] at the paragraph level, it can only classify and rank segments of privacy documents. Therefore, `PI-Extract` can extract data objects at the word and phrase levels while Polisis does not. Moreover, `PI-Extract` can be integrated with Polisis to enhance the user's understanding of privacy documents further. For example, Polisis can be used to extract the paragraphs relevant to the user's query, and then use `PI-Extract` to highlight the important phrases about data objects and practices in the paragraphs.

Many researchers worked on various aspects of evaluation and presentation of privacy policies. Disconnect [84] introduces a set of icons to represent privacy risks of a privacy policy. Privacy Nutrition labels [178] present lengthy privacy policies in a nutrition-label-like form. Kay *et al.* [177] show that the visual elements, such as factoids, vignettes, iconic symbols and typography, increase the attention and retention of the users when reading the software agreements. Other research [182, 292] uses a comic-based interface to draw users' attention to privacy notices and terms of service agreements. [214] evaluates three formats for privacy policies and found that the standardized presentations are not effective in helping users understand companies' privacy practices.

3.4 Background and Problem Formulation

3.4.1 Neural Named Entity Recognition

Named entity recognition extracts such entities as names of people and places, is commonly formulated as a sequence labeling problem, and then solved by Recurrent Neural Networks (RNN) [308]. RNN encodes the text sequentially and can handle long-term dependencies in text while bi-directional long short-term memory (BLSTM) is one of the most widely-used neural architectures for text classification and sequence labeling [202, 205]. In entity recognition, since the label of each token depends on the probability of its neighbors, a conditional random field (CRF) layer is commonly used after the RNN layer to improve the prediction performance [202].

Raw text tokens are converted to real-value vectors before inputting to neural networks by using word embeddings, which comprise the mappings from each word to a single vector. Word embeddings are trained on large datasets of billions of tokens to maximize the coverage of linguistic phenomena. Early word embeddings, such as word2vec [219] and GloVe [245], map words to vectors without context. Recent advances in NLP and computation introduced contextualized word embeddings, such as ELMo [247] and BERT [83], in which the surrounding words are taken into account when mapping a word to a vector, hence improving the prediction performance.

3.4.2 Problem Formulation

We formulate the extraction of personal data objects and actions thereon as a sequence labeling problem: given a sentence of tokens $s = t_1, \ldots, t_n$, find the label l_i for each token t_i, where $l_i \in \{Collect, Not\_Collect, Share, Not\_Share\}$. A personal data object is a text span (a phrase or a word) that expresses a type of user data. Each of such text spans is assigned a data-action label which indicates the action thereon. The labels for text spans are actions on data objects, "collection by 1st party" and "sharing with a 3rd party", and whether the action is performed or not. The 1st party is the company/organization that owns the service, and 3rd parties are companies/organizations other than the 1st party. Determining the labels is based on the data flows: *Collect* and *Share* correspond to the data flows to the 1st and 3rd party, respectively. Table 3.1 shows their definitions. For example, phrase "your personal information" is marked *Not_Share* in "we may not share your personal information with anyone". We use the classic flat entity structure [104] for each label so that text spans with the same label (i.e., same data action) are contiguous and not overlapping. For example,

Label	Action performed on the data object
Collect	Collected or used by the first party.
Share	Collected by a third party.
Not_Collect	Not collected and not used by the first party.
Not_Share	Not collected by a third party.

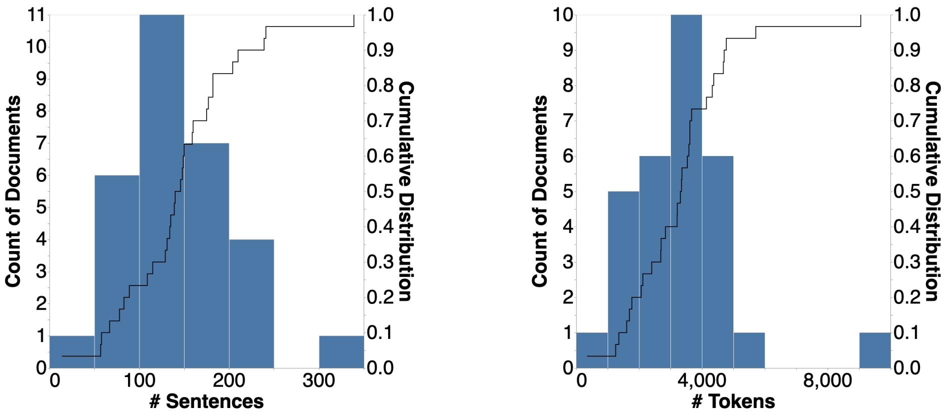
Table 3.1: Types of data actions to extract from text.

the whole "delivery and address information" is labeled instead of each overlapping phrase "delivery information" and "address information".

The labels are used independently without assuming their mutual exclusion or implication. For example, *Share* does not always imply *Collect*, when the service allows a third party to collect and analyze the user's *personal data* instead of doing it by itself, such as in "we do not collect any personal data, but we use Google AdMob that can collect and send it to Google." Furthermore, a pair of negated labels can be used for the same phrase when conditional sharing is performed. "Your personal information" is labeled with both *Share* and *Not_Share* in "we do not share your personal information with third parties without your consent." It is worth noting that handling contradictory policy statements (e.g., a data type is stated to be both collected and not collected) is outside the scope of `PI-Extract`.

3.5 Dataset Construction

While data objects can be extracted using NER models, creating a dataset is challenging because the determination of start and end of data type spans is vague due to the addition of vague words in the sentences. For example, given a sentence "we collect certain information about your location," we can select either *certain information about your location*, *information about your location*, *your location*, or *location*. A state-of-the-art approach [20] opted to use a set of manually-derived patterns to reduce their efforts. This section describes how we created and controlled the quality of a dataset for training and evaluating the performance of NER for extracting data practices.



(a) Number of sentences.

(b) Number of tokens.

Figure 3.2: Cumulative distributions of document lengths in terms of number of sentences and tokens.

3.5.1 Data Practice Dataset Contruction

3.5.1.1 Document Selection

We selected and annotated 30 documents from the 115 online privacy policies in the OPP-115 dataset [303] which cover a variety of data practices and styles of online privacy policies. Although OPP-115 cannot be used directly for our purpose of training NER, it contains coarse-grained paragraph classifications which were used as the starting point of our annotation process. We chose the policies of the top websites in the US [10] as large service providers tend to have long and sophisticated policies and have higher coverage of the linguistic phenomena in the corpus [105]. The websites comprises various business domains such as social network, search engine, banking and e-commerce. Total number of sentences and tokens are 4.3k and 99.1k tokens, respectively. Each policy has 144 sentences and 3303 tokens on average. The cumulative distributions of the number of sentences and tokens are shown in Fig. 3.2.

3.5.1.2 Annotation Scheme and Process

Two annotators labeled the data objects in each sentence with the 4 labels described in Section 3.4 and created annotation guidelines for annotators to create consistent labels.

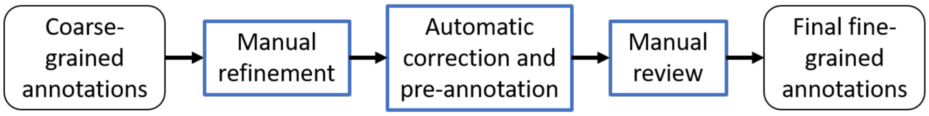
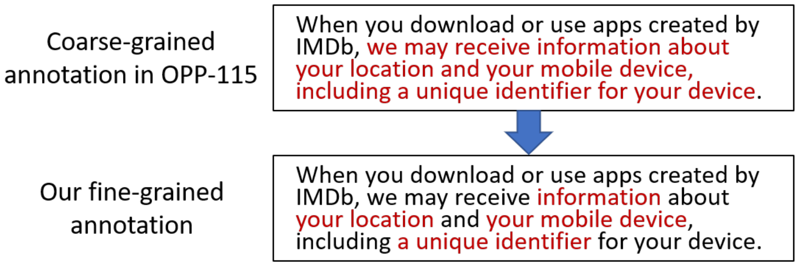

Coarse-grained annotations
Manual refinement
Automatic correction and pre-annotation
Manual review
Final fine-grained annotations

Figure 3.3: Semi-automated annotation process.

The labelers were two of the authors: an advanced PhD student and an industry privacy researcher, and both had more than two years of experience in privacy and security research. First, we created a mini-reference from a subset of 12 documents (40% of the corpus) to develop and evaluate an annotation guideline and process. The main principle is to extract noun phrases from the privacy sentences which express a personal data type that is collected, used or shared by the service provider. The annotation guideline explains corner cases such as how to extract data objects from a complex list. We evolved the guidelines to reflect the new phenomena encountered in the documents while inter-annotator agreement (IAA) was continuously measured to give feedback to annotators. Every time the guidelines were modified, we reflected the changes onto the existing annotations. The guideline document had 4 major updates and its final version (available on GitHub [87]) has 7 pages, 6 high-level principles and 7 rules, each of the rules with multiple examples. After the guidelines and methodology were stabilized and fixed, each annotator followed them to perform the annotation independently on other 18 documents. Finally, they resolved the remaining disagreements by follow-up discussions.

Annotation Revision. To increase the annotation speed and quality (i.e., consistency), we used a semi-automated process that has 4 steps: preprocessing, revision of existing coarse-grained annotations in OPP-15, automated correction/pre-annotation, and final review. These steps were done in sequential order for each document (as shown in Fig. 3.3). We first removed the sentences which do not contain an actual description of data collection or sharing from the dataset to reduce noisy samples. In particular, we removed sentences which are titles or not a complete sentence. A sentence is considered as a title when it matches the corresponding title-cased statement more than 95% or has less than 4 tokens.


Coarse-grained annotation in OPP-115
Our fine-grained annotation

When you download or use apps created by IMDb, we may receive information about your location and your mobile device, including a unique identifier for your device.

When you download or use apps created by IMDb, we may receive information about your location and your mobile device, including a unique identifier for your device.

Figure 3.4: Example of how long labeled texts in the OPP-115 dataset are refined into shorter phrases. The red color denotes personal information.

The similarity is calculated by using the Levenshtein distance with *fuzzywuzzy* [271] library. Furthermore, since the OPP-115 dataset was in the HTML format, we extracted well-formed plain-text sentences from the HTML, such as merging lists into well-formed sentences and aligning annotations between plain text and HTML code.

The annotators created new fine-grained phrase-level annotations based on the existing coarse-grained labeled text spans in the OPP-115 dataset which was created by law experts. The original OPP-115 dataset has a low overall inter-annotator agreement (IAA) of 29.19% F1 on the 4 labels since it was intended to have classified paragraphs rather than labeled text spans. Therefore, we resolved the conflicting annotations, refined the labels which cover long text, and identified additional data objects that the original annotators missed. While having a low IAA, the existing annotations, created by skilled workers, are useful to speed up the process, such as to determine whether or not a sentence contains any data collection or sharing practice.

Our revision of the OPP-115 corpus was done using WebAnno [91] web-based text annotation tool. An example revision is provided in Fig. 3.4 where a long-labeled text is refined into three shorter annotated phrases. Other sentences which do not end with a period or do not start with an alphabet character are also removed since they are typically sentence fragments resulting from preprocessing.

Automated Correction and Pre-annotation. We developed a semi-automatic process that includes automated tools for correction and pre-annotation, which are commonly used to increase the annotation speed and improve the quality of corpora [105, 290, 317]. The limitations and bias of the automated methods were also written in the annotation guideline for annotators to be aware of them and avoid too much reliance on the automatic annotations. These tools were developed on 12 policies and fixed thereafter. They were then used to double-annotate the remaining 18 documents.

Automatized Correction. The automatized correction has 2 steps to create consistently labeled text spans: (i) remove relative and prepositional clauses, and (ii) align annotations with noun chunks. Although including relative clauses can narrow the scope of a data type, they frequently contain nested noun phrases, so how to determine the end of these clauses is unclear. For example, "your personal information that you entered in the forums on our website" would be revised to "your personal information". If we include the relative clause, it is hard to determine whether the annotated text span should end at *the forums* or *our website*. Therefore, removing the relative and prepositional clauses reduces the inconsistencies of the labeled spans. The labeled text spans are then aligned to noun chunks in each sentence. The noun phrase alignment removes inconsistencies in the text spans because it is challenging and tedious for annotators to remember to include all the adjective and pronoun prefixes such as "other" and "additional". The alignment also automatically determines whether the conjunctions (*and* and *or*) in a list of data objects would be included in the annotation or not. We used the Spacy library [100] to recognize and chunk non-nested noun phrases.

Automated Pre-annotation. We leverage automatic extraction in PolicyLint [20] to reduce the effort of finding new data objects. Although PolicyLint has a low recall rate, its high precision is useful to reduce the correction effort of the annotators. In particular, we use the domain-adapted named entity recognition (NER) and Data-Entity-Dependency (DED) trees trained in the same dataset in PolicyLint to recognize data objects and label the action for each text span. Our modifications to PolicyLint are detailed in Section 3.6.

Final Manual Review. After the automatized correction and pre-annotation, the annotators manually reviewed the automatically created annotations. Finally, they hold a discussion to reconcile the disagreements between their labeled policies.

3.5.1.3 Privacy Policy Corpus

The resulting corpus has 4.1k sentences and 97k tokens. The annotators labeled 2,659 data objects in all documents. The exact-match F1 score is used as the IAA metric. This score has been widely used to measure the prediction performance of the NER task [296]. Two labeled spans match only when they have the same boundaries and the same label. One of the annotators is set as the reference and IAA is then computed as the exact match of the other annotator with the referenced person. The IAA was calculated after the final manual review and achieves 98.74% F1 (98.87% precision and 98.61% recall) overall. The IAA does not reach 100% due to the inherent ambiguity in policy documents and different interpretations of the same sentence. The IAA for each document is presented in Table A.3 in Appendix A.6. We spent an average of 1 hour annotating each policy, or 60 hours in total for 2 annotators.

3.6 Data Practice Extraction

3.6.1 Automated Extraction Techniques

3.6.1.1 `PI-Extract`

`PI-Extract` extracts data objects and the data practices by using neural networks which provide more flexibility than the rule-based methods. While rule-based methods rely on the completeness of the list of collection and sharing verbs, neural models leverage the semantics and syntactic knowledge from word embeddings trained on massive corpora. In particular, as described below, `PI-Extract` uses a BLSTM-CRF model based on BERT-Large-Cased contextual word embeddings [83] to achieve the best performance. Below, we describe the design of `PI-Extract` and experiments with different data practice extraction techniques.

In the BLSTM-CNN-CRF architecture [202], the input text is encoded into a dense vector as the concatenation of word embeddings and character-level representations (encoded by a Convolutional Neural Network (CNN)). The embeddings are then inputted to a layer of BLSTM which encodes the sequence in both backward and forward directions. For a given sentence $(x_1, x_2, \ldots, x_n)$ containing n words, an LSTM computes a representation $\overrightarrow{h_t}$, the left context of the word x_t in the sentence. Another LSTM layer computes a representation $\overleftarrow{h_t}$ for the right context. Thus, each word within the sentence is represented as a combination of the left and right contexts, $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$. This representation is then fed to a CRF layer to compute the scores of the labels for each input token with dependency on its neighbors.

`PI-Extract` uses 4 BLSTM-CRF-based NER models to predict the 4 labels in any sentence because each NER model can predict only a single non-overlapping label for each token while different labels can overlap, i.e., a token can have multiple labels assigned to it. Each model is jointly trained on each dataset to recognize both the text boundaries of data objects and the privacy actions (like collection or sharing) performed on them. `PI-Extract` uses the maximum likelihood as the loss function so that the training process maximizes the probability of the correct tag sequences [187].

The BLSTM-CRF network has one 100-dimensional bidirectional LSTM layer. We used L2 regularization for the transitions in the CRF layer with $\alpha = 0.01$. The training phase used a batch size of 20 and an Adam optimizer with a learning rate of 10^{-5} and coefficients ($\beta_1 = 0.9$, $\beta_2 = 0.999$). These parameters are similar to those used in [202]. We experimented with two state-of-the-art pre-trained word embeddings: 300-dimensional GloVe [245] and 1024-dimensional contextualized BERT-Large-Cased [83]. GloVe converts each token to a dense real-number vector regardless of its context while BERT leverages the context in the sentence to generate the output embeddings.

Since it is desirable to balance between high precision and high recall for generic use cases, the model is optimized for the F1 score (i.e., the harmonic mean of precision and recall). The training of the neural models ran for a maximum of 100 epochs and stopped early if F1 did not improve after 10 epochs. `PI-Extract` implemented the neural models using the AllenNLP framework [115].

3.6.1.2 Rule-based Extraction

To create a strong baseline, we implemented a rule-based extraction (`RBE`) method based on the open-source code of PolicyLint [20]. PolicyLint uses patterns of dependency trees of sentences to extract policy statements as 3-tuples $P = (Entity, Action, Data)$ where *Entity* performs an *Action* (*collect* or *not-collect*) on the *Data*. A data structure called *Data-Entity-Dependency* (DED) tree is used to analyze the dependency tree of the sentence to extract the policy statements. A DED tree represents the relation between a *Data* and a *Entity* in a sentence's dependency tree.

`RBE` uses a list of phrases for the corresponding parties to determine the role of an Entity (i.e., a first or third party). The list comprises terms subsumed by the first/third-party phrases (Table 3.2) in the ontologies of PolicyLint [20] and PoliCheck [22]. `RBE` matches the lower-cased words if the phrase is a pronoun, or matches the lemmas otherwise. For example, "authorized third-party service providers" contains lemmas "service provider", and is hence classified as a third party.

`RBE` then determines the label for each *Data* text span based on the role of the *Entity* in each simplified policy statement extracted by PolicyLint, which expresses a data flow to the *Entity*. In particular, the label is *Collect* or *Share* for a first- or third-party *Entity*, respectively. The same action verb can have a different label, depending on the *Entity* role. For example, considering "we may share your personal information with third parties" and "you may be required to share your personal information with us," although they use the same verb *share*, the label of "your personal information" is *Share* in the first sentence

Party	Phrases
1st party	I, we, us, our company
2nd party (user)	you, visitor
3rd party	third party, affiliate, advertiser, business partner, partner, service provider, parent corporation, subsidiary, sponsor, government agency, other company, other organization, other party, other service

Table 3.2: Phrases for determining privacy parties.

but is *Collect* in the second case. Examples of label determination are given in Table A.1 (Appendix A.3).

`RBE` makes several changes to optimize PolicyLint extraction for the `PI-Extract` dataset. `RBE` disables a generation rule of PolicyLint which generates a *Collect* label for every sharing verb since we do not assume any implication between the labels (Section 3.4.2). Furthermore, `RBE` adds the clausal complement (*ccomp* dependency) to negative sentiment propagation to improve the extraction of negated verbs. Given data objects extracted by PolicyLint, `RBE` aligns them to noun chunks following our annotation pipeline (Section 3.5.1.2). On the other hand, the original entity recognition model of PolicyLint is reused because its data-action extraction algorithm was optimized for the data objects extracted by the model.

3.6.2 Evaluation

3.6.2.1 Dataset

We randomly divide the dataset into 23 documents (3035 sentences) for training and 7 documents (1029 sentences) for validation. Denoting a *positive sentence* to be the one with at least one labeled text span, the number of positive sentences and data objects of the dataset for each label are given in Table 3.3. The *Collect* and *Share* labels have the largest number of training instances with 575 and 348 positive sentences, or 1311 and 552 data objects, respectively. *Not_Collect* and *Not_Share* labels have the fewest number

Label	Split Name	# Positive Sents	# Data Objects
Collect	Training	575	1311
Collect	Validation	192	409
Share	Training	348	552
Share	Validation	144	209
Not_Collect	Training	37	56
Not_Collect	Validation	14	22
Not_Share	Training	58	72
Not_Share	Validation	15	21

Table 3.3: Dataset statistics. Positive sentences contain at least one labeled data objects.

of training examples with only 37 and 58 positive sentences, or 56 and 72 personal data phrases, respectively.

3.6.2.2 Metrics

We compute the precision, recall and F1 score for the exact matches in which a predicted span is considered as true positive only if it exactly matches the golden standard span [296]. Since our goal is to extract and visualize the data objects as complete as possible, maximizing F1 (geometric mean of precision and recall) is more desirable than only maximizing the precision.

3.6.2.3 `RBE` Performance

The performance of `RBE` is shown in Table 3.4. Since `RBE` is designed to maximize the precision of recognition, it has low recall and high precision. With train patterns, while the recall rates are only in 27 – 43%, the precision in all of the labels are in 81–100%. The highest precision is 100% for the *Not_Collect* label, and the lowest is 81.34% for the *Collect* label. The overall F1 is 41.81%.

`RBE` is limited by the pre-specified vocabulary, grammar and extraction rules. Its list of collection and sharing verbs is not complete. For example, the verb list does not include *ask*, so it missed data practices in sentences like "we ask for your name when you register to

	Without Train Patterns			With Train Patterns		
Label	Prec.	Rec.	F1	Prec.	Rec.	F1
Collect	83.19	24.21	37.50	81.34	26.65	**40.15**
Share	81.69	27.75	41.43	82.43	29.19	**43.11**
Not_Collect	100.0	18.18	30.77	100.0	27.27	**42.86**
Not_Share	100.0	42.86	**60.00**	90.00	42.86	58.06
Overall	83.74	25.72	39.35	82.59	27.99	**41.81**

Table 3.4: Prediction performance of RBE method. In *With Train Patterns* configuration, RBE was trained on the positive sentences in the training set, in addition to the original PolicyLint samples.

use certain features." Furthermore, RBE missed data practices in sentences that have complex grammars outside of its 16 training patterns, such as "we may enable our advertisers to collect your location." RBE could not extract *Not_Share* data objects in negative-sentiment expressions that are not included in its negated-verb extraction rules, such as *your email address* in "we may provide your physical mailing address (but *not your email address*) to a postal service." RBE also failed to recognize negative sentiments in semantically-negated statements like "*under no circumstance* do we collect any personal data revealing racial origin."

The performance of RBE improved slightly when it was trained on the positive sentences (i.e., sentences with at least one data object) from training data. RBE learned 616 patterns from 1438 sentences which comprise 560 original PolicyLint samples (86 patterns learned) and 878 unique positive sentences (530 patterns learned) from the PI-Extract dataset. The overall F1 score increases by 2.46% when it uses patterns learned from sentences in the training set so the recall rate is improved with more known patterns. We conjecture this limited improvement to come from the fact that RBE was not designed to learn directly from complex grammars in the real-world sentences but rather from sentences with simple building-block patterns.

Word Embeddings	Label	Precision	Recall	F1
GloVe	*Collect*	65.78	54.52	59.63
GloVe	*Share*	44.17	43.54	43.86
GloVe	*Not_Collect*	77.78	31.82	45.16
GloVe	*Not_Share*	55.56	47.62	51.28
	Overall	57.87	50.08	53.69
BERT	*Collect*	64.46	69.19	**66.75**
BERT	*Share*	65.82	49.76	**56.68**
BERT	*Not_Collect*	100.0	50.00	**66.67**
BERT	*Not_Share*	72.73	76.19	**74.42**
	Overall	65.71	62.63	**64.14**

Table 3.5: Prediction performance of neural methods.

3.6.2.4 `PI-Extract` Performance

Since the neural models are more flexible than the rule-based methods of PolicyLint, they have higher overall performance (F1) but lower precision. The neural networks leverage the syntactical and vocabulary knowledge in word embeddings which were trained with very large datasets [262]. The contextualized embeddings in BERT have better performance than the traditional embeddings in GloVe. Our evaluation results are summarized in Table 3.5. When using BERT, the overall F1 score is 64.14%, and F1 is improved 7.1–23.1% across labels, compared with the neural models with GloVe word representations.

Using BERT, the extraction works best on the *Collect* label at 66.75% F1 and worst on the *Not_Collect* label at 56.68% F1. This reflects the recognition accuracy is proportional to the dataset size: *Collect* has the most number of training examples (1311 text spans) while *Not_Collect* has the least (56 text spans). A main reason for the low F1 score is that the vagueness and sophistication of the language used in privacy documents make it difficult to determine the text spans and the actions on them. Since the models with BERT embeddings outperform both GloVe-based configurations and `RBE` by large margins in all labels, we henceforth use BERT-based models for `PI-Extract` unless stated otherwise.

Since low recall rates are shown to make a bad impact on the usability of visual presentation of data practices (Section 3.7), we tried to improve the recall rates of the BERT models by changing the early stopping criterion to stop the training when the *recall rate* did not improve for 10 epochs. However, there is a trade-off between recall and precision. While the overall recall was improved by 3.03%, the overall precision decreased by 4.67% and F1 reduced by 0.88% (Table A.2 in Appendix A.4). Therefore, to make the model to be generic for a wide range of applications rather than being application-specific, we kept the above models with the higher F1.

3.6.2.5 Extraction of Context-free Data Objects

We hypothesize that the low F1 scores of the models were due to the limitation of NER models which were designed to extract context-free named entities rather than context-dependent data objects and practices. We test the performance of NER models to extract context-free data objects without the data actions. We derived a set of data object entities by merging all the data action labels into a single *Data* label. In the preprocessing step, sentences without any data collection/sharing verbs (list of such verbs are from [20]) were removed. Overlapping labeled text spans were resolved by keeping the longest text spans. This dataset has 1,737 sentences, 55.3k tokens and 1,736 entities. The corpus was then split into a training set (1,274 sentences, 39.4k tokens and 1271 entities) and test sets (463 sentences, 15.9k tokens and 465 entities). On the test set, the BERT-based NER model achieved an F1 score of 80.0% (79.2% precision and 80.9% recall). This result provides supports that context-free data objects can be extracted with high accuracy by the NER models and the consistency of the annotations on data objects in our corpus.

We developed a rule-based string matching baseline that matches data objects based on the lemmas of all the data-object terms in the training set. This method has an F1 score of 48.65% with 34.37% precision and 83.19% recall. The recall rate does not reach 100% because the validation set still contained unseen terms such as those that were specific to the type of the service (such as *photograph*) and did not occur in the training set. Furthermore, the training set did not include complete combinations of word forms such as it included *personally identifiable information* but not *personally identifying information*. The precision is low because this method does not distinguish the semantics of sentences. For example, a data object can be used in data usage purpose clauses that do not express data collection or sharing practices, such as the service uses encryption "to prevent unauthorized persons from gaining access to *your personal information*."

3.6.2.6 Performance on Homogeneous Privacy Policies

We evaluate `PI-Extract` on a homogeneous collection of privacy policies that contains policies of services in the same domain. We hypothesized that `PI-Extract` would have better performance on such policies since they share a similar vocabulary of data objects. Specifically, we selected 11 policies of news websites from the `PI-Extract` dataset (listed in Table A.3 in Appendix A.6) to trained the BERT models (described in Section 3.6.1.1) using the *k*-fold cross-validation strategy. Each of the 11 policies was held out once to create a dataset such that the validation set comprises the held-out policy and the remaining 10 privacy policies constitute the training set. `PI-Extract` achieved an average F1 score of 69.56% (79.21% precision and 62.42% recall) which is 5.42% higher than that on the heterogeneous `PI-Extract` dataset. This result indicates `PI-Extract` performance can be improved further by training on a dataset in the same domain as the target application.

3.7 Visual Presentation of Data Practices

3.7.1 Presentation Method

We propose a presentation method, called *data practice annotation* (`DPA`), to highlight and describe the data practices extracted by `PI-Extract` in order to enhance users' understanding of privacy policies. In particular, from the predictions of `PI-Extract`, the personal data objects are highlighted, and actions performed on the data objects are described as text annotations. The data action labels are displayed on the top of the highlighted phrases so that they do not hinder the reading flow of the users on the policy text. The background colors of the text and labels are different for each label. The presentation is implemented in web browsers using Brat annotation tool [289]. An example is shown in Figure 3.1.

Although there is a rich body of research on text highlighting [38, 109, 201, 232, 248], little has been done on the effects of text highlighting and annotation for user comprehension of privacy policies. Wilson *et al.* [304] found that highlighting relevant privacy policy paragraphs can reduce task completion time and positively affect the perceived difficulty of crowdworkers without impacting their annotation accuracy. However, `DPA` is different in both granularity and the presentation method. First, `DPA` annotates policies at a fine-grained phrase level. Second, `DPA` not only highlights personal data types but also provides descriptions of privacy practices performed on the data types. The highlighted data objects help users find them faster because the users need not perform a slow linear search through the text since the highlighted text already stands out. The data practice annotation puts explanation of privacy practices into context and helps users read related policy statements easier.

3.7.2 User Study Design

We design an IRB-approved (Study No. HUM00158893) user study to evaluate the effects of the `DPA` presentation on users' reading comprehension. The purpose of this experiment is to answer the following questions.

- **RQ1**: If correct data practice annotations are presented, do users understand privacy policy text better, as indicated by a higher total score?
- **RQ2**: If erroneous data practice annotations are presented, do users have worse comprehension?

	Domain	#Sents (#Words)	FKG	Question (Question Type)	DPA-Err Error Type
E1	*wealthfront.com*	6 (184)	17.43	Q1-1 (Data action)	Omitted annotation
E2	*ea.com*	6 (133)	14.40	Q2-1 (Data action)	Incorrect data action
E3	*linkedin.com*	14 (300)	12.40	Q3-1 (Data action) Q3-2 (Data type)	Incorrect data action Omitted annotation
E4	*tigervpn.com*	14 (349)	13.07	Q4-1 (Data action) Q4-2 (Data type)	Omitted annotation Incorrect data action
	Average	10 (241)	14.32		

Table 3.6: Domain names, lengths, readability scores, questions and types of annotation errors in `DPA-Err` version of the selected policy excerpts (E1 – E4).

- **RQ3**: If data practice annotations (which are either correct or incorrect) are presented, do users need less effort to read the policy excerpts, as indicated by shorter answering time?

3.7.2.1 Subjects

We recruited 150 crowdsourced workers from Amazon MTurk [15] for the survey. All the participants were required to reside in the United States due to restrictions in our IRB. To ensure the participants are experienced, they were required to have a good performance track record which includes a 90%-or-higher task approval rate and at least 1,000 HITs approved. We screened the participants during the training to ensure users have sufficient English skills to read and understand the instructions and privacy statements. The workers spent 9.6 minutes on average (with a standard deviation of 5.6 minutes) to complete the questionnaire. We paid each worker $2.3 so they earned an hourly wage of $14.3 on average, which is higher than the U.S. Federal minimum hourly wage of $7.25 in 2020 [220].

	`Plain` (n=52)	`DPA-Err` (n=49)	`DPA` (n=49)
Overall	3.69 (1.04)	3.12 (1.07)	4.67 (1.16)
Short Excerpts	1.23 (0.70)	1.49 (0.62)	1.76 (0.48)
Long Excerpts	2.46 (0.90)	1.63 (0.86)	2.92 (0.89)

Table 3.7: Mean (SD) scores. Max possible total scores in Overall, Short Excerpts and Long Excerpts are 6, 2, 4, respectively. *n* denotes the number of samples.

Excerpt	*Collect*				*Not_Collect*				*Share*				*Not_Share*			
	Prec.	Rec.	F1	Sup.	Prec.	Rec.	F1	Sup.	Prec.	Rec.	F1	Sup.	Prec.	Rec.	F1	Sup.
E1	0.83	1.00	0.91	5	0.00	0.00	0.00	1	0.50	1.00	0.67	1	-	-	-	0
E2	-	-	-	0	1.00	1.00	1.00	1	-	-	-	0	-	-	-	0
E3	1.00	1.00	1.00	13	-	-	-	0	0.75	0.50	0.60	6	-	-	-	0
E4	0.88	1.00	0.93	7	0.00	0.00	0.00	5	1.00	1.00	1.00	2	1.00	1.00	1.00	1

Table 3.8: Extraction performance of `PI-Extract` on the 4 policy excerpts. 0% F1 score indicates no prediction made for the label.

3.7.2.2 Instruments

We selected 4 excerpts from real-world privacy policies, each of which comprises one or multiple paragraphs. Each excerpt is self-contained and contains coherent content (e.g., anaphoras refer to other words in the same snippet). The privacy policies are of diverse online service types: financial (*wealthfront.com*), gaming (*ea.com*), professional social networking (*linkedin.com*), and virtual private network services (*tigervpn.com*). These types of businesses are known to collect sensitive data about users' finance, children's personal information, social connections, and data transfers. The policies were downloaded as the latest version in August 2020.

Excerpts of privacy policies were presented instead of the whole privacy policies because it is unrealistic for a user to read a thousand-word privacy policy from start to end [212]. We assume users can always narrow down to the sections of their interest by using a table of contents or information retrieval tools like Polisis [141].

We experimented with policy segments of different lengths (short and long) and different difficulty levels (easy and hard) of policy text. There are 4 segments in the study, a combination of two lengths — short and long – and 2 types of highlights — positive and negated. The short paragraphs have 133–184 words (6 sentences) while long paragraphs have 300–349 words (14 sentences). The reading time is expected to be 0.6–1.5 minutes (assuming an average reading speed of 238 words/minute [44]). With 4 excerpts in the questionnaire, the total task completion time for each participant (including answering the demographic survey, training questions and usability questionnaire) was expected to be about 10 minutes.

To evaluate the difficulty of the excerpts, we use Flesch-Kincaid Grade Level (FKG) [180] to measure their readability. FKG computes the average grade a person is expected to completely understand the written text and was used in readability studies of privacy policies [101, 284]. Three incomplete-sentence section titles with 2 words or less (such as "2.1. Services") were excluded to avoid skewing results. The excerpts have an average FKG of 14.32, indicating 14 years of education are expected for full comprehension. This reading difficulty is similar to the average FKG of 14.42 in a recent large-scale privacy policy survey [284]. The easiest policy passage is *linkedin.com* with an FKG of 12.40 and the hardest is the snippet from *weathfront.com* with an FKG of 17.43. Table 3.6 shows the detailed statistics of the selected policy excerpts.

We used `PI-Extract` to extract the data practices in the excerpts which were previously unseen by the models. The policy snippets contain 1–19 data practice annotations. All 4 data action labels (Section 3.5) have at least one occurrence among all snippets. The prediction performance is 71.1% F1 score on average, ranging from 0.6 – 1.0 F1 score. Table 3.8 provides the number of the data practices and prediction performance for each of the selected excerpts.

Questions. The questions test the comprehension of participants about the content of the excerpts. There are 1 and 2 questions in short and long excerpts, respectively.

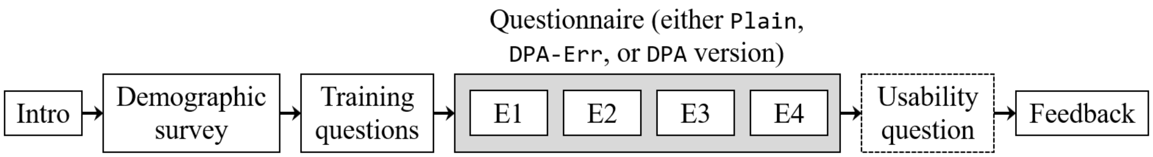


Figure 3.5: Visualization of the user study process. Each participant will be shown either `Plain`, `DPA-Err`, or `DPA` version of the policy excerpts (E1–4) in the Questionnaire. Questions in the shaded box are randomly shown to the users. The question in the dashed box is shown only to users of annotated (`DPA` and `DPA-Err`) versions.

Multiple-choice questions (rather than yes/no questions) were used to reduce noisy randomly-selected correct answers. There are 2 types of questions: (1) select a correct data action performed on a given data type and (2) select a correct data type given a data action and a condition. In the data action questions, the 4 choices are the 4 data actions as described in Section 3.5. In the data-type-selection questions, alternatives were created as data types in a similar context to avoid guessing the correct answer without reading. In long excerpts, the first and second questions are based on the facts in the first and second halves of the snippet in that order. While the questions are the same among all excerpt versions, the correct answers are contained in one of the annotations in the `DPA` version. Table 3.6 lists the types of questions for each excerpt.

To test a deep understanding of the policy text, the questions include conditions or complex data objects which are referenced across sentences so that the respondents need to read carefully to select the correct answer. For example, one question asks for the data practices on the "personal information from children under 13" which was mentioned and defined in different sentences. The questions and excerpts in the `DPA` version are listed in Appendix A.1.

Incorrect Predictions. We created a version (called `DPA-Err`) of the excerpts which contain incorrect annotations to test their effects on user comprehension. These annotations may occur due to imperfect predictions of neural models used in `PI-Extract`. We manually injected incorrect annotations by altering the existing annotations which were asked in the questions. There are 2 types of wrong annotations. The first is *omitted annotation* in which the annotation of the data type asked in the question is missing from the excerpt. The second is annotations with an *incorrect data action* label. We consider common wrong predictions of swapping between *Collect* and *Share* labels, and between negated and positive labels (such as *Not_Collect* and *Collect*). Table 3.6 lists the error types in the `DPA-Err` version.

3.7.2.3 Procedures

At a high level, the study follows a between-subject design so that each participant reads one of the versions of the privacy policy excerpts and were asked questions related to their content. The three versions of the policy segments are `Plain` (raw text), `DPA-Err` (annotated text with injected errors), and `DPA` (annotated text with predictions from `PI-Extract`). Fig. 3.5 shows the visualization of the process of the user study.

After an initial introduction, the experiment comprises 4 main sections: demographic survey, training, main questionnaire, and a usability question. The introductory instructions used neutral descriptions without mentioning the annotation presentation in order to prevent participants from forming potential bias. In the main questionnaire, each respondent was presented with either `Plain`, `DPA-Err`, or `DPA` version of the policy excerpts. Questions from the 4 excerpts were also randomly shown to the participants to avoid fatigue effects on a particular excerpt. For each policy snippet, a brief description of the company was provided to the participants to inform them of the context of the privacy statements. We collected the answering time of the participants for each question which was measured from the beginning of the question until the answer was submitted. Due to the limitation of the survey platform which can only measure the submission time per page, participants were shown one question with the corresponding excerpt at a time. The back button was disabled so that participants could not go back to modify their answers.

Since our purpose is to test the reading comprehension, policy excerpts were presented as images to control the results to be only from reading the text, i.e., avoid mixing answers from using a finding tool with answers from reading. Using a finding tool will entail another factor of users' fluency in using the searching tools. To make the text images display consistently among participants, the crowdsourced job description required to perform the questionnaire on a PC or laptop and we programmed the survey to detect the performance on smartphones to terminate the experiment at the first step. The user study was designed and performed via Qualtrics online survey software [254].

Training Questions. Before the main questionnaire, the participants were given two sample questions to help them get used to the main task. Explanations were displayed if they selected wrong answers and they could not proceed until they answered all questions correctly. The instructions also included a notice of the possibility of erroneous data practice annotations due to incorrect predictions.

Usability Question and Feedback. After the main questionnaire, annotated version participants were asked about the usefulness of the annotated text and provided their ratings on a 5-point Likert scale. A final free form feedback form was also provided.

3.7.3 Experimental Results

We collected a total of 900 responses for the 6 questions from 150 distinct respondents. 52 participants completed `Plain`, 49 did `DPA-Err`, and 49 did `DPA` version. We originally planned to have the same number of workers for each version, but because the participants did the survey simultaneously and some of them left in the middle of the survey, the survey platform did not divide the respondents evenly. All participants completed the survey using a web browser on a desktop operating system and their screens had width and height of at least 1024 and 786 pixels, respectively. In this section, unless noted otherwise, we calculate effect sizes by using Cohen d and the standard deviation is abbreviated as *SD*.

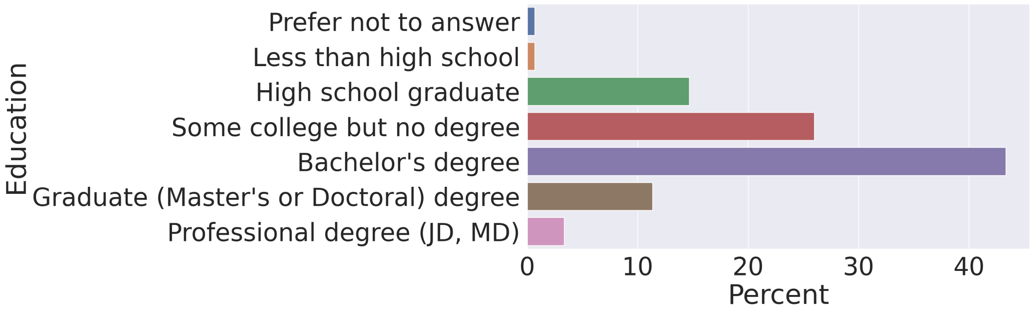
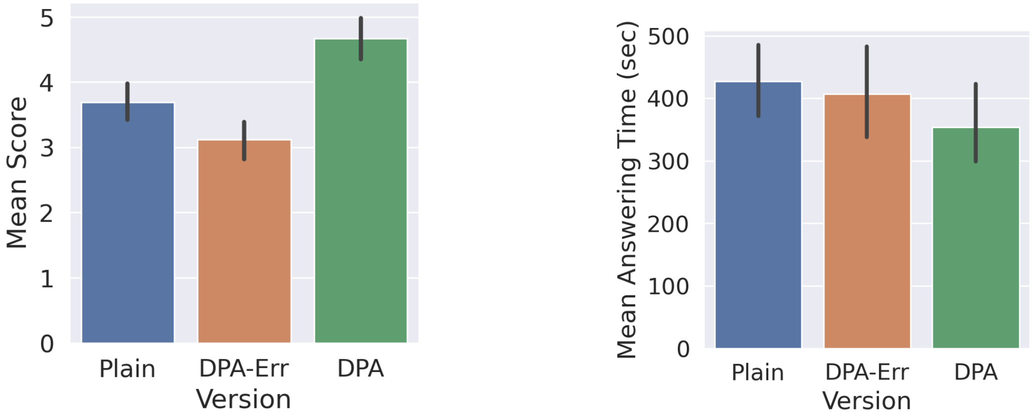


Figure 3.6: Education levels of the participants.



(a) Average total scores (max possible total score = 6).

(b) Average total answering time.

Figure 3.7: Average total scores and answering time of excerpt versions. Error bars are 95% confidence intervals.

Each correct answer gets 1 score so the maximum possible score of the questionnaire is 6. The score and answering time of each question are shown in Fig. A.1 in Appendix A.2.

3.7.3.1 Demographics

Across all the respondents, the average age is 45 years (SD=12.1), 49% are males and 50% are females (1% preferred not to answer). 99% of the participants have at least a high school degree (1% preferred not to answer). 41% of the respondents have either high-school education or some college but with no degree while 58% have a bachelor's degree or higher (Fig. 3.6). 85% of the workers reported being employed.

Error Type	Version	Mean (SD)	p-value (d)
Omitted annotation	`Plain`	1.81 (0.66)	-
	`DPA-Err`	1.18 (0.67)	$<$ .001 (0.94)
Incorrect data action	`Plain`	1.88 (0.70)	-
	`DPA-Err`	1.94 (0.63)	0.68 (0.08)

Table 3.9: Scores on different error types of `DPA-Err`. The max possible total score of the questions of each type is 3.

3.7.3.2 Research Question 1

The data practice annotations in `DPA` version improve the reading performance significantly, as indicated by a significant higher total score ($F(1,99) = 20.06$, $p < .001$, $d = 0.89$). The annotations improve the average total score by 26.6%, from 3.69 ($SD = 1.04$) to 4.67 ($SD = 1.16$) and the effect size $d = 0.89$ is large [57, 267]. The detailed scores are shown in Fig. 3.7a and Table 3.7.

Further analysis shows that the effect of `DPA` is significant on both short policy excerpts ($F(1,99) = 18.92$, $p < .001$, $d = 0.87$) and long snippets ($F(1,99) = 6.63$, $p < 0.05$, $d = 0.51$). The improvement in average total scores of `DPA` is on short snippets (42.6% increase) which is higher than the long excerpts (18.56% increase). `DPA` is most effective on the question *Q2-1* which asks about the data action performed on *personal information from children under 13* of *ea.com* with the correct answer to be *Not Collected*. The effect size on this question is large $d = 1.45$ ($F(1,99) = 53.34$, $p < .001$). We hypothesize that there are fewer annotations in short texts so users spend less time to find the annotations relevant to the question. Table 3.7 shows the scores on the excerpts.

3.7.3.3 Research Question 2

The effect on the overall reading performance of wrong annotations in `DPA-Err` version is significant ($F(1,99) = 7.35$, $p < 0.01$, $d = 0.54$). The average total score was reduced by 15.43% from 3.69 to 3.12. The effects on short and long excerpts are mixed. While `DPA-Err` slightly increases the average score by 21.05% ($F(1,99) = 3.85$, $p = .052$, $d =$

Version	Mean (SD)
Plain	427.06 (215.80)
DPA-Err	407.31 (251.54)
DPA	353.97 (226.30)

Table 3.10: Average total answering time (sec).

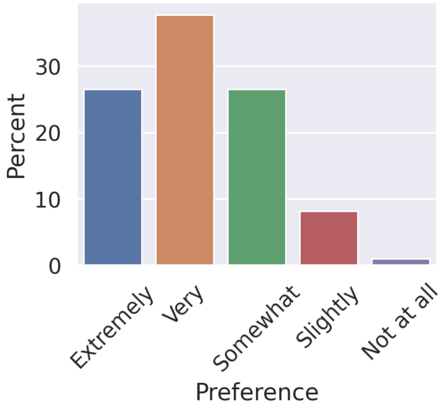


Figure 3.8: Helpfulness of annotations (DPA and DPA-Err).

0.39) on short excerpts, it significantly reduces the score on long excerpts by 33.67% ($F(1,99) = 22.49$, $p < .001$, $d = 0.94$). Table 3.7 lists the scores.

To identify the causes of the negative impacts of incorrect annotations, we further analyzed the effects of DPA-Err when annotations were either omitted or contained an incorrect data action label. While the reduction of the omission incorrectness on performance is significant ($F(1,99) = 22.40$, $p < .001$, $d = 0.94$), the decrease caused by incorrect-action-label annotations is non-significant ($F(1,99) = 0.16$, $p = 0.68$, $d = 0.08$). The omission incorrect type indeed did not add any value to the policy text but action-label-swapped incorrect annotations still helped users find the relevant data types so that they could read the surrounding text to answer correctly. A user reported that s/he "still has to read the sentence, it didn't highlight negatives like *do not... collect*." The detailed scores are listed in Table 3.9.

3.7.3.4 Research Question 3

Annotations do not significantly reduce the effort of reading the policy text, as indicated by the shorter average total answering time. The difference of average total answering time among 3 versions (Plain and annotated versions) is not statistically significant ($F(2,147) = 1.33$, $p = .266$). DPA slightly reduces the average total answering time of

Plain version by 17.11% ($F(1,99) = 2.76$, $p < .10$, $d = 0.33$). The difference of the answering time between DPA-Err and Plain is non-significant ($F(1,99) = 0.18$, $p = 0.67$, $d = 0.08$). The total answering time is shown in Fig. 3.7b and Table 3.10. The answering time for each question is shown in Fig. A.1b in Appendix A.2.

3.7.3.5 Effect of Education Levels

Since the 4 policy segments have different readability scores, we compute the correlation between the user education levels and the answering scores for each policy excerpt. The results show that users with higher education levels achieved higher scores on the Plain version of excerpt E-1, which requires 17.43 years of education to comprehend and is the hardest in the questionnaire. Specifically, users with a bachelor's degree or higher get a significantly higher average score than the other participants with lower education levels. The average score increases by 36.88% from 0.68 to 0.93 ($F(1,50) = 6.05$, $p = 0.017$, $d = 0.69$). However, there was no significant difference for other easier excerpts in the Plain version. The average scores were also not significantly different in DPA and DPA-Err versions. We hypothesize that the annotations made the policy excerpts easier to read, thus reducing the difference of scores between education levels.

3.7.3.6 Qualitative Evaluation

A majority of the participants with the annotated versions (both DPA and DPA-Err versions) found the visual aid helpful. 64.2% of them considered the highlighted text very or extremely helpful while 9.2% considered the annotations provided no or slight help. The DPA version which has relevant annotations was given higher preference: 77.5% of workers considered the highlighted text very or extremely helpful and no participant found the visualization not helpful. Fig. 3.8 shows the distribution in the annotated versions.

The participants of this study also provided free-form comments which confirm the helpfulness of the visual aids. A participant answering the Plain version said the policies were "still not clear, companies need to be required to do a better job." On the other hand, the DPA "was very effective to find information" and "without the highlights it would take many minutes and much more effort to grasp how complicated this all is."

3.8 Discussion and Limitations

3.8.1 Limitations of the Model

PI-Extract is not able to detect implicit data objects and actions which are not stated explicitly in sentences. For example, "if we notice that users in general prefer national political commentary, we might put that content in a special place on the website or in the app" indicates that user preference is collected to promote the political advertisements. However, the model is not able to extract the data and action in such a case. Moreover, personal data types can be mentioned indirectly by referring to other data types in other sentences. For example, in the sentence "when you post comments in response to a story or video on any of our Services, we — and other users — receive *that information*, the phrase "that information" refers to "comments" and requires co-reference resolution to extract. These limitations can be alleviated by using more sophisticated natural language understanding techniques that can model and analyze the semantics of implicit statements and analyze privacy policies as a whole, not only on a sentence basis.

The contiguous non-nested entity annotation cannot capture data types in nested or non-contiguous texts such as when multiple data objects are included in a single list. For example, two data objects "software attributes" and "hardware attributes" are included in a complex phrase "software and hardware attributes". Such nested data types can be annotated by using nested-entity annotation scheme [104], but it will require a significantly more complicated annotation scheme. The annotation scheme also does not cover the conditions and purposes of data actions which are left as our future work.

The dataset focuses only on privacy policies on websites and has not explored other platforms such as mobile and IoT devices. However, we observe that it is common for services to have a single privacy policy that covers multiple platforms, especially for popular online services [305]. Therefore, similar data types are used across the policies in different platforms and can be extracted by the `PI-Extract` models.

Although we hoped NER models can jointly learn to extract personal data objects and the actions performed on them effectively, the overall F1 scores are still low. This is possibly due to insufficient data samples needed for the NER models to learn to distinguish different actions applied to the data types in different contexts. Future advances in natural language processing will improve entity extraction models and require less data, so the performance of `PI-Extract` will be further improved.

Privacy-policy domain-specific word embeddings trained on large corpora of policies were known to provide performance improvements [141]. However, due to the model complexity, training BERT models on million-policy datasets (such as [16, 315]) would require excessive computation. For example, SciBERT [31] needed 7 days on an 8-core TPU v3, and BioBERT [190] required 23 days on 8 Nvidia V100 GPUs. We leave the evaluation of domain-specific BERT models as our future work.

3.8.2 Validity of User Study

Our user study could not fully control the participation of online respondents although we tried to recruit experienced crowdworkers who are more likely to make an adequate effort to complete the survey properly rather than just randomly selecting the answers. However, bias should be reduced because of the between-subject design, random assignment among policy text versions, and the use of multiple-choice questions. It would be better to recruit law experts and interview them to have feedback on the quality of annotation.

The reading environment such as screen resolution was not controlled to be consistent among workers although we tried to enforce the participation via a desktop computer by checking the platform on which the survey was accessed. Furthermore, the study used photos to present to users, preventing them from using the Find tool which is common on browsers. A separate study design to test the effectiveness of the Find tool with DPA is needed because DPA does not require users to know the data objects and data practices in advance while the Find tool is useful only when the user knows the keyword s/he is looking for.

3.8.3 Limitations and Extensibility of Data Practice Annotation

Similar to the effects of text highlighting which depends on the quality of the highlights and the interaction with the learners, privacy practice annotations improve the user comprehension the most when the predictions are correct and users read the surrounding text to understand the sentence. Text highlighting has been shown to improve user retention if the highlights are relevant to the questions, and vice versa [38, 109, 201, 232]. Highlighting could even hurt readers' inference of the text [248].

Wrong predictions from PI-Extract indeed have a negative effect on users, similar to inappropriate annotations which are known to have a harmful effect on reading comprehension [117, 277]. However, even with the presence of the incorrect privacy practice annotations, given annotations with an incorrect data action, users appear to have similar comprehension to the Plain version as shown in the analysis of Research Question 2 (Section 3.7.3.3). We expect that with more sophisticated models, the prediction accuracy will improve and the wrong predictions will decrease.

More annotated privacy policies would improve the extraction performance of data practices further as the PI-Extract dataset still does not fully cover all the data types and grammatical phenomena. We measured the overall F1 given the validation set (Section 3.6.2) and the varied sizes of the training sets. The result shows that the F1 score increased with the number of policies (Fig. A.2 in Appendix A.5). The linear regression indicates that, if this linearly increasing trend continued, a training set of 56 policies would be needed to reach the overall F1 of 80%.

PI-Extract annotation scheme and pipeline are generic and can be extended to capture other aspects of privacy policies such as data usage purposes, data retention and opt-out choices. For example, an additional *Usage_Purpose* label can be used to denote the purpose of data collection or sharing. The relation between each data practice and its purposes can be then annotated by link annotations [91].

3.9 Conclusion

We have sought to automatically extract and present personal data objects and privacy practices performed thereon to help users understand which types of their personal information are collected and shared with third parties in privacy policies. We have constructed a large and fine-grained dataset, based on manual annotations of skilled workers. We have then presented PI-Extract, a fully automated system that uses neural models trained on the corpus to extract data practices from privacy policies and outperforms rule-based techniques. PI-Extract presents the extracted data objects and actions as data practice annotations (DPA) on the policy text. A user study was conducted to evaluate the effect of DPA and incorrect predictions on user comprehension and answering time when reading privacy policy excerpts. DPA made a significant improvement of users' comprehension of the presented policy snippets over the plain text version. The results demonstrate the applicability of PI-Extract in raising privacy awareness and reducing the privacy risks for end users.

The work in this chapter appeared in the 2021 Privacy Enhancing Technologies Symposium (PETS) and can be cited as [45].

CHAPTER IV

PurPliance

4.1 Introduction

The FTC has relied on privacy policies written in natural language as a primary means to check and inform users how and why apps collect, use and share user data [62]. Since purposes of data collection and use/sharing are key factors for users to decide whether to disclose their personal information or not [196], it is important for apps to make the users aware of, and consent to them. For example, users would more likely to agree to provide their location for receiving an app's services rather than for advertising purposes. Moreover, while the purposes of data collection, use and/or sharing are specified in the apps' privacy policies, the apps' actual execution behavior may deviate from their specifications in the policies.

Despite its importance, little has been done on checking the consistency between the *purposes* stated in the privacy policies and the actual execution behavior of apps. Prior studies [20, 21, 279, 302, 310, 316] overlooked the *purposes* and *entities* whose purposes were served. Furthermore, the assumption that data sent to an entity is always used for *any* of the receiver's purposes may not hold when the external service processes the data for the app's purposes. For example, the data sent to an analytic service should be used for the app to analyze its usage trend, not for the analytic service's purposes such as delivering personalized advertisements.

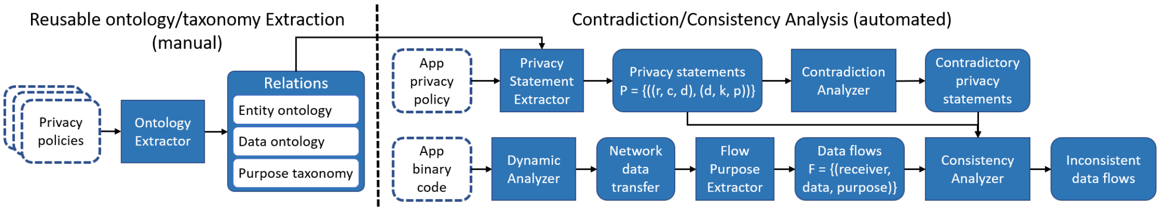


Figure 4.1: `PurPliance` system workflow. Dashed boxes indicate the system inputs.

A key question is then: *Can we automatically check whether the purposes of actual data usage comply with those stated in privacy policies or not?* The first challenge in answering this question is to achieve a clear interpretation of the privacy policy and detect contradictory privacy statements which, if exist, will make the disclosure of data flows ambiguous. The second challenge is to extract the purposes of the actual data flows from the app behavior and compare them with (potentially contradictory) privacy statements.

Analyzing fine-grained purposes of data usage yields a fundamentally different and more complete interpretation of privacy policies than purpose-agnostic approaches, such as PolicyLint [20] and PoliCheck [21]. Let us consider the following policy statement from a popular app on Play Store with more than 1M installations.

Example 1: "We do not share personal information with third parties for their own direct marketing purposes."

`PurPliance` interprets this example as third parties may collect personal data but do not use it to deliver their own advertising, which is part of marketing purposes. Therefore, `PurPliance` flags a contradictory data-usage purpose in another statement stating that the app "may share your personal data with third-party advertising partners to serve personalized, relevant ads." Purpose-agnostic approaches [20, 21] narrowly interpret Example 1 as the app would not share any personal data. Such approaches do not accurately detect the contradiction of the advertising usage purpose and generate lots of false positives because the example would then contradict any other statements about sharing of the user's personal data.

We present PurPliance, an end-to-end fully automated system that detects contradictory privacy statements and inconsistent app behaviors. In the system workflow (depicted in Fig. 4.1), contradiction/inconsistency analysis (right half) is fully automated while ontology extraction (left half) is manual and performed only once. Inspired by the *soundness* (i.e., *no-false-positive*) in software testing with dynamic analysis [147, 148, 280], PurPliance is designed to maximize the precision of detection (i.e., a reported inconsistency should always be true positive), as opposed to maximizing the recall rate.

PurPliance addresses the following three technical challenges. **TC1 (Purpose clause extraction)**: Purpose clauses are written in lengthy and complex phrases, and hence it is difficult to determine their start and end in a sentence. PurPliance leverages neural *Semantic Role Labeling* (SRL) models [172, 273] that are capable of analyzing many more grammatical variations than prior work [20], to extract privacy statement parameters from the semantic arguments of data-practice predicates. Finally, PurPliance extracts uncompounded purposes from complex purpose clauses by analyzing their semantic/syntactic structures and decomposing the clauses into simpler predicate-object pairs and noun phrases. We organize the common purpose clauses extracted from a large collection of privacy policies into a hierarchical taxonomy that defines the relationships among different usage purposes.

TC2 (Data flow extraction): Extracting the purpose of data flows to/from each app is very challenging because the flows take place at a low data level and lack high-level semantics. PurPliance leverages recently-developed datasets and dynamic analysis techniques [166] to infer the purposes and the purpose-served entities of network data traffic from the transferred data and its context. The low-level purposes of data traffic are then mapped to higher-level data-usage purposes in our taxonomy of data purposes.

TC3 (Automated consistency analysis): Automatic detection of contradictory privacy policy statements and inconsistent network data flows requires automated reasoning of these concepts. We introduce the notion of *data-usage purpose* which comprises a purpose-served entity and a usage purpose, and is separated from data collection and sharing. We formalize privacy statements and data flows, and formulate a consistency model to analyze and detect policy contradictions and flow-to-policy inconsistencies.

The evaluation of our end-to-end contradiction detection demonstrates that `PurPliance` is able to detect contradictory sentence pairs in privacy policies with significantly higher precision and recall than PolicyLint [20], a state-of-the-art policy analysis technique. An in-depth analysis shows two main sources of these improvements: 1) semantic-argument analysis improves the extraction of privacy statement tuples and 2) data-usage purpose analysis enhances the expressiveness of the privacy statement tuples to reflect the policy sentences' semantics more accurately. This paper makes the following main contributions:

- *Automatic extraction and classification of data usage purposes in privacy policies.* We developed automatic extraction of purpose clauses based on semantic arguments of the data practice predicates (Sections 4.3.1). We introduced predicate-object pairs to extract simple purposes from a complex clause (Section 4.3.2). We studied data usage purposes in a large privacy policy corpus to construct a purpose taxonomy and develop automatic classifiers. To the best of our knowledge, this is the first large-scale study and classification of data usage purposes in privacy policies.

- *Formalization and automatic extraction of privacy statements and data flows with support for data-usage purposes.* We developed NLP-based automatic methods to extract privacy statements with data-usage purposes from policy sentences (Section 4.4). We adapted existing methods to extract data flows with data purposes from network data traffic (Section 4.5).

- *A formal consistency model with support for data-usage purposes.* We propose a formal model to detect contradictions in privacy policies and flow-policy inconsistencies between privacy policies and mobile apps' data collection (Section 4.6).

- *An end-to-end system* (called PurPliance, open sourced at [88]) that detects inconsistencies between the privacy policy and actual data collection of an app. *A corpus of 108 privacy policies* (publicly available at [88]), containing 5.9k sentences and 189 contradictory sentence pairs, was constructed to evaluate the end-to-end contradiction detection. The results show that PurPliance improves the precision from 19% to 95% and the recall from 10% to 50% compared to PolicyLint. An in-depth analysis shows that PurPliance extracts 88% more privacy statements in 45% more sentences with 9% higher precision than PolicyLint.

- *A large-scale study of policy contradictions and flow–policy inconsistencies in 23.1k Android apps* (Section 4.8). PurPliance found 29,521 potential contradictions in 18.14% of the policies and 95,083 inconsistencies in 69.66% of the apps, indicating the prevalence of inconsistencies of data-usage purposes in mobile apps.

4.2 Related Work

Purpose Analysis in App Behavior. There has been a rich body of work to extract semantics of app behavior to identify potential leakage of sensitive information. Whyper [240], AutoCog [253] and CHABADA [135] analyze and assess the risks of an app's behavior (e.g., permission and API usage) in comparison with the app's description. FlowCog [239] extracts semantics of data flows from an app's GUI to analyze information leaks. NoMoATS [275] inspects the URL and HTTP headers to detect mobile network requests engaged in advertising and tracking. MobiPurpose [166] extracts and infers personal data types and purposes of their data collection from network traffic of Android apps, but it does not check whether the data-collection purposes are legitimate or not.

Privacy Policy Analysis. NLP and ML have been widely used for analyzing natural-language privacy policies. Privee [314] and Polisis [141] analyze privacy policies at the document- and paragraph-level to answer users' questions. However, both are limited by their coarse-grained analyses while our sentence- and phrase-level analyses provide more detailed and comprehensive results. PolicyLint [20] uses dependency parsing to extract privacy statements from policy documents but does not analyze purposes of data collection.

Bhatia *et al.* [35] extract common patterns of purposive statements from privacy policies and use semantic frames to analyze the incompleteness of privacy goals, which include the purposes of data practices [36]. Shvartzshnaider *et al.* [276] analyze information flows in a limited set of privacy policies following the contextual integrity framework [231]. However, these semi-automated methods require the laborious manual efforts of experts or crowd workers.

Behavior-Policy Compliance Analysis of Mobile Apps. Analysis of the (in)consistencies between the actual behavior of mobile apps and their privacy policies has gained considerable interest in recent years. Prior work tracks the collection of users' personal data automatically via Android API calls [279, 310, 316], or via user inputs on app GUI [302]. PoliCheck [21] built upon PolicyLint [20] and the AppCensus dataset [23] improves the accuracy of detecting non-compliances in an app's data flows by taking into account the recipients of the personal data. However, PoliCheck does not consider the business purposes of the data flows. Several researchers focus on narrow app categories, such as paid apps [139] or apps targeting family users and children [235, 259, 260]. They are limited to specialized app categories while `PurPliance` is general and applicable to any type of apps.

Taxonomy of Privacy Purposes. OPP-115 dataset [303] includes 11 classes of data collection and sharing purposes that were manually created in a top-down fashion by law experts. In contrast, we created a hierarchical taxonomy of data-usage purposes by using neural text clustering with contextualized word embeddings to group similar purpose clauses in a large policy corpus. Despite a rich body of work on text clustering [6, 206],

Data Practice	Verbs
Sharing	disclose, distribute, exchange, give, lease, provide, rent, release, report, sell, send, share, trade, transfer, transmit
Collection	collect, gather, obtain, receive, record, store, solicit
Use	access, analyze, check, combine, connect, keep, know, process, save, use, utilize

Table 4.1: List of the SCoU verbs used by `PurPliance`.

Data Action	Sender	Receiver	Data	Purpose	Example
Sharing	Arg0	Arg2	Arg1	Argm-Prp or Argm-Pnc	[We]$_{\text{Arg0}}$ do not [share]$_{\text{V}}$ [your data]$_{\text{Arg1}}$ [with third parties]$_{\text{Arg2}}$ [for their purposes]$_{\text{Argm-Pnc}}$.
Collection	Arg2	Arg0			[We]$_{\text{Arg0}}$ [collect]$_{\text{V}}$ [passwords]$_{\text{Arg1}}$ [for authentication]$_{\text{Argm-Prp}}$.
Use	N/A	Arg0			[We]$_{\text{Arg0}}$ may [process]$_{\text{V}}$ [your contact information]$_{\text{Arg1}}$ [to send you promotions]$_{\text{Argm-Prp}}$.

Table 4.2: Mapping from semantic roles to privacy statement parameters. *V* denotes a predicate (i.e., verb).

we are not aware of any other work that applies text clustering to the analysis of purposes in privacy policies.

4.3 Extraction of Data Usage Purposes

4.3.1 Extraction of Data Usage Purpose Clauses

4.3.1.1 Extraction of Data Practice Predicates and Semantic Arguments

`PurPliance` extracts the purposes of privacy practices by analyzing patterns of semantic arguments, syntactic structures (i.e., parts of speech and dependency trees) and a lexicon of data practices. It first finds data practice predicates (i.e., verbs) that express the action of a privacy practice event such as "collect" and "share". `PurPliance` iterates through the tokens of the sentence and extracts those words whose part-of-speech tags are a verb and whose lemmas are in a manually curated list of Sharing-Collection-or-Use (SCoU) verbs as given in Table 4.1.

We empirically identified common verbs in randomly selected privacy sentences to extend the SoC verbs in PolicyLint [20]. While PolicyLint only distinguishes between collection and sharing of data, we separate some *use* verbs. Although the *use* actions do not explicitly construct personal data flows, they still provide valuable information about data processing purposes. We added a verb to the SCoU list by surveying its usage in randomly selected sentences in our privacy policy corpus. Because every verb has multiple meanings, some of which are unrelated to data collection/sharing/use, there is a trade-off: naively adding verbs increases recall but reduces precision. Therefore, we select verbs that are frequently used to express data practices (i.e., in over 80% of 100 random sentences).

Given a data practice predicate, `PurPliance` analyzes its semantic arguments which are phrases that fill the meaning slots of the predicate and define its details. They answer questions such as "who?", "did what?", "to whom?", and "for which purpose?" of an event expressed by the predicate [172, 188]. Because arguments of the same event are consistent across varying syntactic forms, parameters of privacy statements (such as the receiver and data object) can be extracted accurately even though the same data practice event is expressed in multiple ways with varying grammars. An example of semantic arguments in varying expressions is given in Appendix B.1.

`PurPliance` uses Semantic Role Labeling (SRL), also called *shallow semantic parsing*, to recover the latent predicate-argument structures of sentences. SRL models are trained on corpora called *proposition banks* (PropBank) which contain labels of the semantic roles of sentences. In the corpora, such as OntoNotes 5.0 [255], a specific set of roles is specified for different senses of each verb. Some roles are numbered rather than named to make them more general (e.g., *Arg1* for *object* arguments) while many un-numbered modifier arguments represent the modification or adjunct meanings [42]. The definition of a role may vary with a verb's senses. For example, while *Arg2* typically denotes the instrument of a predicate, *Arg2* of certain data usage verbs like *use* and *store* indicates their purposes.

4.3.1.2 Extraction of Purpose Clauses

We identified semantic arguments that represent purposes based on their specifications in the CoNLL2012 corpus' verb sense frames [255]. The common arguments for purposes are *Argm-Prp* and *Argm-Pnc*, i.e., argument modifier purpose and purpose-not-cause, respectively. Table 4.2 presents some examples. Besides the common purpose arguments, `PurPliance` analyzes additional arguments for certain predicates to identify their purposes, such as *Arg2* of *use* and *save*. The list of these predicate-specific purpose arguments is shown in Table B.1 (Appendix B.1).

A verb may have multiple meanings, such as "save" which means either to save money or to collect (accrue) things. The latter meaning is more relevant in our context of data collection. We verified that arguments of different senses of the data practice verbs have the same meaning for the purpose of our privacy statement extraction, and hence we do not disambiguate the verb senses in this analysis.

We consider three forms of purpose clauses that either (1) start with "to" followed by a base-form verb, (2) start with "in order to" followed by a base-form verb, or (3) start with "for" followed by a gerund (a noun derived from a verb ending with *-ing*) or a noun. The first two forms are the standard identification of purpose clauses in English [169, 181, 209]. The third form is common in privacy policies, such as in "for providing services" or "for the purposes of ..."

4.3.2 Classification of Policy Purposes

4.3.2.1 Uncompounded Purpose Extraction

Since multiple simple purposes are commonly combined into complex purpose clauses, `PurPliance` decomposes them into simple single-purpose parts, called *uncompounded* *purposes*, similar to contextual sentence decomposition [30, 92] used to improve information extraction. Therefore, each complex purpose clause is simplified into a set of uncompounded purposes, each of which is represented by a predicate-object (PO) pair. A PO pair *(p,o)* consists of a predicate (verb) *p* that acts on an object *o*. For example, "to provide and improve our services" is decomposed into (*provide*, *our services*) and (*improve*, *our services*). Similarly, each noun phrase *np* can be converted to a PO pair with an empty predicate *(‘’, np)*. For example, "for fraud prevention and service maintenance" produces "fraud prevention" and "service maintenance". Table B.2 shows some PO-pair examples.

Each PO pair is extracted by first identifying the predicates and then their objects as its arguments. To extract a predicate, `PurPliance` finds words with a *verb* part of speech, excluding subsumptive relation verbs (e.g., *including* and *following*). Predicates also include past participles used as adjectives, such as "*personalized* content." The objects are then the noun phrases in each identified predicate’s arguments. Similarly, `PurPliance` creates PO pairs whose predicates are empty and objects are the longest non-overlapping noun phrases extracted from the purpose clause by using a noun phrase extraction technique [140].

4.3.2.2 Purpose Taxonomy

We extracted uncompounded purpose clauses from a large collection of privacy policies and categorized them into semantically-similar groups to create a taxonomy of purposes. This process of creating a purpose taxonomy is different from data-object and entity ontologies [20] because privacy policies do not have subsumption expressions for purposes as commonly used for data types and entities, e.g., "personal information *includes* email address and name". First, from the privacy policy corpus, purpose clauses were extracted as described in Section 4.4. Purpose phrases with invalid prefixes (not beginning with "to", "for" or "in order to" + V) or empty PO pairs were filtered out. Uncompounded phrases were then created by concatenating the predicate and the object of each PO pair of the extracted purpose clauses. Finally, uncompounded purpose clauses with the number of occurrences greater than a threshold τ were selected to construct a taxonomy.

The uncompounded purpose clauses are grouped into semantically similar groups by using text clustering [206]. Each clause was converted into real-value vectors using *roberta-large-nli-stsb-mean-tokens*, a BERT-based sentence embedding model trained on semantic textual similarity datasets [258]. The vectors were grouped into γ clusters by K-means clustering [263]. The number of embedding groups was chosen heuristically by visualization using t-SNE [203] and by balancing the trade-off between granularity and complexity of the taxonomy.

We chose to use a small number of high-level groups to keep the taxonomy simple while still achieving the goal of detecting contradictions and inconsistencies. From 17k privacy policies, 392k uncompounded purpose clauses were extracted. 6,068 unique uncompounded purpose clauses were then selected using frequency threshold $\tau = 5$. This threshold was empirically chosen to remove noisy rare purpose clauses while shortening the t-SNE visualization time so that we can iteratively develop purpose-clusters without losing common purpose clauses. We conducted an iterative process of adjusting the number of classes and categorizing PO pairs to the selected classes until only a small number of PO pairs do not fit the taxonomy. $\gamma = 16$ was chosen for 15 clusters with a concrete purpose and 1 cluster with *Other* purpose. *Provide ad* and *Personalize ad* are separated for fine-grained classification. Providing ad indicates to only deliver, show, or provide advertising while personalizing ad indicates to customize, personalize, or tailor advertising. Since the purposes in the *Other* class are unrecognized purpose clauses, they do not have relationships (e.g., subsumption) with each other and are thus excluded from the consistency analysis.

Based on the economic activities of businesses [184], the γ low-level classes were further grouped into high-level categories: Production, Marketing, Legality, and Other categories. In the taxonomy, a low-level purpose is an instance of (i.e., has a subsumptive relationship with) the corresponding high-level purpose. For example, *Provide ad* is a

High-level	Low-level	Predicate Patterns	Object Patterns
Production	Provide service	provide, deliver	service, app, product
	Improve service	improve	
	Personalize service	personalize, customize	
		base	location service
	Develop service	track, detect	issue, bug
	Manage service	administer, manage	service, app, product
	Manage accounts	create, manage	account
	Process payments	process, complete	payment, transaction
	Security	detect, investigate, prevent	breach, fraud
		authenticate, verify	user, identity
Marketing	Customer communication	notify	user
		send	update
		resolve	inquiry
	Marketing analytics	analyze	usage, trend
	Promotion	send	promotion, reward
	Provide ad	provide, deliver	advertising, advertisement
	Personalize ad	personalize, target	
	General marketing		marketing
Legality	General legality	enforce	term, right
		comply	law
Other	Other purposes		

Table 4.3: Left half: high- and low-level purposes in the data usage purpose taxonomy; Right half: examples of patterns of the predicates and objects in purpose clauses.

Marketing purpose. In addition, we consider *Personalize ad* to be subsumed under *Provide ad* and *Personalize service*. If a service personalizes ads, then it provides ads, but not vice versa, because an ad can still be displayed without being personalized to the user's interest. The taxonomy is listed in Table 4.3's left half.

4.3.2.3 Data-Usage Purpose Classifier

`PurPliance` classifies purpose clauses by matching patterns of *n*-grams (e.g., words and bigrams) on lemmas of predicates and objects in the PO pairs of purpose clauses. We observe patterns that do not depend on the statement context, so matching such *n*-gram context-insensitive patterns provides precise classification. Moreover, `PurPliance` may classify one clause into multiple categories. For example, "provide personalized services" would be classified into *Provide service* and *Personalize service*.

To develop patterns and evaluate classification performance, we first extracted purpose clauses from all privacy policies and randomly divide them into training and test sets. 198,339 purpose clauses were extracted from our privacy policy corpus of 16.8k unique privacy policies. The training and test sets have 158,671 (80%) and 39,668 (20%) purpose clauses, respectively.

Patterns were developed on the training set which is disjoint from the test set. We randomly selected 1000 sentences in the training set and classified them until reaching a desirable coverage. The patterns covered 46% of the training set and 44% of the test set. The right half of Table 4.3 lists some example patterns on PO pairs.

To evaluate the classifier's precision, we randomly selected purpose clauses from the test set and classified them until each purpose class in the taxonomy contains at least 30 samples. The extracted purposes were then independently verified with the purpose taxonomy (Table 4.3) by two co-authors. Their disagreements were resolved via follow-up discussions. Of 510 randomly-selected samples in the test set, `PurPliance` achieved 97.8% precision on average. This high precision of the classifiers is due partly to the use of strict rule-based matching. The precision score of each purpose class is shown in Table B.3 (Appendix B.3).

4.4 Privacy Statement Extraction

4.4.1 Definition of Privacy Statement

Each sentence in a privacy policy is formalized as a privacy statement which has two components: *Data Collection* which is the transfer of data to a receiver and *Data Usage* which represents the usage of the data and its purpose.

Definition 4.4.1 (Privacy Statement). *A privacy statement is a pair (dc, du) where dc (du) represents data collection (usage). $dc = (r, c, d)$ denotes whether or not a receiver r collects ($c \in \{$collect, not_collect$\}$) a data object d. $du = (d, k, p)$ represents whether or not data d is used for ($k \in \{$for, not_for$\}$) an entity-sensitive data usage purpose p.*

The data usage can be a special *None* value when the statement does not specify any purpose for the data collection or `PurPliance` cannot extract the purpose from a sentence. While a privacy statement can be represented as a flat 5-tuple (r, c, d, k, p), we explicitly separate data collection *dc* from data usage *du* to distinguish the source of a contradiction which is either *dc* or *du*. Furthermore, our contradiction analysis can use hierarchical checking that has a smaller number of rules than that for the high-dimensional flat representation. Moreover, the 5-tuple representation also suffers from a large number of relationships between two tuples which increase exponentially with the number of tuple dimensions. Separating the data usage from data collection creates a constraint that if $c =$ not_collect, then *du* should be *None* because the data object *d* cannot be used without collecting it first.

Entity-Sensitive Data Usage Purposes. We define entity-sensitive data usage purposes as follows to capture the meaning of statements that mention whether the data is used for the purposes of the app itself or a third party. For example, "for third parties' own marketing purposes" is represented as a pair *(third party, marketing)*.

Definition 4.4.2 (Entity-Sensitive Data Usage Purpose). *An entity-sensitive purpose of data usage is a pair (e, q), where e is the entity whose purpose is served, called purpose-served entity, and q is a data usage purpose.*

As an example, "third parties do not collect device identifiers for their advertising purposes" will be translated into a statement *(dc=(third party, collect, device ID), du=(device ID, not_for, (third party, advertising))*. We assume third parties still collect device IDs but the data is not used for third parties' advertising purposes. Because of "their" word, we also assume the data serves third parties' purposes.

Compared to PolicyLint, `PurPliance` adds a new data usage representation *du*, uses a representation of data usage purpose and has a more complete interpretation of privacy sentences. While the *dc* component contains the same parameters as in PolicyLint, `PurPliance` uses a different interpretation of data collection in privacy policy sentences. Given the above sentence, PolicyLint creates *(third party, not_collect, device ID)* but it implies absolutely no collection of device IDs and would flag any other statements about the collection of a related data type.

4.4.2 Extraction of Statement Parameters

`PurPliance` extracts phrases that correspond to the parameters of privacy statements from a sentence in 3 steps: (1) identify data practice predicates (verbs), (2) extract the semantic arguments of each predicate and (3) map these arguments to the parameters.

Receiver Extraction. The receiver and sender of a data practice are determined by either *Arg0* or *Arg2*, depending on the action type (i.e., *collection*, *use*, or *sharing*). Since *Arg0* and *Arg2* are typically the actor and the beneficiary of an action, if the action is collection, *Arg0* is the receiver and *Arg2* is the sender of the data object. Similarly, these roles are swapped if the action is sharing. In the case of data-using actions, there is no sender and *Arg0* represents the entity that uses the data. The mapping from the arguments to the sender/receiver is shown in Table 4.2. The first or third party can also be mentioned implicitly in a sentence depending on the type of the data practice. For example, in "we will not share your sensitive data," the missing receiver is inferred as an *implicit third party* since the type of data action is *sharing*. When a verb is a clausal complement (i.e., its dependency is *xcomp*) and the receiver is an object pronoun, it would be converted to a subject pronoun. For example, in "you authorize us to collect your personal data to provide the services," Arg0 of *collect* is *us* which is then converted to *we*.

Rule	Extracted span*	Created privacy statements
T_1	*(r, not_collect, d, None)*	*((r, not_collect, d), None)*
T_2	*(r, not_collect, d, p)*	*((r, collect, d), (d, not_for, p))*
T_3	*(s, share, r, d, None)*	*((s, collect, d), None)* and *((r, collect, d), None)*
T_4	*(s, not_share, r, d, None)*	*((s, collect, d), None)* and *((r, not_collect, d), None)*
T_5	*(s, share, r, d, p)*	*((s, collect, d), None)* and *((r, collect, d), (d, for, p))*
T_6	*(s, not_share, r, d, p)*	*((s, collect, d), None)* and *((r, collect, d), (d, not_for, p))*

Table 4.4: Privacy statements created from extracted text spans. * *text span = (sender, action, receiver, data, purpose).*

Conversion from Extracted Spans to Privacy Statements. Privacy statements are generated from extracted spans by using transformation rules as listed in Table 4.4. Rules T_1 and T_2 convert spans with a collection verb while rules T_3–T_6 convert spans with a sharing verb. A rationale behind rules T_3–T_6 is that the data collection and sharing are observed only at the client side (i.e., the app), and hence the data collection or sharing on the server side is unknown. In particular, rule T_3 assumes the sender may have the data before sharing it. Similarly, rule T_4 means while the sender does not share data, it may still collect the data. Rules T_5 and T_6 mean the receiver may still collect data, but the data should not be used for the purpose p. For example, given "we do not provide your personal data to third parties for their own marketing purposes," we interpret this statement as personal data can be transferred to third-party service providers, but does not serve the third parties' purposes.

Action Sentiment Extraction. The sentiment of a data practice can be either positive or negated and indicates whether the data action is performed or not, respectively. The sentiment is determined by checking the presence of the negation argument *Argm-Neg*. If the predicate has no *Argm-Neg*, `PurPliance` analyzes its dependency tree to determine its negation using the method in PolicyLint [20]. For example, in "we never sell your data," *sell* has a negated sentiment because it has a negation argument *never*. However, the negation of *use*-verbs does not generate *not_collect* because "app does not use data A" does not mean the app does not collect data A.

Data Object Extraction. `PurPliance` extracts the text spans of the objects of the privacy practice actions using SRL and extracts data object noun phrases using NER [172]. First, argument *Arg1* is mapped to the Data component since it is the object of a verb across data practice action types. Second, the verb argument is then further refined by using NER which is a common technique used to extract data objects [20]. For example, given "we may use your name and street address for delivery", NER extracts "your name" and "street address" from the corresponding argument identified by SRL.

Purpose-Served Entity Extraction. Although it is more accurate to determine the purpose-served entities by performing co-reference resolution [172], `PurPliance` uses keyword matching to extract purpose-served entities. `PurPliance` leverages an observation that "their" commonly means third parties because data-practice statements in privacy policies are frequently between first-party/users and third parties, such as in the sentence "third parties may not use personal data for their own marketing purposes." Similarly, "our" in purpose clauses commonly refers to first parties. If no such entities were found, the purposed-served entity is set to "any party".

Purpose analysis allows more accurate interpretations of certain sharing statements. In particular, when user data is used for "monetary" or "profitable" purposes or when the data practice is to "lease", "rent", "sell" and "trade" the user's data, we interpret that the data is shared for a third party's purposes. To reduce false positives, we only include the *Marketing – Provide ad* purpose which is the most common. When an advertiser collects a data object for advertising purposes, the purpose-served entity is also set to the advertiser.

Exception Clauses. Given a sentence that includes an exception clause which does not contain data objects or entities, if the privacy statement extracted from the sentence has a negated sentiment, PurPliance changes the sentiment of the privacy statement to be positive. For example, "we do not share your personal data with third parties for their marketing purposes without your consent" produces text spans *(we, share, third party, your personal data, (their, marketing))*. This exception clause handling is similar to PolicyLint.

PurPliance generates additional privacy statements in certain cases when a sentence contains exceptions about purposes. If the sentence is negated and contains "other than [purpose clause]," the data will not be used for other high-level purposes. Excluding other high-level purposes which are semantically non-overlapping produces fewer false positives than excluding other low-level purposes.

Similarly, PurPliance creates opposite-sentiment privacy statements for other purposes if the sentence contains "for [purpose clause] only," or "only for [purpose clause]." PurPliance also excludes the data usage for the purposes of third parties given purpose-restrictive phrases such as "only for internal purposes." Although many third parties' purposes can be considered to be outside of "internal purposes", we exclude only *Marketing – Provide ad*, which is the most common, to reduce false positives.

4.5 Data Flow Extraction

4.5.1 Data Purpose Analysis

PurPliance re-implements MobiPurpose approaches [166] to infer the data types and usage purposes from mobile apps' network traffic (Sections 4.5.3 and 4.5.4). The data types and purposes of app–server communication are inferred from the content of the data sent to the server, the destination and the app description. While the semantics of data is vague, resource names (e.g., variables or server names) are assumed to clearly reflect their intentions [166, 275] as it is necessary for effective software engineering [211] and especially important for server names shared by multiple parties. The rationale of feature selection and system design is discussed at length in [166].

Since only the dataset of MobiPurpose [166] is available, we reproduce its inference and adapt the labels to the purposes in our purpose taxonomy. We assume the human annotators of MobiPurpose dataset correctly labeled the purposes of data, so the high agreement of ML with human annotators means that ML predicts the purposes of data with a high probability.

4.5.2 Data Flow Definition and Extraction

Definition 4.5.1 (Data Flow)**.** *A data flow is a 3-tuple* (r, d, p) *where a recipient r collects a data object d for an entity-sensitive purpose p and* $p = (e, q)$ *where e is the purpose-served entity and q is a data-usage purpose.*

App requests are commonly structured in key–value pairs [285], so each structured data sent to the server is decomposed into multiple key–value pairs $\{kv_l\}$. Therefore, each request or response between app app_i and end-point url_j corresponds to a set of low-level flows $F = \{f_k | f_k = (app_i, url_j, kv_l)\}$. The data type d, purpose-served entity e and usage purpose q are then inferred from F.

`PurPliance` distinguishes first and third parties to determine the purpose-served entity e by analyzing the receiver r and the inferred data-usage purposes q. For example, an app's "supporting" services like content delivery networks (CDNs) use data for the app's *first-party* purposes rather than for another party's. While there are many combinations of r (e.g., *First-party* or *Third-party*) and q (one of 5 purposes, Table 4.6), to avoid false positives, we conservatively set *e=Advertiser* only when *r=Advertiser* and *q=Provide ad* or *Personalize ad* (i.e., an advertiser uses collected data for its advertising purposes). For other cases, such as when an app uses "supporting" third-party services (i.e., *r=Third-party* and *q=Provide service*), data usage still serves the first-party's purposes, and hence we set *e=First-party*. The receivers of data flows are resolved by checking the data's destination URL with the package name, the privacy policy URL and well-known analytics/advertisement lists [4]. Note that purpose-served entity e is not supported by MobiPurpose.

Data Type	Precision	Recall	F1	Support
Identifiers	0.98	0.92	0.95	141
Geographical location	0.98	0.94	0.96	67
Device information	0.98	0.89	0.93	45
Network information	1.00	0.92	0.96	26
User profile	0.89	1.00	0.94	16
Average	0.97	0.93	0.95	59

Table 4.5: Data type extraction performance.

PurPliance uses dynamic analysis to exercise the apps and capture their network data traffic. It has 2 advantages over static analysis: (1) real (not just potential) execution, hence reducing false positives, and (2) destination of data, which can be determined dynamically on the server side.

By analyzing purposes of data flows, PurPliance can distinguish more fine-grained intentions of data usage than entity-only approaches like PoliCheck [21]. In particular, the 1st party can collect data for its own marketing purposes. For example, *Wego Flights* app sends a client ID to its own server at *srv.wego.com* with the request path $/analytics/visits$, so the collection of user ID can be inferred to be for the app's purpose of Marketing Analytics. The sent data's semantics is especially useful to distinguish data transfer to a business partner of the app which is not a popular advertisement network or analytic service provider.

4.5.3 Data Type Extraction

Using the corpus from MobiPurpose [166] which contains manually-annotated data types for key–value pairs of apps' network traffic, we identified patterns of the key-values for each data type. The corpus has 5 high-level data types (listed in Table 4.5) that are common in app data communication: identifiers (i.e., hard/software instance and advertising IDs), network information (e.g., types of network), device information (e.g., device types and configurations), location (e.g., GPS coordinates) and user account information (e.g., user name, password and demographics). MobiPurpose dataset does not distinguish types of ID (i.e., advertising, hardware and instance ID) which are frequently used by developers for overlapping purposes. The distinction can be achieved by finer-grained data type labels. However, developing such a dataset is beyond this paper's scope.

Data Type Features. There are 2 types of patterns: special strings and bag-of-words. The key–value strings are first matched by special-string patterns which comprise unigrams, bigrams, regular expressions and bags of words. If no match is found, an English Word Segmentation model [165] was used to segment the key–value pairs into separate words and construct a bag of words. For example, "sessionid" is separated into *session* and *id*. The occurrence of the word *id* indicates this is an identifier. These patterns become 6-component feature vectors where each component is whether there is any matched pattern or not. The last component is set to 1 if there is no matched pattern for the 5 data types. We tried 4 types of classifiers (Logistic Regression (LR), Multi-Layer Perceptron (MLP), Random Forest (RF) and Support Vector Machine (SVM)) to classify these features. The best-performing classifier is found to be Random Forest with 200 estimators.

Performance Evaluation. The corpus is randomly divided into a development set (80%) for developing string patterns and a test set (20%) for evaluating 5 data-type classifiers. We remove types with too few (i.e., less than 20) samples. The classifiers achieve 95% F1 score with 97% precision and 93% recall rates on average. The precision is more than 89% on all data types. The high accuracy indicates the regularity in key and values which were programmatically produced by the apps. The lowest recall is of the *device information* data type because the classifier misclassifies some samples which look like device IDs, such as *clientId: Huawei+Nexus+6P*, but are actually a device model. The detailed performance results of the classifiers are provided in Table 4.5.

Purpose Class	Prec.	Rec.	F1	Sup.
Production - Provide service	0.76	0.81	0.78	15
Production - Personalize service	0.85	0.66	0.74	18
Production - Security	0.81	0.73	0.77	16
Marketing - Provide ad	0.86	0.86	0.86	76
Marketing - Marketing analytics	0.77	0.85	0.81	72
Average	0.81	0.78	0.79	39

Table 4.6: Purpose prediction performance on the data flows in the test set. The total number of samples is 1413. The classifiers are tuned for the extraction precision. The metric columns are Precision/Recall/F1/Support in this order.

4.5.4 Data Traffic Purpose Inference

Data Usage Purpose Features. `PurPliance` uses the same features as those in MobiPurpose to predict the purposes of each key–value pair in the transferred data. There are 6 features in 3 groups based on the destination URL, sent data and app package name. The first group of features are based on the usage intention embedded in the semantics of the destination URL and sent data which have a form of $scheme://host/path$. The second feature group encodes the characteristics of the data types in the sent data such as the number of key-value pairs. The third feature group shows the relation between the app and the server. For example, the data sent to $cbc2015.prod1.sherpaserv.com/services$ by app $com.sherpa.cbc2015$ is likely to the app's server. They are encoded in 291-dimensional vectors (Table B.4 in Appendix B.5).

Purpose-Classification Dataset. The purpose classification models were trained on MobiPurpose corpus [166] which contains Android apps' network data traffic. We obtained a total of 1413 samples. The data types and purposes of each key–value pairs contain labels created by 3 experts. We aggregated purpose labels into a single purpose label by using majority votes, following the method in the original paper [166]. Specifically, a sample is classified as a purpose p as the most common label from the annotators. The dataset has 24 categories in MobiPurpose taxonomy. We manually mapped them to 7 classes in

PurPliance (as shown in Table B.6, Appendix B.7). The final 5 purpose classes are listed in Table 4.6.

Performance Evaluation. Similar to data type classification, we experimented 4 types of machine learning models: LR, MLP, RF and MLP. The MLP uses ReLU activation and Adam optimizer with a fixed learning rate of 10^{-5}. The Random Forest has 200 estimators. We used random search for the hidden layers of 2-layer MLP, regularization strength (C) of Linear Regression models with range 0.1–10 estimators (range 100–200) for Random Forest. The evaluation was done on the dataset using 10-fold cross validation. Similar to [166], we removed purpose classes that have too few (i.e., less than 20) samples, such as the Other purpose class.

The average F1 score is 79% (81% precision and 78% recall). *Provide ad* and *Marketing analytics* have the highest F1 scores of 86% and 81%, respectively. The lowest F1 score is of the *Personalize service* class since it is challenging to distinguish this class from other classes such as *Provide service*. The results are given in Table 4.6.

We perform an ablation study to evaluate the effectiveness of the features used to predict the purposes. The results (listed in Table B.5 in Appendix B.5) show that the type of the transferred data is the most effective feature that improves the F1 score by 4%. The number of key–value pairs also improves F1 by 2% since there is a correlation between this feature and the data purposes (e.g., analytic services often collect more key–value pairs (10+) than the rest [166]).

4.6 Consistency Analysis

This section formalizes the detection of purpose inconsistencies within privacy policies (called *policy contradictions*) as well as those between the policies and the actual data collection and sharing behavior of the corresponding apps (called *flow-policy inconsistencies*).

4.6.1 Semantic Relationships

Each parameter in a privacy statement is mapped to an ontology (e.g., data object ontology and purpose taxonomy) which defines relationships among the terms used. We extend the semantic equivalence, subsumptive relationship and semantic approximation of PoliCheck [21] to data-usage purposes as listed in Table 4.7. R_1 is defined in Definition 4.6.4, $R_2 - R_4$ are defined in Definition 4.6.5, and $R_5 - R_9$ are defined in Theorem 4.6.6 (proved in Appendix B.4).

Definition 4.6.1 (Semantic Equivalence)**.** *$x \equiv_o y$ means that x and y are synonyms, defined under an ontology o.*

Definition 4.6.2 (Subsumptive Relationship)**.** *Given an ontology o represented as a directed graph in which each node is a term and each edge points from a general term y to a specific term x included in y (i.e., x "is a" instance of y), $x \sqsubset_o y$ means there is a path from y to x and $x \not\equiv_o y$. Similarly, $x \sqsubseteq_o y \Leftrightarrow x \sqsubset_o y \vee x \equiv_o y$ and $x \sqsupset_o y \Leftrightarrow y \sqsubset_o x$.*

Definition 4.6.3 (Semantic Approximation)**.** *The semantic approximation relationship between two terms x and y, denoted as $x \approx_o y$, is true if and only if $\exists z$ such as $z \sqsubset_o x \wedge z \sqsubset_o y \wedge x \not\sqsubseteq_o y \wedge y \not\sqsubseteq_o x$.*

Definition 4.6.4 (Purpose Equivalence)**.** *Two data-usage purposes are semantically equivalent $(e_i, q_i) \equiv_\pi (e_j, q_j)$ if and only if there exist ontologies ε and κ such that $e_i \equiv_\varepsilon e_j \wedge q_i \equiv_\kappa q_j$.*

Definition 4.6.5 (Purpose Subsumption)**.** *$(e_i, q_i) \sqsubset_\pi (e_j, q_j)$ if and only if there exist ontologies ε and κ such that $e_i \sqsubset_\varepsilon e_j \wedge q_i \equiv_\kappa q_j$ or $e_i \equiv_\varepsilon e_j \wedge q_i \sqsubset_\kappa q_j$ or $e_i \sqsubset_\varepsilon e_j \wedge q_i \sqsubset_\kappa q_j$.*

Theorem 4.6.6 (Purpose Semantic Approximation)**.** *Given two data-usage purposes $p_i = (e_i, q_i)$ and $p_j = (e_j, q_j)$, there exist ontologies ε, κ, and π such that*

1. *$e_i \equiv_\varepsilon e_j \wedge q_i \approx_\kappa q_j \Rightarrow p_i \approx_\pi p_j$,*

Relation	$e_i \cdot e_j$	$q_i \cdot q_j$	$p_i \cdot p_j$
R_1	$e_i \equiv_\varepsilon e_j$	$q_i \equiv_\kappa q_j$	$p_i \equiv_\pi p_j$
R_2	$e_i \equiv_\varepsilon e_j$	$q_i \sqsubset_\kappa q_j$	$p_i \sqsubset_\pi p_j$
R_3	$e_i \sqsubset_\varepsilon e_j$	$q_i \equiv_\kappa q_j$	$p_i \sqsubset_\pi p_j$
R_4	$e_i \sqsubset_\varepsilon e_j$	$q_i \sqsubset_\kappa q_j$	$p_i \sqsubset_\pi p_j$
R_5	$e_i \equiv_\varepsilon e_j$	$q_i \approx_\kappa q_j$	$p_i \approx_\pi p_j$
R_6	$e_i \sqsubset_\varepsilon e_j$	$q_i \approx_\kappa q_j$	$p_i \approx_\pi p_j$
R_7	$e_i \approx_\varepsilon e_j$	$q_i \equiv_\kappa q_j$	$p_i \approx_\pi p_j$
R_8	$e_i \approx_\varepsilon e_j$	$q_i \sqsubset_\kappa q_j$	$p_i \approx_\pi p_j$
R_9	$e_i \approx_\varepsilon e_j$	$q_i \approx_\kappa q_j$	$p_i \approx_\pi p_j$

Table 4.7: Data-usage purpose relationships. $p_i = (e_i, q_i)$ and $p_j = (e_j, q_j)$. $\cdot$ denotes a relationship placeholder. $R_1 - R_4$ are definitions, $R_5 - R_9$ are theorems.

2. $e_i \sqsubset_\varepsilon e_j \wedge q_i \approx_\kappa q_j \Rightarrow p_i \approx_\pi p_j$,

3. $e_i \approx_\varepsilon e_j \wedge q_i \equiv_\kappa q_j \Rightarrow p_i \approx_\pi p_j$,

4. $e_i \approx_\varepsilon e_j \wedge q_i \sqsubset_\kappa q_j \Rightarrow p_i \approx_\pi p_j$, *and*

5. $e_i \approx_\varepsilon e_j \wedge q_i \approx_\kappa q_j \Rightarrow p_i \approx_\pi p_j$

4.6.2 Policy Contradictions

Definition 4.6.7 (Privacy Statement Contradiction)**.** *Two privacy statements $t_k = (dc_k, du_k)$ and $t_l = (dc_l, du_l)$ are said to contradict each other iff either dc_k contradicts dc_l or du_k contradicts du_l.*

PurPliance's consistency analysis comprises two steps. Using the Definition 4.6.7 of contradiction between two privacy statements, it checks the consistency of dc and du tuples in this order. The consistency of $dc_k = (r_k, c_k, d_k)$ and $dc_l = (r_l, c_l, d_l)$ is analyzed by a Data Collection consistency model. PurPliance leverages the PoliCheck consistency model in this analysis. However, the PoliCheck consistency model cannot check the two policy statements if both have a positive sentiment (i.e., $c_k = c_l =$ collect) or the two receivers do not have either a subsumptive or semantic approximation relationship. In such cases, since no contradiction was detected, PurPliance checks the consistency of data usage

Rule	Logic	Example
C_1	$d_k \equiv_\delta d_l \wedge p_m \equiv_\pi p_n$	(Device ID, k, Advertising) (Device ID, ¬k, Advertising)
C_2	$d_k \equiv_\delta d_l \wedge p_m \sqsubset_\pi p_n$	(Device ID, k, Advertising) (Device ID, ¬k, Marketing)
C_3	$d_k \sqsubset_\delta d_l \wedge p_m \equiv_\pi p_n$	(Device ID, k, Advertising) (Device info, ¬k, Advertising)
C_4	$d_k \sqsubset_\delta d_l \wedge p_m \sqsubset_\pi p_n$	(Device ID, k, Advertising) (Device info, ¬k, Marketing)
C_5	$d_k \sqsupset_\delta d_l \wedge p_m \sqsubset_\pi p_n$	(Device info, k, Advertising) (Device ID, ¬k, Marketing)
C_6	$d_k \equiv_\delta d_l \wedge p_m \approx_\pi p_n$	(Device ID, k, Advertising) (Device ID, ¬k, Personalization)
C_7	$d_k \sqsubset_\delta d_l \wedge p_m \approx_\pi p_n$	(Device ID, k, Advertising) (Device info, ¬k, Personalization)
C_8	$d_k \sqsupset_\delta d_l \wedge p_m \approx_\pi p_n$	(Device info, k, Advertising) (Device ID, ¬k, Personalization)
C_9	$d_k \approx_\delta d_l \wedge p_m \equiv_\pi p_n$	(Device ID, k, Advertising) (Tracking ID, ¬k, Advertising)
C_{10}	$d_k \approx_\delta d_l \wedge p_m \sqsubset_\pi p_n$	(Device ID, k, Advertising) (Tracking ID, ¬k, Marketing)
C_{11}	$d_k \approx_\delta d_l \wedge p_m \sqsupset_\pi p_n$	(Device ID, k, Marketing) (Tracking ID, ¬k, Advertising)
C_{12}	$d_k \approx_\delta d_l \wedge p_m \approx_\pi p_n$	(Device ID, k, Advertising) (Tracking ID, ¬k, Personalization)
N_1	$d_k \equiv_\delta d_l \wedge p_m \sqsupset_\pi p_n$	(Device ID, k, Marketing) (Device ID, ¬k, Advertising)
N_2	$d_k \sqsubset_\delta d_l \wedge p_m \sqsupset_\pi p_n$	(Device ID, k, Marketing) (Device info, ¬k, Advertising)
N_3	$d_k \sqsupset_\delta d_l \wedge p_m \equiv_\pi p_n$	(Device info, k, Advertising) (Device ID, ¬k, Advertising)
N_4	$d_k \sqsupset_\delta d_l \wedge p_m \sqsupset_\pi p_n$	(Device info, k, Marketing) (Device ID, ¬k, Advertising)

Table 4.8: Logical forms of logical contradictions (C) and narrowing definitions (N). k and $\neg k$ abbreviate *for* and *not_for*, respectively. The data flow has data type $f_d = \mathit{IMEI}$ and purpose $f_q = \mathit{Personalize\ ad}$.

statements du_k and du_l using a Data Usage consistency model. We extend the PoliCheck model [21] for data usage purposes as follows.

The contradiction conditions and types of two data usage tuples $du_k = (d_k, \mathit{for}, p_m)$ and $du_l = (d_l, \mathit{not\_for}, p_n)$ are listed in Table 4.8. There are 16 cases and 2 types of contradictions: *logical contradictions* (C_1–C_{12}) and *narrowing definitions* (N_1–N_4). Logical contradictions occur when du_l states the exclusion of a broader purpose from data usage while du_k states the usage for a purpose type in a narrower scope. On the other hand, narrowing definitions have the not-for-purpose statement (where $k = \mathit{not\_for}$) in a narrower scope than their counterparts. Narrowing definitions may confuse readers and automatic analysis when interpreting the privacy statements, especially when the two statements are far apart in a document.

	X does not collect Y	X collects Y
X does not collect Y for Z	Consistent	Consistent
X collects Y for Z	Contradictory	Consistent

Table 4.9: Privacy-statement comparison when one of the statement has no data usage purpose specified ($du = None$).

When two privacy statements are compared, if one of them has no data-usage purpose specified (i.e., du = $None$), PurPliance flags a contradiction only if they have forms $((r_k, not\_collect, d_k), None)$ and $((r_l, collect, d_l), (d_l, for, p_l))$, i.e., the positive-sentiment statement has $k_l = for$. Following this rule, "X does not collect Y" does not contradict "X does not collect Y for Z" as they are translated to *((X, not_collect, Y), None)* and *((X, collect, Y), (Y, not_for, Z))*, respectively. Table 4.9 lists the cases of this rule.

Example 2. Given two statements: "we use your personal data only for providing the App" and "advertisers may use your device ID to serve you with advertisements," a contradiction is detected as follows. Due to the keyword *only for*, PurPliance excludes third parties' *Marketing* purposes that are not for *providing* the app and translates the first sentence to 1 positive and 1 negated statement: s_1^1 = *(we, collect, personal data), (personal data, for, (anyone, Provide service))*, s_1^2 = *(third party, collect, personal data), (personal data, not_for, (third party, Marketing))*. The second sentence is translated to s_2 = *(advertiser, collect, device ID), (device ID, for, (advertiser, Provide ad))*. Since *device ID* $\sqsubset$ *personal data*, *advertiser* $\sqsubset$ *third party* and *Provide ad* $\sqsubset$ *Marketing*, the first sentence's negated statement s_1^2 contradicts s_2 of the second sentence under rule C_4. PolicyLint will not flag these sentences because it considers only the collection tuples which are all positive sentiments in these sentences.

4.6.3 Flow Consistency Analysis

Definition 4.6.8 (Flow-relevant Privacy Statements)**.** *A privacy statement* $t_f = ((r_t, c_t, d_t), (d_t, k_t, (e_t, q_t)))$ *is relevant to a flow* $f = (r, d, (e, q))$ *(denoted as* $t_f \simeq f$*) if* *and only if* $r \sqsubseteq_\rho r_t \wedge d \sqsubseteq_\delta d_t \wedge e \sqsubseteq_\varepsilon e_t \wedge q \sqsubseteq_\kappa q_t$. *Let* T_f *be the set of flow-*f*-relevant privacy statements in the set of privacy statements* T *of a privacy policy, then* $T_f = \{t \mid t \in T \wedge t \simeq f\}$.

Definition 4.6.9 (Flow-to-Policy Consistency)**.** *A flow* f *is said to be consistent with a privacy policy* T *iff* $\exists t \in T_f$ *such that* $c_t = collect \wedge k_t = for$ *and* $\nexists t \in T_f$ *such that* $c_t = not\_collect \vee k_t = not\_for$.

A data flow is inconsistent with a privacy policy if the Flow-to-Policy Consistency condition is not met. For each flow extracted from app behavior, `PurPliance` first finds the flow-relevant privacy statements T_f and classifies the flow as consistent or inconsistent using the above definitions. Although finer-grained consistency types can be used, such as Clear and Ambiguous disclosures as in PoliCheck, we leave it as future work. For brevity, the definitions only include cases where data-usage purposes are specified. The conditions on purposes are not checked if the data purpose is unspecified (i.e., *du=None*).

Example 1 creates a privacy statement *((third party, collect, personal_data), (personal_data, not_for, (third party, Marketing)))*. Transferring the user device IMEI number to an advertiser's server creates a data flow *f=(advertiser, IMEI, (advertiser, Provide ad))*. Because *IMEI* $\sqsubset$ *personal_data* (via *device_identifier*), *advertiser* $\sqsubset$ *third party*, and *Provide ad* $\sqsubset$ *Marketing* (relationship in the purpose taxonomy), the data flow is inconsistent with the privacy statement.

4.7 System Implementation

Semantic and Syntactic Analysis. `PurPliance` uses a neural SRL model [12, 273] trained on OntoNotes 5.0 [250, 251, 255], a large-scale corpus with 1.7M English words of news, conversations and weblogs and 300K proposition annotations. Each token is encoded into vectors depending on its context by using BERT-base-uncased contextualized word embeddings [83, 297]. Spacy with *en_core_web_lg* language model [100] was used for syntactic analysis and dependency parsing. Analyzing 16.8k privacy policies took 2 hours on 1 machine equipped with 2 Nvidia Titan Xp GPUs.

Data Object and Entity Ontologies. The consistency analysis logical rules require all entities and objects to be mapped into ontologies to check their subsumptive relationships. `PurPliance` extends the data object and entity ontologies based on PoliCheck to check their subsumptive relationship. Similar to the addition of SCoU verbs, we only add data objects and entities that are frequently used in data-practice statements to avoid noise from those used in unrelated sentences. `PurPliance` extracts data objects and entities by using a domain-adapted NER model trained on PolicyLint's dataset of 600 manually-annotated sentences (see Appendix B.8 for details).

Policy Crawler and Preprocessor. We developed a crawler and preprocessor to collect the privacy policies of Android apps. Its implementation is described in Appendix B.6.

Network Data Traffic Collection. `PurPliance` used a tool based on the VPN server API on Android [136] to capture apps' HTTP(S) traffic which is the most common protocol in app–server communication [102]. A system certificate was installed on rooted phones for capturing encrypted traffic. Each app was exercised with human-like inputs generated by deep-learning-based Humanoid [195], built atop Droidbot automation tool [194]. For each app, the experiment ran for at most 5 min and stopped if there was no traffic generated for more than 2 min. These timeouts were empirically determined for a good trade-off between data coverage and the number of apps that we want to explore. We used 5 smartphones with Android 8.

4.8 Evaluation

4.8.1 Data Collection

App Selection. We first selected the top 200 free apps for each of 35 categories on Google Play Store, excluding Android Wear and second-level Game categories [1]. This step resulted in 6,699 unique apps. Second, from a collection of 755,879 apps crawled from Google Play Store in May 2020, we randomly selected additional 28,301 apps that are different from the top apps in the first step and have been updated since 2015. To this end, 35k unique apps were selected. After removing apps with an invalid privacy policy, our final app corpus comprises 23,144 apps with a valid privacy policy.

Privacy Policy Corpus. We create a policy corpus as follows. We removed 6,182 duplicate policies from apps that share the same policy from the same developer. To reduce noise from titles (such as policy section titles), sentences with title-cased or all capitalized words or with less than 5 tokens are removed. Our final privacy policy corpus has 16,802 unique policies with 1.4M sentences. The categories with the most and least apps are Game (3,889 apps/2,797 policies) and Libraries & Demo (166 apps/121 policies), respectively. Fig. B.1 (Appendix B.9) shows their distribution over app categories.

Capturing Network Traffic. We capture the traffic of only the apps which have a valid policy to analyze the app-flow consistency. We intercepted 3,652,998 network requests of 18,689 apps over 33 days. Among those, we discarded traffic with empty-body requests or not from apps with valid policies and apps which became unavailable from Play Store at the time of testing. The final dataset has 1,727,001 network requests from 17,144 unique apps. The number of apps that generated traffic is lower than the selected apps because they either work offline or our automated input generation did not generate any input which triggered any requests to the servers, or the apps require login preventing our tool from using the service. These apps contacted 19,282 unique domains (164,096 unique end-point URLs) and sent 24,918,567 key-value pair data to remote servers. The distributions of network data requests across domains and app categories are described in Appendix B.10.

4.8.2 Privacy Statement and Flow Distributions

PurPliance extracts 874,287 privacy statements from 142,231 sentences in 15,312 policies (93.6% of 16,362 apps with data flows extracted). Of these, 225,718 (25.8%) statements from 43,421 (30.5%) sentences contain extracted purpose clauses. PurPliance recognized 112,652 privacy statements with a non-Other purpose class. The most common purposes are *Provide Service* and *Improve Service* which appear on 72.6% and 59.6% of the apps' policies, respectively. Fig. 4.2 shows the distribution of privacy statements' purposes.

Using the models developed in Section 4.5, 701,427 unique data flows from 16,362 apps were extracted. Each data flow comprises a single key–value pair in the captured traffic of each app. 432,078 (61.2%) have a non-Other purpose and 282,984 (40.3%) have both non-Other purpose and data type. The Other class is for data types or purposes which our classifier was unable to infer such information as a key–value of encrypted data. The most common data types are Device Information and Identifiers which appear in 95.7% and 87.3% of the apps, respectively. *Marketing Analytics* and *Provide Ad* are the most frequent purposes found in 94.1% and 78.7% of apps' data flows, respectively. These results indicate that apps commonly collect both identifiable and anonymous information of devices to deliver relevant advertisements and perform data analytics. The distributions of the purposes and data types are shown in Figs. 4.2 and 4.3, respectively.

There is a mismatch between the distribution of purposes of data flows and that of privacy statements. Although the most common data-flow purposes are advertising and marketing analytics that are present in more than 78.7% of the apps, these purposes are found in privacy statements of only 56.5% and 33.4% of the apps, respectively. The significantly lower presence of the purposes in privacy policies indicates that declarations of data-usage purposes for advertising and analytics are frequently omitted in apps' policies.

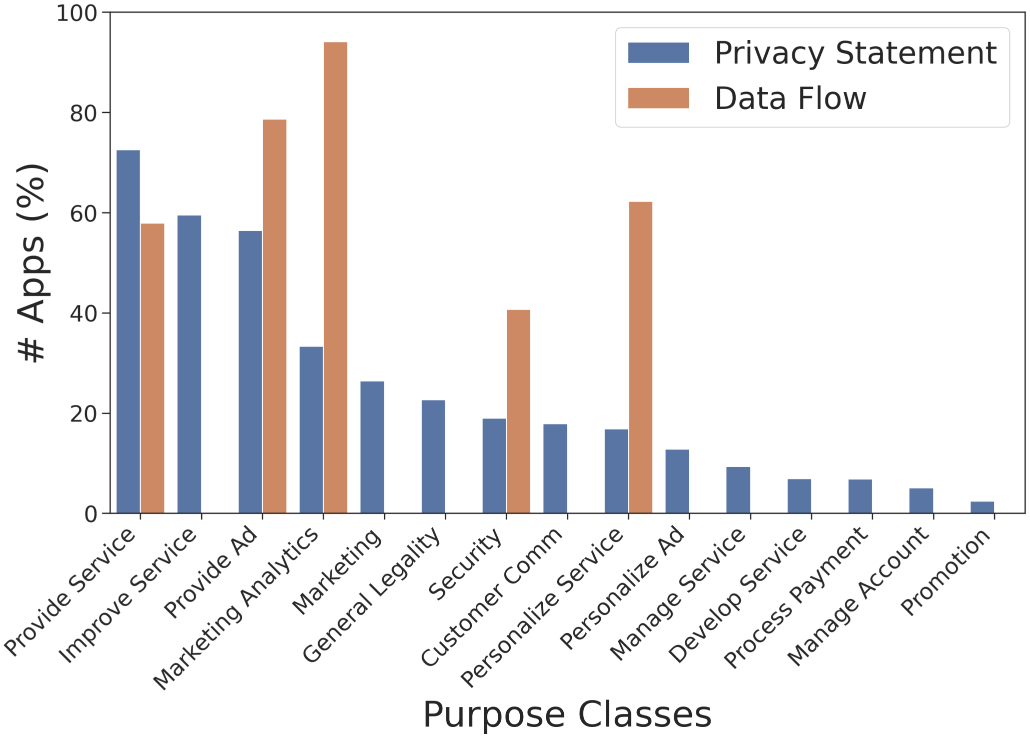


Figure 4.2: Distribution of purpose classes in the privacy statements and data flows of mobile apps.

4.8.3 End-to-end Detection of Contradictions

Evaluation Metrics. We evaluate `PurPliance`'s end-to-end detection of contradictory sentence pairs in privacy policies. Testing the performance at the sentence level assesses the usability of the system better than at the low-level privacy statement tuples. A human analyst would need to read whole sentences to understand the context of a detected contradiction so that s/he can verify and fix it. Therefore, a low false positive rate will help human analysts reduce their effort of reviewing many non-contradictory sentences.

Dataset Creation. We create a ground-truth dataset of 108 policies selected from the privacy policy corpus (Section 4.8.1). To increase the diversity of the policies, we select policies of apps with different levels of popularity as popular apps may have more resources to create their policies than less popular ones. In particular, we randomly select 36 apps in each of the 3 segments based on the number of app installs: greater than 1M (3,144 apps),

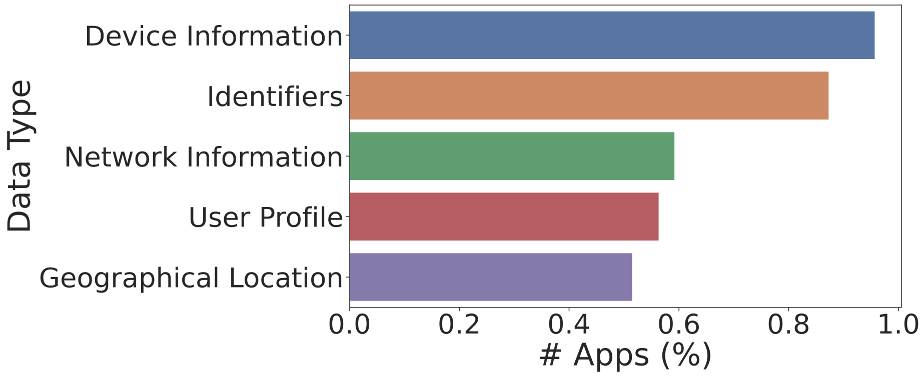


Figure 4.3: Distribution of data types in apps' data flows.

from 10k to 1M (10,482 apps), and less than 10k (3,176 apps). To have diverse document structures, we exclude similar policies created from templates. They are detected by a high TF-IDF cosine similarity [207] (greater than 0.95) and then manual verification that they have no significant differences other than the company/developer names. Documents that are not a valid privacy policy (e.g., terms of service or home pages), due to errors in data collection and pre-processing, are also excluded from the selection process.

Each privacy policy is independently annotated by 2 co-authors: an advanced PhD student and a researcher at a major global company, both with more than 3 years of experience in privacy research. We carefully read the policies and interpret the policy sentences as fully as possible to identify pairs of contradictory data-practice statements (detailed steps are described in Appendix B.11.1). Any disagreements were then resolved during follow-up discussions after every 10 policies were annotated. The annotation took two annotators 108 hours in total (30 minutes/policy/analyst on average).

There are 189 pairs of contradictory sentences in 47 (43.5%) policies. Of these policies, 32 (68.1%) contain 1–3, 12 (25.5%) contain 4–9, and 3 (6.4%) contain more than 9 sentence pairs. The dataset has 5,911 sentences where each policy has an average of 110.8 (96.4 standard deviation) sentences. The selected apps and their statistics are provided in Table B.7 (Appendix B.11.2).

Config	Precision	Recall	F1
PolicyLint	0.19	0.10	0.13
PolicyLint-PO	0.23	0.18	0.20
PurPliance-SRL	0.46	0.24	0.32
PurPliance-PA	0.60	0.43	0.50
PurPliance	0.95	0.50	0.65

Table 4.10: Detection of contradictory sentence pairs.

Experimental Configurations. To comparatively analyze the effects of the main components of PurPliance, we introduce the following configurations. PurPliance-PA is a *purpose-agnostic* version that does not extract purpose clauses and, thus, uses only non-purpose transformation rules T1, T3, T4 in Table 4.4. PurPliance-SRL is PurPliance-PA with PolicyLint's ontologies and data-practice verb list. Based on PolicyLint that uses the default parameters in its open-source repository [19], PolicyLint-PO leverages PurPliance's more complete SCoU verb list and data-object/entity ontologies.

Evaluation Results. PurPliance has 95% precision and 50% recall, which are significantly higher than 19% precision and 10% recall of PolicyLint. There are three main sources of PurPliance's improvements over PolicyLint. First, the semantic argument analysis improves the extraction of privacy statement tuples and increases both precision and recall so PurPliance-SRL improves F1 score from 20% to 32% compared to PolicyLint-PO. Second, a more complete data-practice verb list and data-object/entity ontologies improve the coverage of sentences so F1 of PurPliance-PA increases from 32% to 50% compared to PurPliance-SRL. The more complete verb list and ontologies also increase the performance of PolicyLint-PO from 13% to 20% F1 score compared to PolicyLint. Third, the analysis of data-usage purposes improves the detection of contradictions and increases the precision of PurPliance from 60% to 95% while recall is also enhanced from 43% to 50% compared to PurPliance-PA. The results are listed in Table 4.10.

Config	Precision	# Statements	# Sentences
PolicyLint	0.82	85	47
PurPliance	0.91	160	68

Table 4.11: Performance of privacy statement extraction.

The analysis of data-usage purposes improves PurPliance's F1 from 50% to 65% compared to PurPliance-PA. First, false positives are reduced because of the inclusion of purposes in interpreting sentences and more accurate interpretation of data-selling practices (e.g., *sell* and *rent*). For example, PurPliance does not flag sentences "we do not sell personal data" and "we may disclose personal data to comply with the law" because of different sharing purposes (*Marketing* vs. *Legality*), while purpose-agnostic approaches do. Second, the recall rate is enhanced because purpose-contradiction sentence pairs, such as "we use your personal data only for providing services" and "advertisers may collect personal data to deliver advertising", cannot be detected without purposes analysis.

The low precision of PolicyLint configuration is due mainly to the fundamental change of the interpretation of privacy statements. PolicyLint ignores purposes in statements and thus creates many false positives. For example, PolicyLint's interpretation of "we do not share your personal data for marketing" as "we do not share your personal data" contradicts many other data collection/sharing statements in the policy. Moreover, our metrics are more fine-grained than those used in PolicyLint [20], signifying the impact of PolicyLint's incorrect extraction. To characterize contradiction types in policies, PolicyLint [20] measured the accuracy of detecting contradictions between pairs of *sets of sentences* where sentences in a set generate the same privacy statement tuples. However, a sentence set may include both true-contradictory and false-positive ones.

The recall rate of PurPliance is still limited for three main reasons: complex sentences (29.5%), cross-sentence references (25.3%), and incompleteness of data-object ontologies (11.6%). In complex sentences, data-practice statements are often buried among other unrelated clauses (such as conditions and means of collection). The complex meaning of multiple clauses makes the separation of data-collection statements challenging. For example, "in the event of a corporate merger, your personal data is part of the transferred assets," implies the transfer of personal data without using any data-practice verb. In addition, the sentence-level analysis cannot resolve data types or entities that are defined in other sentences, such as "this information" in "we do not collect this information."

In-depth Analysis. We compare the performance of extracting privacy statement tuples, which is an important intermediate step of `PurPliance` and PolicyLint. As shown in Table 4.11, the results on 300 randomly selected sentences from the privacy policy corpus demonstrate that `PurPliance` significantly outperforms PolicyLint in extracting the privacy statement tuples. `PurPliance` has a 9% higher precision (increased from 82% to 91%), extracts 88% more privacy statements and covers 45% more sentences than PolicyLint. Note that the precision at this step is lower than the final contradictory detection because of further filtering in the later steps of contradiction analysis. The detailed experimental procedures are described in Appendix B.11.3.

4.8.4 Analysis of Policy Contradictions and Flow-to-Policy Inconsistencies

`PurPliance` detected 29,521 potentially contradictory sentence pairs in 3,049 (18.14%) privacy policies. Of these sentence pairs, 2,350 (7.97%) are purpose-specific, i.e., purpose-agnostic systems will miss them. For flow-to-policy inconsistencies, `PurPliance` detected 95,083 (13.56%) potentially inconsistent flows between the actual behavior and privacy policies in 11,399 (69.66%) of the apps with data flows extracted. Fig. 4.4 shows the distribution of the purpose-specific contradiction types.

The most common contradiction types are C_1 and N_1, indicating the problematic discussion of broad data-object and purpose terms in purpose-negated statements. For example, many apps state the collected personal data is not used for third parties' marketing purposes but also mention other contradicting usage purposes. The contradictions show that privacy policies frequently contain ambiguous descriptions of their data-usage purposes. Similarly,

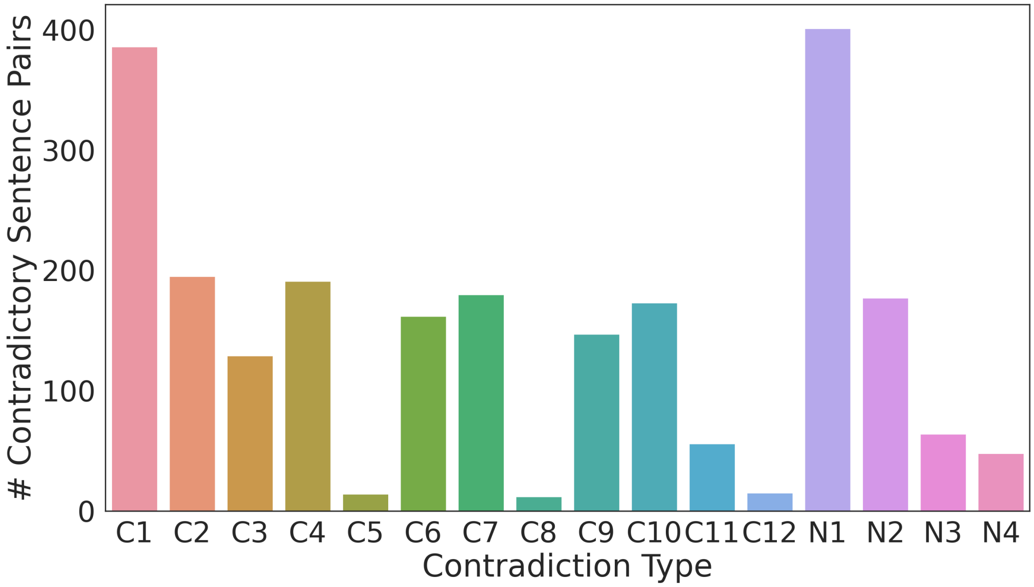


Figure 4.4: Distribution of potential purpose contradictions.

the high number of apps containing detected flow-to-policy inconsistencies indicates a prevalence of inconsistencies in mobile apps.

4.8.5 Findings

Finding 1. We found an issue with statements about the collection of personal data *for internal purposes only* in 28 apps, many of which have 100k-10M installs. Their policies state that "your Personal information we collected is used for internal purposes only." However, it contradicts with "we do not rent or sell your Personal information to third parties outside without your consent," because the exception clause "without your consent" indicates the sharing of personal data with third parties for third parties' purposes. On the other hand, the apps transferred a unique id and geographical location to a third-party domain with a path *client/v2/ads-service/ads*. Therefore, such data flows were inconsistent with the policies.

Finding 2. A common privacy policy template, used in 211 (0.92%) apps in our corpus, contains contradictory statements. The policy claims that their "agents and contractors may not use your personal data for their own marketing purposes." However, the policy states later that the app employs "3rd party ad serving systems" which "allow user data to be utilized for advertising communication purposes displayed in the form of banners and other advertisements on [app name] apps, possibly based on user interests." While ad-serving systems are one of their contractors, they use the personal data for their advertising purposes (which is subsumed under marketing purposes), and user data includes user personal data, hence these statements are contradictory with respect to the purposes of marketing and advertising.

Finding 3. Apps promise that the sharing is not for marketing but later say they will. For example, a popular education app with 10M+ installs states "we do not share your personal data with third parties or corporate affiliates for their direct marketing purposes." However, the policy also states "we allow our service providers (including analytics vendors and advertising networks) to collect information about your online activities through cookies. These third parties may use this information to display advertisements on our application and elsewhere online tailored to your interests." However, displaying targeted advertisements are direct marketing and online activities (such as browsing history) that can uniquely identify a person and can thus be considered as personal data [37]. Therefore, the latter statement is contradictory to the first statement of no direct marketing purpose.

4.9 Discussion

While PurPliance is designed to have low false positives with reasonable coverage, systematic evaluation of its recall rate is challenging because labeling privacy policies is very complex and expensive. SRL still remains a challenging task in NLP [142]. State-of-the-art SRL models [238] achieved only 87% F1 score with 85.5% recall rates. Furthermore, the SRL model used in PurPliance was trained on a generic dataset [255] and has not yet been adapted to the privacy-policy domain. Thus, its performance may be limited.

However, creating a domain-adapted SRL model requires a significant effort due to the complexity of the semantic arguments [255] and large model sizes [12]; this is part of our future inquiry.

PurPliance's extraction of data flows from network traffic has two limitations. First, it cannot decode certificate-pinned traffic which, however, constitutes only $< 5\%$ of the traffic generated by top free apps [166]. Second, the input generator used in PurPliance also cannot exercise login-required apps that use external verification information. Using advanced techniques to exercise certificate-pinned and login apps will improve the coverage of an app's execution paths, thus enhancing PurPliance's recall rate. For example, recently available TextExerciser [143] can be used to generate inputs for the analysis of apps requiring a login. Although PurPliance does not capture the traffic of certificate-pinned and login-required apps, this limitation does not increase false positives, that we aim to minimize. Therefore, we leave this as our future work.

Our analysis is based on client-side information only, so it has limitations in detecting the ultimate purpose of processing on the servers. Although the analysis assumes meaningful names of app resources such as package names and URL hosts/paths, they do not always reveal the true purposes of data flows, so the extraction cannot determine purposes of certain data flows (i.e., increase false negatives). However, predicting the purposes of app behavior still provides evidence of the presence of data-usage purposes which is useful for our goal of detecting inconsistencies.Determining the exact usage purposes of data requires knowledge of server-side processing since usage information is lost once the data is received by the servers. Therefore, the detection needs to be verified by humans such as regulators and service lawyers. Since the data purpose classification has already been discussed at length and evaluated in MobiPurpose [166], developing more sophisticated and accurate data-purpose extraction is beyond the scope of PurPliance.

4.10 Conclusion

We have presented a novel analysis of data purposes in privacy policies and the actual execution of mobile apps. We have developed `PurPliance`, a system for automatic detection of contradictions and inconsistencies in purposes between privacy policies and apps' data transfer. Our evaluation results have shown `PurPliance` to significantly outperform a state-of-the-art method and detect contradictions/inconsistencies in a large number of Android apps.

The work in this chapter appeared in the 2021 ACM Conference on Computer and Communications Security (CCS), and can be cited as [46].

CHAPTER V

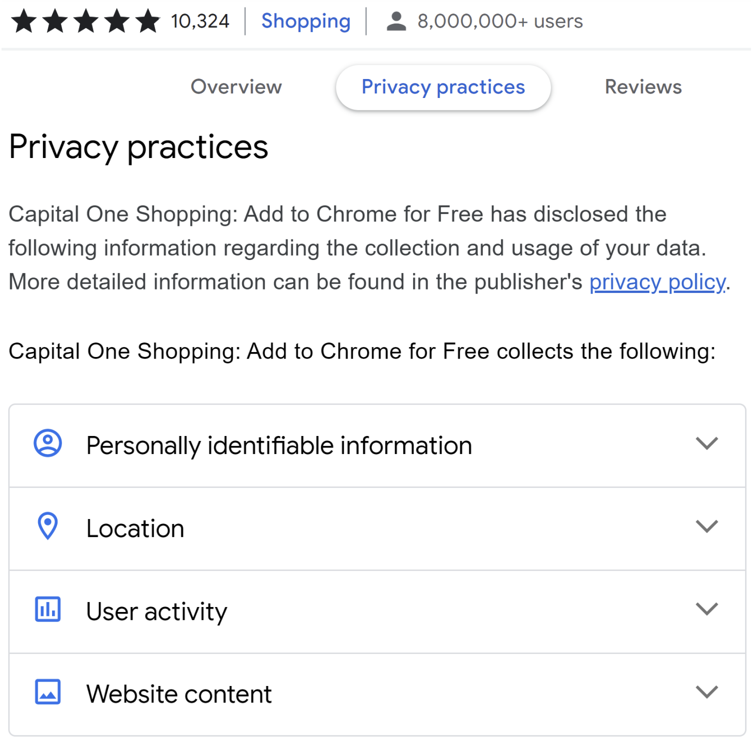
ExtPrivA

5.1 Introduction

While web browser extensions have been widely used to extend the functionality, and enrich user experience, of web browsers, they pose significant privacy risks to the users. Due to their integration with web browsers, extensions can collect highly sensitive data, such as personally identifiable information (PII) and any content that the users input to a web page [246]. These types of data can then be collected by the extensions themselves or transferred to unwanted/unknown/unauthorized third parties [80].

Major extension stores have strict requirements on extensions' privacy practices to reduce privacy risks for users [150, 154, 222]. For example, the Chrome Web Store requires extensions to provide privacy-practice disclosures via the developer Dashboard along with the privacy policies [156]. Fig. 5.1 shows an example of Dashboard privacy-practice disclosures.

Discrepancies between the different forms of privacy disclosures and extensions' behavior are considered to be a serious violation of the Store's developer program policies [157]: *"Any discrepancies between the developer dashboard disclosures, your privacy policy, and the behavior of your item would be a violation of the Chrome Web Store's developer program policies. This can result in the suspension of all the items owned by*



★★★★★ 10,324 | Shopping | 8,000,000+ users

Overview | Privacy practices | Reviews

Privacy practices

Capital One Shopping: Add to Chrome for Free has disclosed the following information regarding the collection and usage of your data. More detailed information can be found in the publisher's privacy policy.

Capital One Shopping: Add to Chrome for Free collects the following:

- Personally identifiable information
- Location
- User activity
- Website content

Figure 5.1: Dashboard privacy disclosures of a Chrome extension.

the publisher, deactivation of the existing user-base, and ban of the entire publisher entity (including related accounts)."

Because of the "non-discrepancy" requirements, data collection for potentially benign purposes may still violate an extension's privacy policy if the collected data is not disclosed in the policy. For example, if an extension claims *not to collect or use user data*, then it would violate the privacy disclosures even when the extension collects the users' location and keystrokes only for debugging and product-analytics purposes.

Prior work has largely overlooked the inconsistencies between web extensions' execution behavior and their stated privacy policies. Due to their lack/inability of determining the legitimacy of data transfer, prior policy-agnostic detection techniques [53, 175, 286] can only analyze common malicious leakage of user data. For example, they can only detect obvious malicious behavior (such as uninstalling other extensions [175]) or check whether the privacy leakage is either accidental or intentional [286].

The main question we aim to answer is: *Can we automatically detect the inconsistencies of the actual data collection of a browser extension with its stated privacy practices?* We propose, ExtPrivA, an end-to-end system that extracts the stated privacy practices and performs a *fine-grained analysis* of data flows to detect any inconsistencies between the actual data practices and the privacy disclosures of web browser extensions. Similar to software testing based on dynamic analysis [147, 148, 280], we aim to minimize false positives for (maximally) correct detection of inconsistencies. Specifically, ExtPrivA addresses the following 3 technical challenges:

TC1 – Detect contradictions of privacy statements of heterogeneous privacy disclosures. Checking the (in)consistency between privacy policies and actual data collection requires unambiguous interpretations of privacy disclosures, i.e., detecting any contradictions between the privacy statements. Analyzing different types of privacy disclosures poses a significant challenge due to the differences between the definitions of data types (i.e., *ontologies*) in different privacy disclosure forms. We derived a formal representation of privacy statements from the free-form privacy policies and template-based privacy disclosures specified via the Developer Dashboard (which we will henceforth call *Dashboard disclosures*). Finally, based on the extension Store's data-type specifications, we derived a unified ontology to leverage a state-of-the-art privacy-analysis technique [46] to detect contradictions between the privacy policies and Dashboard disclosures.

TC2 – Extract actual data collection from extensions' behavior. Since extensions do not automatically execute their functionality while their data traffic only contains low-level key–values, ExtPrivA triggered an extension's functionality and inferred data types from its data traffic to extract its actual data-collection practices. ExtPrivA emulated user interactions on a combination of real-world web pages and a *honeypage* to elicit the behavior of extensions that generated data traffic from the extensions to external servers. We developed a *request-initiator analysis* to isolate the data traffic initiated by extensions.

Finally, ExtPrivA extracted the data types by analyzing the key–value pairs in the HTTP(S) requests' body and URL query strings.

TC3 – Detect flow-to-policy inconsistencies. Since data flows and privacy statements are expressed in different semantic granularities (i.e., low and high level), it is challenging to analyze their relationship and check the (in)consistency. From the extension Store's developer policies, ExtPrivA extracts a data-object ontology used in the privacy-practice disclosures to analyze the relationship between data types in the flows and privacy statements. Finally, we establish the consistency conditions between the data flows and the privacy statements represented in a formal model to detect their inconsistencies.

We evaluate the accuracy of ExtPrivA in extracting privacy statement or data flow, and end-to-end detection performance via the manual verification of two annotators. Our result shows a precision of higher than 90% in the intermediate extraction and an end-to-end detection precision of 85%.

ExtPrivA is used to analyze the (in)consistencies of the privacy disclosures and data-collection behavior of 47,207 extensions that provide Dashboard disclosures on the Chrome Web Store. It identified 525 contradictions in the Dashboard disclosures and privacy policies of 360 extensions which made their privacy disclosures ambiguous. Finally, we found 820 extensions with 84.6M users experiencing 1,290 data flows that are inconsistent with their Dashboard disclosures.

This paper makes the following main contributions.

- A novel fine-grained analysis that detects the inconsistencies between a web browser extension's actual data practices and its privacy disclosures. The analysis also identifies ambiguities in the privacy disclosures by detecting the contradictory privacy statements between free-form privacy policies and template-based Dashboard disclosures.

- An end-to-end automated framework, called ExtPrivA, that analyzes flow-to-policy (in)-consistencies of browser extensions. It extracts privacy statements from the disclosed

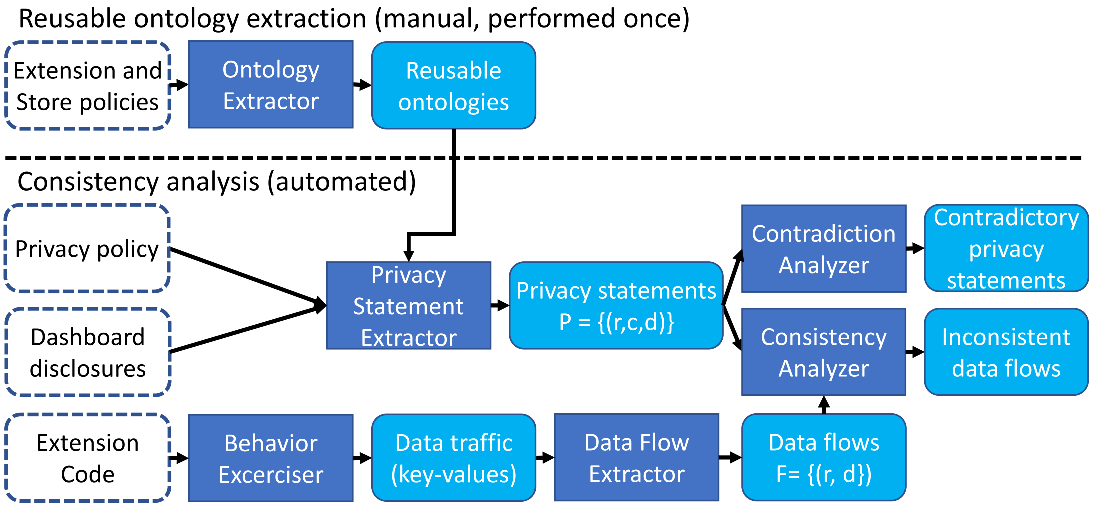

Reusable ontology extraction (manual, performed once)
Extension and Store policies
Ontology Extractor
Reusable ontologies
Consistency analysis (automated)
Privacy policy
Dashboard disclosures
Privacy Statement Extractor
Privacy statements P = {(r,c,d)}
Contradiction Analyzer
Contradictory privacy statements
Consistency Analyzer
Inconsistent data flows
Extension Code
Behavior Excerciser
Data traffic (key-values)
Data Flow Extractor
Data flows F= {(r, d})

Figure 5.2: `ExtPrivA` analysis pipeline.

privacy practices (Section 5.4), performs dynamic analysis to extract data traffic (Section 5.5), and extracts data flows from the transferred key-values (Section 5.6). Finally, the system detects contradictory privacy statements and inconsistencies between the data flows and the privacy statements by using a formal model (Section 5.7). Fig. 5.2 shows the analysis pipeline. Our evaluation demonstrates that `ExtPrivA` detects contradictory statements at a 91.7% precision, and detects flow-to-policy inconsistencies at an 85% precision.

- A large-scale study of 47.2k extensions on the Chrome Web Store (Section 5.9). Despite the strict vetting process of the Store, we still found a large number of extensions that had contradictory statements and flow-to-policy inconsistencies in their privacy disclosures and data-collection behavior, posing high privacy risks to millions of users. This finding highlights the critical issues of privacy-practice disclosures of browser extensions in practice.

5.2 Related Work

5.2.1 Detection of Privacy Leakage

Researchers proposed various ways of examining the execution of web browser extensions to detect their privacy leakage, i.e., unexpected execution of privileged API and the flows of sensitive data to servers or disk storage. Hulk [175] introduced two ways to trigger malicious behavior, specially structured web pages called *honeypages*, and an event-handler triggering fuzzer, to detect affiliated fraud, credential theft, ad injection or replacement, and social network abuse. Starov *et al.* [286] analyzed and decoded network traffic of extensions to find the leakage of users' sensitive data, such as browsing history and search-engine queries. However, they did not determine whether such data collection violates the extensions' privacy policies or not.

Other researchers focus on JavaScript analysis techniques. ExtensionGuard [51], Mystique [53] and JTaint [307] presented JavaScript taint analysis schemes to detect the leakage of sensitive information in the data flows during the extension execution. Somé [282] analyzed the vulnerabilities in message passing interfaces of web extensions that can be exploited by web applications to access privileged browser APIs and sensitive user information. DoubleX [25] developed static analysis techniques to detect vulnerable internal data flows of an extension which can be exploited by attackers.

Another thread of research developed ML-based classifiers to classify whether a certain extension behavior is malicious or not. Aggarwal *et al.* [5] created a classifier based on Recurrent Neural Network (RNN) to classify whether a sequence of API calls indicate the stealing of sensitive user information or not. Zhao *et al.* [312, 313] attempted to determine legitimacy of extensions' data flows based on their main functionality provided by the extensions.

None of the prior studies has analyzed the privacy policies of browser extensions to detect flow-to-policy inconsistencies. They rely on an expert analysis of the execution and network logs to determine whether the detected sensitive data transfer is malicious or not [53], but such a manual analysis greatly limits the scalability of the detection and types of the data leakage that can be detected. Furthermore, prior network-traffic-based analyses [175, 286] did not consider the receivers of the data traffic of web browser extensions, and hence may suffer from false positives where an extension sends user data to its servers to provide its functionality, not for malicious purposes. `ExtPrivA` avoids these limitations by analyzing privacy-practice disclosures and extracting data types/receivers in data traffic of extensions to determine the legitimacy of their data practices.

5.2.2 Analysis of Privacy Statements

Recently, researchers analyzed the statements in privacy policies of mobile apps and online services. PI-Extract [45] extracted fine-grained data types and collection/sharing actions performed thereon in website privacy policies. PolicyLint [20] detected contradictory policy statements in privacy policies of mobile apps. However, none of these addressed the privacy statements and policies of browser extensions that require platform-specific interpretation and analysis.

5.2.3 Flow-to-policy Consistency Analysis

Inconsistencies between the privacy policies and the actual data collection of mobile apps have been the subject of recent research. Zimmeck *et al.* [316] conducted a static analysis on Android apps to detect inconsistencies between their collected data types with those stated in the apps' policies. PoliCheck [21] improved the consistency analysis by considering the receivers in data flows from mobile apps to external receivers. PurPliance [46] modeled data-usage purposes to detect the inconsistencies in the data-usage purposes between the stated privacy statements and actual data collection of Android apps. However, none of the prior works have analyzed the flow-to-policy inconsistencies of browser extensions whose execution model is fundamentally different from mobile apps.

5.3 Background

5.3.1 Extension-Platform Privacy Requirements

In addition to privacy policies, major extension stores require extensions to provide easy-to-read disclosures of their privacy practices to users. In particular, Google has required developers to declare the types of data their extensions collected via the developer dashboard since January 2021 [156]. The privacy policies are free-form documents while the Dashboard disclosures are based on a common template that share among all extensions on the Store. Developers must also certify that they follow the Limited Use policy under which the transfer of user data to ad platforms, or for personalized advertising, is prohibited [154]. The Dashboard disclosure form is shown in Fig. C.1 (Appendix C.3). In this paper, we use the term *privacy policy* to distinguish the free-form policy documents from the template-based *Dashboard privacy-practice disclosures* of the Chrome Web Store.

5.3.2 Extension Architecture

Browser Selection. Since Google Chrome has been the most popular browser and the architecture of Chrome extensions has been adopted by other major browsers, we select Chrome as the representative extension architecture for further examination. At the time of this writing, Google Chrome constitutes 67% and Chrome-based browsers (Chrome, Edge, and Opera) constitute 79% of the desktop browser market share [287], a significantly larger share compared to other browsers (e.g., Firefox and Safari have less than 10% each). Furthermore, the Firefox and Safari browsers have adopted a cross-browser extension API, called *WebExtensions* [185, 227]. Therefore, the design principles of our analysis pipeline should apply to other non-Chrome browsers.

Main Components of Extensions. A Chrome browser extension comprises of four main executable components: background scripts (or *background pages*), content scripts, web-accessible resources (WARs) and pop-up pages. A JSON manifest file declares the components and how they are executed with respect to a web page. Background scripts (Manifest V2) and service workers (Manifest V3) run in a "background" (i.e., hidden from users without any UI) environment which has a lifetime independent of other user-facing web pages and can access privileged Chrome extension APIs. The scripts are executed asynchronously to handle events generated by other components. Content scripts are injected into a web page so they can read and modify the DOM tree which is inaccessible by the background pages [55]. The content scripts are executed at the start or end of loading the host web page. On the other hand, the JavaScript included in the WAR resources is loaded and runs in the same context with the host pages. Pop-up pages execute upon a user's click on the extension icon to interact with users. The components in an extension can communicate with each other and with the host web page via message passing.

Extension Identification. An extension in a browser is uniquely identified by an extension ID that can be used to access the resources included in the extension's package. Resources of an extension, such as its content scripts, have URL prefixed with *chrome-extension://<extension-id>*. The ID is generated randomly when the extension is loaded to the browser but can be made to be a fixed value by specifying a *key* value in the extension manifest [127].

Execution Entries. Since extensions are event-driven applications that have multiple entry points to trigger their functionality, the manifest file provides extensions with a means to statically declare their static entry points [128]. An extension can specify the URL patterns where the content scripts are executed when loading a web page from a pattern-matched URL. In addition, extensions can declare the event handlers for extension actions (i.e., mouse clicks on the extension icon on the browser menu bar) and the activation of the extension's context-menu items. Furthermore, an extension can declare the URL-match patterns on which it has effect. These URL match patterns are included in the *host permissions* and *web accessible resources* to restrict the web pages to which the extensions have access and the pages which can access the resources (e.g., JavaScript and CSS) included in the extension package, respectively.

Restriction of Network Access. To enhance the browser's security, only background scripts and pop-up pages bypass the same cross-origin resource sharing (CORS) policy and can send information to any servers without any restriction. By contrast, content scripts are subject to the CORS policy of the host web pages [123]. Unless the server side allows CORS requests, content scripts cannot directly request resources or send information to an arbitrary external server other than the origin of the currently visiting URL.

5.4 Analysis of Privacy-Practice Disclosures

The privacy statements of an extension comprises the statements from template-based Dashboard disclosures and free-form privacy policies. In this section, we describe the analysis and extraction of formal privacy statements from these two forms of privacy disclosures.

5.4.1 Privacy Statement Definition

To simplify the analysis of privacy policies, we limit the analysis to the statements about data collection, i.e., whether a receiver r collects a data type d or not, as formally defined next. Since privacy-practice disclosures specify a fixed policy for data-usage purposes, called a *Limited Use* policy [157], analysis on data-usage purposes can be done separately.

Definition 5.4.1 (Privacy Statements)**.** *A privacy statement is a tuple* $s = (r, c, d)$ *where* r *is a receiver that collects or does not collect (*$c \in \{collect, not\_collect\}$*) a data type* d*.*

5.4.2 Analysis of Dashboard Disclosures

We extract privacy statements from an extension's Dashboard privacy-practice disclosures as follows. Let $D = \{d_i\}$ be the set of data types that the extension declares to collect and T be the set of all possible data types that an extension can declare. We assume that if an extension does not declare its collection of a data type $d_i \in T$, then it will not collect d_i. The set of data types U not collected by the extension is then derived by excluding the stated data types from T: $U = T \setminus D = \{d'_i | d'_i \in T \wedge d'_i \notin D\}$. From D and U, the following privacy statements will be created: $S = \{(r, collect, d_i) | d_i \in D\} \cup \{(r, not\_collect, d'_i) | d'_i \in U)\}$. For example, given the Dashboard disclosures in Fig. 5.1, where the extension states that it collects only PII, Location, User Activity, and Website Content, the corresponding privacy statements are $S = S_c \cup S_n$ where $S_c = \{(extension, collect, d_i) | d_i \in \{PII, location, user\_activity, site\_content\}\}$ and $S_n = \{(extension, not\_collect, d'_i) | d'_i \in T \setminus \{PII, location, user\_activity, site\_content\}\}$.

At the time of writing this paper, the Chrome Web Store specifies a total of 9 data types that an extension can declare [156]. Therefore, `ExtPrivA` extracts a total of $|T| = 9$ privacy statements for each extension which comprise $|D|$ positive-sentiment statements for the declared data types D and $9 - |D|$ negative-sentiment statements for the undeclared data types in the extension's privacy-practice disclosures. Since the privacy-practice disclosures follow fixed declaration templates, D is extracted from the privacy-practice disclosures using regular expressions. The data types specified by the Chrome extension's policies are listed in Table C.2 and Fig. C.1 (Appendix C.3).

5.4.3 Analysis of Free-form Privacy Policies

Privacy Statement Extraction. Given the privacy policy of an extension, `ExtPrivA` adopts `PurPliance` [46], a state-of-the-art privacy-policy analysis technique, to extract privacy statements from the sentences in the document. For each sentence, the parameters of privacy statements (data type, collection action and receiver) are determined by an NLP pipeline. The system first identifies the sharing-collection-and-use verbs in the sentence, and then uses the semantic role labeling to extract the semantic arguments (e.g., subjects and objects) of each verb. The data types are extracted from the verbs' objects by a named entity recognition (NER) model.

Since the NLP pipeline was originally designed for Android apps, `ExtPrivA` addresses the following challenges to handle the differences between the privacy policies of Chrome extensions and Android apps.

Extension-Scope of Privacy Statements. While Android apps typically have dedicated privacy policies, many browser extensions are found to use generic privacy policies that cover the data practices shared by the web services developed by the same developer. For example, the extension policies frequently contain both privacy statements about the websites and the extensions. However, the current sentence-based privacy-policy analysis techniques [20, 46] cannot distinguish the scope of each statement (i.e., whether the statement is about the website or the extension) owing to the lack of a holistic whole-document analysis. Therefore, we exclude statements that do not mention extensions to reduce false positives. In particular, we include only the sentences that contain the keyword "extension".

Extension Data Ontologies. Since the data-type ontologies modeling data types and their relationship of Android apps are different from the relationships of browser extensions, we augment them with the high- and low-level data types of the Web Store (Table C.2). Similar to the domain adaptation [193] in NLP, this addition is necessary because privacy policies of Android apps do not include certain extensions-specific data types, such as Website Content and Web History.

5.5 Analysis of Extension Execution

`ExtPrivA` analyzes the data collection of an extension in three main steps: 1) exercise the extension functionality, 2) extract the network traffic initiated by the extension, and 3) extract the data flows that comprise the receivers & data types from the raw data traffic.

We describe the first two steps in the rest of this section while presenting the last step in Section 5.6.

5.5.1 Triggering Extension Functionality

5.5.1.1 Candidate URL Extraction

Since extensions do not have access to all websites by default, `ExtPrivA` first identifies the URLs of the websites that an extension has access to. To generate these URLs, `ExtPrivA` analyzes the extension manifest and extracts the URLs patterns for the background scripts, content scripts and WAR resources declared in the *host_permissions* and *matches* keys in the manifest.

`ExtPrivA` generates a set of candidate URLs from each URL pattern. A pattern is first decomposed into 4 components, following the manifest format [125]: scheme, subdomain, domain, and path. `ExtPrivA` then synthesizes the candidate URLs that match the specified URL patterns by substituting wildcard components with common valid values such as *www* for a subdomain. For example, from *https://*.example.com/subpath/**, a candidate URL *https://www.example.com/subpath/* is generated. Inspired by Hulk [175], for those patterns that match unspecified domains and paths such as <*all_urls*> and *https://*/**, `ExtPrivA` selects top website domains in two categories, search and shopping, on which extensions commonly execute from the Tranco list [189]. These URLs are listed in Table C.1.

5.5.1.2 Test pages

To test the extensions, we use two types of web pages: real pages and a *honeypage*. The former is real-world web pages that are served either from the Internet and a web-page replay server. In contrast, the latter is a specially-crafted web page that is based on prior extension analysis work [307] and contains various HTML elements to trigger common functionality of extensions. Real pages are useful for extensions that execute based on the structure of websites where the *honeypage* cannot be replicated. For example, *honeypage* cannot emulate real complex websites like Amazon shopping pages.

5.5.1.3 User Interaction Emulation

To trigger the data-collection functionality of extensions that operate upon user activities, we design interaction templates based on the browser, mouse and keyboard actions. These actions are the main user-interaction categories expected by extensions to execute their functionality [281]. We re-implement [281] and for each template, we add further customization for each web page. For example, a different element selector is used depending on whether the browser is accessing a Google search result page or an Amazon product page. `ExtPrivA` performs the following templates after the web page is fully loaded:

Text Selection and Mouse Actions. To elicit potential data collection on the text selected on a web page, `ExtPrivA` selects a word and activates the extension via the extension icons on the menu bar and/or the context menu. `ExtPrivA` performs mouse scrolling and clicking to select the text to trigger extensions that operate upon mouse events (like clicks and double-clicks). For example, a dictionary extension shows the definition of a selected word after the user selects the word, clicks the right mouse button and selects the extension icon on the context menu. When the web page is the *honeypage* or a replayed page, the text of a fixed element is selected. Otherwise, `ExtPrivA` selects the first word of the <*body*> element that is expected to exist on any web page.

This interaction template already includes a click on the extension icon on the browser menu bar. This interaction is called *extension action* (Manifest V3) or *browser action* (Manifest V2) and is one of the main entry points that trigger the functionality of extensions. For example, a shopping assistant shows the information of products on *amazon.com* only when the user clicks on its menu-bar icon.

Keyboard Input and Form Submission. To trigger functionality of extensions that monitor keyboard events, ExtPrivA inputs a keyword into a form field on the *honeypage*, issues a copy command via *ctrl+c* and submits the input form to our server endpoint. For example, spelling checkers may monitor keyboard typing and suggest correction. Inspired by the *bait* technique [2], ExtPrivA inputs a special value, called *bait*, that is used to detect the extension's collection of keyboard input.

Interaction with New Tab Pages. To trigger the extensions that provide a customized new tab page, ExtPrivA opens a new tab and types in a keyword. For example, *Infinity New Tab* extension opens a customizable tab that lets users enter a search term and shows current weather forecast. This kind of extensions may collect user location and/or search terms without the user's awareness.

5.5.2 Data Traffic and Initiator Analysis

It is challenging to extract the data traffic originated from an extension because the HTTP requests sent from the browser do not differentiate between those sent by extensions and those sent by web pages. Therefore, ExtPrivA extracts extensions' data traffic by analyzing the request-initiator scripts and the HTTP Origin header as follows.

First, ExtPrivA leverages the call-stack information of script-initiated network requests provided by the network-activity inspection of Chrome's DevTools. In DevTools, an initiator of a network request can be one of 6 types such as a JavaScript script or HTML parser [81]. An extension's script, like other extension's resources, has its URL in the form of *chrome-extension://<extension-id>/path-to-script* (identifying extension IDs is described in Section 5.8). Since the initiator information of a script contains call frames including the URLs of the initiated scripts in the call stack, if the script URL is prefixed by a *chrome-extension* scheme and matches the extension ID, the traffic is initiated by the extension. For content scripts which execute in the web pages' contexts, DevTools captures content scripts' requests, and indicates *chrome-extension://* initiators for content scripts but not for injected inline-scripts.

Second, ExtPrivA utilizes the *Origin* HTTP request header which is non-programmatically modifiable to indicate the security contexts that cause the browser to initiate an HTTP request [226]. This header is set to *chrome-extensions://<extension-id>* if the request is initiated by an extension. External scripts in the background or pop-up pages of the extension do not have URLs with the chrome-extension scheme, and thus cannot be identified by using the call stacks in the script initiators. Using the Origin header can identify the requests initiated by such embedded external scripts.

5.5.3 Extraction of Key–Value Pairs

ExtPrivA parses HTTP requests in the extension traffic intercepted in the prior step into key–value pairs since structured responses are widely used by web services [111]. The key–values are extracted from the sent cookies, URL query strings and request bodies of HTTP POST messages. In our dataset, while most of the traffic is plaintext, when encountering encoded traffic, ExtPrivA attempts to decode the data by using multiple rounds of Base64 decoding. This decoding is based on the technique used by Starov *et al.* [286].

Unlike automatic data such as IP addresses as part of the IP protocol or information in the HTTP security headers, sending data via URL parameters or the POST body requires a significant effort to obtain and set the values correctly. In particular, obtaining and adding personal data to URL parameters require developers' effort, unlike the IP addresses that browsers automatically set. Therefore, the occurrences of these values in the transferred data are unlikely created accidentally by the extension developers. To further reduce false positives of unintentional data leakage, we exclude key–value pairs in HTTP headers (other than the Cookie header) because the headers may include information automatically set by the browser rather than intentionally set by the extension.

We filtered out key-values sent to the servers that had the same hostname with the currently visited web page because the web page had already collected the user's data. For example, if a user visits host *H=www.example.com*, we exclude extensions' traffic to *H*. It was unclear whether the extensions were leaking data because the user already shared data with *H*. However, filtering same-host traffic creates no false positives and excludes only 0.81% (381/47,207) of extensions.

5.6 Data Flows

5.6.1 Data Flow Definition

Given the data traffic of an extension collected in Section 5.5, `ExtPrivA` extracts data flows that formally represent the data-collection behavior of the extension. A data flow is formalized in the following definition.

Definition 5.6.1 (Data Flow)**.** *A data flow is a tuple $f = (r, d)$ where a receiver r receives a data object d.*

5.6.2 Extraction of Data Flows

5.6.2.1 Extraction of Data Types

We select data types and design a rule-based extractor as follows.

Data-Type Selection. Of the 9 data types of the Store, we choose to extract 4 context-free data types whose meanings do not depend on their usage contexts: Website Content, Web History, Location and User Activity. It is challenging to extract context-dependent data types because the lack of the server-side information makes it practically impossible to determine the ultimate usage purposes of the collected data. For example, given a form on a web page, a user can input any content such as his/her first name (PII) or a generic search term (non-PII). Depending on the data-usage purpose, the extension might deliberately

High-level Type	Low-level Type	Matching Pattern
Web History	Page Title*	Exact match of page title
	Page URL	Exact match of page URL
	Page Hostname	Exact match of page hostname
Website Content	Hyperlink*	Hyperlinks in <a> elements
	Website Text*	*bait* text value
	Product ID	Product ID on shopping sites
Location	IP address*	IP addresses of testbed servers
	Region*	<city_name>, <zip_code>
	GPS Coordinates*	Coordinates of testbed servers
User Activity	Mouse Click*	*ui.click* events
	Keystroke Logging*	*ui.input* events/partial *bait* input

Table 5.1: List of the high-level and low-level data types supported by `ExtPrivA`. * marks the examples of low-level data types provided by the Chrome Web Store [156].

collect the first name (and other PII) or only generic Website Content. Since extracting context-dependent data types requires the determination of the semantics of the data type's context, we leave their extraction as future work.

`ExtPrivA` extracts 11 low-level data types under the high-level data types as listed in Table 5.1. Since the Store provides only several examples rather than an exhaustive list of each low-level data type, we add the following examples for their privacy significance and relevance to our experiments. *Page URL* is one of the "browsing-related data", a definition of the Store for the Web History, and can be used to exactly determine the page that a user visited. Similarly, *Page Hostname* reveals a user's browsing habits while extensions frequently break a page URL into a hostname and a URL path before sending them to external servers. Finally, *Product ID* is considered separately for analyzing shopping-assisting extensions during user visits to shopping sites like *amazon* and *ebay*. Data types listed in Table C.2 are all examples in Store policies when this chapter was written.

While adding the low-level data types widens the scope of the high-level types, we avoid any addition that makes the high-level data types overlap and become ambiguous. In particular, some low-level data types overlap (such as Page URL and Page Hostname) but one low-level data type does not simultaneously fall into different high-level data types.

Extractor Design. The extraction of data types from a key-value pair is formulated as a classification problem. For each low-level data type, we create a classifier that determines whether the key-value contains the data type or not. To achieve low false positives (i.e., high precision), we design classifiers based on pattern-matching rules as follows.

To extract Website Content and Web History data types, `ExtPrivA` searches for the content and the URL of the currently visited web page in the transferred key-values. For example, if the traffic contains an exact match of the URL of the web page, the extension collects the currently visited URL or the Web History data type. Similarly, for certain websites, we search for an ID in the URL such as an item ID on amazon URLs (e.g., *amazon.com/dp/ABC* where the last part of the URL, *ABC*, is the item ID). Inspired by the *bait* technique [2], in addition to the existing website content, we search for the *bait* value contained in the *honeypage* in the traffic. The *bait* is selected to avoid collision with other common keywords in the traffic key-values so that its occurrence in the traffic indicates the collection of the Website Content.

To detect the collection of User Activity, we rely on API documentation and the *bait* technique. Specifically, we found that extensions utilized popular the Sentry monitoring library [160] to monitor the keyboard input and mouse clicks. In particular, "ui.click" and "ui.input" are used for a mouse click and keyboard input events, respectively. Furthermore, after `ExtPrivA` inputs a *bait* keyword W via keyboard, if only part of W, but not the whole W, exists in the traffic, we consider the extension monitored keystrokes.

Development of Data-Type Matching Rules. We follow the widely-used bootstrapping procedure in which the set of patterns is built iteratively with minimum human intervention [7, 166]. To create the seed patterns for extracting a data type T, we first performed an exploratory study on the data traffic of the extensions that disclosed their collection of T.

Using a set of patterns, we found a set of matching key-value pairs where we discovered the new patterns. The process is then repeated while retaining only the most reliable patterns after each iteration. The final patterns were found to change only slightly with carefully-tuned seeds [166] and are listed in Table 5.1.

5.6.2.2 Extraction of Data Receivers

Given a data type extracted from a key-value pair, the *receiver* of the corresponding data flows is set to the *extension* that sent the data and the external server where the data is sent to, regardless of the ownership of the external server. Because a key-value is transferred to an external server by the execution of an extension, the extension must first collect the data from the browser or web pages before sending it to the external server. The data types extracted by `ExtPrivA` (Table 5.1) are dynamic data that require the execution of a script or API call to retrieve their values, rather than static/hard-coded data like an extension version. For example, when an extension sends a user's mouse clicks to *google-analytics.com* for its development-analytics purposes, the extension is considered collecting the user activity even if it does not own the Google Analytics server.

Even when an extension directly shares user data with third parties, it poses high privacy risks to users if the users are not aware of the collection of their data due to the execution of the extension. For example, when a translation extension transmits the user-selected text on a web page to an external spelling-checking service, the user needs to be aware of such data collection to avoid inadvertently selecting sensitive data, such as an email with a trade secret, to be sent to an external spell checker.

5.7 Detection of Inconsistencies

5.7.1 Semantic Relationships

ExtPrivA detects the inconsistencies between an extension's actual data collection and its privacy-practice disclosures by analyzing the (in)consistencies between the extracted privacy statements (Section 5.4) and data flows (Section 5.6). As data flows and privacy statements are expressed in different terms and granularity, in order to check their (in)consistencies, ExtPrivA leverages ontologies of data types and receiving entities that represent the relationship between terms to perform logical comparisons between the statements and flows. An ontology can be represented as a directed graph where an edge points from a more general term to a more specific term. For example, there is an edge from Website Content to Hyperlink data type. Inspired by prior work [20, 21, 46], the semantic relationships and the consistency condition are defined as follows.

Definition 5.7.1 (Semantic Equivalence)**.** *Two terms x and y are semantically equivalent in an ontology o, denoted as $x \equiv_o y$, if and only if they are synonyms in o.*

Definition 5.7.2 (Subsumptive Relationship)**.** *Two terms x and y have a subsumptive relationship (i.e., x "is an instance of" y) in an ontology o, denoted as $x \sqsubset_o y$, if there are a series of terms x_1, $x_2, \ldots, x_{n-1}$ such as $x \sqsubset_o x_1$, $x_1 \sqsubset_o x_2$, ..., and $x_{n-1} \sqsubset_o y$. Similarly, $x \sqsubseteq_o y \Leftrightarrow x \equiv_o y \vee x \sqsubset_o y$.*

5.7.2 Privacy-Statement Contradictions

Two privacy statements are said to be *contradictory* if their data or receivers have subsumptive relationships with each other while the statements have opposite sentiments (positive vs. negative). For example, a contradiction occurs between "we do not collect your personal data" and "we may collect your location" because *location* subsumes under *personal data* while keeping the receivers the same. We leverage the logical contradiction rules in PolicyLint [20] to detect such contradictions and formalize them as follows.

Definition 5.7.3 (Policy Logical Contradiction)**.** *Two privacy statements* $(e_i, collect, d_k)$ *and* $(e_j, not\_collect, d_l)$ *are contradictory if (*$d_k \sqsubseteq_\delta d_l$ *or* $d_l \sqsubset_\delta d_k$*) and* $e_i \sqsubseteq_\varepsilon e_j$*.*

A main challenge in detecting contradictions between Dashboard disclosures and privacy policies lies with the differences between their data-type ontologies that comprise the sets of data types and their subsumptive relationships. Specifically, the Dashboard data types are defined by the Chrome Web Store and follow narrower definitions than those used in the privacy policies that contain broader statements about the websites, services and extensions. For example, the term "personally identifiable information" in privacy policies includes "IP addresses" [210] while the Store's definition does not [157].

To resolve these differences and analyze privacy statements uniformly, we treat the collection of the data types in Dashboard disclosures as normal sentences so that they are comparable with the statements in the privacy-policy counterpart. For example, the collection of Location in Fig. C.1 is treated as "we collect your location." Therefore, we use the privacy policies' ontologies that are broader than the Store's ontologies to analyze the privacy-statement tuples in both privacy policies and Dashboard disclosures. In particular, we add the data-type nodes and subsumptive-relationship edges in the Store's ontology graph into broader privacy-policy ontologies.

An advantage of this approach is that the unified ontologies can be used to detect the contradictions among the statements of the same privacy policies. While this approach excludes the generated negative-sentiment statements that require a complete declaration of all data types (Section 5.4.2), ignoring these statements do not generate any false positives.

5.7.3 Flow-to-Policy Consistency

Definition 5.7.4 (Flow-Relevant Privacy Statement)**.** *A privacy statement* $s_f = (r_f, c, d_f)$ *is said to be relevant to a flow* $f = (r, d)$ *if and only if the flow's receiver and data object are subsumed under the corresponding terms of the statement, i.e.,* $r \sqsubseteq_\varepsilon r_f$ *and* $d \sqsubseteq_\delta d_f$*.*

Definition 5.7.5 (Flow-to-Policy Consistency)**.** *A data flow f is said to be consistent with a set of privacy statements $S = \{s\}$ if and only if the set of flow-relevant privacy statements $S_f \subset S$ contains a positive-sentiment and no negative-sentiment privacy statement, i.e., $\exists s_f = (r_f, c, d_f) \in S_f$ s.t. $c = collect$ and $\nexists s'_f = (r'_f, c', d'_f) \in S_f$ s.t. $c' = not\_collect$.*

Informally, given privacy disclosures that comprise a set of privacy statements, a flow is consistent with the disclosures if there is a positive-sentiment statement that states the collection of the data type in the flow while there is no negative-sentiment statement that describes the "non-collection" of the data. For example, a flow *f = (extension, selected text)*, where the *selected text* in the currently visiting web page is collected by the *extension*, has a relevant statement that "we collect the website content" because *website content* includes the *selected text* (i.e., *selected text* $\sqsubset$ *website content*) and *we* $\equiv$ *extension*. The flow is then consistent with the disclosures if there is not any relevant statement that states otherwise.

A flow-to-policy inconsistency occurs when the Consistency Condition (Definition 5.7.5) is not satisfied. We classify the types of the inconsistencies into Correct Disclosure and Incorrect Disclosure. A Correct Disclosure occurs when the Consistency Condition holds and an Incorrect Disclosure happens if the condition does not hold. For example, a flow *(extension, selected text)* is inconsistent with privacy disclosures if there is negative statement *(extension, not_collect, website content)*.

We focus on the inconsistencies between the extension behavior and the Dashboard disclosures since they follow the same extension-specific data-type ontologies defined by the Chrome Web Store. Comparing the data-collection behavior with the privacy policies requires to resolve the semantic gap between the data types defined in the Store and the common policies (Section 5.7.2) while the flow extraction is designed based on the Store's data-type ontologies. Furthermore, the inconsistencies between the data flows and privacy-policy documents have already been studied before [20, 46]. Finally, because the complete list of data types is defined by the Store, this flow-to-policy consistency analysis utilizes

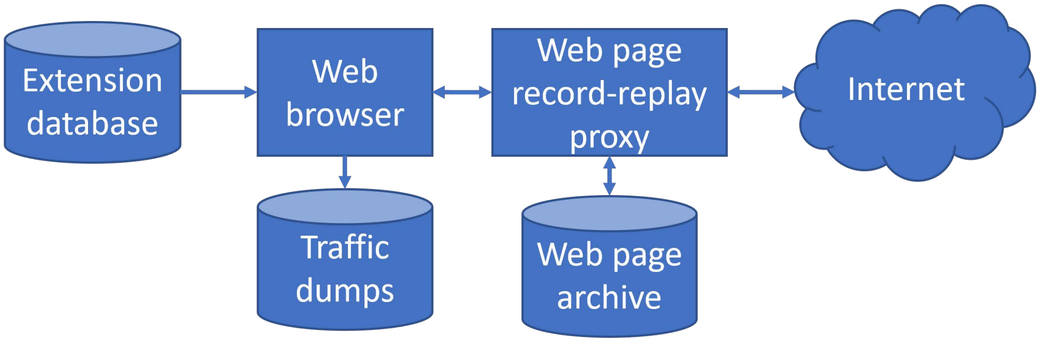


Figure 5.3: ExtPrivA extension analysis testbed.

the negative-sentiment privacy statements for the undeclared data types as described in Section 5.4.2.

5.8 Implementation

5.8.1 Privacy-Disclosure Extraction

We extracted privacy statements from template-based Dashboard disclosures by parsing the Privacy-Practice page of each extension while we leveraged the open source PurPliance [46] to extract privacy statements from the privacy policies. For each extension, ExtPrivA parsed the overview page to obtain the URL of the privacy policy. The system then pre-processed and extracted the sentences from the policy using both rule-based methods and neural NLP models. The crawling and pre-processing of extensions' privacy policies are described in Appendix C.2.

5.8.2 Data Flow Analysis

Analysis Pipeline. ExtPrivA performs dynamic analysis and captures data traffic of the extensions using an analysis pipeline as shown in Fig. 5.3. Each extension is initially loaded to a clean browser instance that disables updates and other unnecessary background traffic such as user-metrics reporting to avoid noisy traffic, following the measurement procedure of Chromium telemetry framework [130]. The browser then records the traffic of the extensions and loads web pages via a web page record-replay proxy. The browser employs mechanisms to avoid bot detection that has been known to affect the real behavior of websites [32, 171].

ExtPrivA utilizes the Playwright browser automation tool [217] to drive an instance of the Chromium web browser. Extensions are loaded to browser instances that display to a virtual X11 frame buffer (Xvfb) as the browser does not support to load extensions in the headless mode. The keyboard keystrokes are sent to the browser instances via the X11 server using the *xdotool* [278]. To trigger an extension action (i.e., a click on the extension menu-bar icon), ExtPrivA instruments the manifest to set the shortcut keys to perform the extension actions because the browser automation tool can only interact with web pages' contents but not the interface of the browser.

To make the experiments reproducible, we employ a web page replay (WPR) proxy that is a modified version of the Chrome Web Page Replay tool [122] to record, replay and passthrough network requests and responses. The WPR proxy replays website contents for reproducibility while allowing the browser extensions to communicate with the Internet to capture the extensions' realistic behavior. Since the WPR proxy passes through dynamic requests that were not pre-recorded, it tests extensions on dynamic contents. Furthermore, replaying static-contents avoids heavy traffic to websites. In the record mode, the WPR proxy records the responses of web pages by using the browser with no installed extensions. In the replay mode, the proxy passes through requests which are not found in the WPR proxy's recorded request store. In particular, the most commonly visited web pages were recorded and replayed. We set up the browser to whitelist the SSL certificates for the WPR proxy to capture and replay encrypted HTTPS traffic.

Traffic Interception. To intercept the traffic generated by the JavaScript's XHR requests, ExtPrivA utilizes the Chrome DevTools Protocol (CDP) [124] to extract the network traffic from a web page to servers. ExtPrivA creates a CDP session via Playwright to send commands and receive events from the DevTools in the browser instance. Specifically, ExtPrivA enables network tracking functionality of the DevTools and extracts information from the network events. The request information contains a *request initiator* which can be the DOM parser or a script. Since the browser treats a background page as a regular web page, ExtPrivA captures network traffic of background pages of Chrome extensions separately.

Determine Extension IDs. To accurately extract the network traffic originated from an extension, ExtPrivA determines the extension's ID which is unique in the browser instance. The system adds a *key* value to the manifest to make the extension ID non-randomized [127]. ExtPrivA then extracts the extension ID from the preference configuration file in the browser's *user data directory* and also verifies the loaded extension path.

5.8.3 Testbed

To perform experiments in a large dataset of extensions, we create a distributed experimental framework to run the dynamic analysis on multiple machines. The testbed is replicated and run in identical and isolated environments. The framework is based on Docker Swarm [159] and the browser is started with arguments to make it run in the resource-constrained docker environments [216].

5.9 Evaluation

We performed an in-depth analysis of the flow-to-policy inconsistencies of the extensions on the Chrome Web Store. Presented below are the experimental setup and results.

5.9.1 Extension Selection

We designed a crawler to collect extensions on the Chrome Web Store. By following the Store's *sitemaps* [151], the crawler systematically visited and extracted the source code and description of each extension. The data collection was done by a server located in the US on Feb 21, 2022, and took 18 hours to complete. The total number of extensions collected is 134,196.

In the following experiments, we consider the 47,207 (35.18%) extensions that declare the Dashboard disclosures. Of all extensions, 35,316 extensions have privacy policies and 12,484 of them have no Dashboard disclosures. The disclosures have been required for publication of an extension on the Web Store since March 2021 [157]. We also observed a significant increase of extensions with Dashboard disclosures, from 31,839 extensions in a crawl in May 2021. Therefore, we assume that all extensions on the Store will gradually include Dashboard disclosures.

5.9.2 Policy and Flow Characterization

5.9.2.1 Dashboard Disclosures

The majority of extensions state not to collect any user data while a significant number of extensions state collection of only 1 data type. Of the extensions with Dashboard disclosures, 33,787 (71.57%) state that they do not collect or use any user data while 15.97% of the remaining extensions state collection of only 1 data type. As shown in Table 5.2a, this kind of extensions (i.e., those that collect only 1 data type) is the most common.

For each data type, the number of extensions that declared the data collection is also small. The most common data type collected by the extensions is Website Content (13.5%) while the least common is Health Information (0.29%). Table 5.2b shows the distribution of the collected data types.

5.9.2.2 Privacy Policies

Of the 47,207 extensions with Dashboard disclosures, 22,832 (48.37%) contain privacy policies. `ExtPrivA` extracted 8,012 extension-related privacy statements from 2,091

# Data Types	% Exts.
0	71.57
1	15.97
2	5.47
3	3.84
4	2.03
≥ 5	1.12

(a) Data types per extension.

Data Type	# Exts. (%)
Website Content	6,392 (13.54)
PII	4,545 (9.63)
User Activity	4,533 (9.60)
Authentication Info.	3,005 (6.37)
Location	2,562 (5.43)
Web History	2,549 (5.40)
Personal Comm.	929 (1.97)
Financial & Payment	509 (1.08)
Health Information	135 (0.29)

(b) Distribution of data types.

Table 5.2: Data-type distribution on Dashboard disclosures.

Candidate URL	Total
<all_urls>	7841
https://www.youtube.com	1068
https://www.coolstart.com	1002
https://www.google.com	893
https://www.mystart.com	804
https://www.facebook.com	686
https://www.amazon.com	651
https://mail.google.com	646

Table 5.3: Top candidate URLs.

# Flows	# Exts.
1	618
2	169
3	121
4	83
5	11
Total	1002

Table 5.4: Distribution of data flows in extensions.

extensions' privacy policies. Because of the exclusion of the statements that do not mention browser extensions, `ExtPrivA` did not include the policies from the remaining extensions.

Of these privacy statements, 6,238 (77.86%) have a negative sentiment and 1,774 (22.14%) have a positive sentiment. 1,538 extension policies contain negative sentiment statements that discuss broad categories of data. Of the statements with a negative sentiment, the data object "personally identifiable information" or "PII" appears in 1,280 of these extensions. This high percentage highlights the significance of negative privacy statements as 83.22% (1,280/1,538) of the policies that contain a negative sentiment exclude the collection of a broad data type.

5.9.3 Data Traffic and Flows

5.9.3.1 Experimental Setup

Given an extension E, `ExtPrivA` first identifies the candidate URLs to activate the extension's functionality (Section 5.5.1.1). The system then visits each of the identified URLs in a clean browser instance with the extension E installed at each start-up while disabling other extensions to reduce execution and traffic noise. For each URL, the system visits a real page and a *honeypage*. If the URL has been recorded by the Web Page Replay proxy, the network requests are redirected to the proxy to reduce loads on the server side while improving the reproducibility of the experiments. Since the number of the candidate URLs can be large, for each extension, `ExtPrivA` visits the URLs until either all URLs or a maximum of 10 URLs are visited.

For each URL, the browser waits until the home pages are fully loaded by waiting until there are no network connections within a timeout of 5 seconds or a maximum of 30 seconds. Because the experimental servers used a fast Internet connection, we empirically found that these timeouts were sufficient to completely load most of the web pages. The page loading heuristics are commonly used in the empirical settings and provided as the default in the web browser automation tools [129, 217]. Finally, `ExtPrivA` interacts with the browser to activate the functionality of the extension (Section 5.5.1.3). It is worth noting that an experiment does not raise false positives if the extension is not successfully loaded or its functionality is not activated. The analysis was performed on a cluster of 8 machines with 1.18TB of RAM in a university in the US and took 70 hours to complete.

5.9.3.2 Extension URL Patterns

From the 47,207 extensions that provide Dashboard disclosures, we extracted 129,218 candidate URLs on 28,618 domains. The distribution of the domains has a long tail with only 248 domains with frequency greater than 100. The most common extracted domains are *yahoo.com* and *google.com* that involve a large number of country-specific subdomains for their services. The third most common domain is *coolstart.com* that hosts a new-tab page for numerous new-tab-customization extensions. The most common extracted candidate URLs are shown in Table 5.3.

5.9.3.3 Extracted Key–Value Pairs

`ExtPrivA` activated the extensions' functionality, captured their network traffic and extracted 680,923 key–value pairs sent from 3,904 extensions to 3,280 external server endpoints each of which is a combination of a host and a path. To activate an extension's functionality, `ExtPrivA` visited 5.1 candidate URLs on average (1.82 *SD*). The most common host of the endpoints is *www.google-analytics.com* (80,171 (11.77%) key-value pairs). The high percentage of traffic to Google Analytics indicates its popularity among the extensions for data collection.

5.9.3.4 Data Flow Characterization

Flow Data Types. From the traffic key–value pairs, `ExtPrivA` extracted 1,706 unique data flows for the data types received by 1,002 extensions. Each extension collects 1.7 data types on average (1.04 *SD*). The most common data types extracted from the extensions are the URL and hostname of the currently visited web pages which are under the Web History high-level data type. Such data types are privacy-sensitive as they can be easily used to construct the users' web browsing habit. The distribution of the extracted data types over the extensions is shown in Table 5.4.

5.9.4 Evaluation of Detection Performance

We now evaluate the performance of the extraction and consistency analysis. Since our goal is to minimize the false positives, we focus on the evaluation of system precision by verifying the correctness of randomly-selected samples. The verification was done manually by two PhD students with no less than 3 years of experience in user-privacy research. The annotators first agreed on a common annotation scheme/verification workflow, and then worked independently to verify the correctness of the system output. Finally, they held a follow-up meeting to reconcile the differences, if any. Since dynamic analysis cannot exercise all behaviors of an extension while making a ground-truth dataset of privacy policies and flows requires significant effort and time [166, 303], we leave the recall-rate evaluation as future work.

5.9.4.1 Performance of Contradiction Detection

To evaluate the performance of contradiction detection, we verified the pairs of the contradictory privacy statements detected by `ExtPrivA`. The annotators read the corresponding sentences of positive and negative statements to assess whether the extracted statements were indeed contradictory or not. Each privacy statement could be generated from different sentences in the privacy policies where the sentences were slightly different in grammar but expressed the same (non)collection of the same data types. When the sentences were ambiguous due to the lack of context, we traced back to the extension privacy-practice disclosure pages on the Store and the privacy policies to fully understand the statements.

The result shows that the detection achieves 91.7% precision. Of the 60 randomly selected sentence pairs of 56 extensions, only 5 were false positives. Some of these false positives were due to the lack of cross-sentence analysis such as co-reference resolution. For example, a statement of the *#fastset For Social Media* extension privacy policy applied to another different extension of the same developer but such a mention could only be understood by reading the preceding sentences.

5.9.4.2 Accuracy of Data-Type Extraction

We verified the data types extracted from key-value pairs by attempting to understand the intention of the data traffic from the context which includes the web page being visited, the extension's description and external sources. In particular, we used the Chrome DevTools to obtain other key-values in the data traffic and traced back to the request-initiated script to identify the data source of the key-values. Inspired by a prior mobile-app traffic analysis [166], we leveraged two properties of key-value pairs to infer the data types: 1) naming conventions indicate the data types of a key-value pair and 2) external knowledge such as the extension description can be a strong indication of data-collection purposes. For example, given *key=regionName* and *value=<city_name>*, the key-value pair likely represents the transfer of the user's geographical location.

We evaluated the extraction's precision on 330 randomly selected samples (30 samples per data type), showing a $\geq$93.3% precision. Since the extraction is based on strict matching rules, a match is likely a correct occurrence of the data type in a key-value pair. One of the lowest precisions is the Hyperlink data type that indicates collection of the hyperlinks of the currently visited web page, because, in some cases, the links were inserted by the extensions, not the original web page content. The precision for each data type is provided in Table C.3 of Appendix C.4.

5.9.4.3 Accuracy of End-to-end (In)consistency Detection

Given a detected inconsistency, we attempted to reproduce the result by using only built-in tools of the browser to evaluate the end-to-end performance of the inconsistency detection. We installed the extension on a clean browser instance and captured the network traffic of the tab and the background pages via Chrome DevTools to reproduce the data-collection behavior of the extension. We read the privacy disclosures and verified the correctness of the privacy-statement extraction. Finally, we assessed whether the data-collection behavior violated the privacy disclosures or not.

Inconsistencies were detected with a precision of 85%. We were able to reproduce 51 of the randomly selected 60 detected inconsistencies of 59 extensions. Manual verification of each inconsistency took 15 minutes on average, so the two annotators spent 30 hours

Type	Positive Stmt.	Negative Stmt.	# Contradictions
1	Privacy policy	Dashboard	388
2	Dashboard	Privacy policy	73
3	Privacy policy	Privacy policy	64
	Total		525

Table 5.5: Number of contradictory pairs of privacy statements per statement type. *Stmt* stands for a privacy statement.

in total. Most of the false positives were due to a non-data-collection callback function of an extension script that was included as one of the initiators of the network traffic. For example, an extension installed an HTTP-request event handler to check the occurrence of a URL pattern in the HTTP requests even if it did not do any data collection.

5.9.5 Findings

5.9.5.1 Finding 1: Dashboard disclosures and privacy policies contain contradictory statements

ExtPrivA detected 525 pairs of contradictory privacy statements in the privacy policies of 17.22% (360/2,091) extensions that have extension-related statements (Section 5.9.2.2). Each contradiction comprises one positive statement and one negative statement, either or both of which are from the privacy policies. Because Dashboard disclosures are template-based, there are no contradictions when both statements are from the Dashboard disclosures. The distribution of the contradiction types is shown on Table 5.5.

The most common contradiction type comprises one statement from the Dashboard disclosures that states the non-collection of any data and another statement from the privacy policy that claims the collection of certain data. This type constitutes 73.90% (388/525) of the detected contradictions. Such discrepancies between the privacy policies and Dashboard disclosures are a serious problem and can thus result in the suspension/removal of the extensions from the Store [157].

These extensions had a total of 27.3M users where 16 extensions have more than 100k users and 4 have more than 1M users. For example, AdBlock, that has more than 10M users, declared that it would not collect or use any user data on the Dashboard disclosures but its privacy policy stated that "when the AdBlock extension communicates with AdBlock servers, we receive the computer's IP address" [158]. The policy also mentioned that "after six months we will remove any identifying information such as IP address from our log files and databases," i.e., AdBlock databases record IP addresses.

There are multiple potential causes of the contradictions. First, the manual creation of these policies is error-prone, especially when the authors of the privacy policies and Dashboard disclosures are different. Second, the developers might reduce the number of declared data types in their Dashboard disclosures to reduce the time to publish the extension on the Chrome Web Store. At the same time, the privacy policy tends to contain a comprehensive description of data-collection practices to cover future use-cases.

5.9.5.2 Finding 2: A significant number of extensions fail to fully declare the data types that they collect from users in their privacy disclosures

ExtPrivA detected 820 extensions with more than 84.6M users that have a data flow inconsistent with their Dashboard disclosures. These extensions constitute 81.84% (820/1,002) extensions with an extracted data flow. The inconsistent data flows are 75.62% (1,290/1,706) of the total extracted flows. Each extension contains 1.57 inconsistent data flows and 103,190 users on average. This result underscores the incomplete privacy disclosures on the Chrome Web Store that pose high privacy risks to a large number of users.

While ExtPrivA detects inconsistencies in various types of extensions, the inconsistent extensions do not spread evenly over extension categories. The most common categories are Productivity and Shopping with 425 and 261 extensions, respectively. These categories tend to collect and use more data to customize and enhance web pages. For example, they

Data Type		# Extensions
High-level	Low-level	(Inconsistent/Total)
Web History	Page URL	505/616
	Page Hostname	304/345
	Page Title	53/70
	Total	672/800
Website Content	Hyperlink	149/229
	Product ID	139/215
	Website Text	70/116
	Total	303/472
Location	IP Address	36/53
	Region	11/24
	GPS Coordinate	4/5
	Total	48/69
User Activity	Mouse Click	12/18
	Keystroke Logging	7/15
	Total	14/23
Total		820/1002

Table 5.6: Distribution of the extracted data types and detected inconsistencies. Each row reports # of extensions that have inconsistencies and ones that have data flows extracted.

include shopping price analytics, new-tab customization and translators. The distribution of the inconsistent extensions over the categories is shown in Fig. 5.4 and Table C.4 (Appendix C.5).

Case Studies. As a case study, the *CapitalOneShopping* [49] extension (8M+ users), using its background page, sent the full URLs of the currently visiting product pages of shopping websites (like *ebay* and *amazon*), the time of visits and a persistent user ID to its server at *track.capitaloneshopping.com*. The extension automatically collected these pieces of information every time we visited a new product page even without clicking the extension icon. While the information might be used for providing the extension's services, such as offering coupons, it is "browsing-related data" under the definition of the Web History data type and could be easily used to determine the browsing paths of users.

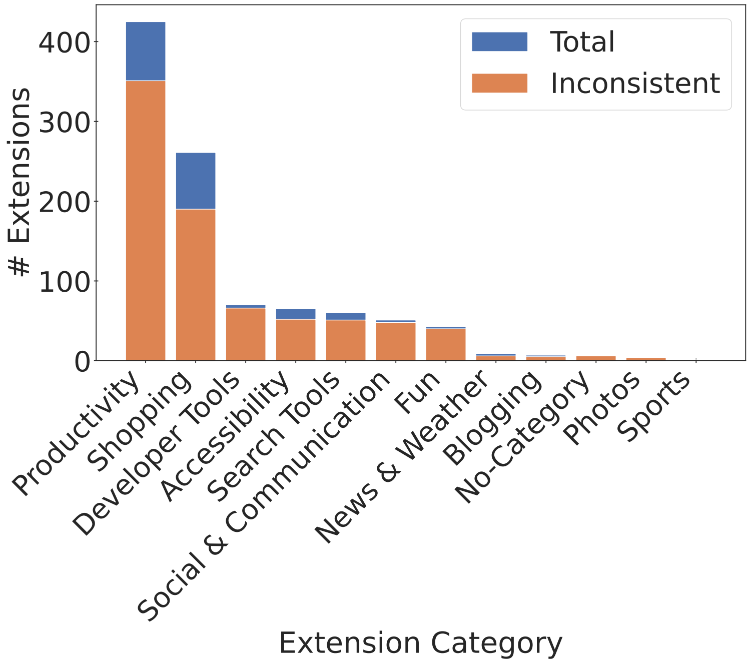


Figure 5.4: Distribution of inconsistent extensions over categories.

Although the extension disclosed the collection of Web Content, it omits the actually-collected Web History from its Dashboard disclosures.

Similarly, the *SearchPreview* extension [270] (100k+ users) declared not to collect or use any user data but the data flows showed that the content of Google search result pages was transferred to the extension's server. The extension used the information to display the previews (thumbnails) of search results on Google search pages. In particular, the full URL of the Google search page and URLs of the search results were transferred to *searchpreview.de*. Since a Google search page URL includes the query term, browser and language information, the collected information is Website Content. However, such data collection was omitted from the Dashboard privacy disclosures.

Potential Root Causes. A direct cause of the flow-to-policy inconsistencies is that extensions tend to declare a limited number of user data types that they collect, effectively omitting their data collection practices. In particular, 56.95% (467/820) of the inconsistent extensions disclosed "no user data collection or use" in their Dashboard disclosures while the extensions still collected certain data types. Because declaring the collection of more data types incurs a longer security review time on the Chrome Web Store, we conjecture that the extensions attempted to reduce the number of the collected data types to shorten the extension publishing time. For example, 71.57% (33,787/47,207) of extensions disclosed not to "collect or use any user data" while the other 15.97% disclosed to collect only one data type.

Furthermore, the most common data types that are collected without declaration is Web History that occurred in 672/1,002 extensions. The extensions collected the URL or title of currently visited web page while excluding them from the privacy disclosures. We hypothesize that the extensions might omit the sensitive web browsing history data type to avoid being suspicious to users and increase their installation rate.

Similarly, the extensions commonly used analytics and monitoring libraries but might not fully configure the libraries to limit the data they collect. From the data traffic of extensions, we found that many extensions collect user data for analytic purposes yet failed to disclose them in the privacy disclosures. For example, 14% of the extracted flows were sent to Google Analytics or Sentry analytics libraries [160]. Therefore, we recommend the developers to minimize the data that is collected by the external analytics libraries and services.

5.10 Limitations and Future Work

Applicability of Analysis Techniques. Detection of the inconsistencies between privacy disclosures and extension execution is critically important for all stakeholders in the web browser ecosystem. The removal of an extension from a marketplace would cause substantial loss to the developers. The deviation of actual data collection and usage from the stated practices is not expected by the users and causes loss of users' trust in the web browsers' privacy protection. Finally, extension stores can leverage the analysis techniques/tools to audit and detect privacy breaches.

As an end-to-end framework, ExtPrivA can be easily integrated into the Chrome Web Store's vetting process or an IDE to help developers verify that their extensions operate consistently with their stated privacy policies. As extensions may use third-party libraries, it is hard and expensive for developers to check the consistency manually. A benefit of dynamic analysis is that it can provide the inputs to reproduce the inconsistencies and facilitate the debugging process. Furthermore, even with an extensive vetting process, our results show the Chrome Web Store to still misses extensions that provide misleading privacy-practice disclosures. We plan to communicate our findings to the developers and Chrome Web Store to help them fix the inconsistencies in their extensions.

Analysis of Non-Chrome Browser Extensions. Since most of the ExtPrivA pipeline utilizes black-box analysis methods, ExtPrivA can be extended to detect the inconsistencies of extensions in non-Chrome web browsers, such as Firefox and Safari. First, the free-form privacy policies of non-Chrome browser extensions are not different from those of Chrome-based browsers. Extracting privacy policies only needs to handle the differences in the extension description web pages on different extension stores. Similarly, while the internal API of Chrome that extracts the initiators of network traffic does not automatically translate to Firefox and Safari, these browsers have equivalent APIs, such as the traffic initiator of Firefox network analysis [64]. Finally, the browser automation tool [217] of ExtPrivA testbed already supports multiple browsers. We leave the multi-browser support for Firefox and Safari extensions, which accounted for only 18% for the desktop browser market share (less than 10% each) [287], as future work.

Detection of Contradictions in Privacy Disclosures. The existence of contradictions between the privacy policies and the Dashboard disclosures highlights the inadequacy of manual checking for the (in)consistencies of privacy disclosures. However, ExtPrivA cannot fully analyze the privacy policies because of the inherent limitation in determining the scope of policy statements, i.e., whether a sentence is about browser extensions or not.

Some recent approaches [20, 46] still analyze privacy policies at the sentence level due to the lack of a holistic analysis of the entire documents. While solving this problem requires advances in both NLP and privacy-policy analysis, recent development of ML models specialized in privacy policies [8] will help detect the contradictions more effectively.

Compliance of Data-usage Purposes. To comply with the Chrome Web Store developer policies, extension developers must agree with the Limited Use policy that prohibits the data types collected by extensions from being used or transferred to advertisers for advertising purposes [154]. Therefore, one can check the compliance with this policy by analyzing the data-usage purposes of the extensions. However, since determining the purposes of data collection without server-side information is still challenging [46, 166] and there is no prior work on analyzing the purposes of data usage for browser extensions, we leave this analysis as future work.

Limitations of Data-Flow Analysis. Due to the limitations of dynamic analysis, `ExtPrivA` cannot exercise all execution paths of an extension to generate the transfer of all possible data types. Similar to software testing, it is challenging to generate inputs to completely activate all functionalities of a sophisticated extension. Furthermore, `ExtPrivA` cannot generate credentials to support login-required websites. With improvements from the research in extension/JavaScript dynamic analysis [53, 175] and input generation [143], `ExtPrivA` can cover more data flows of each extension to detect more inconsistencies.

Analysis of Login Extensions. Supporting extensions that require login is challenging because the account registration process is complex and services frequently deploy bot-prevention to avoid automated account creation and detect fake identities. While recent techniques can generate input text for login Android apps [143], automatically performing account registration and login on web apps requires further advances in web page analysis and NLP. Although supporting login extensions can improve the coverage of extracting data-collection behavior of extensions, lack of login-extension support does not increase the false-positive rate that we aim to minimize.

5.11 Conclusion

We have presented a novel system, ExtPrivA, to detect inconsistencies between the privacy disclosures and the actual data collection of browser extensions. ExtPrivA is an end-to-end system that performs a fine-grained analysis of data collection of browser extensions to detect their flow-to-policy inconsistencies. It first analyzes contradictions between privacy statements in the Dashboard disclosures and privacy policies. It then extracts data flows and analyzes their (in)consistencies with privacy statements using a formal model.

Using ExtPrivA, we have conducted a large-scale study of 47,207 extensions that provide Dashboard disclosures on the Chrome Web Store. Of these, ExtPrivA detected 1,290 inconsistent data flows of 820 extensions with more than 84.6M users. ExtPrivA has also detected 360 extensions that contain 525 pairs of contradictory privacy statements in their Dashboard disclosures and privacy policies. These detected inconsistencies highlight critical issues in the privacy notices of web extensions that may mislead users about their privacy practices. We hope these findings will help all the involved parties remove/minimize such inconsistencies and enhance the users' privacy in the web browser ecosystem.

CHAPTER VI

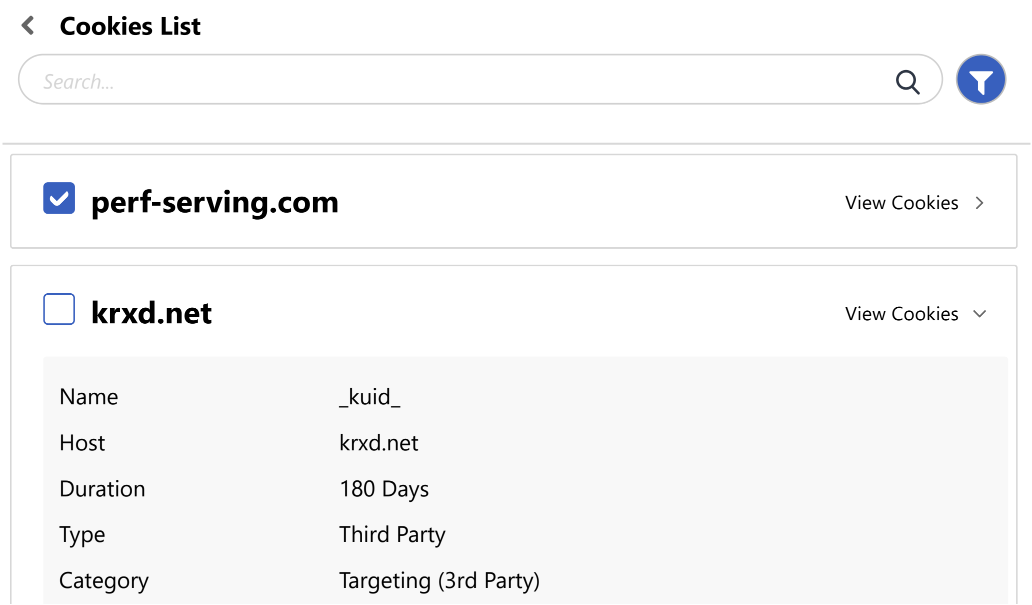
ConsentChk

6.1 Introduction

Privacy laws commonly forbid data collection without user consent. In the EU, the General Data Protection Regulation (GDPR) [97] and ePrivacy Directive (ePD, or "cookie law") [294] have mandated online services to receive user consent before collecting user data. Similarly, in the US, the Federal Trade Commission Act (FTC Act) prohibits deceptive business practices [234] such as apps' collection and sharing of users' data without receiving their approval [59, 61]. The California Consumer Privacy Act (CCPA) further requires services to provide users with choices to opt out of the sale of their data.

Websites have utilized consent services of third-party companies, called *Cookie Management Platforms* (CMPs), to obtain user consent to meet the GDPR requirements [306]. CMPs produce consent management libraries that allow users to set their consent preferences of every cookie on a website, allowing users to control the collection of their personal data. For example, Fig. 6.1 shows an example that unsets the consent for the cookies from the *krxd.net* advertiser.

Since the placement of unconsented cookies is unlawful, it is important for users and websites to detect the inconsistencies between the data collection via cookies and their consent/rejection by the user. For example, an inconsistency occurs when a website still places cookies from *krxd.net* to track a user during his/her visit to the website even after



☑ perf-serving.com

□ krxd.net

Name	_kuid_
Host	krxd.net
Duration	180 Days
Type	Third Party
Category	Targeting (3rd Party)

Figure 6.1: A cookie setting that allows users to set their consent/rejection of cookies from individual trackers. However, the website is not guaranteed to honor the users' choices.

the user rejected them. The inconsistencies mislead users, and thus, violate not only the GDPR but also other consumer protection laws, such as the FTC Act. In fact, the European regulators and the FTC have fined companies for their data collection without users' consent [39, 59, 61, 78]. Unlike other requirements specific to the GDPR, using unconsented cookies is unlawful globally, and hence we need their understanding outside of the EU.

While prior studies found cookie banners often ignored user consent, they lacks in-depth analyses of the placement of unconsented cookies from a global standpoint. Bollinger *et al.* [41] discovered new types of consent violations without their in-depth analysis in different environments. They analyzed the usage of rejected cookies only on Cookiebot while ignoring OneTrust which had a significant more market share. Furthermore, they did not evaluate violations of user consent in different regions of the world. Other prior work did not precisely identify the consent-violated cookies, thus not providing any concrete evidence of the user consent violations. Matte *et al.* [208] studied the European Transparency and Consent Framework (TCF) and considered only *cookie categories*, each of which contains tens to hundreds of cookies, not the compliance of *each cookie*.

The main question is then: *To what extent do websites use unconsented cookies to collect user data?* We present an automated framework, called `ConsentChk`, that detects three types of cookie consent violations by checking the (in)consistency between the cookie consent/rejection and the actual usage of each cookie on a website. `ConsentChk` extracts cookie consent preferences on websites in non-GDPR environments and provides a systematic categorization of consent violations via a formal analysis. It detects the usage of rejected cookies for 2 more CMPs, thus covering 3X more market shares than state-of-the-art studies [41, 208].

`ConsentChk` addresses several technical challenges in detecting cookies that violate user consent. First, we create a preference button detector to activate cookie preference menus on websites even if they do not show cookie banners by default and employ various UI customizations. `ConsentChk` then accepts/rejects the consent and extracts the consent for each cookie. Second, checking of consent violations must ensure the consent-violated cookies to fall under the user consent's scope. To address this challenge, we determine the consent scope based on the consent cookie's domain and define consent enforcement conditions to minimize false positives. Finally, `ConsentChk` systematically analyzes the inconsistencies by constructing a formal model to cope with the various types of mismatches between the cookie flows and the user cookie preferences. Inspired by the soundness of dynamic analysis in software testing [147, 148, 280], our primary goal is to minimize false positives so that each detected inconsistency may indeed be a consent violation.

Using ConsentChk, we have examined 101,703 websites out of the 200k top global websites from the UK and the US and found wide-spread violations of user consent. We analyzed the cookie blocking mechanisms to identify the root causes of consent violations. Finally, we created a browser extension to help end-users audit the inconsistencies and honor their cookie rejection.

This paper makes the following main contributions:

- Formal definition and construction of a model to systematically analyze the inconsistencies between user cookie consent and the cookie usage of a website. We categorize all possible types of consent violations after users accept/reject cookies as: (i) usage of rejected cookies, (ii) omitted consent choices, and (iii) ambiguous cookie consent due to contradictory consent preferences.

- A generic approach that leverages linguistic features of preference buttons to automatically activate the cookie preference menus of any CMPs. The preference-button classifier achieves the high detection rate of 85.96% top-3 score.

- Extraction of cookie flows under the scope of user consent.

- An end-to-end (E2E) framework, called `ConsentChk`, that detects the violations of user consent per cookie on a website. An E2E evaluation demonstrates that the detection of consent violations has a high precision of >91%.

- *A large-scale study on* 101,703 *top global websites.* `ConsentChk` detected 82.20% (4,973/6,050) of websites in a UK-based measurement and 81.86% (4,134/5,050) of websites in a US-based measurement that collected cookies with rejected consent. We measured the rejected cookie usage violations of CMPs with a 3X more market share than prior work. Our findings demonstrate that CMPs are not always effective in blocking unconsented cookies as they promise, and thus allow websites to collect users' data without their consent.

As shown in Fig. 6.2, `ConsentChk` comprises four main components which are detailed in the remainder of the paper: cookie preference menu activation (Section 6.3.1), consent preference extraction (Section 6.4), cookie flow extraction (Section 6.5.1) and flow-to-preference consistency analysis (Section 6.5.2). Our in-depth evaluation and findings are reported in Sections 6.6 and 6.7.

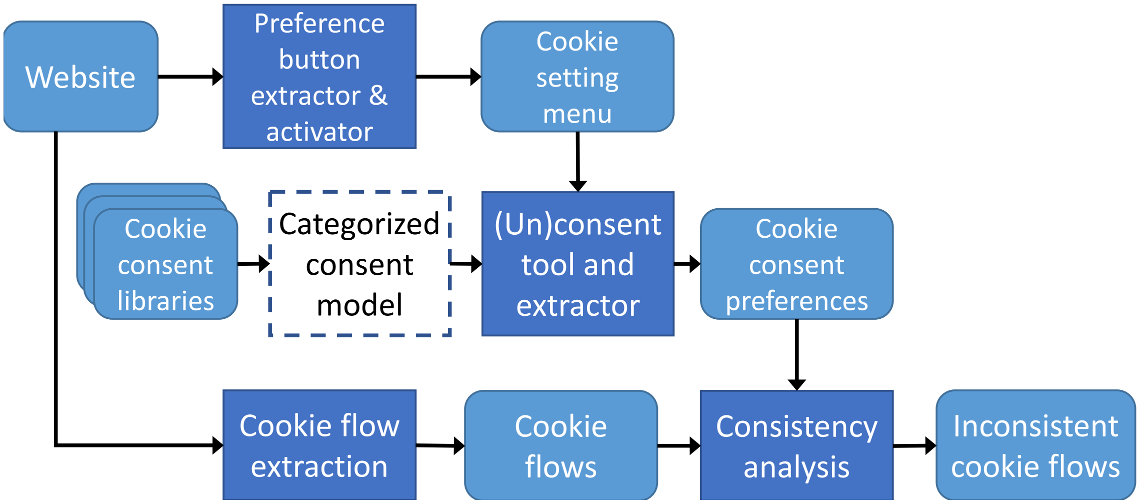

Website
Preference button extractor & activator
Cookie setting menu
Cookie consent libraries
Categorized consent model
(Un)consent tool and extractor
Cookie consent preferences
Cookie flow extraction
Cookie flows
Consistency analysis
Inconsistent cookie flows

Figure 6.2: Given a website, `ConsentChk` analyzes the actual enforcement of user consent and outputs the detected inconsistent cookie flows. The dashed box represents an one-time manual step that creates a reusable consent setter for each consent library. Other steps/boxes are fully automated.

6.2 Related Work

Cookie-level Analysis. There has been limited research on the usage of consent-violating cookies. The Sweepatic security platform detects the cookies that are set without user consent [291], but it can only detect the cookies *before* the user consents/rejects. Bollinger *et al.* [41] found new types of consent violations but did not present in-depth analyses and measurements from global vantage points. They identified ambiguous cookie declarations that were included in multiple categories but did not quantify their actual usage when the cookies were simultaneously rejected and accepted on websites. Their study also identified the existence of undeclared cookies but only measured the usage of these cookies when all cookie consents were accepted. Finally, their measurement of the usage of rejected cookies only supports Cookiebot, without analyzing OneTrust which is significantly more popular. In contrast, we improve cookie consent detection/extraction, add measurements from a vantage point outside of the EU, and analyze more CMPs.

Category-level Analysis. Matte *et al.* [208] check the compliance of cookie banners with the Interactive Advertising Bureau (IAB) Europe TCF consent framework. However, their reliance on the standardized API of TCF makes it inapplicable to other CMPs which are widely used on websites in the world. Furthermore, they require *manual* acceptance/rejection of cookie consent to the cookie banners and did not provide a fine-grained analysis of each individual cookie. Finally, they assume that websites would follow the enforcement of the IAB without checking the actual cookie usage on websites.

The ineffectiveness of cookie settings has also been reported. Sanchez-Rola *et al.* [265] found that the number of cookies on websites even increased after the user rejected cookie consent. Papadogiannakis *et al.* [241] investigated the tracking based on cookie syncing and browser fingerprinting after users rejected all cookie consents to find the tracking activities continued even after the users' rejection. Liu *et al.* [198] found that the bidding behavior of advertisers was not statistically different even when the user rejected cookie consents and opted out of the sale of personal information.

All of these focused only on *coarse-grained* tracking practices, i.e., the number of third parties when the user rejects or accepts all tracking cookies. In contrast, `ConsentChk` analyzes the (in)consistencies of tracking with user consent at a *fine-grained* level, i.e., for *each cookie* specified in the cookie settings. Since each cookie category comprises tens to hundreds of cookies and trackers (Section 6.6.3), compared to the cookie-category analysis, `ConsentChk` reduces significant effort in locating the problematic cookies and trackers. Furthermore, a coarse-grained analysis only based on the number of cookies may yield false positives because non-rejectable cookies (e.g., necessary cookies) are still accepted and the usage of cookies varies with web pages (homepage vs. sub-pages) [145], and hence a decrease in number of cookies does not always reflect the blocking of cookies.

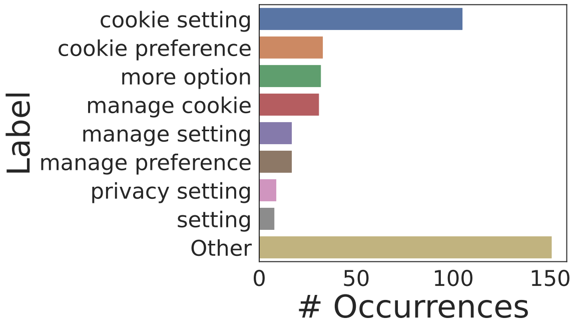


Figure 6.3: Distribution of labels of cookie preference buttons.

G	Feature (Dimension)	HTML attributes	D_G
G_1	# n-grams and keywords (3)	aria-label, class, id, text	12
G_2	# tokens $> n_t$ or not (1)	aria-label, text	2
G_3	Has consent library API (1)	class, href, id, onclick	3
		Total	17

Table 6.1: Preference button detection features. G and D_G stand for a feature group and its dimension, respectively.

6.3 Automated Setting of Cookie Consent

Detecting cookie consent violations requires automatically setting the cookie consents but activating the preference menus, before any analysis, remains challenging due to flexible HTML implementations. Several tools have automated the interactions with consent notices [66, 116, 298] but they are based on hard-coded JavaScript/CSS API of CMPs that lack of support for customized preference buttons. Khandelwal *et al.* [179] attempt to click every element and then detect preference menus after each click, but clicking every button is inefficient. Bollinger *et al.* [41] accept/reject cookie consent by using GDPR-specialized Consent-O-Matic tool [298], but this tool is not designed for global websites that do not show cookie banners for first-time visitors.

`ConsentChk` activates preference menus by using a *preference-button extractor* and a *menu activator*. When accessing a web page, if a cookie setting menu is not detected, the extractor analyzes HTML elements to extract the candidate preference buttons that may activate a menu. To increase its coverage, the activator then tries to click the top *k* candidate buttons while handling the incorrect states when clicking incorrectly-detected non-preference buttons. Specifically, if clicking a non-preference button navigates to another page or activates no consent preference menus, `ConsentChk` detects it and returns to the initial URL and tries other buttons.

We define a *cookie preference button*, also called *preference button*, to be an HTML element that, upon a user click, displays a menu for the user to set the cookie consent. For each web page, the preference-button extractor finds visible HTML elements that represent a button or link in all *iframes* contained in the page. We consider *a*, *button*, *div* and *span* elements which are commonly used to represent links and buttons [54, 199]. For *div* elements, we only select leaf elements to avoid many unrelated elements without affecting the detection accuracy.

Since `ConsentChk` extracts, sets and verifies the consent settings in the later stages of the pipeline, no false positives will be generated if the activator clicks an incorrectly-detected button. For example, if a web page does not contain any preference button or clicking a button causes the website to hide the cookie settings, `ConsentChk` would miss the settings in the worst case but does not create any false positive.

In what follows, we describe the development and evaluation of a machine learning (ML) model to classify whether an HTML element is a preference button or not.

6.3.1 Preference Button Classifier

It is challenging to extract preference buttons because depending on the context, button labels can be shortened and omit important keywords like *cookie* or *preference*. For example, a button in a cookie banner is labeled only "here" in "click *here* to change your preferences." Many buttons even have a label "do not sell my personal information". Moreover, since the button labels have a long-tail distribution (as shown in Fig. 6.3), a naïve approach to matching the button labels with the most frequent keywords will have low coverage. On the other hand, extraction solely based on consent libraries' API is not sufficient because preference buttons may undergo various website customizations.

6.3.1.1 Data Collection

We randomly selected 8k websites from the top 10k global websites for developing features and training the classifier. The remaining 2k websites were set aside to evaluate the model performance. We use the Tranco list [189] generated in July 2021 (ID: 9QK2) and accessed websites from UK. While we accessed the websites from the UK to maximize the cookie banners, we aimed to label the preference buttons on both the cookie banners and the footer of the websites for the extractor to work on non-GDPR environments.

Two authors of this paper, who are PhD students with more than 2 years of experience in privacy and security research, manually visited the home pages of 1,000 randomly selected websites from the 8k websites to identify cookie preference buttons. The annotators independently double-annotated the home pages of the websites and resolved disagreements through follow-up discussions. For each website, we recorded a snapshot of the home page HTML and the CSS selectors that uniquely identify the preference buttons. We kept only those websites with English home pages, and excluded duplicate websites that redirected the browser to the same websites. Examining each website took 2 minutes on average, or 67 hours in total for the two annotators.

This way, we create a training set of 298 web pages containing 436 preference buttons out of 71,020 all links/buttons. Many websites only show a cookie banner without any choice or only a binary accept/reject option. While the percentage of positive samples (i.e., preference buttons) is small, it reflects the portion of preference buttons on real websites relative to numerous other links/buttons.

Group	Examples
Unigrams	adchoice, adjust, change, choice, choose, configure, consent, cookie, customise, customize, manage, option, personal, preference, privacy, review, setting, update, view
Bigrams	configure consent, set preference, advanced setting, privacy setting, update preference, personal information, manage preference, california sell, privacy preference, sell personal, consent detail, manage setting, change privacy, view cookie
Keywords	change consent, change setting, consent choice, consent tool, cookie consent, cookie preference, cookie setting, customize setting, manage cookie, review cookie

Table 6.2: Examples of n-grams and high-frequency keywords extracted from the button labels.

6.3.1.2 Feature Selection

We derive 3 classification feature groups based on the HTML attributes: aria-label, class, id, and inner text. The attribute *aria-label*, an accessibility feature of the web for marking buttons with labels for those users with disability to use with screen reader programs [67], is especially useful in cases where the button is displayed as a non-textual icon. The features are shown in Table 6.1 and the feature vector has 17 dimensions in total.

Feature group G_1 is the number of occurrences of selected unigrams, bigrams and high-frequency keywords in the button labels. The *n*-grams are extracted from the inner-text and aria-label of the preference buttons in the training set. Each *n*-gram is the combination of lower-case lemmas from the text. We also separate the most frequently-used bigrams into a set of high-frequency keywords. Table 6.2 lists some examples of the *n*-grams and keywords.

Feature group G_2 indicates whether the number of tokens (excluding stop words) of a button label is greater than a threshold n_t or not. This feature avoids long paragraphs that include many preference-button keywords. We empirically choose the threshold $n_t = 9$ as the number of tokens of the longest preference-button labels in the training set.

Finally, feature group G_3 indicates use of a cookie consent library API, and hence the presence of such an API strongly indicates that the website provides cookies settings based on the consent library. We identify cookie library APIs used in HTML *class*, *href*, *id*, *onclick* handler attributes by third-party libraries such as *ot-sdk-show-settings* of OneTrust.

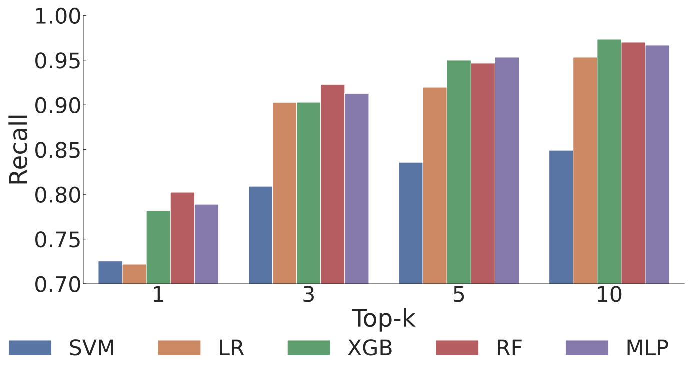


Figure 6.4: Top-k scores of 10-fold validation of ML models.

6.3.1.3 Performance Metric

We report the top-*k* score, a widely-used metric for evaluating information retrieval systems [141], which represents the portion of websites containing preference buttons detected from the top *k* classification results. A website is successfully detected if one of its preference buttons is among the top *k* buttons with the highest classification probabilities. While the `ConsentChk` button activator can try multiple preference-button candidates, it is desirable to reduce the number of trials `ConsentChk` needs to perform to open a preference menu, i.e., achieving a higher top-*k* score at a lower *k*.

6.3.1.4 Model Selection

We try multiple ML algorithms to find the best performing classifier based on top-*k* scores ($1 \leq k \leq 10$) of 10-fold cross validation on the training set. The models include logistic regression (LR), single-hidden-layer multi-layer perceptron (MLP), random forest (RF), support vector machine (SVM) and XGBoost. We use grid search to find the regularization *C* (SVC and LR), the number of decision trees (RF and XGBoost) and hidden layer size (MLP) that maximize top-1 scores. Appendix D.1.1 lists the tuning ranges.

A random forest (RF) with 100 decision trees has the best top-1 score of 80.22%, so we select this model for later experiments. It outperformed other models on top-1 and top-3 scores while having similar top-5 scores with other classifiers Fig. 6.4 shows the top-k scores of the models. Furthermore, an ablation study of the random forest model shows that feature groups G_2 and G_3 greatly increase the accuracy by increasing the top-1 scores by 2.4% and 2.7%, respectively. Appendix D.1.2 shows the scores of each feature group in the ablation study.

6.3.1.5 Performance Evaluation

To obtain an unbiased estimate of the classifier performance, similar to the training set, two authors annotated 200 randomly selected websites from the 2k held-out domains in Section 6.3.1.1. The resultant test set has 57 web pages containing 65 preference buttons of 10,614 buttons in total. The results demonstrate the high performance of the MLP model with top-1, top-3, top-5 and top-10 scores of 77.19%, 85.96%, 85.96%, and 89.47%, respectively.

6.4 Consent Preference Extraction

After a cookie setting menu appears, a user can set the consent for the cookies on the website. In this section, we describe the extraction of the cookie consent and the design of an (un)consent tool.

6.4.1 Definitions

Considering a cookie in a browser's cookie store as a tuple of key-values, `ConsentChk` distinguishes cookies in a browser cookie store by their *name*s, *domain*s and *path*s. This cookie distinction follows the storage model in the cookie specification [29]. Other cookie attributes, such as *value* and *expiration time*, may change over time for the same cookie.

Definition 6.4.1 (Cookie Equivalence)**.** *Let* $c_k = (n_k, d_k, p_k)$ *and* $c_l = (n_l, d_l, p_l)$ *denote two cookies, where n, d, and p represent the name, domain, and path of each cookie,* *respectively. The cookies c_k and c_l are equivalent, denoted as $c_k \equiv c_l$, if and only if $n_k = n_l \wedge d_k = d_l \wedge p_k = p_l$.*

Informally, a "cookie consent preference" is the acceptance or rejection of the consent of a cookie's usage. After the consent preferences are set, the cookies can be divided into two potentially overlapping sets of approved or rejected cookies.

Definition 6.4.2 (Cookie Consent Preference)**.** *A cookie consent preference $p_i = (c_i, s_i)$ is a pair of a cookie c_i and a consent choice s_i where $s_i \in \{consent, not\_consent\}$ indicates whether the consent for usage of c_i is approved or rejected, respectively. A user's cookie preferences P on a website are the set $P = \{p_i | p_i = (c_i, s_i)\}$ that associates each cookie $c_i \in C_s$ with his/her consent choice s_i.*

Definition 6.4.3 (Approved and Rejected Cookies)**.** *The set, A_c, of the cookies that are consented (approved) by the user is represented as $A_c = \{c | (c,s) \in P \wedge s = consent\}$, and the set, R_c, of the cookies that are not consented (rejected) by the user as $R_c = \{c | (c,s) \in P \wedge s = not\_consent\}$. Then, the set of cookies P_c in the cookie preference set P is the union of the approved and rejected cookies $P_c = \{c | (c,s) \in P\} = A_c \cup R_c$.*

6.4.2 Categorized Consent Analysis Framework

We observe that CMPs commonly group consent settings into categories of cookies to simplify the consent process. Therefore, we derive a *categorized consent* analysis framework that groups cookies into categories and provides the list of cookies of each group. In this framework, the set of cookies used on a website is divided (by purposes or vendors, for example) into subsets, called *cookie categories t_k*. A consent choice of cookie category t_k applies to all cookies in that category. For example, the consent rejection of *krxd.net* domain cookie category (Fig. 6.1) applies to all cookies from that domain. Fig. 6.5 depicts our framework.

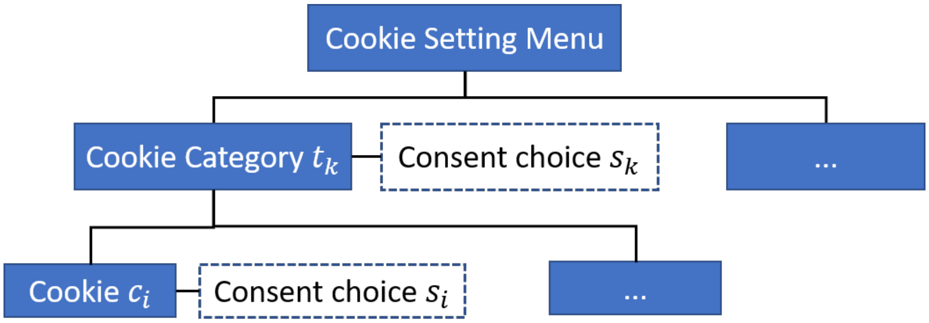


Figure 6.5: Categorized consent analysis framework.

6.4.3 (Un)consent Tool and Consent Extractor

Using the categorized analysis framework, the cookie consent preferences are extracted into 2 steps: extracting (1) cookie-category consent and (2) cookies in each category. Since websites employ various designs and customizations, automatically setting cookie consents via the UI is challenging. The UI control of a "cookie consent setting" is implemented using different UI elements, such as check boxes, sliders, or switches. By using the analysis framework, `ConsentChk` supports different numbers of cookie categories of different consent libraries but requires only an one-time analysis for each cookie library. For example, Cookiebot supports 4 cookie categories while Termly supports 6 cookie categories.

Automatic (Un)consent Tool. To analyze a specific cookie setting instance, its UI controls need to be mapped to the components in the analysis framework. The main manual effort is to map the HTML elements to the corresponding category consent categories. For each CMP, we analyze its UI variants in the cookie setting dataset. We use the Chrome DevTools to identify CSS selectors that uniquely identify UI elements on the layout. Although the identification of the mapping is done manually, we need the manual mapping only once for each of the limited number of cookie setting layouts provided by each cookie library.

Cookie Consent Preference Extractor. After setting the consent on the UI, to extract the consent of each cookie recorded by the cookie library, we extract the consent for each category and the list of cookies for each category. Combining these two lists, we get the consent for each individual cookie. For example, OneTrust stores consents of categories in *OptanonConsent* cookie and the lists of cookies per cookie category in *en.json*.

6.5 Cookie Consent Violation Detection

6.5.1 Cookie Flows

Informally, a *cookie flow* represents the data collection of a website by transferring a cookie placed on the user's web browser to the website's server. It is formalized as follows.

Definition 6.5.1 (Cookie Flow)**.** *A cookie flow is the data collection using a cookie c by a receiver r and represented by a pair $f = (r, c)$.*

To extract the set of cookie flows $F = \{(r_i, c_i)\}$ during a website visit, `ConsentChk` extracts the cookies sent to servers via the network debugging functionality of Chrome DevTools Protocol that reports all HTTP(S) requests with associated cookies [151]. Dynamic analysis is an advantage in that it reveals 'real' occurrences of cookie flows rather than finding only potential ones, thus reducing false positives. For example, a website may block the use of third-party cookies by preventing the loading of third-party scripts and frames without actually removing the cookies. Checking on the actual cookies being used by the websites overcomes the limitations of prior work that either extracts all cookies in the browser regardless whether the cookies were transferred to the servers or not [265].

We assume that a website should enforce a cookie consent preference if 1) the consent is recorded in the browser and 2) the web page being visited is within the scope of the consent. These *Consent-Enforcement Conditions* ensure the consent of a cookie applies only when the user browses the web pages — not other websites — to which the user gave his/her consent. For example, the user's consent given to website *a.com* does not apply to website *b.com*.

The scope of a consent preference is defined by the domain of the consent cookie (such as *OptanonConsent* of OneTrust) or a local storage object which records the user consent. For example, the consent cookie with domain *.a.com* applies to all subdomains of *a.com*, such as *subpage.a.com*.

6.5.2 Cookie Consent Violation Types

6.5.2.1 Rejected Cookie Usage

A Rejected Cookie Usage violation occurs when the website uses a cookie that is rejected and not approved by the user. Using explicitly rejected cookies violates user consent and consumer-protection laws such as the GDPR and the FTC Act. We separately consider the cases when the cookie has contradictory consent preferences, e.g., the cookie's consent is approved and rejected at the same time in two different cookie categories (see Section 6.5.2.2).

Definition 6.5.2 (Rejected Cookie Usage)**.** *A cookie flow $f = (r, c)$ has a Reject Cookie Usage violation regarding a set of cookie preferences $P = A_c \cup R_c$ if and only if the cookie in the flow is rejected and not approved in P, i.e., $c \notin A_c \wedge c \in R_c$.*

6.5.2.2 Ambiguous Cookie Consent

When the categories of cookie settings overlap, a cookie can be both rejected and approved by a user. The ambiguous consent choices may happen when the CMP does not properly classify cookies into non-overlapping categories. For example, *oclc.org* contained a cookie named *GPS* with host *youtube.com* that was used to track the user's location [63] and included in both always-active Strictly Necessary and optional Targeting cookie categories. So, when a user rejects this cookie in the Targeting category, the rejection contradicts the consent given in the Strictly Necessary category.

The ambiguous consent choices violate the GDPR and the FTC Act, because they mislead users to reject a cookie in one category but the cookie is still consented in another category. The contradictory preferences are also akin to the problematic contradictory statements in privacy policies identified by the FTC and prior studies [20, 46, 249].

While websites may set precedence over cookie preferences to automatically resolve contradictory preferences, such a precedence is not always stated explicitly to users. For example, consented cookie preferences may take a higher precedence over unconsented preferences, i.e., $(c, consent)$ and $(c, not\_consent)$ can be resolved to $(c, consent)$. However, we have not found any description of the cookie settings of randomly sampled 50 websites from the cookie setting corpus (Section 6.3.1.1), indicating such an automatic precedence may not have been implemented or stated explicitly.

Definition 6.5.3 (Contradictory Consent Preference)**.** *Two cookie preferences* (c_k, s_k) *and* (c_l, s_l) *are said to be contradictory if and only if* $c_k \equiv c_l \wedge s_k = consent \wedge s_l = not\_consent$.

An Ambiguous Consent violation occurs if the website uses a cookie with a contradictory consent preference. The cookie usage violates user consent because the user already rejected the cookie. However, we distinguish this violation type from the Rejected Cookie Usage which does not have any consent for the cookies.

Definition 6.5.4 (Ambiguous Consent)**.** *A cookie flow* $f = (r, c) \in F$ *is said to have an ambiguous consent if the cookie* c *is both approved and rejected by the user, i.e.,* $c \in A_c \wedge c \in R_c$.

6.5.2.3 Consent Choice Omission

Omitting consent choices of a cookie prevents users from giving consent to the cookie, so using the cookie lacks user consent and violates privacy laws such as the GDPR and CCPA regulations which require opt-out choices for the collection and sale of user data on a website. The Consent Choice Omission violation is different from the *unclassified* cookie category (frequently used by CMPs like Cookiebot and Termly) which still informs users of the usage of the cookies in this category.

Definition 6.5.5 (Consent Choice Omission)**.** *A cookie flow $f = (r, c)$ has its consent choice omitted if and only if the cookie c is neither approved nor rejected by the user, i.e., $c \notin A_c \wedge c \notin R_c$.*

6.5.2.4 Correct Cookie Consent Enforcement

A cookie flow is said to be consistent with user consent preferences if the user approves and does not reject the cookie. So, a website correctly enforces user consent if all of the cookies used on the website are consistent with the user preferences. However, checking the correctness of cookie consent enforcement requires to check all possible cookies that the website may use on all of its web pages over time.

Definition 6.5.6 (Flow-to-preference Consistency)**.** *A cookie flow $f = (r, c)$ is said to be consistent with the cookie preference set P if and only if the cookie in the flow is approved and not rejected, i.e., $c \in A_c \wedge c \notin R_c$.*

Definition 6.5.7 (Correct Cookie Consent Enforcement)**.** *A set of cookie consent preferences P is correctly enforced by a website if and only if all cookie flows $F = \{f_i\}$ of the website are consistent with P.*

6.5.3 Implementation

6.5.3.1 Mapping Cookie Declarations to Browser Cookies

`ConsentChk` maps the cookie declarations in the cookie settings to the cookies used by the website by matching the cookie names and domains. In the simplest case, the cookie and the declaration have exactly matched names. However, when cookie names in the declarations are specified as a pattern, such as *_gaxxx* or *_ga###*, we assume an ‘x’ or ‘#’ to match any single character. However, a single ‘#’ in the end of the declaration matches any alpha-numeric string. For example, the declared cookie name "_gatxxx" matches cookie "_gat123". These kinds of cookie name patterns only constituted 3.8%

cookies declarations in our large-scale study (Section 6.6). Similarly, the domain names of a cookie and a declaration match each other if the declared domain is a suffix of the cookie domain. This domain matching scheme is similar to the standard cookie domain matching specification [29].

6.5.3.2 URL-to-Cookie-Domain Matching

In order to verify the consent cookie (such as *OptanonConsent* of OneTrust) to match the URL of the currently visiting web page, `ConsentChk` matches URLs with cookie domains by following the cookie-matching implementation of Chromium's networking stack [152, 153] as it is the browser used in our experiments. While the cookie domain-matching algorithm is specified in RFCs [29], browsers have their own implementations [153]. For example, a consent cookie with domain *.example.com* matches all web pages with URLs under the corresponding subdomains like *https://subsite.example.com*.

6.6 A Large-Scale Study

6.6.1 Experimental Setup

6.6.1.1 Cookie Library Selection

We follow the approaches of Bollinger *et al.* [40] to select the CMPs that declare individual cookies. A CMP is selected if it provides 1) a cookie list of each category and 2) the declaration (name and domain) of each cookie. From the list of 44 popular CMPs on the top 1M sites reported by BuiltWith [47], we selected four cookies libraries: OneTrust, CookiePro, Cookiebot, and Termly. These are the top CMPs that represent a 3.53% market share. Each of OneTrust and Cookiebot has a >1% market share while CookiePro and Termly have 0.23% and 0.12%, respectively. The libraries we select is the same as those in [40], because we use the same selection criteria and the CMPs have not changed much

CMP	Market Share	Cookie List?	Cookie Decl.?
Osano	2.18%	✓	✗
OneTrust	1.98%	✓	✓
CookieYes for WP	1.30%	✗	✗
WP CookieNotice	1.21%	✗	✗
Cookiebot	1.20%	✓	✓
IAB Europe TCF	1.15%	✗	✗

Table 6.3: The most popular CMPs with more than 1% market share on the top 1M websites as reported by BuiltWith [47]. The last two columns denote the criteria for the CMPs to be suitable for analyzing consent violations of each cookie. *WP* and *Decl.* stand for WordPress and declaration, respectively.

Consent Library	Cookie Category Consent	Cookie Declaration
OneTrust	*OptanonConsent* cookie	*en.json*
Cookiebot	*CookieConsent* cookie	*cc.js*
Termly	*TERMLY_API_CACHE* local storage	*cookies* (JSON)

Table 6.4: Consent storage objects of the CMPs.

since their work. Table 6.3 shows the market shares and the satisfied criteria of the CMPs. Since CookiePro is now part of OneTrust, we report results for the remaining three CMPs. While companies have been fined for not including full descriptions of cookies, we expect the number of CMPs including cookie lists per category for automatic analysis to increase in the future.

The selected CMPs support different cookie categories. The four commonly supported categories are *Necessary*, *Functional*, *Analytics* and *Targeting*. While Cookiebot and Termly use 4 and 6 fixed cookie categories, respectively, OneTrust support varying cookie categories. For example, *scientificamerican.com* uses *Social Media Cookies*, a customized category of OneTrust. These CMPs store the consent of categories and cookie lists per category in consent cookies and local storage, e.g., OneTrust uses the *OptanonConsent* cookie. Table 6.4 lists the data objects that store the categorical consent and cookie declarations of the CMPs. See Appendix D.2 for the decoding of consent cookies.

6.6.1.2 Measurement Setup

Website Selection. From the top 200k global websites in the Tranco list May 2022 (ID: PZ6PJ), we select 101,703 websites, which have an English home page and were loaded successfully, for further analysis with `ConsentChk`. We use the most up-to-date list at the measurement time to avoid the domains that become non-existent. Some websites in the list failed to load due to various issues like non-website ad-serving domains. The language of the websites is determined by a neural-network-based language detector after converting the web pages to plain text [131, 266].

Measurement Procedure. Given a website domain W, `ConsentChk` first opens a clean web browser instance, visits the home page of W and detects a cookie preference button to open the cookie-setting menu. The system then rejects the cookie consent, reloads the home page where the cookie preferences were submitted, and checks the Consent-Enforcement Conditions to ensure the consents have been recorded. After this step, the website is expected to enforce the consents. Finally, `ConsentChk` visits other 5 sub-pages that have hyperlinks on the home page and their URLs matching the domain of the consent cookie. This step generates more cookie traffic because the website may use additional cookies on its subsites [145]. We assume the cookie preferences take full effect starting from the loading of next page as the cookie blocking methods provided by CMPs [69, 74] can block cookie-loading scripts of third parties only during web page loading but cannot block already-loaded scripts.

We used `ConsentChk` to reject all rejectable cookie categories to minimize the number of consents given to a website. Checking all consent combinations, which grow exponentially with the number of consent choices, is infeasible and inefficient to detect consent violations. See Appendix D.3 for implementation details of the (un)consent tool.

We choose the value k=5 for `ConsentChk` to try the top-5 preference-button candidates as a trade-off between the coverage and experimentation duration. On average, each page contains 232 buttons and links, so using only the top 5 links/buttons reduces the experimentation time significantly while still achieving a high recall rate. Increasing *k* forces the system to check all *k* buttons on the websites that do not contain any preference buttons and increases the experimentation time significantly.

The crawler uses a 60-second timeouts to load the pages. We found this timeout sufficient to completely load most of the web pages with the fast network of our servers and cloud providers. In case the timeouts are too short for a cookie to load, there are no false positives because the inconsistencies require the presence (not the absence) of a cookie. The crawler uses Playwright browser automation [217] to control the Google Chrome web browser. It also utilizes techniques provided by an automatic browsing plugin to avoid being detected by bot detection (i.e., stealth mode) [32].

We measured the websites from IP addresses in the UK and the US. Unless stated otherwise, the reported results are from the UK. The experiments were conducted in distributed experiment framework based on Docker Swarm [159] on 8 machines with 1.08TB RAM and 128 task queue workers. The cookie-consent scanning of the 101,703 websites took 40 hours to perform the measurements from the two locations.

6.6.2 Extraction Results

Of the selected 101.7k websites, `ConsentChk` successfully analyzed the flow-to-consent consistency of 6,050 (5.95%) websites. OneTrust and Cookiebot are the most popular, appearing on 5,219 and 778 sites, respectively. Termly was detected only on 53 sites. These detected cookie settings reflect the relative market shares of the CMPs.

From the analyzed websites, `ConsentChk` extracted 709,259 cookie declarations. The number of cookie declarations per website varies greatly with an average of 117.2 (147.9 *SD*), ranging from 2 to 2,658 cookies. The system extracted 7.7M cookie flows of 124,830 unique cookies from 32,196 web pages.

Consent Violation Type	UK		US	
	# Cookies	# Websites	# Cookies	# Websites
Rejected Cookie Usage	43,697	82.20% (4,973/6,050)	39,446	81.86% (4,134/5,050)
Consent Choice Omission	52,804	85.02% (5,144/6,050)	70,547	89.05% (4,497/5,050)
Ambiguous Consent	398	4.20% (254/6,050)	345	4.02% (203/5,050)

Table 6.5: Detected consent violations of cookie usage.

6.6.3 Findings

6.6.3.1 Finding 1: The Majority of Websites Use Rejected Cookies

ConsentChk found 82.20% (4,973/6,050) of the websites still used at least one cookie a user rejected. Each website used an average of 10.3 unconsented cookies. The total number of the cookies is 43,697 that constitute 42.3% of all the extracted cookies. This finding indicates that users cannot opt out of cookie usage even with explicit consent rejection.

Consent-Violated Cookies. The most common consent-violated cookies are tracking cookies of Google (_ga and _gid) and Meta (_fbp). This finding highlights an important fact that users cannot opt out of tracking even when they *explicitly reject* the consent of such types of cookies. Prior studies [208, 241, 265] reported that tracking still continued despite the users' rejection but their coarse-grained analysis could not pinpoint the exact violating cookies. Table 6.7 shows the most common cookies with a Rejected Cookie Usage violation.

Our evaluation of the Rejected Cookie Usage violations has a significantly higher coverage than prior work. The total market share of our supported three CMPs is 3.53% which is 3X more than those of Bollinger *et al.* [41] and Matte *et al.* [208]. [41] only measured Cookiebot and [208] only studied TCF, which have 1.2% and 1.15% market shares, respectively.

Consent-Violated Trackers. To quantify the third-party trackers that own Incorrect Enforcement cookies, we count the number of the websites on which each tracker placed

CMP	# Violations
OneTrust	4,370 (83.73%)
Cookiebot	553 (71.08%)
Termly	50 (94.34%)

Table 6.6: CMPs with Rejected Cookie Usage.

its cookies. The most common cookie domains are found to be *doubleclick.net* and *youtube.com* which are tracking domains of the same owner *google.com*. Table 6.8 lists the top trackers.

Website Categories. To assess the distribution of the website categories that contain the detected Incorrect Enforcements, we categorized the websites using the FortiGuard Web Filter Categories [106]. The most common category is Business (1,515 sites). We conjecture that business websites frequently provide cookie settings to comply with cookie laws but the CMPs do not strictly enforce the consent, making `ConsentChk` detect a high number of Rejected Cookie Usage violations on their sites. Fig. 6.6 shows the top website categories with this violation type.

Cookie Categories. To analyze the cookie categories displayed to users in cookie settings, we perform simple aggregation of synonymous category names. Categories containing "required", "essential" are aggregated into the "Necessary" category. Similarly, "targeting" categories which commonly mean "targeted advertising" are aggregated into the "Advertising" category. While the category names has a long tail distribution, we opt not to aggregate other minor label variations because they comprise a very small percentage of the dataset.

Of the cookie categories in which `ConsentChk` detected Rejected Cookie Usage, the number of cookies per categories is 36.71 on average (64.29 *SD*) and varies greatly from 1 to 2,409. Similarly, the number of trackers is 9.90 on average (14.46 *SD*) and ranges from 1 to 160. The category with the most number of cookies and trackers is Advertising. Table 6.9 shows the detailed statistics. This high number of cookies per category highlights the need

Cookie Name	# Websites
_ga	2,542
_gid	2,442
_fbp	1,374
IDE	938
YSC	875

Table 6.7: Top cookies with Rejected Cookie Usage.

Tracker	# Sites (%)
doubleclick.net	1,128 (22.68%)
youtube.com	882 (17.74%)
linkedin.com	803 (16.15%)
linkedin.com	791 (15.91%)
nr-data.net	413 (8.30%)

Table 6.8: Top trackers of rejected-usage cookies.

Category	# Cookies		# Trackers	
	Mean	SD	Mean	SD
Advertising	60.18	80.69	19.49	19.77
Performance	22.42	37.04	3.99	6.02
Functional	23.83	67.23	5.35	5.04
Statistics	41.96	39.57	8.00	5.23
Overall	36.71	64.29	9.90	14.46

Table 6.9: Categories and trackers of rejected-usage cookies. *SD* stands for standard deviation.

to pinpoint the exact consent-violated cookies rather than debugging the whole category. Fig. D.2 (Appendix D.4) shows the distribution of cookie declarations per category in cookie settings.

Violation Rates of CMPs. Of the three measured CMPs, Termly has the highest violation rate of 94.34% (50/53). However, OneTrust has the highest absolute number of the violating websites with 4,370 (83.73%) sites since it was deployed more widely. Table 6.6 shows the violation rates of CMPs.

6.6.3.2 Finding 2: Cookie Settings Provide Users With Contradictory Cookie Preferences

`ConsentChk` detected 1,107 cookie settings on 11.17% (676/6,050) websites that can be used to set contradictory cookie consent preferences. These settings are for those cookies that have the same *Name* and *Host* values but are included in two different categories

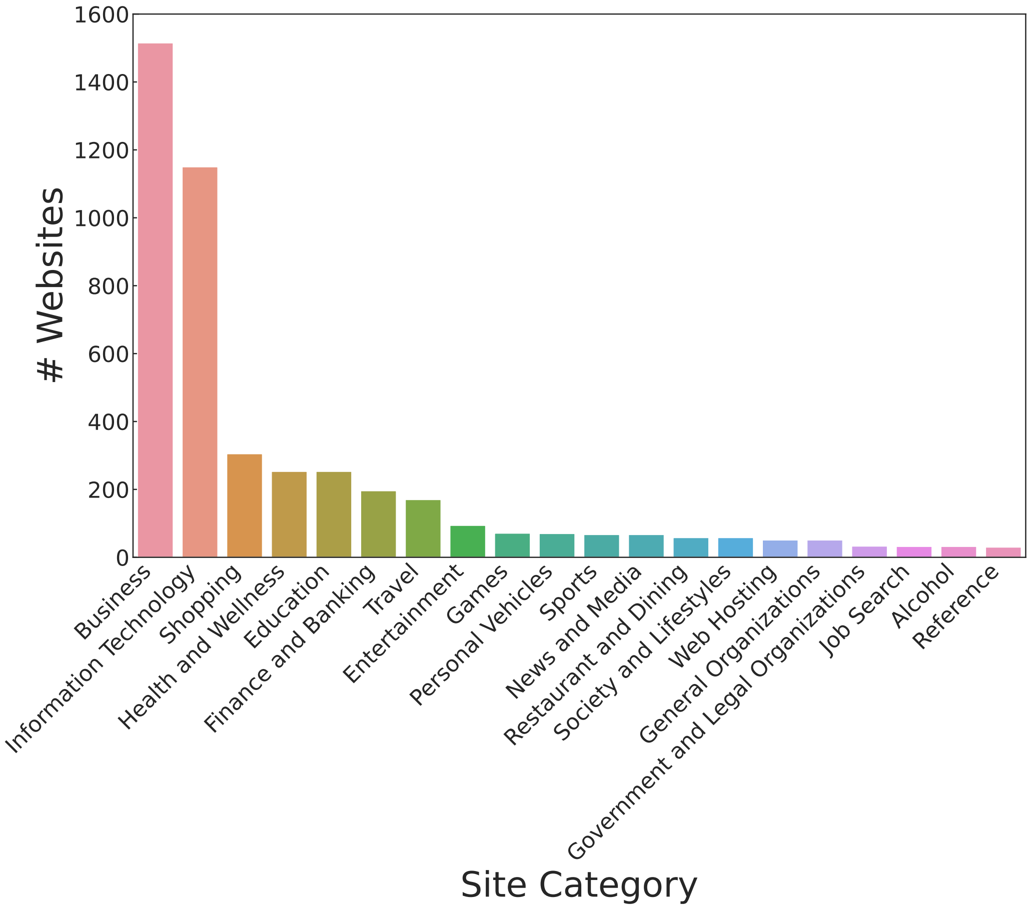


Figure 6.6: Categories of websites with Rejected Cookie Usage.

that can be set to contradictory consent preferences. Such contradictory settings make the interpretation of cookie notices ambiguous and confuse users and automatic analyzers.

In the majority 97.65% (1,081/1,107) of the detected contradictory-setting pairs, each pair contains an "always-active" cookie category *and* another category that can be rejected. The minority of the pairs allow users to reject/approve consent on both categories. For example, *location_data* cookie of *cosmopolitan.com* was listed in both always-active Necessary and rejectable Functional categories, and hence can be accepted and rejected simultaneously on the website.

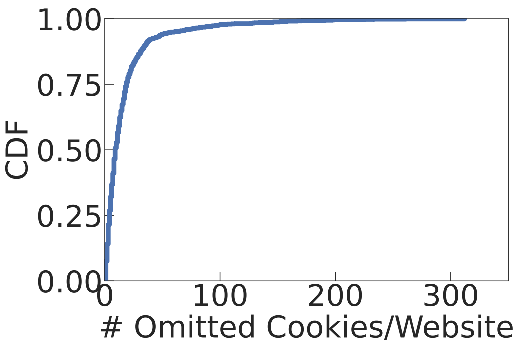
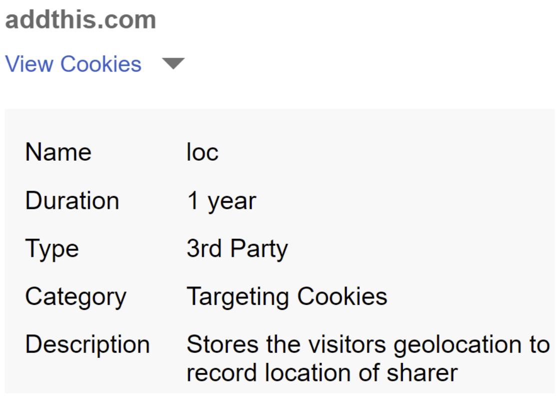


Figure 6.7: CDF of omitted cookies per website.



Name	loc
Duration	1 year
Type	3rd Party
Category	Targeting Cookies
Description	Stores the visitors geolocation to record location of sharer

Figure 6.8: Geographic data type of cookie *loc* of *addthis.com*.

6.6.3.3 Finding 3: Websites Contain Cookies with Ambiguous Consent Caused by Contradictory Consent Preferences

`ConsentChk` captured usage of 398 unique cookies with an ambiguous consent in 4.20% (254/6,050) websites (Table 6.5). These cookies were included in 2 different categories in which one was *Approved* while the other was *Rejected*. A detected cookies with ambiguous consents indicate that the cookie libraries do not guarantee the consistency of the user's choices. These contradictory preferences and ambiguously-consented cookies actually *did occur*. Prior work [41] reported this kind of cookies which was included in multiple categories but did not quantify its actual usage on websites.

6.6.3.4 Finding 4: Most Websites Fail to Completely Declare the Cookies Used on Them

`ConsentChk` found the cookies flows that were not listed in the cookie declarations of 85.02% (5,144/6,050) websites. Because a main functionality of cookie settings is to inform users of the privacy practices of the website, failure to declare cookies may mislead the users to believe that the website is not collecting certain types of data (e.g., location and time of visits). The omitted preferences may be due to the dynamics of websites that may load additional cookies at runtime. For example, the websites with the least number of cookie declarations (e.g., *technicalseo.com*) omitted most of the cookies used while keeping a few cookies of the integrated cookie libraries. However, our finding indicates the prevalence of the omitted cookie flows, even when `ConsentChk` has not made any complex interaction with the web pages, such as login or form submission. The number of omitted cookies per website ranges from 1 to 109 with an average of 10.2 (10.6 *SD*). Fig. 6.7 shows the cumulative distribution of the number of the omitted cookies per website.

6.6.3.5 Finding 5: Few Websites Correctly Enforce Cookie Consent Preferences

Only 6.25% (378/6,050) websites were found to correctly enforce the cookie consent preferences of users. Due to the limitation of dynamic analysis that cannot verify all possible cookie flows on a website, `ConsentChk` provides only an upper bound of the correct enforcement of the cookie consent preferences. Generating more cookie flows on the websites might detect additional inconsistencies of these preferences. However, missing them does not increase the false positives that we are trying to minimize.

6.6.3.6 Finding 6: Consent Violations Do Not Change Significantly When Accessing from Outside of the EU

While many websites outside of the EU did not show cookie banners for first-time visitors and even completely hid their cookie settings, `ConsentChk` still detected a similar number of consent violations to that in the EU. `ConsentChk` found cookie settings on 5,050 websites when accessing from the US, which is 16.5% less than the UK-based measurement. However, compared to the UK-based measurement, the percentage of the Consent Choice Omission is higher at 89.05% (4,497/5,050) while the Rejected Cookie Usage is slightly lower at 81.86% (4,134/5,050). The relative order of frequency the consent violation types does not change where the Consent Choice Omission is the most common violation. Table 6.5 shows a comparison of the measurements from the UK and the US.

6.6.4 Root Cause Analysis

While prior work [41, 291] has identified several causes of violations, we conducted an in-depth analysis of cookie-blocking mechanisms to identify additional potential causes as follows.

Incorrect Integration of Cookie-Blocking Scripts. One of the most common ways of blocking 3rd-party cookies is to add a cookie-category tag provided by CMPs to 3rd-party scripts on the website. This kind of cookie-blocking tags automatically blocks the loading of the 3rd-party scripts when the corresponding cookie category is rejected [24, 69, 74]. For example, OneTrust provides a special class, like <*script class="optanon-category-C0002"*>, which blocks the execution of the inside JavaScript code if cookie category *C0002* (Functional cookies) is rejected. Similarly, Termly provides special HTML attributes such as *data-categories="analytics"* to mark and block the scripts that belong to the Analytics category when this category is rejected. Since cookies are transferred along with HTTP requests to the servers that host the advertisers' scripts or tracking pixels, preventing the loading of embedded code effectively blocks the usage of the cookies. This blocking only prevents the usage of the cookies while not deleting them from the browser.

However, our sampled websites show that many of them left 3rd party scripts uncontrolled by the CMPs, and hence 3rd-party cookies were still loaded even after a cookie category was rejected. Similarly, websites appeared not to block their own 1st-party cookies, especially the analytics cookies shown in Table 6.7. For example, on *scientificamerican.com*, we manually rejected *Performance cookies* to measure the websites' traffic which included 1st-party analytics cookies. However, analytics cookies, such as the popular Google Analytics *_ga*, were still present. The Google Analytics cookie is crucial to track users throughout the site, so the websites have an incentive to keep it to track their users.

Integrating the cookies takes time and effort, especially if the websites are continuously updated. So, if the website developers neglect the cookie-blocking integration, the blocking mechanism will not work. Therefore, the faulty CMP integration is likely to be one of the causes of the Rejected Cookie Usage.

Incompleteness of Automatic Cookie Scanners. While CMPs scan for 3rd-party iframes/scripts on a website to declare 3rd-party cookies in the cookie settings and automatically block them, we found many cookies only appear in certain pages with embedded content, making automatic cookie detection challenging. For example, only web pages embedding a YouTube video use cookies from *youtube.com*, so the automatic scanning will miss YouTube cookies if it does not scan the video web pages. YouTube cookies are among the top consent-violated cookies (Table 6.8). Therefore, the incompleteness of the CMPs' automatic scanners is likely to cause both Rejected Cookie Usage and Consent Omission violations.

6.6.5 Data Types of Cookies

We found that the data type of a cookie is not just a generic type like "cookies" or "unique identifier" but includes more specific data objects contained in the cookie's value. For example, if a cookie contains the latitude and longitude of a user, then its data type is "precise geographic location" or "GPS coordinates", instead of just "cookies" or "unique ID". A cookie may contain multiple data types if its value is complex, e.g., a set of key–value pairs that comprise multiple data values.

Fig. 6.8 shows an example cookie description embedded in the OneTrust cookie settings on *businesswire.com*. The cookie *loc* with domain *addthis.com* contains data type "geolocation" of users to help *addthis.com* track their location when its share buttons are clicked. We decoded this cookie with Base64 decoding and found our postal code, state and country names.

6.7 Evaluation

We evaluate the detection performance of ConsentChk, the ambiguity of cookie-declaration mapping and analyze ConsentChk in comparison with prior work.

6.7.1 End-to-end Detection Performance

We evaluate the precision of detecting Rejected Cooke Usage and Consent Choice Omission violation types by manually rejecting cookie consent on the detected websites. We randomly selected 40 websites from those with these 2 violation types in Section 6.6. The sites include 34 OneTrust, 5 Cookiebot and 1 Termly CMPs. We could not evaluate the recall rate because the dynamic analysis cannot reveal all possible consent violations that a website may have.

For each website W, we manually created cookie flows after cookie rejection. Using a clean instance of the Chrome browser, we navigated to W, rejected any rejectable cookie categories, and visited sub-pages of W. Finally, we recorded the consented/rejected cookie preferences and the transferred cookies by using the DevTools network monitor.

To verify the correctness of the detection, we checked whether each rejected cookie with Rejected Cookie Usage discovered by ConsentChk was captured in the manual browsing or not. Similarly, we checked whether cookies with detected Consent Choice Omission were unspecified in the cookie settings of the website or not. We automated simple checking, such as when cookie names and domains exactly match those in the cookie declarations. However, two annotators manually checked and discussed ambiguous cases (e.g., cookie names were declared as *_ga_#*), to determine whether the detection was correct or not.

Our results demonstrate the low false positives of the rule-based detection pipeline. We manually reproduced and verified 92.1% (257/279) cookie flows with Rejected Cookie Usage and 91.2% (364/399) flows with Consent Choice Omission detected by ConsentChk. All the reproduced flows were correct detections. The remaining cookie flows were not

# cookies/declaration	# declarations
0	207,857
1	25,035
≥ 2	130

# declarations/cookie	# cookies
0	25,961
1	23,593
2	888

Table 6.10: Mapping between declarations and browser cookies.

reproduced due to the random placement of advertising cookies and the inclusion of random IDs in the cookie names.

Ambiguous Consent Choice Detection. We randomly selected 30 websites with detected Ambiguous Consent Choice violations. These sites use 26 OneTrust, 3 Cookiebot and 1 Termly. We checked the correctness of the detected Ambiguous Consent Choice cookies by reading the cookie declarations on the UI of 87% (26/30) sites and storage objects of CMPs, such as *en.json*, of the remaining sites. This way, we verified all the detected violations to be correct.

6.7.2 Mapping of Cookie Declarations

We analyze the ambiguity of the mapping from cookie declarations to browser cookies using the scheme in Section 6.5.3.1. There exist certain ambiguities due to underspecification of cookie names and domains in the cookie settings. For example, the host names of cookies may be declared as *Glassdoor* which may match both *glassdoor.com* or *glassdoor.ie*. However, our evaluation results show such an ambiguity makes minimal impact on the detection accuracy because trackers usually generate/use unique cookie names to distinguish their own cookies.

An ambiguity occurs when the 1:1 mapping does not hold, e.g., one declaration maps to multiple cookies, or vice versa. Incorrect mappings may create false positives, such as when a cookie rejection maps to multiple cookies, causing some to be unnecessarily rejected. The mapping between cookie declarations and browser cookies is inherently ambiguous because a cookie declaration only specifies a "host" which does not strictly follow the specification of cookie domains [29] and there is no specification for interpreting a "host" used in cookie settings. Note that this ambiguous mapping differs from Ambiguous Enforcement.

We found that the majority of mappings are 1:1, thus making the mapping ambiguity low. As shown on Table 6.10, of the declarations and cookies with a mapping, only 0.06% of declarations map to multiple cookies and 1.8% of cookies map to multiple declarations. Because each cookie is uniquely identified by name, domain and path, we found that a tracker commonly used different cookie names to differentiate its own cookies under the same domain. Furthermore, except for common names like "uid" or "uuid", cookies of different trackers are frequently found to have different names. This unique naming of cookies reduces the mapping ambiguity.

6.7.3 Comparison to Cookie-Category Analysis

While the enforcement of users' cookie consent preferences to the cookie placement of websites has been analyzed before, prior studies [208, 241, 265] provided only a *coarse-grained* analysis at the level of *cookie categories*, each of which consists of tens to hundreds of cookies. As shown in Section 6.6.2, each inconsistent cookie category comprises 36.71 cookies on average and up to 160 trackers, `ConsentChk` saves significant amounts of time and effort to identify the problematic cookies over prior work that only detects violation at the cookie-category level. Furthermore, a cookie contains not only a tracking ID but also other data types, such as the user's geolocation (Fig. 6.8). Therefore, analysis at the cookie-category level cannot precisely capture the receivers and data types that are leaked even after a user rejected the use of cookies.

6.8 Browser Extension

We have developed `ConsentEnforcer`, a browser extension, to help end-users *audit* and *enforce* their cookie preferences on the websites they visit. (This will soon be open-

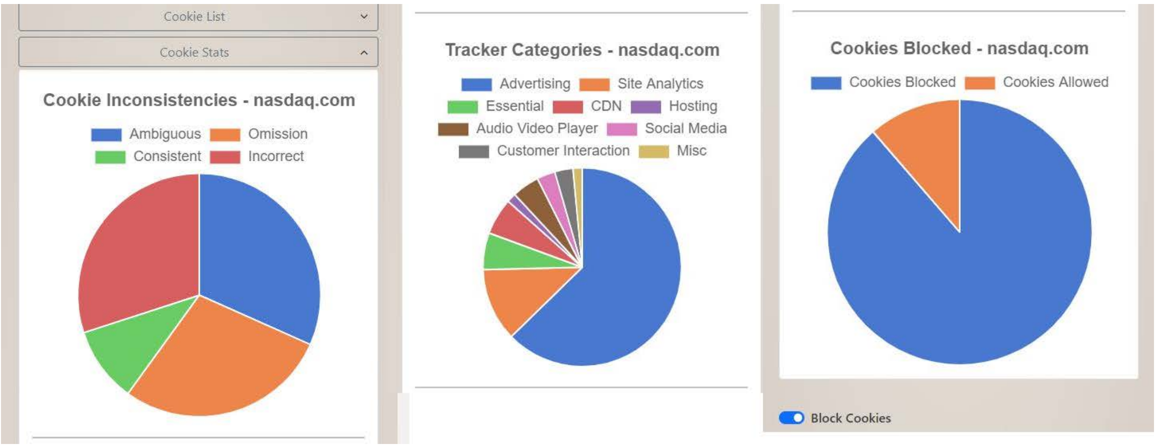


Figure 6.9: User interface of `ConsentEnforcer` extension.

sourced at [65].) The extension comprises a cookie auditor and a consent enforcer. The auditor collects and displays the detected flow-to-preference inconsistencies and categories of the cookie receivers on a website. `ConsentEnforcer` enforces the user's rejection by removing rejected cookies from HTTP(S) requests and responses. By blocking the rejected cookies in the network traffic, `ConsentEnforcer` blocks only the cookies under the scope of the user consent while keeping other cookies' behavior unchanged to avoid disruptive user experience. Fig. 6.9 illustrates the interface of `ConsentEnforcer`.

We conducted a small-scale IRB-approved user study of 38 Amazon Mechanical Turk (MTurk) workers to evaluate users' perceptions towards cookie inconsistencies and usability of `ConsentEnforcer`. 97.37% of the participants would feel their privacy was violated if a website continued to exchange cookies even after the users' rejection. 21.05% of the users even claimed they would go as far as filing a complaint to a website that violated their consent/rejection. Of the 12 users who downloaded and installed our extension, 83.33% found the extension easy to operate. The survey results demonstrate the usefulness of `ConsentEnforcer` to end-users. We leave a large-scale user study as future work.

6.9 Discussion

A general automated analysis tool will benefit all stake-holders of the web. First, users can use it to detect suspicious behavior of websites. Second, regulators can automate the checking of non-compliant behaviors on websites. Lastly, good-intention businesses can leverage the tool to ensure the cookie consent libraries provided by 3rd parties work as intended to comply with the law.

Considering the prevalence of the problematic cookie consent management detected thus far, we recommend website owners to verify the actual execution of the cookie consent libraries integrated in their websites. Since the enforcement of users' cookie consent preferences provided by cookie consent libraries may not be guaranteed, `ConsentChk` can be used as an independent auditing tool for websites to verify that the cookie consent management solutions operate as intended. Providing problematic and ineffective cookie settings makes users confused and lowers their trust. Moreover, it violates cookie laws when a website does not honor the users' cookie consent [208]. Unfair practices are the grounds for regulators to penalize the service [112].

Additional support of new libraries requires the analysis of the cookie-setting UI, and decoding of the consent cookie and structured data objects. It took us an average of 30 hours to add support for a cookie library, but we expect the cookie library developers to spend much less time for the layout extraction as they can leverage their own source code, instead of reverse-engineering the UI and data objects. Furthermore, although the cookie setting layouts are extracted manually, the extraction is done only once for each cookie layout.

Due to the limitation of dynamic analysis that cannot generate all possible cookie uses, `ConsentChk` can only detect the violations while it cannot prove that a website completely honor user consents. This limitation is inherent to all techniques based on dynamic analysis such as software testing. Missing flows will reduce the number of detected incorrect-enforcement cookies. However, this incompleteness does not increase false positives of the incorrect-enforcement detection because each detected incorrect-enforcement has true occurrences of the captured cookie flows and extracted cookie preferences.

Despite the detection of consent violations, it is challenging to identify their root causes without server-side information. They can be caused by incorrect enforcement of the consent libraries or a buggy implementation of the server-to-server communication. Therefore, the detected inconsistencies need manual verification to determine their root causes.

6.10 Conclusion

We have presented `ConsentChk`, an automated system that detects cookie consent violations for global websites. It automatically detects cookie preference buttons to extract cookie preferences even on non-EU websites where cookie settings are hard to find. We have constructed a formal model to systematically analyze the (in)consistencies between user consent preferences and actual cookie usage of websites. The formal model and the automated system lay a foundation for automatically detecting cookie consent violations. Finally, we found the majority of the studied websites to have consent violations in both measurements from inside and outside of the EU. This finding suggests the existence of systemic issues of CMPs and highlights the need for automatically auditing cookie usage so that websites can ensure the compliance with the regulations in any part of the world.

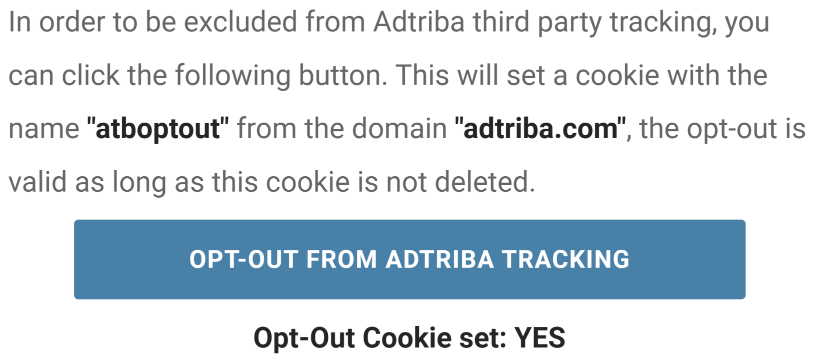
CHAPTER VII

OptOutCheck

7.1 Introduction

Online trackers, such as ad platforms and analytics service providers, leverage various tracking techniques to collect users' browsing history across websites, posing serious privacy concerns to users and regulators. As a result, the trackers' privacy policies often provide users an opt-out link or button to reject targeted advertisements and/or their data collection [72, 183]. Fig. 7.1 shows an example where an ad platform states to stop tracking users via unique-identifier cookies after the user opts out.

Inconsistencies between a stated opt-out policy and its actual tracking behavior pose high privacy risks to users since the data collection occurs/continues even after they opt out, contrary to their expectation. These inconsistent privacy practices can also be deemed



In order to be excluded from Adtriba third party tracking, you can click the following button. This will set a cookie with the name **"atboptout"** from the domain **"adtriba.com"**, the opt-out is valid as long as this cookie is not deleted.

Opt-Out Cookie set: YES

Figure 7.1: Example opt-out setting and policy statements. A user opts out of tracking by clicking the opt-out button that creates a cookie to record the user's opt-out choice.

deceptive and illegal by regulators. Federal Trade Commission (FTC) has fined several ad networks for their short-lived opt-out cookies [112], deceptive policy statements about a complete cookie opt-out [114], and falsified statements on browser cookie settings [113]. Therefore, checking (in)consistencies between the stated privacy policies and the corresponding data practices is important as it benefits all of users, companies and regulators; users will be reassured of their privacy protection, regulators can prevent trackers' deceptive mechanisms, and tracker companies will be forced to comply with their stated privacy policies.

The main research question to answer is then: *Do opt-out settings really opt users out of an online tracker's data practices as stated in its opt-out policy?* To answer this question, we address the following three challenges that originate from the complexity and vagueness of the opt-out policies specified in legal language and the variability of non-standardized opt-out links/buttons. First, the semantic extraction and analysis of opt-out policy statements are difficult due to the complexity of website user interface and the legal language used in privacy policies. Second, analyzing the data collection and tracking behavior requires activating an opt-out choice, extracting data flows and inferring data-usage purposes of trackers after the opt-out setting is enabled. Finally, verifying (in)consistencies between the opt-out policies and the data-collection practices needs to reconcile the different (i.e., high vs. low) levels of granularity between the policy statements and data flows.

Unlike prior work on the opt-out choices provided by content-publishing websites, we study trackers' opt-out of tracking services as third-parties on the content websites. Prior work [28, 137, 138, 229] has mainly studied the usability of opt-out choices and the extraction of generic opt-out hyperlinks on content-providing websites, rather than direct opt-out settings of online trackers. A recent study of compliance of cookie banners [208] does not apply to the cookies on websites *other* than those hosting the banners, thus covering a different scope from our work. Moreover, none of prior studies has checked the (in)consistencies between the opt-out settings and the corresponding policy statements. They assumed that trackers always honored users' opt-out preferences once the opt-out cookies were set [72, 183].

To fill these gaps, we present an automated framework, `OptOutCheck`, that analyzes (in)consistencies between opt-out policy statements and the corresponding data practices of online trackers.

First, given a tracker's website, `OptOutCheck` automatically discovers its opt-out buttons/links that record a user's preference of opting out of the service's tracking and data collection. From the sentences next to an opt-out button, `OptOutCheck` extracts the policy statements about the privacy practices for opted-out users (called *opt-out policies*). It identifies 5 classes of opt-out policies such as *No-tracking* and *No-data-collection* by analyzing the semantic arguments, syntactic dependencies and text patterns of the policy sentences. For example, a tracker may not use unique-ID cookies to track an opted-out user.

Second, `OptOutCheck` extracts the data flows from a user's browser to a tracker's servers after the user activates the opt-out choices. To this end, `OptOutCheck` simulates a user's click on opt-out buttons, identifies opt-out cookies and determines the cookie domains enforced by the opt-out policies. `OptOutCheck` then identifies the tracking and data-collection behavior by analyzing the data types and usage-purposes of the key–values sent via cookies and URL parameters to the tracker's servers after an opt-out.

Finally, `OptOutCheck` formalizes policy statements, data flows and subsumptive relationships of data types to define the condition under which a data flow is consistent with a privacy policy. We derive logical rules to check the satisfaction of this condition based on the opt-out policy classes and the data types in the data flows to detect flow-to-policy inconsistencies. Inspired by the soundness of dynamic analysis tools in software testing [147, 148, 280], we aim to minimize false positives (i.e., maximize the precision) so that the reported inconsistencies should always be true positives. In a large-scale study, OptOutCheck found multiple inconsistencies of popular online trackers which we manually verified, demonstrating OptOutCheck's scalability and effectiveness.

This paper makes the following main contributions:

- Classification of opt-out policies and creation of automatic classifiers for policy sentences. We create a dataset by categorizing policy statements that describe the data-collection policies after a user chooses to opt out. We develop automatic classifiers based on natural language processing (NLP) that achieve ≥84.6% precision on the previously-unseen samples.

- Extraction of data-collection behavior of trackers after a user opts out. We create a dataset and derive a classifier to identify opt-out cookies that achieves 95% precision on the test set. We develop techniques to extract the scope of opt-out policies based on opt-out cookies, extract the matching data traffic, and infer the data types collected by a tracker.

- A formal analysis of (in)consistencies between the opt-out policy statements and data flows conditioned on users' opt-out (Section 7.8). We derive formal consistency conditions and logical rules to detect the inconsistencies based on the classification of opt-out policies and data flows.

- An end-to-end (E2E) automated framework, OptOutCheck, that detects (in)consistencies between the actual data practices and the stated opt-out policies of online trackers.

- A large-scale study of opt-out choices of 2,981 online trackers. Of the 165 trackers for which OptOutCheck detected opt-out buttons and opt-out cookies, 11 trackers were found to track and collect user data despite their policy statements to stop the tracking and/or data collection after the user's opt-out. These trackers were present on 3.65% of the top 10k websites on average, and tracked a significant amount of web traffic. Since

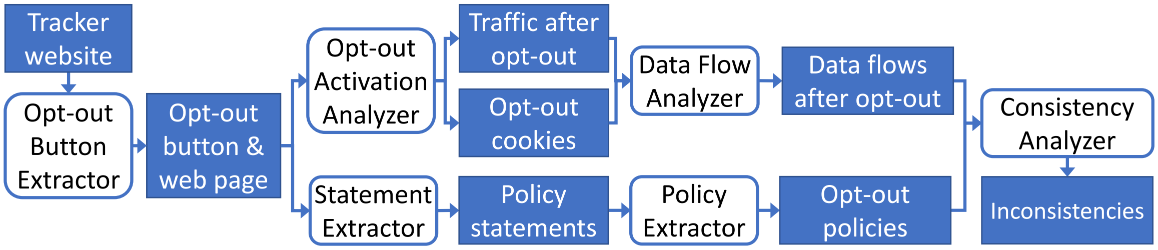


Figure 7.2: `OptOutCheck` workflow.

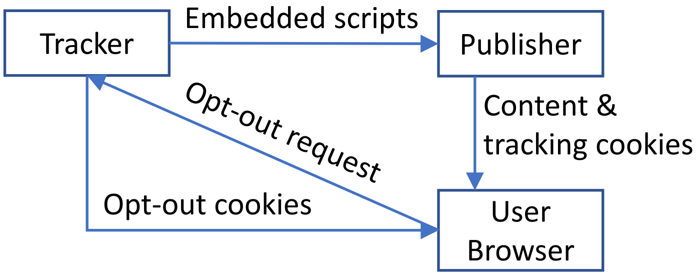


Figure 7.3: Trackers' data flows.

the inconsistencies are direct violations of the trackers' own privacy policies while the trackers collected user data without the users' consent, regulators may impose heavy fines for their deceptive privacy practice and unlawful data collection.

The rest of the paper details `OptOutCheck`'s analysis pipeline depicted in Fig. 7.2. `OptOutCheck` first searches for opt-out buttons in privacy-policy web pages (Section 7.4). It then extracts the corresponding opt-out policy statements (Section 7.5), opt-out cookies (Section 7.6), and data flows after opting out of trackers' services (Section 7.7). Finally, the system checks the conditions to detect inconsistencies, if any (Section 7.8).

7.2 Background

7.2.1 Trackers and Tracking Mechanisms

Trackers are the companies that collect information about users who browse the web [120]. The most common types of trackers are advertisers and data analytics services that collect user data to create online behavioral advertising (OBA). Other types of trackers are site analytics and social media who track users to understand the patterns of users' activity for the website to improve and provide services [86, 119]. As depicted in Fig. 7.3, we consider data flows among users, trackers and publisher websites. When a user accesses a content-providing website, besides the publisher's own contents the user wants to read, the browser also loads trackers' cookies and scripts. Trackers offer users opt-out choices on their websites so that they can request not to track or collect their data.

The most common online tracking technology used in practice and stated in privacy policies is the HTTP cookies placed on users' devices [183, 261]. Members of Digital Advertising Alliance (DAA) in USA and Canada agree not to use Flash and similar local-storage-based tracking tools unless an opt-out mechanism is publicly provided [48]. There are also other advanced web tracking mechanisms that are harder to detect, such as canvas fingerprinting, ever cookies, and cookie syncing [3].

We consider *third-party cookies* that are the cookies in domains other than those of the websites being accessed regardless of domain ownership [110, 236, 299]. We use the term "domain" to indicate a pay-level domain that a consumer or business can directly register, and is typically a subdomain followed by an effective top-level domain (public suffix) [108, 189]. The effective top-level domains are extracted by using *tldextract* library [186].

7.2.2 Opt-out Mechanisms

Placing anonymous opt-out cookies in the users' web browsers to signal their choices is the *de facto* mechanism used by trackers [183]. It is possible to have a persistent identifier for opt-out purposes, but trackers can now easily track users who contradict the purpose of opt-out. Many trackers' privacy policies even describe their opt-out mechanisms explicitly [183], such as 'this will set a cookie with the name "atboptout" from the domain "adtriba.com"' as depicted in Fig. 7.1. Furthermore, tracking blocking tools, such as those of Network Advertising Initiative (NAI) [162], DAA [48], and Evidon Global Opt-out [73]), use this method.

Anonymous opt-out cookies remain the most common opt-out mechanism for advertisers [27, 264] and were explicitly described in the privacy policies we had surveyed, and hence we only consider cookie-based tracking and opt-out mechanisms. Although other forms of tracking like fingerprinting are used by trackers, a recent study found that fingerprinting is not stable owing to the changes of user fingerprints over time, so trackers even employ cookie re-spawning to enable reliable tracking of users [107]. Another opt-out mechanism uses server-side storage to store the users' consents on the server side [96]. It requires a long-term ID for each user, such as the user's ID or email address, and needs to performs synchronization between server-side consent storage and cached local cookies on the user's browser. Since we consider the opt-out settings provided by ad platforms that do not require the user to login or input his/her email address, this opt-out mechanism is outside of our scope.

7.3 Cookie Crawler

We have developed a crawler that automatically visits web pages, performs user interaction and records HTTP cookies set by both JavaScript and HTTP responses. The crawler is set up to use a university-based vantage point and user-behavior-emulating browser configurations to reduce measurement bias as websites containing trackers may behave differently when they detect a visitor to be a bot [170]. Specifically, the crawling is conducted in an 8-node computing cluster located in a US university network. The crawler is built upon Playwright [217] that automates the Google Chrome web browser and emulates realistic human browsing behavior to circumvent trackers' bot-protection mechanisms [32].

Each web page visit waits until there is no network activity for at least 0.5 second or a 30-second timeout expires, which is a common heuristic used by web automation tools for loading dynamic web pages [129, 217]. (See Appendix E.2 for the rationale of the loading timeouts.) Furthermore, if the loading fails, to avoid transient network errors, the web page load was retried at most three times with a 2-minute waiting time between two retries.

7.4 Extraction of Opt-out Buttons

This section describes the detection of the actionable choices provided by trackers for users to opt out of their data collection and tracking. We define an *opt-out activation button* (also called *opt-out button*) as a clickable HTML element that, upon its click, will record the user choice/preference of opting out of the trackers' services. Similarly, an opt-out page is a web page that contains an opt-out button. Such pages can be an iframe embedded in another web page. Furthermore, while many websites instruct users to use opt-out tools providing self-regulatory groups such as NAI [163] and DAA [14], `OptOutCheck` does not analyze them because these tools do not contain any specific definition of a tracker's opt-out. Moreover, members of these groups frequently provide their own definitions of opt-out which are usually stricter than the minimum requirements of NAI and DAA [183].

Given a tracker domain, `OptOutCheck` uses a three-stage pipeline to extract its opt-out button. It first identifies the candidate web pages that may contain an opt-out button or a link to an opt-out page by searching for keywords related to "*opt out*" in the entire website. `OptOutCheck` then detects opt-out button candidates from the web pages. Finally, `OptOutCheck` validates opt-out buttons by extracting the opt-out cookies after clicking the candidate buttons.

7.4.1 Extraction of Opt-out Page Candidates

7.4.1.1 Challenges

As trackers have incentives to keep users from opting out of their tracking [168], they tend to make it difficult to detect opt-out pages of their websites. The opt-out pages can be placed deep down in the website's hierarchy with very few links to the pages. For example, an ad platform website may have multiple policies, such as privacy and cookie policies, but only one of them has a hyper link that points to an opt-out page. Another challenge is to search websites that support multiple languages where the links to switch the language must be discovered.

Because checking the availability of an opt-out page requires exhaustive crawling of the whole website, we leverage search engines that systematically index web pages of trackers' websites to find opt-out page candidates. Although search engines may not crawl all websites in real time, the privacy policies and opt-out links do not change very frequently [17]. Finally, the results in this step are refined further in other detection steps in the `OptOutCheck`'s pipeline, thus avoiding/minimizing potential false positives.

7.4.1.2 Query Term Design

We derive a query term for Google Programmable Search Engine [132, 134] to search for the web pages that contain keywords related to the opt-out of trackers' websites. Specifically, we use query term *"opt out opt-out site:<tracker-domain>"* where the *<tracker-domain>* is substituted for the website domain of a tracker, such as *site:adblade.com*. The query includes *opt out* without any quote to search for variations of *opt out* such as *opted out* or *opting out*. The term "opt-out" helps detect opt-out pages with that term appearing on their URLs instead of their contents. The search engine then looks for these "opt out" variations and the exact "opt-out" term in both URL and website's content [221, 272].

The query is designed to have better coverage rather than maximizing precision because the later steps in the pipeline (e.g., opt-out button detection) will filter out unrelated non-opt-out pages. So, the query term avoids restricting the search with the *exactTerms* or *orTerms* parameters [133]. We also try to use the minimum number of customized parameters as using more parameters is found to make the output results less stable over time. For example, restricting to English-only web pages produced no results in some query executions. The Google search may still miss web pages, but it will only increase false negatives (i.e., no detection of opt-out buttons) without increasing false positives (i.e., incorrect detection).

7.4.1.3 Evaluation

We evaluate the extraction performance on trackers that are known to have opt-out buttons on their websites. Specifically, we randomly selected 100 trackers from the Evidon Global Opt-out list [73]. We excluded trackers' opt-out pages that were not accessible, possibly due to the outdated opt-out-page URLs in the Evidon database. Finally, we extracted opt-out pages of 43 trackers to create the dataset.

We observed that the search engine is effective in finding the opt-out pages. The opt-out pages are included in the top-1 and top-3 results in 34/43 (79.07%) and 40/43 (93.02%) of the search queries, respectively. There are three cases where the search engine could not detect the opt-out pages. A website places its privacy policy in PDF where the opt-out link is not clickable. Another non-English tracker uses "don't track" instead of "opt out" keyword. Finally, one website disallows crawling of its privacy policy using *robots.txt* specification [155]), thus preventing search engines and automated web crawlers from detecting the opt-out button placed in the privacy policy. Because the results lower than the top 3 did not improve the opt-out page detection, `OptOutCheck` uses only the top-3 results from the search engine for a further analysis.

7.4.2 Opt-out Button Detection

We derive patterns to extract opt-out button candidates by following the Snowball bootstrapping procedure which has been widely adopted for extracting information in web and mobile environments [7, 146, 166, 167]. Specifically, we construct patterns of the attribute values of the HTML elements that represent opt-out buttons. A key step is that after each iteration, only the most reliable rules are kept for the next iteration. Therefore, the set of extraction rules improves as it iterates. This is detailed in Algorithm 1.

Let E be a set of extraction rules where each rule $e \in E$ is a tuple of *(element-selector, attribute, value pattern)* that matches the *value pattern* with the value of an *attribute* of the elements selected by the CSS *element-selector*. In these tuples, the *attribute* is an

HTML tag's attribute or text content. To avoid mixed effects on different types of HTML elements, each of the rules applies on one tag and one attribute. Specifically, an *element-selector* is a CSS selector that selects only one type of HTML element rather than a list that selects multiple different HTML tags. Similarly, an *attribute* denotes a single attribute of an HTML element rather than a list of attributes. The *value pattern* can be a regular expression or a function that performs complex matching on the value of the element's attribute. An example rule is *('a', text-content, '^opt[-]out')* that matches any anchor element (i.e., hyperlinks) with text content starting with either "opt-out" or "opt out". The regular expression matching is case-insensitive to handle varied capitalization in the opt-out button's labels. An element's text content represents only a human-readable text, not invisible elements [225].

The seed set contains 4 rules to extract elements *a*, *button*, *input* and *span* with text content starting with "opt out" or "opt-out". These HTML tags are commonly used to implement buttons in web pages [54, 199]. As in prior research [166], we also observe that the seed rule set does not significantly affect the final rules if the matching frequency thresholds are tuned properly.

Following the bootstrapping algorithm, we added patterns that use *id*, *class*, *value*, *onclick* and *href* of these elements. The final extraction rule set contains 14 rules with a frequency cutoff threshold of 10 (i.e., the rules with less than 10 matches are excluded). We use two patterns: starting with "opt out" variants and contain the "opt out" function identifier (e.g., "optoutToggle"). The matching patterns use dashes, underscores and spaces as the delimiters.

7.4.3 Opt-out Choice Activation

To activate an opt-out choice, `OptOutCheck` attempts to click the opt-out button candidates until an opt-out cookie is detected or a maximum of 5 candidates have been tried. If clicking a link does not create an opt-out cookie, the crawler returns to the original

Algorithm 1 OptOutCheck's bootstrapping procedure for extracting opt-out buttons from a large corpus of web pages.

```
Initialize E to a set of seed extraction rules
While E does not grow
    Use the rule set E to detect opt-out buttons
    Generate new rules based on the detected buttons
    Keep only reliable rules; the resulting rule set is E'
    Set E = E'
Output: a set of rules to extract opt-out buttons
```

page and tries the next candidate button. Appendix E.1 describes the implementation of a button-clicking action while Section 7.6 introduces the definition of opt-out cookies.

To reduce the number of link-clicks, OptOutCheck ranks the opt-out button candidates based on the classifier's confidence (i.e., the classification probability). The system prioritizes the matched patterns on the displayed text content which is a user-facing feature. Furthermore, OptOutCheck excludes the candidates based on the URLs that are informational opt-out web pages commonly used by trackers or industrial opt-out tools, such as the DAA and NAI websites. Similarly, hyperlinks that points to the currently visiting page are also removed.

7.5 Opt-out Policy Analysis

This section describes the automated extraction of opt-out policies from the opt-out web pages of trackers. Automated analysis of opt-out policies is necessary because manual inspection is impractical to cover thousands of advertisers' privacy policies and account for their regular/frequent updates.

7.5.1 Interpretation and Formal Definitions

7.5.1.1 Interpretation of Opt-out Policies

We consider an opt-out statement to be equivalent to a negative-sentiment statement, i.e., a statement "opt out of *S*" is equivalent to "not *S* after opt-out" where *S* is a statement about data collection. For example, "you can opt out of receiving targeted ads" is equivalent to "you will not receive targeted ads after opt-out."

Due to the ambiguity of language in privacy policies, we make the following interpretation of common opt-out statements. Like the interpretation in prior work [183], we assume "no tracking" to indicate that user data can still be collected but will not be associated with the device, such as by using unique-ID cookies. Tracking can be defined as "collecting data over multiple different web pages and sites, which can be linked to individual users via a unique user identifier" [176]. In addition, since a cookie is always sent to its ad provider's server whenever the browser makes a request to the server [29], if the advertiser states that it will stop placing cookies on the user's browser (except for the opt-out preference cookies), or the user can opt out of the advertiser's cookies, we interpret it to be equivalent to "stop data collection." Finally, we interpret "targeting" term to be equivalent to "targeted advertising," so "opting out of targeting" means that interest-based advertising will not be displayed to the users.

7.5.1.2 Formal Definitions

Inspired by prior work in privacy-policy analysis [20, 46], we formalize the statements in privacy policies as follows to analyze the (in)consistencies between privacy policies and actual data-collection behavior.

Definition 7.5.1 (Policy Statement)**.** *A policy statement is a pair* (dc, du) *where dc represents data collection and du is data usage.* $dc = (r, c, d)$ *denotes whether a receiver r does or does not collect (*$c \in \{collect, not\_collect\}$*) a data object d.* $du = (d, k, p)$ *represents whether a data object d is used for or not for (*$k \in \{for, not\_for\}$*) a data usage purpose p of the receiver.*

Definition 7.5.2 (Semantic Equivalence)**.** *x and y are semantically equivalent, denoted as* $x \equiv_o y$*, if and only if they are synonyms defined under an ontology o. Similarly,* $x \not\equiv_o y$ *denotes nonequivalent concepts in an ontology o.*

Opt-out Policy Class	Policy Statement Set
No-tracking	{ ((r, collect, d), (d, not_for, tracking) }
No-data-collection	{ ((r, not_collect, d), None) }
No-data-collection-for-oba	{ ((r, collect, d), (d, not_for, targeted_ad)) }

Table 7.1: Opt-out policy classes and the corresponding sets of policy statements. In the policy statement sets, data type $id\_data \equiv_\delta$ "unique identifier", $d \equiv_\delta$ "data", and receiver $r \equiv_\delta$ "first party" under an ontology δ. "oba" stands for online-behavioral advertising.

Grammatical Role	Example	Policy Statement	Policy Class
Object	You can opt out of tracking and our unique cookie identifiers here.	(we, collect, data), (data, not_for, tracking)	*No-tracking*
Main clause	If you opt out, we will no longer use cookies to collect your data for targeted advertising.	(we, collect, data), (data, not_for, targeted ad)	*No-data-coll.-for-oba*
Adverbial clause	If you want us to stop collecting your data, please opt out here.	(we, not_collect, data), None	*No-data-collection*
No "opt" predicate	Please do NOT collect information about me using cookies and other tracking technologies.	(we_implicit, not_collect, data), None	*No-data-collection*

Table 7.2: Examples of opt-out policy clauses, their grammatical roles with respect to the *opt* predicate, the extracted policy statements and opt-out policy classes. The opt-out policy clauses in each sentence are underlined.

A policy statement only captures the semantics of the sentences that describe data collection, sharing or use. Other policy sentences that do not specify explicit data practices, such as "we will stop showing targeted advertising", are not modeled since it is unclear which data is collected or used. The data usage *du* can be a special value *None*, indicating that the usage purpose is not specified.

7.5.2 Opt-out Policy Classes

In order to analyze the (in)consistencies between opt-out policies and data practices, we categorize the policy statements according to their stated data practices and purposes. Inspired by prior work on privacy policies of online trackers [72, 183], we consider 5 types of opt-out policies: no tracking (*No-tracking*), no data collection (*No-data-collection*), no data collection for targeted advertising purposes (*No-data-coll.-for-oba*), no displaying online behavioral advertising (*No-display-oba*) and *Other*. The *Other* class includes samples that do not belong to any other classes such as opt-out of the sale of information, stop receiving marketing emails/text messages, and opt-out instructions.

Our opt-out policy taxonomy covers two main types of data practices, *user-activity tracking* and *user-data collection*, while a data practice's purpose is either *for delivering OBA* or *unspecified*. However, the *No-tracking* class is not divided further based on the data-usage purpose because a statement about tracking is seldom coupled with data-usage purposes.

Using the definitions in Section 7.5.1.2, we formalize the opt-out policy classes in such a way that each policy class comprises policy statements that have semantically equivalent terms. For example, *No-tracking* class is a set of policy statements in the form *(r, collect, id_data), (id_data, not_for, tracking)* where *id_data* can be substituted by a synonym such as "unique identifier" and *r* can be a synonym of "first party". Of the opt-out policy classes, statements about stopping displaying online behavioral advertising (*No-display-oba*) are not formalized for the flow-to-policy consistency analysis because they do not explicitly express any data collection. The opt-out policy classes and the corresponding privacy-statement sets are listed in Table 7.1.

7.5.3 Automated Opt-out Policy Classification

The extraction of opt-out policies from a policy sentence is formulated as a binary classification problem. For each opt-out policy class, we create a classifier that determines whether a sentence expresses the opt-out policy or not. As the result, a sentence may contain one or multiple opt-out policies. For example, "to opt out of our tracking and data collection, please click the button below" contains two policies: *No-tracking* and *No-data-collection*.

The rest of this section details the main steps of the automated classification pipeline: identify opt-out predicates, extract opt-out policy clauses, and classify the policy clauses.

7.5.3.1 Opt-out Predicate Identification

The pipeline begins with the extraction of opt-out predicates (verbs) describing the action that a user needs to take for an opt-out. The most common form of such predicates is a verb with lemma *opt*. In addition, `OptOutCheck` also looks for nouns with lemma *opt* and traverses up the dependency tree to identify the action performed on the noun. For example, in sentence "if you do not want to see OBA, please *click* our opt out here," *opt* is a noun and *click* is extracted.

7.5.3.2 Opt-out Policy Clause Extraction

To extract the clauses that express the data-collection policies for an opted-out user, the system identifies the clauses that have one of the following grammatical roles with respect to an opt-out predicate: object, main clause, and adverbial clause. In an exceptional case when a sentence does not have any opt-out predicate, but its context is clearly about opt-out policies (e.g., the sentence is the label of an opt-out button), we treat the whole statement as an opt-out policy clause. Table 7.2 lists examples of the opt-out policy clauses and their roles in a sentence.

The system primarily extracts the opt-out policy clauses from a sentence by analyzing the semantic arguments of the *opt-out* predicates. Specifically, we design `OptOutCheck` to analyze the following arguments of each opt-out predicate: object (*Arg1*), instrument (*Arg2*), adverbial (*Argm-Adv*), purpose (*Argm-Prp*) and purpose-not-cause (*Argm-Pnc*). A semantic argument answers questions like "who?", "did what?", "to whom?", and "for which purpose?" of an event expressed by the predicate [172, 188]. The definitions of these arguments are given in the OntoNotes 5 linguistic corpus [255].

As a complement to the semantic-role analysis, `OptOutCheck` analyzes the syntactic dependencies in the sentence with respect to the *opt-out* verbs. In particular, it searches for the main clause of each *opt-out* predicate by analyzing the syntactic dependency tree of the sentence [172]. For example, the verb *opt* in "if you opt out" does not have any semantic arguments, and hence `OptOutCheck` looks for its main clause "we will no longer use cookies to collect your data" and treats it as an opt-out policy clause.

7.5.3.3 Opt-out Clause Analysis

`OptOutCheck` classifies a sentence into the opt-out policy classes by identifying data objects, data-collection sentiment (i.e., collect or not) and advertising data-usage purposes in an opt-out policy clause. To identify the *No-data-collection* policy for "opt out of" phrases, `OptOutCheck` identifies negative data-collection actions on data objects in the object argument *Arg1* of an *opt* predicate. `OptOutCheck` uses a named entity recognition (NER) model [172] to accurately extract data objects (such as *cookie* and *unique cookie identifiers*). In addition, `OptOutCheck` uses patterns of syntactic dependencies to identify data-practice noun phrases. Data-collection noun phrases such as "use of cookie" and "collection of data" are identified by searching for data objects (e.g., *cookie* and *data*) with a *pobj* (object of a preposition) dependency with respect to data-usage actions (e.g., "use" and "collection"). For example, "opt out of unique cookie identifiers" and "opt out of our use of information about you" are classified as *No-data-collection*.

Since cookies are the means of data collection, a negative-sentiment action performed on cookies is an indication of the *No-data-collection* policy, such as "we will stop placing cookies on your browser." The common actions on cookies are *drop*, *place*, and *set*. The negative sentiment of a data-collection action is indicated by the existence of a negation-modifier dependency, a *Argm-Neg* semantic argument, or a negative-sentiment modifier such as "no longer" and "stop".

Since the sentences in close proximity to opt-out buttons have a context related to opt-out choices, the occurrence of certain keywords is a good indicator of policy classes. Specifically, to extract *No-tracking*, the classifier looks for nouns and verbs related to tracking, such as *tracking*, *identifier* and *disassociate*. Similarly, advertising-related keywords, such as *target*, *advertising* and *marketing*, indicate advertising data-usage purposes. The advertising purposes also distinguish *No-data-collection* from *No-data-coll.-for-oba*.

7.5.4 Development of Opt-out Policy Classifiers

We create a manually-annotated dataset as the ground truth to develop matching patterns for the opt-out policy classifiers as follows.

7.5.4.1 Tracker Selection

We crawled cookies of the top 5k websites in the US as of October 2020, ranked by the SimilarWeb analytics service [200]. This selection is to ensure the privacy policies of the online trackers to be subject to the same legal and regulatory requirements, such as the Notice and Choice framework in the US [112]. Furthermore, we excluded pornography websites, using a blocking list [230], since these websites use specialized trackers [300] and are not our focus.

We selected a dataset of 120 popular third-party cookie domains. From the 180 cookie domains that were present on at least 100 websites, we chose the top 100 third-party cookie domains and other 20 randomly selected domains from the remaining cookie domains to cover both the most popular and less popular cookie domains. The number of domains was limited by the resources needed to analyze and annotate the cookie domains. Appendix E.3.1 provides details of the cookie collection and domain selection.

7.5.4.2 Opt-out Button Identification

From the selected cookie domains, we traced back to the websites of the trackers that own the cookie domains and manually extracted the opt-out buttons on each website. From the home page, we searched for the privacy policies (e.g., for the website visitors, corporate customers, and end-users) and then identify the opt-out settings contained in the policies. Since opt-out buttons were not ambiguous, this extraction was done by one advanced PhD student and took an average of 45 minutes for each domain, or 90 hours for 120 cookie domains. The details of the extraction process are provided in Appendix E.3.2.

Of the analyzed trackers, 80 provided opt-out choices. The most common form is single-click opt-out buttons. 76 (95.00%) of the settings have a single step, i.e., a single click, to opt out. The remaining settings need 2 steps: select an opt-out preference option and then click the submit button.

7.5.4.3 Opt-out Policy Corpus

From the identified opt-out web pages, we selected the sentences next to the opt-out buttons and classified them into the opt-out classes. Since privacy policy sentences were vague and complex, the classification of the sentences was done by two PhD students with no less than 3 years of experience in user-privacy research. It took an average of 3 minutes for each sentence on average, or 20 hours for both annotators. The inter-annotator agreement is 94%. We held a follow-up meeting to reconcile the differences.

The final opt-out policy corpus contains 246 sentences in 80 trackers. *No-display-oba* is the most common opt-out policy with 49 (19.92%) occurrences. *No-data-collection* constitutes 23 (9.35%) instances. The least common policy with 18 (7.32%) samples is *No-tracking*. The imbalance between the opt-out policy classes and the *Other* class reflects the small percentage of the opt-out policy statements compared to descriptive opt-out instructions in practice. The number of sentences per opt-out policy class is listed in Table 7.3.

7.5.4.4 Automatic Classifiers

Using the dataset, we derived two classifiers for *No-tracking* and *No-data-collection* policies that are the only opt-out policies that can be verified by observing the behavior of the trackers on the client side. Other classes related to online-behavioral advertising purposes are hard to verify without knowing the processing purposes on the tracker servers.

Policy Class	# Sentences
No-tracking	18 (7.32%)
No-data-collection	23 (9.35%)
No-data-coll.-for-oba	23 (9.35%)
No-display-OBA	49 (19.92%)
Other	139 (56.50%)
Total	246 (100%)

Table 7.3: Opt-out policy dataset. A sentence may contain multiple opt-out policies.

Metric	Train	Test
Precision	0.98	0.97
Recall	0.74	0.74
F1	0.84	0.84
Support	649	279
# Samples	7,649	3,279

Table 7.4: Opt-out cookie classifier performance on the training and test sets.

The classifiers achieved an average F1 score of 86.04% with precision $\geq$ 88% on the policy corpus. The high inter-annotator agreement and the high F-1 scores demonstrate the consistency of the interpretation of the policy classes and the regularity of the sentence patterns. It is worth noting that due to the data sparsity, i.e., small number of samples per opt-out policy class, we use the dataset as a training set for developing the matching patterns while Section 7.9.3 will evaluate their performance as part of the consistency analysis pipeline. Table E.1 (Appendix E.3.3) shows the detailed performance of the classifiers.

7.5.5 Implementation

7.5.5.1 Opt-out Policy Statement Identification

`OptOutCheck` extracts opt-out policies from the policy statements that describe the data collection practices after a user clicks on the opt-out button. For example, as shown in Fig. 7.1, ad platforms would cease their tracking after the user opts out. Identifying these sentences is challenging because of the flexible design and implementation of websites.

We observe that the opt-out policy statements are commonly placed nearby (e.g., in the surrounding paragraphs). This assumption is close to the expectation of FTC [112]. Therefore, given an opt-out page identified in Section 7.4, `OptOutCheck` converts the web page into plain text [252, 266] and extracts 10 sentences (5 before and 5 after) surrounding the position of the opt-out button. Furthermore, to reduce unrelated statements, except for labels of opt-out buttons, policy sentences without any "opt" predicate (e.g., *opt-out*, *opt out* and *opting out*) are excluded.

7.5.5.2 Natural Language Analysis

`OptOutCheck` uses the neural-network-based language pipelines of Spacy NLP library [9, 197] to parse and create the dependency trees of privacy policy sentences. The semantic arguments are analyzed by using a semantic role labeling model (SRL) of the AllenNLP library [12], which is based on *Roberta-base* contextualized word embeddings and trained on the CoNLL2012 (OntoNotes 5) large-scale natural language dataset [273]. Finally, we use PurPliance [46] to analyze privacy-statement parameters such as data-collection actions and data objects. To improve the data-type extraction, we augment its data-object NER model with terms related to cookies that are commonly used in the privacy policies of online trackers.

7.6 Opt-out Cookie Extraction

To check whether a tracker's data collection practices follow its opt-out policies or not, it is necessary to determine that a user's opt-out preference has been recorded by the tracker. Since we focus on the opt-out mechanism based on anonymous cookies, we define *opt-out cookies* as the cookies that online trackers use to record a user's opt-out choice [11, 70, 99]. These cookies are created upon clicking an opt-out button for the trackers to enforce their opt-out data collection policies on web pages where the cookies present.

Automated extraction of opt-out cookies is necessary as privacy policies rarely include specifications of these kinds of cookies. The mapping from a tracker to cookie domains using a predefined list is also not guaranteed to be complete and up-to-date. Furthermore, a differential analysis of the cookies before and after an opt-out is not sufficient for extracting opt-out cookies because the opt-out button may redirect the user to the tracker's home page where other cookies — unrelated to opt-out cookies — are added.

7.6.1 Opt-out Cookie Classifier

OptOutCheck takes a hybrid approach to extract opt-out cookies where a cookie is matched with a predefined opt-out cookie registry and then an automatic classifier if not found. The exact-match approach leverages the opt-out cookie registries provided by automatic opt-out tools: Evidon Global Opt-out [73], DAA Protect My Choice [14], and Google Keep My Opt-Outs [126]. Any cookie that has its name, domain and value matched the registries is determined as an opt-out cookie. The extraction excludes *session cookies* because the tracker should remember the opt-out choices of users over multiple browsing sessions. In what follows, we describe a classifier that uses the pattern of a cookie's name and value to determine whether it is an opt-out cookie or not.

7.6.1.1 Opt-out Cookie Dataset

To develop and evaluate the opt-out cookie matching patterns, we derive a ground-truth dataset that contains the cookie names and values from the exact-match registries. We excluded cookies with a non-anonymous identifier value, which is empirically identified as a combination of 10–20 alpha-numeric characters, while keeping cookies with anonymous values that comprise only zeros and dashes. This process resulted in 928 opt-out cookies from 795 trackers.

We then mixed the opt-out cookies with 10k cookies randomly sampled from the crawling of the top 5k websites as described in Section 7.5.4.1. These additional cookies are considered negative samples (i.e., non-opt-out cookies) because the crawling process did not perform any opt-out, i.e., we assume the browser does not have any opt-out cookies unless the user explicitly opts out.

Stratified partitioning was then performed to split the dataset into training and test sets with a 70–30% ratio. The patterns are developed on the training set and evaluated on the test set. The final dataset contains 10,928 cookies with 7,649 and 3,279 samples in training and test sets, respectively. The number of samples and supports in the dataset are shown in Table 7.4.

7.6.1.2 Opt-out Cookie Patterns

The matching rules comprise two types of patterns based on cookie names and cookie values. First, the patterns in cookie names include the spelling and abbreviation variants of "opt out", such as "opt-out" and "OptedOut". The abbreviation pattern "oo" does not simply match when it is a substring; it matches only if "oo" is either the whole string or surrounded by delimiters like "_". We exclude the cookies whose string values can be converted to *False* in common programming languages, such as *0* or *false*. For example, cookie *optout=false* does not indicate an opt-out. Second, a cookie is considered for an opt-out purpose if its name indicates a unique user ID, such as "uid" and "uuid", *and* its value is not unique such as a single-digit number like "-1" or "nan". These special values of a tracking cookie can be used to indicate the opt-out preference. It is worth noting that opt-out cookies must have both appropriate *key* and *value*, e.g., cookie named "uuid" is not an opt-out cookie until its value becomes "-1".

7.6.1.3 Performance Evaluation

As shown in Table 7.4, the classifier achieves a high F1 score of 84% (97% precision and 74% recall) on the test set. As the dataset is highly unbalanced, these metrics are computed only for positive samples. We aim to minimize the false detections (i.e., maximize precision), so we consider the performance is good enough when the precision on the training set was greater than 95%. We conjecture that this high accuracy comes from the regularity of the naming of opt-out cookies created by programmers. It is worth noting that `OptOutCheck` does not recognize cookies with obfuscated names and values but this limitation does not increase the false-positive rate of the system.

7.7 Data Flow Analysis

We now describe how `OptOutCheck` extracts the actual data-collection behavior of a tracker from its network traffic to detect the inconsistencies, if any, between its actual behavior and opt-out policies.

7.7.1 Data Flow Definition

We consider the data objects and purposes in the data-collection behavior of a tracker, which is formalized as follows.

Definition 7.7.1 (Data Flow)**.** *A data flow is a 3-tuple (r, d, p) where a recipient r collects a data object d for the receiver's purpose p.*

The receivers of network traffic are determined by the destination hosts in the intercepted URLs. For example, the data sent to hosts owned by tracker T has the receiver $r = T$. A data object d is the data type transferred via the network, such as a "unique identifier" or "user location". A data-usage purpose p is the purpose of collecting and using the data object such as "for delivering OBA" or "for product research and analytics."

7.7.2 Extraction of Key-Values

In order to extract key–value data pairs from cookies and URL parameters in the HTTP traffic, `OptOutCheck` addresses two challenges: 1) ensure captured traffic falls under the scopes of the corresponding opt-out policies and 2) avoid cookies that are only stored in the browser but not transferred to the servers.

7.7.2.1 Opt-out Policy Scopes

To analyze the data collection on opt-out choices, `OptOutCheck` considers only cookies and URL parameters sent to the URLs that fall under the scope of opt-out policies. In particular, these URLs are the ones that match the domains of the tracker's opt-out cookies (which are determined in Section 7.6). Although the scope of opt-out choices may span beyond the opt-out cookies' domains, because a tracker must own the domain of an opt-out cookie, we assume a data flow to follow the opt-out policy if its domain matches the top-level domain of an opt-out cookie, called an *opt-out domain*. For example, if the opt-out cookie is *opt_out=1* under domain *ads.tracker.com*, the opt-out domain is *.tracker.com*.

The domain matching follows the domain-match specification [29]. Moreover, the longest matching URL paths take the precedence if there are multiple matched domains and paths found [82].

7.7.2.2 Cookie Transfer Interception

`OptOutCheck` intercepts the cookies and URL parameters transferred from a web browser to the trackers' servers in the HTTP requests made by the browser during each web page visit. By capturing the cookies transferred via network traffic, the data in the cookies is guaranteed to be collected by the trackers, rather than being only stored and unused in the browser. To determine the expiration time of the cookies intercepted in the HTTP requests which contain only the keys and values of the transferred cookies, they are resolved to the cookies stored in the browser by matching their names, values, domains, paths and request URLs. We use the HTTP request interception feature of the web browser automation tool where the interception is performed before the traffic is encrypted in the HTTPS protocol.

7.7.3 Extraction of Data Flows

From the extracted key–value pairs, `OptOutCheck` infers the data objects d and data-usage purposes p of data-flow tuples formalized in Definition 7.7.1. For example, a data flow associated with the collection of a unique-ID cookie *uid* used by a tracker T is $(T, uid, tracking)$. Since the automatic opt-out policy extractors extract only *No-tracking* and *No-data-collection* opt-out policies (Section 7.5.4.4), we focus on detecting the data types that reflect the tracking and data collection of a tracker as follows.

7.7.3.1 Detection of Tracking Identifiers

`OptOutCheck` detects the cookies that contain unique identifiers for tracking purposes. A data flow for such a tracking cookie is *(<tracker>, unique ID, tracking)* where *<tracker>* is the tracking cookie's owner. Unique IDs (known as unique user identifiers or tracking IDs) are widely used for tracking users [52, 242, 265].

Since automatic detection of identifier cookies has been developed before [94, 121, 215], we assume cookies and URL parameters containing unique IDs are used for tracking purposes. While it is not possible to determine the ultimate usage purposes of these IDs without the information at the server side, unlike automatic data collection such as logging of IP addresses on HTTP servers, setting cookies and URL parameters requires significant effort, and hence the collection of such data is unlikely to be accidental. For example, the collection of cookie named *uid* containing a 16-digit identifier that does not change throughout a user's browsing activity is likely to track users by assigning each user with a unique user ID.

`OptOutCheck` determines a cookie to have a unique ID using a set of criteria that are empirically determined and evaluated by Englehardt et al. [94]. The heuristics leverage two main properties of a unique ID cookie — *unique across browser instances* and *persistent over time*. There are 5 criteria as follows. First, cookies are *long-lived*, i.e., their expiration time is longer than three months. This time threshold is the same as that in the work of Englehardt et al. [94]. Second, their values are *constant* throughout web browsing (i.e., visits to different websites by the same browser instance) to avoid varying non-ID values like timestamps and the browsing history. Third, the cookie values are of *constant length* across different measurements. Fourth, cookies have *user-specific* values which are unique among different browser instances. Finally, cookie values have *high entropies*, i.e., its values change significantly across measurements. A cookie is filtered out if the RatcliffObershelp-similarity [244] score of its values in different measurements is higher than 0.55. Note that `OptOutCheck` reuses the threshold values from [94] and developing better thresholds is outside of this paper's scope.

`OptOutCheck` parses and decodes URL parameters into key–value pairs in order to determine the data types collected by the trackers. As the values can be encoded in various data formats [52], `OptOutCheck` attempts to decode the URL parameters and cookie values in JSON and base64 formats. The same heuristics of detecting unique IDs for cookies apply for URL parameters except the long-lived criterion as URLs do not have expiration time.

7.7.3.2 Detection of General Data Collection

In addition to the unique IDs, `OptOutCheck` detects the collection of other user data types such as location and web browsing history. Inspired by the *bait* technique [2], the system looks for the known values of the crawling servers' IP addresses, location (e.g., city and state names), browser/OS versions, and URLs of the visited web pages in the values of the extracted key–value pairs. Their existence is the indication of data collection by a tracker. For example, a tracker is collecting user location if its cookie contains a key–value pair *region=<city_name>* containing the name of the city where the crawling server is located.

7.8 Opt-out Flow-to-Policy Consistency

This section presents a formal model to analyze the consistency between the policy statements and data flows from web browsers to trackers which are conditioned upon users' opt-out.

7.8.1 Subsumptive Relationship

The formal representations of opt-out policy statements (Definition 7.5.1) and data flows (Definition 7.7.1) are based on the concepts of receiving entities (i.e., receivers), data objects and purposes that have subsumptive relationships with each other. For example, a relation "personal data includes email addresses" translates to that *email address* is subsumed by *personal data*. `OptOutCheck` leverages the subsumptive relationships in the ontologies of PolicyLint [20] that are derived from subsumptive phrases of a large number of privacy policies. The relationship between the policy terms are formalized as follows.

Definition 7.8.1 (Subsumptive Relationship)**.** *Concept x is subsumed by another concept y, denoted as $x \sqsubset_o y$, if and only if $x \not\equiv_o y$ and there is a path from y to x in an ontology o represented as a directed graph in which each node is a term and each edge points from a general term y to a specific term x included in y, i.e., x "is a" instance of y. Similarly, $x \sqsubseteq_o y \Leftrightarrow x \sqsubset_o y \vee x \equiv_o y$.*

7.8.2 Consistency Model

Informally, a data flow is consistent with a privacy policy T which consists of a set of policy statements t_s, if there is a policy statement that discloses the data object and purpose of the data flow and there is no policy statement that discloses otherwise (e.g., uncollection of the data). The consistency condition is formalized as follows.

Definition 7.8.2 (Flow-relevant Policy Statements)**.** *A privacy statement $t_f = ((r_t, c_t, d_t), (e_t, k_t, q_t))$ is relevant to a flow $f = (r, d, p)$ (denoted as $t_f \simeq f$) iff $\wedge r \sqsubseteq_\rho r_t \wedge d \sqsubseteq_\delta d_t \wedge p \sqsubseteq_\kappa p_t$. Let T_f be the set of flow-f-relevant policy statements in the set of policy statements T of a privacy policy, then $T_f = \{t_f \mid t_f \in T \wedge t_f \simeq f\}$.*

Definition 7.8.3 (Flow-to-Policy Consistency)**.** *A flow f is said to be consistent with a privacy policy T iff $\exists t_f \in T_f$ such that $c_t = collect \wedge k_t = for$ and $\nexists t'_f \in T_f$ such that $c'_t = not\_collect \vee k'_t = not\_for$.*

A data flow is inconsistent with a privacy policy if the Flow-to-Policy Consistency condition is not satisfied. For example, an opt-out policy *((ad_platform, collect, data), (data, not_for, tracking))* is inconsistent with a data flow *(ad_platform, user_ID, tracking)* when the ad platform still retains a user ID cookie *uid=<unique_ID>* for tracking users after an opt-out even though the opt-out policy states that they will cease their tracking practice.

For the sake of brevity, the definitions are for policy statements with a specified usage purpose. If the data usage purpose *du* of a policy statement is unspecified, i.e., $du = None$, the conditions on the data usage purpose are ignored during the checking.

7.8.3 Inconsistency-Detection Rules

`OptOutCheck` detects two types of consistency corresponding to the two opt-out policy classes. If the opt-out policy is *No-tracking*, the collection of unique IDs for tracking purposes after the user opted out is inconsistent. If the policy is *No-data-collection*, the collection of any data (such as unique ID, user location, web page URLs and IP address) is inconsistent.

The following theorem formalizes an inconsistency when a tracker still collects unique IDs for tracking purposes after users' opt-out. Its proof is given in Appendix E.4.

Theorem 7.8.4 (Unique-ID Tracking Inconsistency)**.** *The collection of unique IDs for tracking purposes after users' opt-out is inconsistent with a No-tracking or No-data-collection opt-out policy.*

7.9 Large-scale Study

7.9.1 Tracker Selection

We select widely-used tracker lists that provide the websites of trackers' owner companies and privacy policies to derive a tracker dataset. In particular, we use the tracker

Tracker Database	# Trackers
WhoTracksMe	3,194
Disconnect	1,393
Evidon	796
DuckDuckGo	229
Merge	4,021

Table 7.5: Sizes of the tracker databases.

Filtering Step (Removal)	# Trackers
Fail-to-load pages	3,319
Duplicate home pages	3,097
Duplicate site domains	2,981

Table 7.6: Tracker-list filtering steps, starting from the merged tracker list.

databases provided by WhoTracksMe [120, 176], Disconnect Tracking Protection [86], Evidon Global Opt-out [73] and DuckDuckGo Tracker Radar [89]. These databases have 229–3,194 trackers as shown in Table 7.5. The WhoTracksMe database contains trackers from usage data collected via the Ghostery extension's users from May 2017 to March 2022. The Disconnect database is created and updated by using manual reviews of trackers' scripts/privacy policies and error reports from the companies labeled as trackers [85]. The Evidon Global Opt-out tool contained 796 trackers at the time of this writing. Finally, we extracted privacy-policy URLs of 229 trackers from the 2022 March crawl of the DuckDuckGo Tracker Radar database. We did not use tracker domains in ad-blocking lists such as EasyList [90] because many of them were resolved to only file servers without obvious connection to the trackers' privacy policies. By uniquely identifying each tracker by its pay-level domain, merging the three selected lists yields a list of 4,021 unique trackers. The number of trackers the crawler successfully loads a home page is 3,319.

Finally, we remove trackers with home pages redirected to the same web domains, leaving 2,981 trackers. This step is to avoid those ad platforms that provide multiple different ad services. For example, 29 home pages of Google ad services have the same *google.com* domain. We do not exclude non-English home pages at this stage to avoid the removal of multilingual trackers which may have a non-English home page but an English privacy policy. Table 7.6 shows the number of trackers extracted throughout the filtering steps.

7.9.2 Extraction of Opt-out Buttons

From the selected 2,981 trackers, the Google Programmable Search Engine yielded 14,059 links for 71.72% (2,138/2,981). Only 2% of tracker websites disallowed the Google search engine by using *robots.txt*. Refining the search results to only the top-3 links and removing links to PDF files (e.g., PDF privacy policies) yielded 5,323 links for opt-out page candidates of 71.05% (2,118/2,981) trackers.

Extracting opt-out buttons from the opt-out page candidates led to opting out 195 trackers, i.e., detected an opt-out button and found opt-out cookies after clicking the button. After excluding 30 trackers with non-English opt-out pages, `OptOutCheck` identified 265 opt-out cookies from 165 trackers. Using only the pattern-based classifier, it could still identify 254 opt-out cookies from 160 trackers, demonstrating the effectiveness of opt-out-cookie patterns. Table 7.7 shows the trackers after each opt-out choice extraction step.

Performance Evaluation. We evaluate the recall rate of the opt-out button extractor by randomly selecting 50 trackers in the Tracker dataset (Section 7.9.1) and manually identifying the opt-out choices provided by these trackers. Of these, we found 10 trackers providing opt-out buttons (other 4 trackers were excluded because their opt-out buttons led to nonexistent web pages or the policies were not written in English). The opt-out button extractor extracted 5 buttons with a precision of 100% and a recall rate of 50%. The majority of the missing cases were due to the opt-out buttons required multiple steps to active such as visiting another web page, clicking a checkbox, and submitting the opt-out.

7.9.3 Extraction of Opt-out Policies

`OptOutCheck` found sentences related to opt-out policies in most of the 165 trackers with an English opt-out web page in Section 7.9.2. Specifically, the system analyzed 1,369 opt-out-related sentences in the privacy policies of 152 trackers. It then extracted

Extraction/Analysis Step	# Trks.
Have opt-out page links	2,118
Successfully opted out	195
Have English opt-out pages	165
Opt-out policies extracted	42
Data flow extracted	33
Inconsistencies detected	11

Table 7.7: Number of trackers during opt-out choice analysis. *Trks* stands for trackers.

Policy Class	# Sents. (# Trks.)	Precision
No-tracking	27 (21)	84.6%
No-data-coll.	28 (26)	85.2%
Total	54 (42)	

Table 7.8: Extracted policy classes. *Sents* stands for sentences.

55 opt-out policies from 54 sentences of 42 trackers (a tracker may contain multiple policy statements). The most common policy class is *No-data-collection* with 26 trackers.

Two authors manually verified the extracted policies that were *unseen* by the opt-out classifiers in the training set. The results show that the classifiers achieved high precision rates of 84.62% (22/26) and 85.19% (23/27) for the *No-tracking* and *No-data-collection* classes, respectively. Table 7.8 shows the policy classification results.

7.9.4 Extraction of Data Flows

7.9.4.1 Measurement Procedure

We analyze the differences of cookies on publisher websites between before and after opting out of a tracker T to detect the changes in the data-collection behavior of T. This process avoids false positives due to the cookies set by the tracker's own website when `OptOutCheck` visited it for opting out. These cookies may entail first-party data collection of T that is unrelated to T's third-party tracking services. Specifically, `OptOutCheck` first visits a set of publisher websites using a clean instance of a web browser and records the set T_c of cookies under T's opt-out domains. It then visits the tracker's website and activates the opt-out choices provided by T. `OptOutCheck` confirms that the opt-out has been set successfully by checking the presence of the tracker's opt-out cookies. Finally, `OptOutCheck` visits publisher websites again and records the values of the cookies in T_c.

Due to the randomness of placement of online advertisements, `OptOutCheck` sequentially visits a set of candidate web pages S until it finds 10 web pages that send requests containing the cookies of T, or S is exhausted. We use the *WhoTracksMe* and DuckDuckGo Tracker Radar cookie databases that contain the lists of trackers detected on top websites to generate S for each tracker.

7.9.4.2 Extracted Data Flows

Of the 165 trackers with opt-out buttons, `OptOutCheck` found 129,286 candidate websites for 146 trackers where their cookies may have been placed. Each tracker has an average of 582 (*SD* 1,026) candidate websites.

Following the measurement procedure in Section 7.9.4.1, `OptOutCheck` scanned 476 websites and extracted 52 data flows from 4,341 for 33 trackers. Unique identifiers are the most common data type and found on 98% of the flows. The other data type is the information about the user's IP address and city name included in cookie *geode* of *udmserve.com*.

7.9.5 Opt-out Choice Inconsistencies

7.9.5.1 Detected Inconsistencies

`OptOutCheck` detected 11 trackers that had conducted tracking and data collection inconsistently with their opt-out policies after activating the opt-out choices. Two authors independently verified the results by manually following the measurement procedure (Section 7.9.4.1 and checking the existence of tracking cookies using Chrome DevTools. We determined the purposes of cookies from cookie names, values and cookie description (if there is any). All of the detected inconsistencies were confirmed to be correct. Appendix E.5 provides details of the detected inconsistent flows, opt-out policies, opt-out cookies and domains.

Although the number of the detected inconsistent trackers is low, they tracked a significant amount of web traffic while the inconsistencies are direct violations of the trackers' privacy policies. On average, each tracker was present at 0.64% (*SD* 1.27%) across all page loads and on 3.65% (*SD* 6.57%) of the top 10k websites where they were included as a third-party in March 2022 [120]. Given that there were 4.95 billion Internet users [288], these inconsistencies might affect a significant number of users.

Case Studies. Criteo, which was present on 21% of the top 10k websites [120], contains multiple statements describing how its opt-out choice works such as "disable Criteo services will result in the deletion of the cookies dropped by Criteo in your browser you are currently using that allows us to recognize your browser or device" and "the termination of the collection of your personal data." Therefore, the opt-out policies are *No-tracking* and *No-data-collection*. However, after clicking "disable Criteo services" and the opt-out cookie *optout=1* was set, cookie *uid* was still retained with a unique ID. Both of these cookies were under *.criteo.com* domain.

Underdog Media instructed users to "opt out of our Underdog Media hosted technology by clicking here." After clicking the opt-out button, the website confirmed the status of "opt-out for Underdog Media hosted 3rd Party Cookies." Therefore, this opt-out policy was classified as *No-data-collection*. The button set an opt-out cookie *optout=Thank_You* but the tracker still retained multiple cookies to collect data from users. One of the cookies was *geode* that contained the IP address and city name of the browser.

As another example, *adtriba.com* instructed users that "to be excluded from Adtriba third party tracking, you can click the following button." This opt-out policy was classified as *No-tracking*. However, users even with an opt-out cookie *atboptout=1* were still tracked. The tracker still retained an *atbgdid* cookie that contains a device ID [118]. This cookie was under *.adtriba.com* domain and existed on publisher websites even before our visit to *adtriba.com* for opting out, so it was likely used for third-party tracking purposes. However, because the policy is *No-tracking*, we expect all tracking cookies to be removed after an opt-out.

7.9.5.2 Root Cause Analysis

The inconsistencies could be due to an incomplete/buggy implementation of opt-out choices since trackers might not always develop and test this feature completely. In all the detected inconsistencies, the opt-out cookies were successfully set after clicking the opt-out button, demonstrating that the trackers made an effort to record the opt-out preferences. However, the tracking cookies were still retained, so we hypothesize that the trackers are not successful at making the opt-out choice fully functional.

Since the trackers have incentives to keep users from opting out of their tracking, they might also attempt to make the opt-out process unnecessarily complex for the end-users. We found that 3 trackers in the detected inconsistencies did not automatically delete their tracking cookies. For example, *criteo.com* retained *uid* cookie after an opt-out although the cookie did not reappear after its deletion. However, since many trackers automatically deleted their tracking cookies upon opt-out, there should not be any difficulty of automatic deletion of a tracker's own cookies. Therefore, it is unreasonable to require average end-users to open Chrome DevTools to manually search and delete the tracking cookies while retaining the necessary opt-out cookies.

Regardless whether the inconsistencies were accidental bugs or deliberately created by the trackers to mislead the users, since the opted-out users revoked their consent of tracking and/or data collection, the tracker companies conducted inconsistent data practices without the opted-out users' consent. Therefore, the companies may face heavy fines from regulators due to the deceptive privacy practices and unlawful data collection. It is a tracker's responsibility to ensure the consistency between its stated privacy policy and the actual data practices of its services. Given the detection of such inconsistencies by

`OptOutCheck`, the trackers, developers and regulators can investigate and resolve their root causes.

7.10 Notification to and Responses from Trackers

Of the 11 detected inconsistent trackers, we informed 10 trackers of the detected inconsistencies in their opt-out choices. We excluded *deepintent.com* because it drastically updated the website and removed the opt-out choice at the time we contacted the trackers. Each notification email included our interpretation of the opt-out policies, our detected opt-out and tracking cookies, and the steps we took to reproduce the inconsistencies for each tracker. All of the emails appeared to be delivered successfully.

One of the trackers responded to our notification and made changes to their privacy policies to correct the detected inconsistency. In particular, Taboola's Privacy Team confirmed our finding of their opt-out inconsistency. They updated their opt-out method to immediately delete the tracking cookie *t_gid* after an opt-out, which also sets an opt-out cookie DNT=1, to stop tracking users. They changed the opt-out button to point to a dedicated opt-out portal.[1] Specifically, they said "To avoid any confusion, and in an excess of caution, we have since updated the opt-out in our privacy policy so that it goes directly through Taboola's Data Subject Access Request Portal[1] instead and the user's t_gid cookie is deleted straight away." Two researchers in our team manually verified that their changes fixed the opt-out inconsistency. It is worth noting that Taboola is categorized as "very prevalent" and ranked at 37/920 top most prevalent trackers on the Web by WhoTracks.me. Its cookies are present in 2.4% of all page loads on the top-10k sites [120].

[1] `https://accessrequest.taboola.com/`

7.11 Limitations and Future Work

One of the bottlenecks of OptOutCheck is the analysis of the web-based privacy policies for accurate extraction of opt-out buttons and policies. A major challenge there is the extraction of the sequence of user actions to activate an opt-out choice and the relevant policy statements from complex website content. Specifically, existing textual information extraction techniques are not applicable to extract multi-step interactions and complex legal statements from general non-standardized website layouts. Although we have developed heuristics to extract policy statements from the sentences next to an opt-out button, a holistic analysis of the whole privacy-policy web pages will likely improve the recall rate. For example, *adform.com* placed the opt-out buttons on the sidebar far away from the opt-out policy statements, preventing/hindering OptOutCheck's extraction of the opt-out policies. However, the document-level analysis needs advances in natural language understanding and information extraction that have been studied extensively for decades [207].

OptOutCheck analyzes policies on a sentence basis and hence misses several cases due to the references to a previous sentence, such as in "we will stop this process when you opt out" where "this process" refers to the data collection for targeted advertising in the previous sentence. A holistic analysis of multiple sentences or the whole document may yield a better analysis and improve the recall rate. We did not check contradictions in opt-out privacy policies either, because the opt-out choice descriptions are usually short, and hence unlikely contain contradictions. Furthermore, the opt-out policy corpus in Section 7.5.4 is still small with little support. We plan to use ML-based opt-out policy classifiers trained on a larger dataset to cover flexible grammars in privacy policies. This is part of our future inquiry.

We have not addressed other storage mechanisms such as HTML5 LocalStorage, and advanced web tracking mechanisms such as canvas fingerprinting [3] due to the vagueness of their privacy policies. While the opt-out policies provided the definitions of cookie-based data collection and/or tracking, concrete descriptions of other technologies were often omitted. So, we leave the analysis of other tracking technologies as future work.

It is challenging to analyze data types and usage purposes of cookies without knowing their server-side processing. Unlike well-defined programming API (e.g., Android API specification), most cookies have no such specification of their purposes and value ranges. Furthermore, for security and performance reasons, the values of cookies are usually not human-readable but encrypted or encoded. Despite these challenges, researchers attempted to extract the purposes of transferred data from client-side information only [166, 275]. So, we leave the analysis of complete purposes of cookies as future work.

Major cookie-blocking desktop web browsers (e.g., Edge and Firefox) do not block *all* third-party cookies by default [218, 223]. Cookies used for certain purposes, such as analytics, are not blocked by Edge using the default browser settings. For example, we found empirically that both Firefox and Edge still allowed *adnxs.com*'s tracking cookies on *cnn.com*. Furthermore, the browsers do not completely prevent data leakage via URL parameters. Therefore, `OptOutCheck`'s framework is still valid for these browsers.

Although our corpus focuses on websites in the US and privacy policies written in English, `OptOutCheck` is applicable to inconsistent trackers in other languages and countries. The implementation of opt-out choices and related policy statements may vary with the requirements of local privacy laws. However, analyzing and comparing the differences of regulations between different countries are outside the scope of this paper.

Automatic detection of the discrepancies between the stated privacy policies and actual data-collection behavior of trackers benefits all stakeholders of the Web ecosystem. First, regulators can readily scan trackers for critical violations to protect users. Second, the end-to-end automated framework can be easily integrated into the workflow of companies to assess the potential privacy risks in their system and gain more trust from users. Finally, users can avoid privacy risks due to misleading statements in the privacy policies of online trackers.

We have already set up a website at [103] to increase user awareness of the inconsistencies of opt-out choices of online trackers discovered so far. We will soon inform these trackers of our findings.

7.12 Related Work

While there has been research on the cookie consent settings and opt-out choices in the privacy policies of publisher websites [28, 137, 138, 208] the scope of `OptOutCheck` about online trackers is very different. Likewise, prior work on flow-to-policy consistencies of Android apps [20, 21, 46] does not directly applicable to online services. We summarize prior research on online trackers and their opt-out policies.

Opt-out Choices of Online Trackers. Balebako *et al.* [27] measured the effectiveness of privacy tools including the opt-out cookie mechanism and found that the opt-out cookies were effective in limiting OBA. Sakamoto *et al.* [264] studied the opt-out cookie mechanism provided by ad agencies for opting out of OBA to find that the advertisers continued to track users when the users started to browse again. However, these are limited to evaluating the effectiveness of opt-out tools without systematically considering opt-out policies such as *No-tracking* and *No-data-collection* for opted-out users.

Komanduri *et al.* [183] examined privacy policies of members of Digital Advertising Alliance (DAA) and Network Advertising Initiative (NAI) to evaluate their compliance with the self-regulatory principles on top 100 websites and reported non-compliance instances. They found 93% of 74 surveyed policies provide their own definitions of opt-out and 57% provide the opt-out definitions stronger than the minimum requirements of DAA and NAI. However, they assumed that the opt-out preferences would be honored by the advertisers. Cranor *et al.* [72] manually analyzed 75 privacy policies of advertisers who were members of DAA, and found the policies kept silent on many consumer-relevant practices. Our tools analyze the policies automatically and go beyond the members of DAA and NAI. Although these studies laid a foundation for analysis from a legal perspective, they did not develop any automated method to extract information from privacy policies.

Measurement of Online Trackers. Numerous researchers have studied the network of online trackers. Englehardt *et al.* [93] conducted large-scale measurements of online trackers on the top 1M websites. Lerner *et al.* [192] conducted longitudinal measurements of third-party web tracking for 10 years and found increasing prevalence and complexity of third-party tracking on the web. Iordanou *et al.* [164] analyzed the data flows across the borders of EU nations and found that the majority of tracking flows cross countries in Europe but are well confined within the GDPR jurisdiction. Yang *et al.* [309] compared web tracker ecosystems on desktop and mobile environments. However, the prior work has not analyzed the trackers' privacy policies and verified whether the tracking practices followed the opt-out policies or not.

7.13 Conclusion

We have presented `OptOutCheck`, an end-to-end automated framework that detects inconsistencies between the actual data practices of online trackers and their policy statements regarding user opt-out choices. We have classified opt-out policies and created automatic NLP-based classifiers to extract the policies from trackers' opt-out web pages. Based on the patterns of HTML elements, `OptOutCheck` identifies opt-out buttons, simulates users' opt-out, detects opt-out cookies, and extracts data flows from the cookies and URL parameters sent to tracker servers. Finally, we have constructed a formal model to detect the inconsistencies between the opt-out policies and the associated data flows. A large-scale study shows that trackers still continue the same data practices that contradict their stated opt-out policies even though these inconsistencies are violations of the trackers' own policies and may lose the users' trust in their services. `OptOutCheck` has laid a foundation for automatic detection of discrepancies between the opt-out choices and the actual data practices of online services.

CHAPTER VIII

Conclusions and Future Directions

8.1 Conclusion

In this dissertation, we systematically identify and assess privacy risks associated with the privacy notices and opt-out choices of mobile apps and web services, from the user interface to actual data collection/sharing. We address privacy issues of the end-user presentation of privacy policies, the data practices of mobile apps and browser extensions, as well as the opt-out choices of websites and online trackers. The inconsistencies detected by the systems/tools in Chapters IV – VII are potential breaches of consumer protection laws such as the GDPR, CCPA and FTC Act. In each of the assessments, we highlight the limitations of the state of the art in evaluating user privacy risks and propose automatic systems to address them. We summarize the main contributions of this thesis as follow.

First, starting from the end-users, we inform users of privacy practices via the presentation of privacy policies. In Chapter III, we introduce `PI-Extract` to automatically extract data practices from privacy policies and help users better understand them through an easy-to-read presentation. We demonstrate that the system can perform the extraction at an accuracy higher than a state-of-the-art method while the proposed data-practice annotations significantly improve users' comprehension.

Second, at the app-behavior level, we analyze the flow-to-policy (in)consistency of mobile apps and web browser extensions. In Chapter IV, we extract the purposes of data usage from privacy policies and data flows, and then construct a formal model to detect inconsistencies of data flows with the stated purposes in the privacy policies. The system detects multiple contradictions in the privacy policies and flow-to-policy inconsistencies of mobile apps. In Chapter V, we develop a system to automatically analyze the data collection of web browser extensions and check the consistency of their actual data collection with their stated privacy practices. We find multiple browser extensions performing data practices inconsistent with their privacy disclosures.

Finally, we create systems to assess the consent/opt-out choices provided by websites and online trackers. In Chapter VI, we design `ConsentChk` to analyze cookie consent settings to detect inconsistencies in the cookie preference enforcement of websites. The system detects and quantifies critical issues such as incorrectly enforced cookie preferences and contradictory cookie preferences, indicating the prevalence of problematic cookie consent management. In Chapter VII, we introduce `OptOutCheck` to identify the inconsistencies between the opt-out policies and the opt-out settings of online trackers. The system detects inconsistencies between data practices and opt-out policies of multiple popular trackers that are critical violations of the trackers' own policies and lose user trust in online services.

8.2 Future Research Directions

While this dissertation has embarked on mitigating privacy risks for end-users, there still remain multiple questions to be answered. Discussed below are several samples of them.

8.2.1 Holistic Analysis of Privacy Policies

The detection of flow-to-policy inconsistencies in Chapters IV, V and VII can be improved by developing more advanced semantic analysis and formal models for privacy policies. First, the extraction of policy statements is still limited to the sentence level and cannot analyze references across multiple sentences and even different parts of the policies. We need to investigate a more holistic analysis that can handle multiple sentences and the whole privacy policy at once. This direction will involve the creation of large datasets and the development of advanced ML-based semantic extraction with optimizations for privacy policies.

Second, the current formal representations of policy statements only model the *data types* and *purposes* of data-(un)collection statements but more complete representations of policy statements are needed to better model the policies and detect inconsistencies more accurately. For example, most privacy policies include statements for multiple platforms (such as websites, apps, and extensions) but determining the conditions and/or the platform-scopes of data collection is necessary for checking the flow-to-policy consistency of a particular platform such as browser extensions. Therefore, we need to extend the formal models to other important aspects, such as the *conditions* and *scopes*, of the statements in privacy policies.

8.2.2 Data-type and Purpose Inference

The flow-to-policy consistency analysis in Chapters IV – VII relies upon the inference of data types and purposes from low-level key-value pairs in data traffic but the extraction is still limited due to the lack of server-side information. More sophisticated data-flow inference techniques will extract more data flows and improve the recall rates. We need to improve the extraction by analyzing the description/declaration of the purposes of cookies and creating datasets for training ML-based extractors. Specifically, we will need to support the remaining 5 contextual data types for `ExtPrivA` by analyzing input fields (e.g., <*input*> attribute *type="password"* indicates password input (authentication information)), performing static analysis on extension code (e.g., registration prompts in extensions' pop-up pages) and analyzing extension behavior after signing in to websites.

8.2.3 Integration to Development Environments

Our end-to-end automated systems can be easily integrated into the workflows of companies. However, *how to integrate them into integrated development environments (IDE) to support developers to build Android apps and debug websites, and evaluate the usability of such tools* is still an open problem. These tools will greatly help app developers avoid privacy-related bugs and reduce the privacy risks of end-users.

8.2.4 Privacy-Risk Assessment of Novel Environments

While the systems developed in this thesis have addressed a wide range of mobile and web apps, other environments, such as the Internet of Things (IoT), smart homes and wearable devices, have not yet been studied. Since these novel apps have become increasingly popular and are collecting privacy-sensitive user data, we need to assess privacy risks in these environments.

8.2.5 Automatic Checking of Compliance with Regulations

We need to develop an automated technique to check the compliance of apps' privacy policies and actual data collection against privacy regulations such as the GDPR and CCPA. Rather than only checking certain aspects of the regulations such as the requirements of cookie notices, it would be more generic and desirable to model the privacy requirements stated in the regulations. The models of regulation requirements would be different from the existing formal models of the data collection and sharing in privacy policies. An automatic checking tool would greatly help companies avoid non-compliance with privacy regulations.

APPENDICES

APPENDIX A

PI-Extract

A.1 User Survey Instruments

The following is the questions used in the `DPA` version of the survey described in Section 3.7. The `Plain` version is the same except does not have the highlighted text while `DPA-Err` version has annotations which are ommitted or contain an incorrect action label.

[Introduction]

We would like to understand your opinion about the presentation of privacy policies of websites.

By continuing you agree with the collection of your answers in the survey. Your responses for this survey are used for academic research purposes only.

The survey will take 5-10 minutes to complete.

[Demographic Questions]

What is the highest level of school you have completed or the highest degree you have received?

○ Less than high school degree

○ High school graduate (high school diploma or equivalent including GED)

○ Some college but no degree

○ Bachelor's degree in college (4-year)

○ Graduate (Master's or Doctoral) degree

○ Professional degree (JD, MD)

○ Prefer not to answer

What is your gender?

○ Male

○ Female

○ Prefer not to answer

Are you employed?

○ Yes

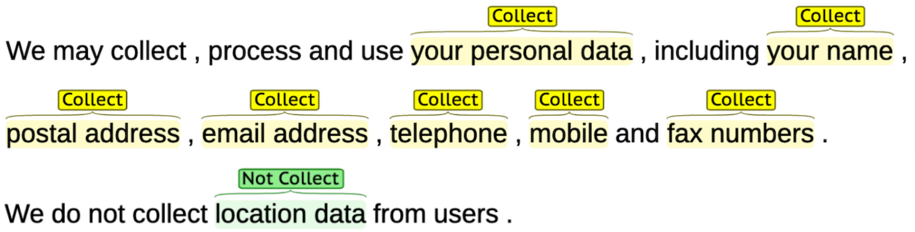
○ No

○ Prefer not to answer

What is your year of birth? **[A text box is presented]**

[Training Questions]

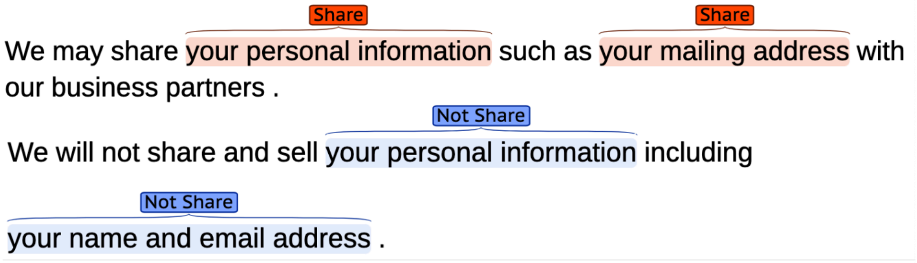
To help you understand privacy policies faster, the following sentences highlight the data that the company collects or does not collect from users.



We may collect , process and use your personal data , including your name , postal address , email address , telephone , mobile and fax numbers .

We do not collect location data from users .

To help you understand privacy policies faster, the following text highlights the user's data that the company shares or does not share with other businesses.

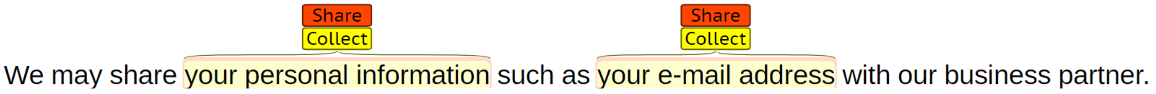


We may share your personal information such as your mailing address with our business partners .

We will not share and sell your personal information including your name and email address .

Read the sentence and answer the questions about a company's privacy policy. You can leverage the highlights to answer faster.

Note that some highlights sometimes may be missing or contain an incorrect label.


Share
Collect
Share
Collect

We may share your personal information such as your e-mail address with our business partner.

May the company collect your personal information?

○ Yes

○ No

May the company share your e-mail address?

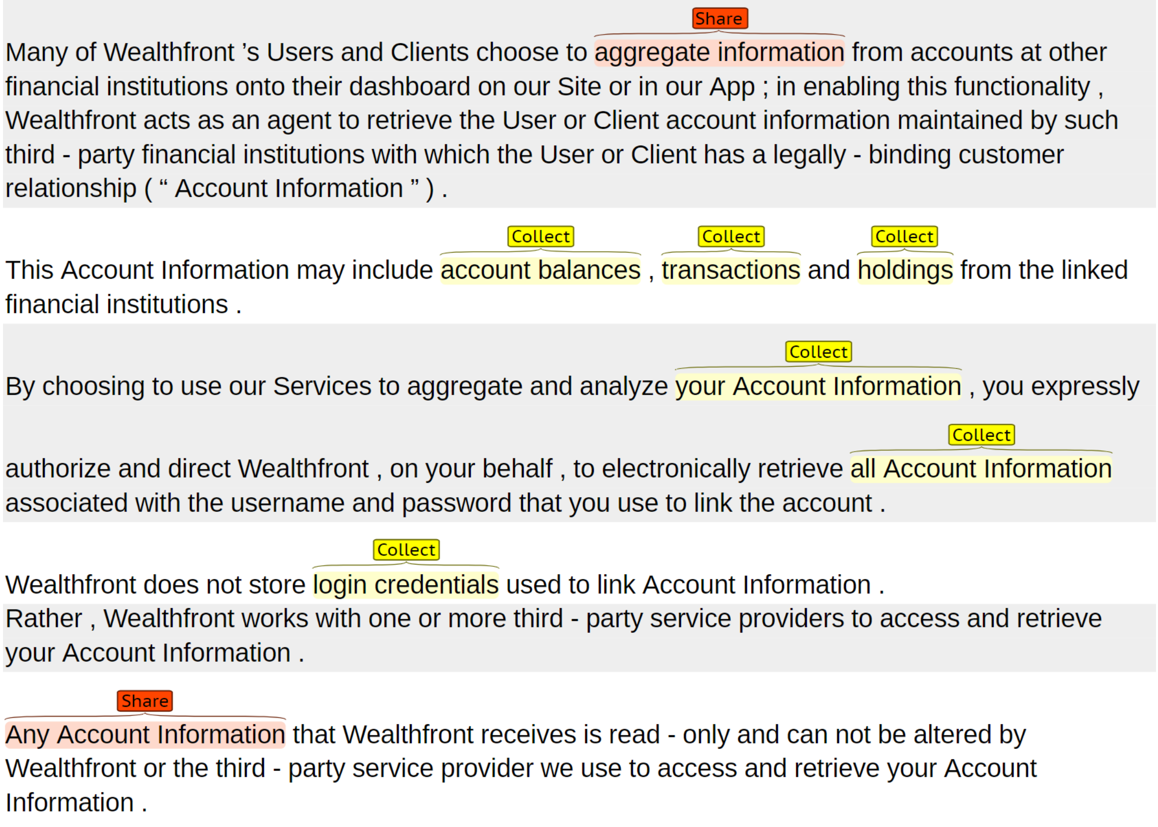
○ Yes

○ No

[Main Questionnaire]

[Excerpt E1]

Read the following paragraph from privacy policy from a financial service and answer the question below.


Share
Collect
Collect
Collect
Collect
Collect
Collect
Share

Many of Wealthfront 's Users and Clients choose to aggregate information from accounts at other financial institutions onto their dashboard on our Site or in our App ; in enabling this functionality , Wealthfront acts as an agent to retrieve the User or Client account information maintained by such third - party financial institutions with which the User or Client has a legally - binding customer relationship (" Account Information ") .

This Account Information may include account balances , transactions and holdings from the linked financial institutions .

By choosing to use our Services to aggregate and analyze your Account Information , you expressly authorize and direct Wealthfront , on your behalf , to electronically retrieve all Account Information associated with the username and password that you use to link the account .

Wealthfront does not store login credentials used to link Account Information .
Rather , Wealthfront works with one or more third - party service providers to access and retrieve your Account Information .

Any Account Information that Wealthfront receives is read - only and can not be altered by Wealthfront or the third - party service provider we use to access and retrieve your Account Information .

As stated in the paragraph, which of the following practices is true about your transactions from linked financial instutions?

○ Collected by the service

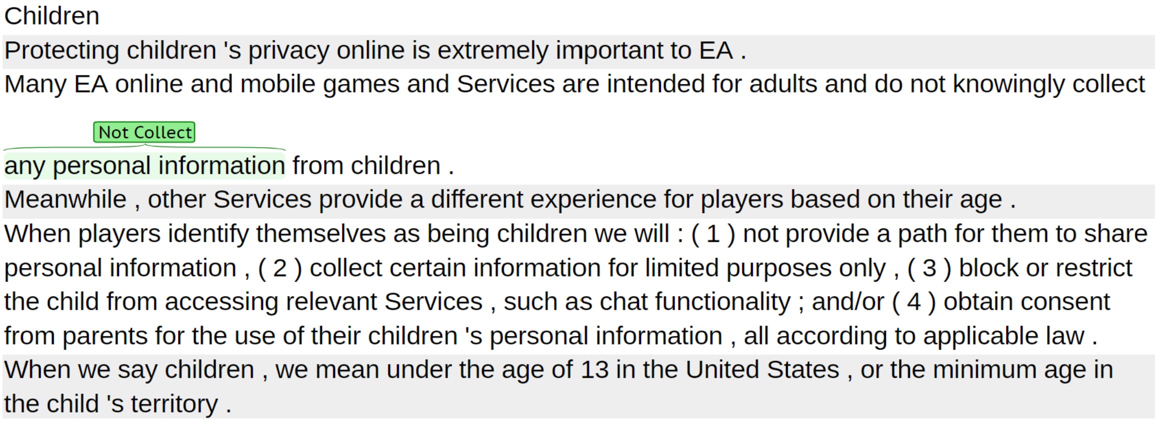
○ Not collected by the service

○ Shared by the service

○ Not shared by the service

[Excerpt E2]

Read the following paragraph from the privacy policy of a gaming service and answer the question below.



Children

Protecting children 's privacy online is extremely important to EA .

Many EA online and mobile games and Services are intended for adults and do not knowingly collect any personal information from children .

Meanwhile , other Services provide a different experience for players based on their age .

When players identify themselves as being children we will : (1) not provide a path for them to share personal information , (2) collect certain information for limited purposes only , (3) block or restrict the child from accessing relevant Services , such as chat functionality ; and/or (4) obtain consent from parents for the use of their children 's personal information , all according to applicable law .

When we say children , we mean under the age of 13 in the United States , or the minimum age in the child 's territory .

As stated in the paragraph, which of the following practices is true about personal information from children under 13 in the United States?

○ Collected by the service

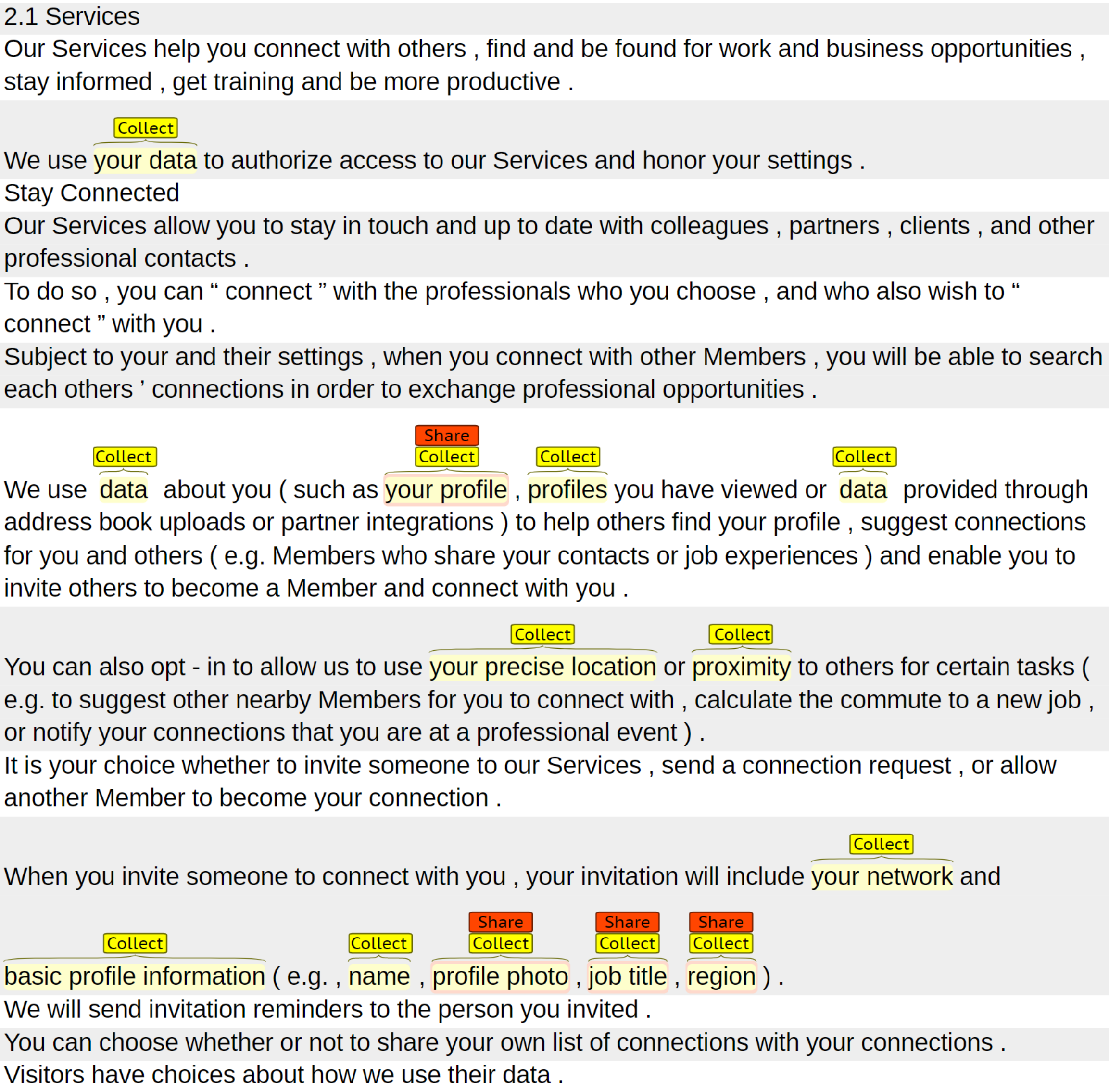
○ Not collected by the service

○ Shared by the service

○ Not shared by the service

[Excerpt E3]

Read the following paragraph from the privacy policy of a professional social network and answer the questions below.



2.1 Services

Our Services help you connect with others , find and be found for work and business opportunities , stay informed , get training and be more productive .

We use your data to authorize access to our Services and honor your settings .

Stay Connected

Our Services allow you to stay in touch and up to date with colleagues , partners , clients , and other professional contacts .

To do so , you can “ connect ” with the professionals who you choose , and who also wish to “ connect ” with you .

Subject to your and their settings , when you connect with other Members , you will be able to search each others ’ connections in order to exchange professional opportunities .

We use data about you (such as your profile , profiles you have viewed or data provided through address book uploads or partner integrations) to help others find your profile , suggest connections for you and others (e.g. Members who share your contacts or job experiences) and enable you to invite others to become a Member and connect with you .

You can also opt - in to allow us to use your precise location or proximity to others for certain tasks (e.g. to suggest other nearby Members for you to connect with , calculate the commute to a new job , or notify your connections that you are at a professional event) .

It is your choice whether to invite someone to our Services , send a connection request , or allow another Member to become your connection .

When you invite someone to connect with you , your invitation will include your network and basic profile information (e.g. , name , profile photo , job title , region) .

We will send invitation reminders to the person you invited .

You can choose whether or not to share your own list of connections with your connections .

Visitors have choices about how we use their data .

As stated in the paragraph, which of the following practices is true about your precise location?

○ Collected by the service

○ Not collected by the service

○ Shared by the service

○ Not shared by the service

As stated in the paragraph, which of the following is shared with another person who you invite to connect?

○ Your job title

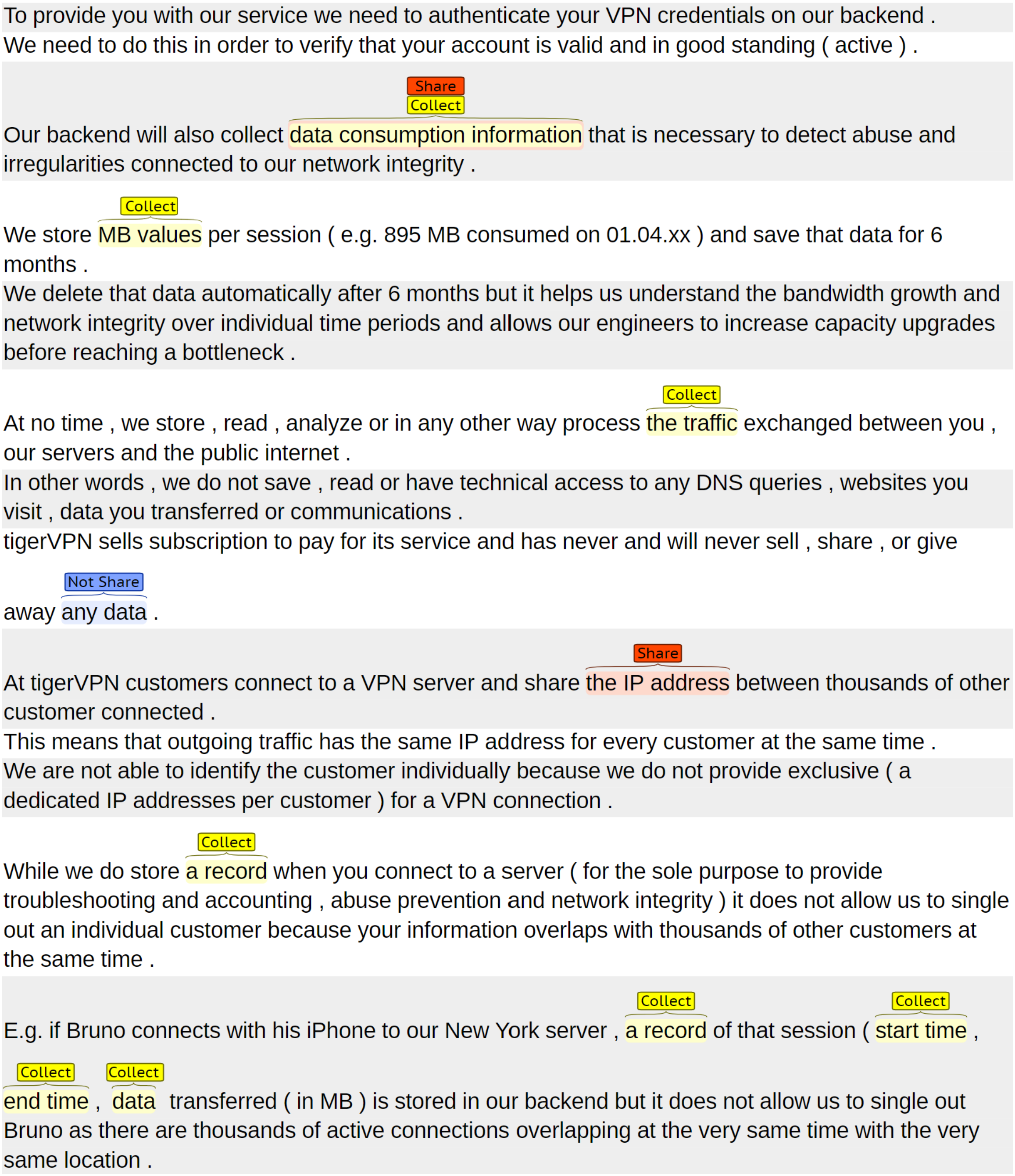
○ Your address

○ Your professional skills

○ Your preferred social networks

[Excerpt E4]

Read the following paragraph from the privacy policy of a virtual private network website and answer the questions below.



To provide you with our service we need to authenticate your VPN credentials on our backend .

We need to do this in order to verify that your account is valid and in good standing (active) .

Our backend will also collect data consumption information that is necessary to detect abuse and irregularities connected to our network integrity .

We store MB values per session (e.g. 895 MB consumed on 01.04.xx) and save that data for 6 months .

We delete that data automatically after 6 months but it helps us understand the bandwidth growth and network integrity over individual time periods and allows our engineers to increase capacity upgrades before reaching a bottleneck .

At no time , we store , read , analyze or in any other way process the traffic exchanged between you , our servers and the public internet .

In other words , we do not save , read or have technical access to any DNS queries , websites you visit , data you transferred or communications .

tigerVPN sells subscription to pay for its service and has never and will never sell , share , or give away any data .

At tigerVPN customers connect to a VPN server and share the IP address between thousands of other customer connected .

This means that outgoing traffic has the same IP address for every customer at the same time .

We are not able to identify the customer individually because we do not provide exclusive (a dedicated IP addresses per customer) for a VPN connection .

While we do store a record when you connect to a server (for the sole purpose to provide troubleshooting and accounting , abuse prevention and network integrity) it does not allow us to single out an individual customer because your information overlaps with thousands of other customers at the same time .

E.g. if Bruno connects with his iPhone to our New York server , a record of that session (start time , end time , data transferred (in MB) is stored in our backend but it does not allow us to single out Bruno as there are thousands of active connections overlapping at the very same time with the very same location .

As stated in the paragraph, which of the following practices is true about your data?

○ Collected by the service

○ Not collected by the service

○ Shared by the service

○ Not shared by the service

As stated in the paragraph, which of the following may be stored by the service when you connect to a server?

○ The IP address you used

○ A record of your session

○ The messages you sent

○ A unique ID of your device

[Usability Question]

How do the highlighted words help you identify the personal information collected or shared by the company?

○ Not at all helpful

○ Slightly helpful

○ Somewhat helpful

○ Very helpful

○ Extremely helpful

[Feedback]

What is your feedback about this survey (if you have)? **[A text box is presented]**

A.2 Scores and Answering Time

Scores and answering time in user study are shown in Fig. A.1.

A.3 Data Action Examples in `RBE`

Examples of data actions in `RBE` are given in Table A.1.

Entity Role	Data Action	Example
First party	Collect	*We may collect your personal information from Analytics tools.*
Third party	Share	*Our business partners may collect your demographic information.*

Table A.1: Examples of data actions, based on simplified policy statements of PolicyLint, used in `RBE`.

Word Embeddings	Label	Precision	Recall	F1
BERT	*Collect*	62.31	71.15	66.44
BERT	*Share*	55.12	54.07	54.59
BERT	*Not_Collect*	77.78	63.64	70.00
BERT	*Not_Share*	76.19	76.19	76.19
BERT	Overall	61.04	65.66	63.27

Table A.2: Recall-optimized BERT models.

A.4 Recall-optimized BERT models

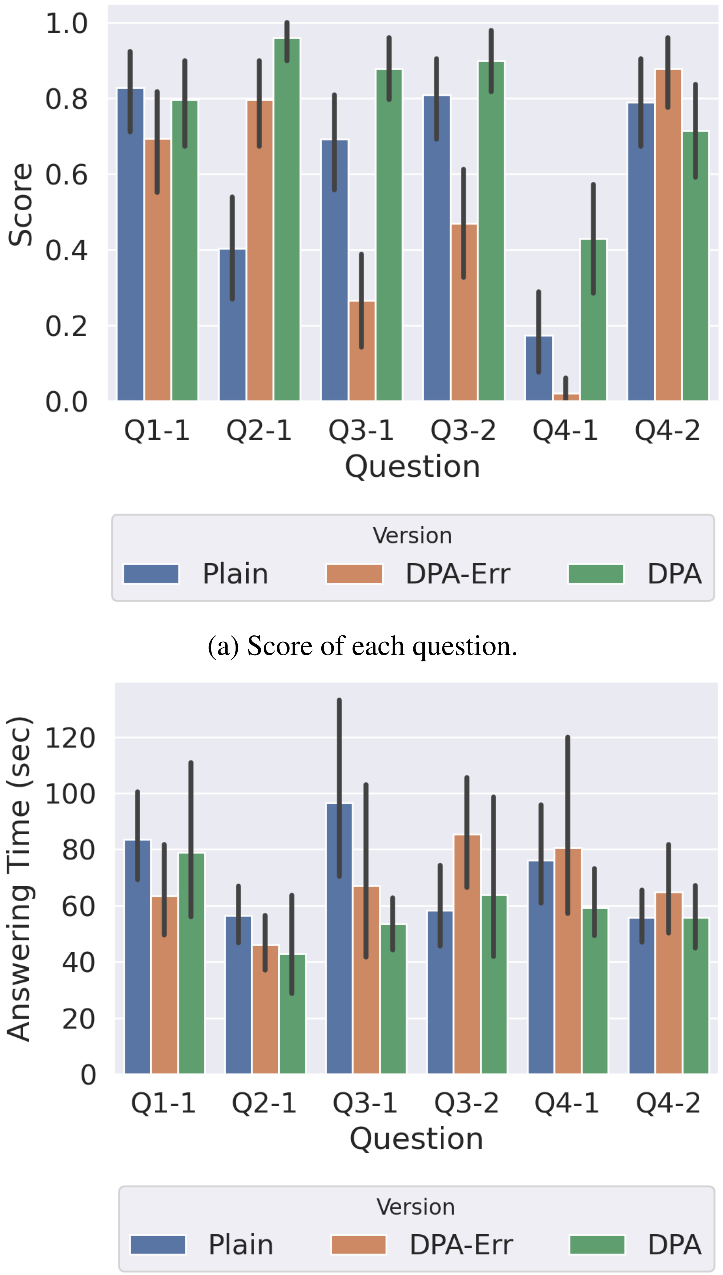
The performance of recall-optimized BERT models is shown in Table A.2.

A.5 Dataset Coverage

The performance of `PI-Extract` for varied dataset sizes is shown in Fig. A.2.

A.6 Corpus IAA and Statistics

The IAA between annotators, number of sentences and tokens of each document in the corpus are shown in Table A.3.



(a) Score of each question.

(b) Answering time of each question.

Figure A.1: Score and answering time of each question in the user study. Error bars are 95% confidence intervals.

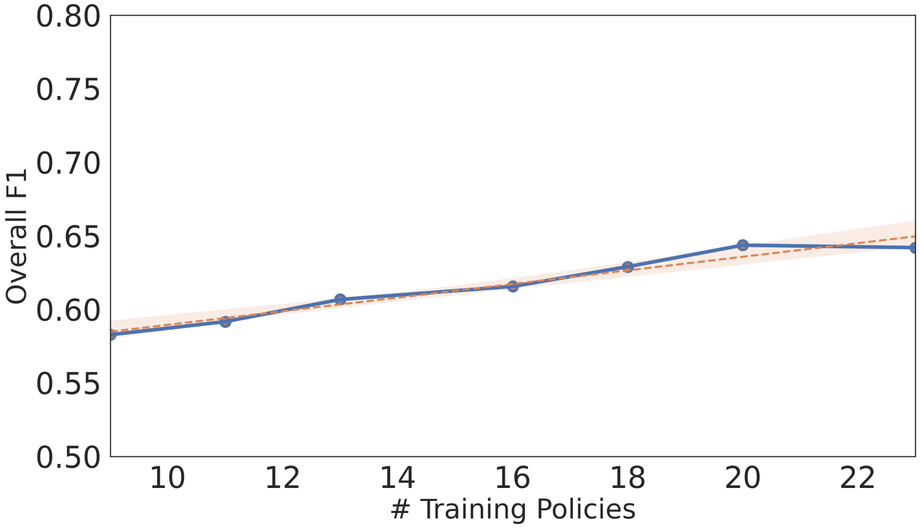


Figure A.2: Overall F1 when increasing the training set size. The linear regression line is dashed and the shade region shows its 95% confidence interval.

Website	Precision	Recall	F1	Support	# Sentences	# Tokens
bankofamerica.com	95.73	91.06	93.33	123	187	4618
yahoo.com	97.83	93.75	95.74	48	76	1573
nytimes.com*	97.96	96.00	96.97	150	200	4317
barnesandnoble.com	97.35	97.78	97.56	225	310	8944
google.com	97.48	98.31	97.89	118	123	3151
instagram.com	97.92	97.92	97.92	96	148	3511
reddit.com	96.83	99.19	97.99	123	163	3536
thefreedictionary.com	100.00	97.30	98.63	37	58	1230
playstation.com	98.68	98.68	98.68	76	135	3484
ted.com	98.41	100.00	99.20	62	54	1336
pbs.org*	100.00	98.48	99.24	66	119	2659
aol.com*	100.00	98.68	99.34	76	135	3291
washingtonpost.com*	100.00	98.73	99.36	79	156	3227
sciencemag.org*	98.77	100.00	99.38	80	128	3195
geocaching.com	100.00	98.78	99.39	82	140	2630
walmart.com	98.84	100.00	99.42	85	228	4589
theatlantic.com*	99.03	100.00	99.51	102	153	4049
gamestop.com	99.12	100.00	99.56	112	169	4295
foxsports.com*	100.00	99.13	99.56	115	126	3590
uh.edu	100.00	100.00	100.00	10	14	343
imdb.com	100.00	100.00	100.00	33	109	2355
thehill.com*	100.00	100.00	100.00	41	53	1669
steampowered.com	100.00	100.00	100.00	56	70	1760
ticketmaster.com	100.00	100.00	100.00	59	147	2054
minecraft.gamepedia.com	100.00	100.00	100.00	73	101	2806
msn.com*	100.00	100.00	100.00	78	86	2090
mlb.mlb.com*	100.00	100.00	100.00	103	122	3606
fool.com	100.00	100.00	100.00	108	183	4734
amazon.com	100.00	100.00	100.00	111	143	3307
esquire.com*	100.00	100.00	100.00	132	228	5700
Total	-	-	-	2,659	4,064	97,649

Table A.3: IAA and statistics of privacy policies in the corpus. *-marked websites were used in the evaluation of `PI-Extract` for policies in the same domain (Section 3.6.2.6).

APPENDIX B

PurPliance

B.1 Semantic Arguments of Purpose Clauses

Semantic arguments of an event do not change even though the syntactic structure of the sentence changes. For example, let us consider the following sentences which express a data-usage event:

- [We]$_{\text{Arg0}}$ do not [share]$_{\text{V}}$ [your personal data]$_{\text{Arg1}}$ [with third parties]$_{\text{Arg2}}$ [for targeted ads]$_{\text{Argm-Pnc}}$;
- [Third parties]$_{\text{Arg0}}$ may not [collect]$_{\text{V}}$ [your personal data]$_{\text{Arg1}}$ [to deliver targeted ads]$_{\text{Argm-Pnc}}$.

While the purpose of *delivering targeted ads* is stated differently in noun and verb phrases starting with *for* and *to*, it is consistently an *Argm-Pnc* (purpose-not-cause) argument of the predicate. The data object *your personal data* is also an *Arg1* in both cases.

Table B.1 lists the predicate-specific semantic arguments of purpose clauses used in addition to the common arguments *Argm-Prp* and *Argm-Pnc*.

Predicates	Argument
use, save, check	Arg2
analyze	Argm-Adv
save, receive, solicit, record	Arg3
receive	Arg4
disclose, give, sell, send, transmit, provide	C-Arg1

Table B.1: Predicate-specific semantic arguments of purpose clauses used by `PurPliance`.

B.2 Examples of Predicate-Object Pairs

Table B.2 shows examples of purpose classification with PO pairs.

Purpose clause	PO pairs
To provide personalized services	(provide, personalized services), (personalize, services)
To comply with laws	(comply, laws)
For promotional purposes	(, promotional purposes)
For scientific purposes	(, scientific purposes)

Table B.2: Examples of purpose classification with PO pairs.

B.3 Policy Purpose Prediction Performance

The performance of policy purpose prediction is shown in Table B.3.

B.4 Purpose Approximation Proof

The following is the proof of Theorem 4.6.6.

Proof.

1. Because $q_i \approx_\kappa q_j$, exists q' such as $q' \sqsubset_\kappa q_i$ and $q' \sqsubset_\kappa q_j$. Therefore, $(e_i,q') \sqsubset_\pi (e_i,q_i)$ and $(e_i,q') \sqsubset_\pi (e_j,q_j)$. The existence of $p' = (e_i,q')$ implies $p_i = (e_i,q_i) \approx_\pi p_j = (e_j,q_j)$.

2. Because $q_i \approx_\kappa q_j$, exists q' such as $q' \sqsubset_\kappa q_i$ and $q' \sqsubset_\kappa q_j$. Therefore, $(e_i,q') \sqsubset_\pi (e_i,q_i)$. Also, because given $e_i \sqsubset_\varepsilon e_j$, so $(e_i,q') \sqsubset_\pi (e_j,q_j)$. The existence of $p' = (e_i,q')$ implies $p_i = (e_i,q_i) \approx_\pi p_j = (e_j,q_j)$.

High-level	Low-level	Precision
Production	Develop service	100.0
	Improve service	100.0
	Manage account	100.0
	Manage service	100.0
	Personalize service	83.3
	Process payment	100.0
	Provide service	100.0
	Security	100.0
Marketing	Customer comm.	80.0
	General marketing	100.0
	Marketing analytics	100.0
	Personalize ad	100.0
	Provide ad	100.0
	Promotion	100.0
Legality	General legality	100.0
Other	Other purposes	100.0
	Average	97.8

Table B.3: Policy purpose prediction performance on test set.

3. The proof is similar to (2) with the roles of entities e_i and e_j swapped with purposes q_i and q_j, respectively.

4. The proof is similar to (3) with the roles of entities e_i and e_j swapped with purposes q_i and q_j, respectively.

5. Because $e_i \approx_\kappa e_j$, exists e' such as $e' \sqsubset_\varepsilon e_i$ and $e' \sqsubset_\kappa e_j$. Because $q_i \approx_\kappa q_j$, exists q' such as $q' \sqsubset_\kappa q_i$ and $q' \sqsubset_\kappa q_j$. Therefore, $(e', q') \sqsubset_\pi (e_i, q_i)$ and $(e', q') \sqsubset_\pi (e_j, q_j)$. The existence of $p' = (e', q')$ implies $p_i = (e_i, q_i) \approx_\pi p_j = (e_j, q_j)$.

□

B.5 Data Flow Purpose Features

Features used for inferring usage purposes of data flows are listed in Table B.4. The ablation study results are shown in Table B.5.

Group	Feature	Explanation	Dimension
Sent data	(G1) URL bag-of-words	Bag of words extracted from the request URL.	140
	(G2) Sent data bag-of-words	Bag of words extracted from the sent HTTP(S) data.	140
Data characteristics	(G3) Sent data types	Enumeration of data types in the sent data.	6
	(G4) Number of key-values	Number of key-value pairs in the sent data.	1
	(G5) Number of data types	Number of data types in the sent data.	1
App-specific info	(G6) App-destination similarity	The package name has long common substrings with the URL.	3

Table B.4: Features used in the purpose classification for data flows.

Features	Precision	Recall	F1
G.1	0.69	0.67	0.68
G.1,2	0.73	0.68	0.70
G.1,2,3	0.75	0.73	0.74
G.1,2,3,4	0.77	0.75	0.75
G.1,2,3,4,5	0.79	0.76	0.77
G.1,2,3,4,5,6	0.81	0.78	0.79

Table B.5: Ablation study of the purpose classification features. The performance is on the test set.

B.6 Privacy Policy Crawler and Preprocessor

A crawler was developed to scrap the privacy policies of Android apps. Given an app ID, the crawler first searches for the privacy policy URL in the metadata of the app on Google Play Store. A full HTML version of the web page is scrapped by using Google Chrome controlled by Puppeteer web driver [129] so that dynamically rendered privacy notice contents are downloaded correctly. Finally, PolicyLint's open-source privacy policy HTML pre-processing tool [18] was used to remove extraneous GUI elements and HTML tags and extract a plain-text version that contains well-formed sentences of the privacy policy. If the privacy policy classifier determines that the downloaded document is not a privacy policy, the crawler searches for a privacy link within the page and repeats the HTML downloading and extraction process.

A classifier based on Support Vector Machine (SVM) is developed to determine whether the downloaded web document is a privacy policy or not. The model is trained on a set of 375 documents (199 positive and 176 negative examples). The training and validation used 5-fold cross validation while 15% of the documents were held out for testing. The classifier achieved F1 scores of 98.12% and 96.49% for validation and testing, respectively. Similar

	MobiPurpose purpose class	`PurPliance` purpose class
1	Search nearby places	Production - Provide service
2	Geosocial networking	Production - Provide service
3	Network switch notification	Production - Provide service
4	Geotagging	Production - Provide service
5	Transportation information	Production - Provide service
6	Map and navigation	Production - Provide service
7	Recording	Production - Provide service
8	Location-based game	Production - Provide service
9	Alert and remind	Production - Provide service
10	Third-party login	Production - Provide service
11	Geo localization	Production - Provide service
12	Reverse geocoding	Production - Provide service
13	Location spoofing	Production - Provide service
14	Network optimization	Production - Provide service
15	Interface customization	Production - Personalize service
16	Location-based customization	Production - Personalize service
17	Signed-out user personalization	Production - Personalize service
18	Anti-fraud	Production - Security
19	Authentication	Production - Security
20	User/device tracking for data analytics	Marketing - Marketing analytics
21	Data collection for analytics	Marketing - Marketing analytics
22	Data collection for advertising	Marketing - Provide ad
23	User/device tracking for advertising	Marketing - Provide ad
24	Data collection for advertising personalization	Marketing - Personalize ad

Table B.6: Conversion from purpose classes in MobiPurpose [166] to `PurPliance` taxonomy. This table does not present full `PurPliance` taxonomy but relevant classes with ones in MobiPurpose.

to PolicyLint [20], we filtered out sentences that do not contain any data practice verbs or data objects, and sentences that start with an interrogative word.

B.7 Mapping Purposes of MobiPurpose to PurPliance's Purpose Taxonomy

The conversion from purpose categories in MobiPurpose to the data-usage taxonomy of `PurPliance` is listed in Table B.6.

B.8 Domain-adapted NER Model

`PurPliance` uses a domain-adapted NER model to extract the data objects and entities from sentences. We retrained the NER component of the Spacy *en_web_core_lg* language model [100] on PolicyLint's dataset of 600 manually annotated sentences. 150 sentences were randomly selected while the other 450 have one of the 9 subsumptive relationship patterns. Similar to the procedure used in PolicyLint [20], we trained the model on the training set of 500 samples until the loss converges after 180 epochs. The data object recognition performance on the test set of 100 samples achieves an 83.1% F1 score (82.26% precision and 83.95% recall).

B.9 Distribution of Apps and Policies

Fig. B.1 shows the distribution of apps and unique policies per app category.

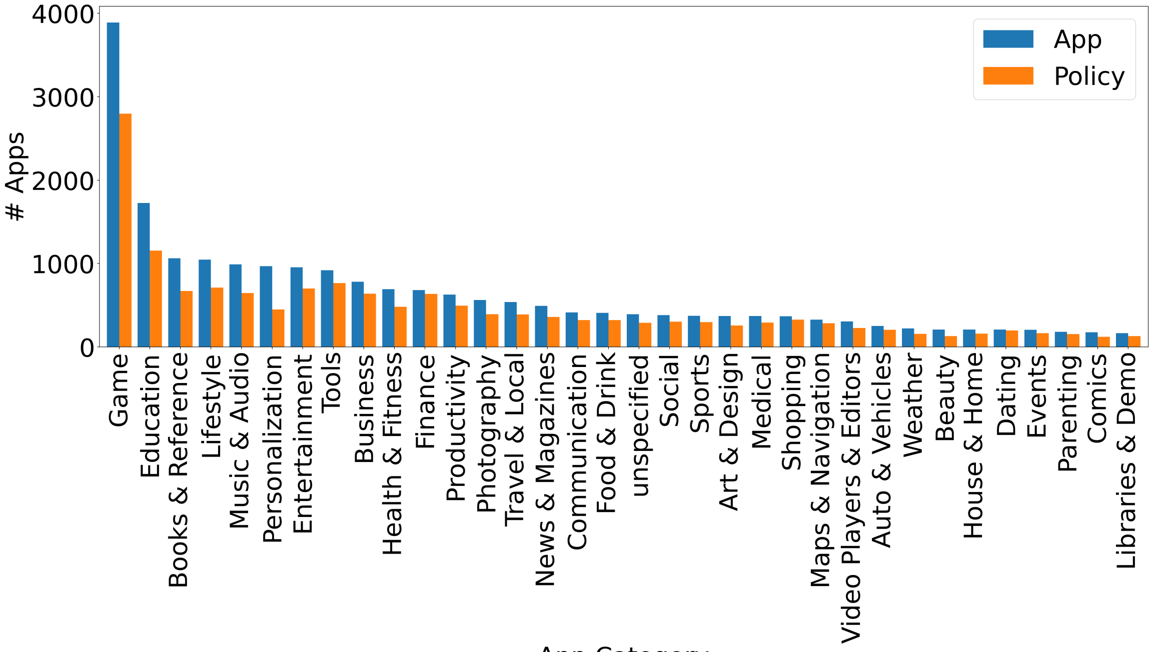


Figure B.1: Distribution of apps and unique policies per app category.

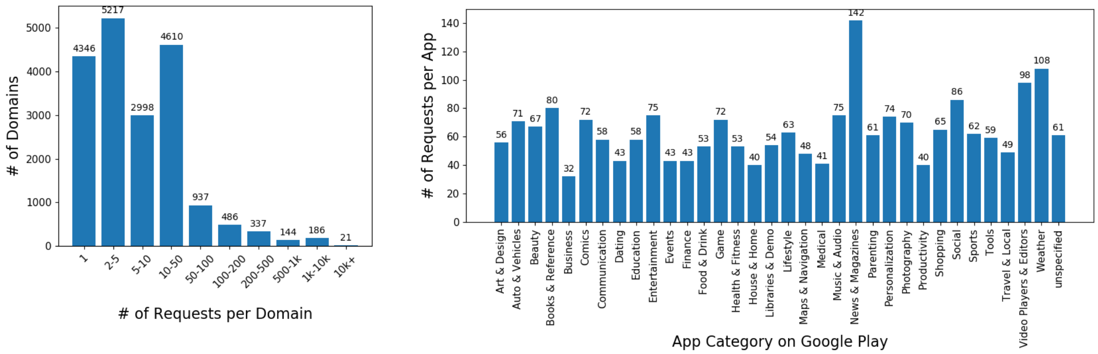


Figure B.2: Data statistics of 1,727,001 network requests intercepted. The left figure shows the distribution of requests among domains. The right figure shows the distribution of requests among app categories on Google Play.

B.10 Distribution of Captured Traffic over App Categories

Statistics of network traffic intercepted are shown in Fig. B.2. The top 3 contacted domains are *googleads.g.doubleclick.net* with 230,309 requests, *pagead2.googlesyndication.com* with 86,767 requests, and *csi.gstatic.com* with 73,939 requests. The traffic distribution has a long tail: 13,269 (68.8%) domains were contacted by only one app and 12,561 (65.1%) domains have less than 10 network data requests.

B.11 Dataset for End-to-end Contradiction Detection

B.11.1 Annotation Procedure

Contradictions in each privacy policy are identified as follows. We first look for any sentences that contain negated sentiment either in data collection/sharing or in purpose clauses as their occurrences are much less frequent than positive ones. For each negated sentence found, we try to find as many contradictory positive statements as possible. The common keywords in negated statements include "not", "never", "only for", "only to", "solely". However, because negated statements are expressed in various ways and their meanings can only be determined by the context, we need to read the whole policies to search for negated statements. To fully interpret the policy sentences, we checked the meaning of each word and the contextual sentences of each identified statement to understand the specific intention and meaning of the terms in the sentence. We also consulted external regulatory texts for the definition of certain data types when necessary.

B.11.2 Dataset

The apps selected for the evaluation of end-to-end contradiction detection and their statistics are shown in Table B.7. The app with the most contradictory sentence pairs is *au.com.realestate.app*. It contains a statement that "We do not collect sensitive information as defined under the Privacy Act 1988("Privacy Act")." However, because *sensitive information* is a type of *personal information* as defined by the Privacy Act [26, 293], the broad negated-sentiment statement have a narrowing-definition contradiction with many other sentences about collection/sharing of *personal information*.

B.11.3 Evaluation of Privacy Statement Extraction

Experimental Procedure. We compare `PurPliance`'s performance in extracting privacy statements from policy document sentences with PolicyLint [20], a state-of-the-art extraction method. To avoid test data leakage [174], 285k (20%) sentences in the corpus were set aside as the test set while `PurPliance` was developed and fine-tuned on the other 80% sentences. 300 sentences were then randomly selected from the test set for evaluation. The privacy statements of `PurPliance` and PolicyLint from their parameter extraction step are used. We used PolicyLint's public implementation without any changes. The NER models used by both systems are trained on the same dataset, and hence they have similar capabilities. In addition, since PolicyLint does not support purpose extraction, purposes extracted by `PurPliance` and their combinations are not counted in this evaluation. Three of the authors annotated the sentences. We used majority votes and held discussions to reach a consensus about the correctness of the extracted statements.

Metrics. Since our goal is to minimize false positives, the precision and the number of extracted statements are used as the main performance metrics. Different from creating a dataset of contradictory sentences in Section 4.8.3, there are a large number of possible text spans that express a data type or a receiving entity in each sentence and limitations of the contiguous entity annotation. Therefore, it requires a significant amount of effort to create a complete dataset of annotations of all policy statements and control its quality [45].

Results and Analysis. Our results show that `PurPliance` extracts more privacy statements with higher precision than PolicyLint. The precision of `PurPliance` is 0.91, higher than 0.82 of PolicyLint. `PurPliance` extracted 160 statements from 68 sentences which are 88% more statements and cover 45% more sentences than PolicyLint. Table 4.11 shows our experimental results.

An in-depth analysis shows the most common incorrect extraction of both systems is caused by the erroneous recognition of data objects and receivers by NER models. Furthermore, since both systems do not analyze the semantics of sentences, they extract data-collection practices from non-data-collection statements such as "data protection laws in Europe distinguish between organizations that process personal data ..." However, both systems employ further filtering in the later steps of their pipelines so trivial incorrectness would not increase false positive rates of the whole system significantly.

`PurPliance` extracts more statements than PolicyLint because it can cover many grammar variations which are not included in PolicyLint's 16 sentence templates of data collection and sharing. For example, PolicyLint missed all policy statements from "we do not sell, trade, or otherwise transfer to outside parties your personal identifiable information," because it did not recognize the long list of multiple data action verbs.

	App	# Sent-Pairs	# Sentences	# Installs
1	au.com.realestate.app	31	264	1,000,000
2	com.rfi.sams.android	18	468	10,000,000
3	com.toongoggles.tv	13	137	100,000
4	in.followon.alumni	9	143	100
5	com.birthday.flowers.images	8	71	1,000
6	com.SuperAwesome.DragonVillageBlast	7	213	100,000
7	com.qarasoft.kosho	7	121	50,000
8	com.crazyplex.hotcoffeemaker	6	148	100,000
9	com.innovle.qtix	6	86	5,000
10	com.colorflash.callerscreen	5	77	1,000,000
11	com.mobibah.afanoromolovesms	5	35	10,000
12	com.theepochtimes.news	4	145	100,000
13	net.playtouch.becomeapuppygroomer	4	122	10,000
14	com.spicyyoghurt.pixiegame.free	4	37	100
15	com.appwallet.magictoucheffect	4	32	10,000
16	com.squareup	3	474	10,000,000
17	com.tappx.flipnsave.battery	3	285	1,000,000
18	com.greatclips.android	3	280	5,000,000
19	com.fishcrackergames.WhatBread	3	52	500
20	com.fontskeyboard.fonts	3	46	5,000,000
21	com.qvq.simpleball	3	45	500
22	com.grab.yourbaby	3	43	5,000
23	com.pdffilereader	3	36	1,000,000
24	com.visionsmarts.pic2shop	3	19	1,000,000
25	com.gi.talkingrapper	2	365	1,000,000
26	com.olo.kneaders	2	224	10,000
27	com.geeko.ivrose	2	163	1,000,000
28	com.ilsc.mygreystone	2	156	100
29	me.nextplus.smsfreetext.phonecalls	2	137	5,000,000
30	com.eivaagames.Bowling3DPro	2	79	1,000,000
31	com.dumpgames.virtual.single.dad.simulator.happy.father	2	75	500,000
32	theme.space.galaxy.planet.shining.aircraft.launcher.wallpaper	2	36	100
33	comethru.event.organizer	1	281	10
34	com.bravolang.chinese	1	272	1,000,000
35	com.sia.id00145	1	173	100
36	com.journedelafemme.bestwomanslove	1	89	1,000
37	com.lily.times.basset2.all	1	80	1,000,000
38	com.appybuilder.bmkbmk767.purerelationship	1	71	100
39	com.lwsipl.archightech.launcher	1	66	500,000
40	com.lexilize.notme	1	58	500
41	com.polaroid.cube.plus	1	56	1,000
42	kynguyen.app.mirror	1	43	1,000,000
43	com.repsi.heartrate	1	35	1,000,000
44	appinventor.ai_mssrnick.almohana	1	27	100
45	air.com.miracle.SeaRescue	1	24	500
46	photo.editor.collage.maker.photoeditor	1	12	1,000,000
47	net.moderndefense	1	10	10,000
	Total	189	5911	

Table B.7: Selected apps with contradictory sentence pairs. *# Sent-Pairs* stands for the number of contradictory sentence pairs.

APPENDIX C

ExtPrivA

C.1 List of Testing URLs

The testing URLs used by ExtPrivA to generate candidate URLs are listed in Table C.1.

C.2 Privacy Policy Crawling

To obtain the privacy policy documents, for each extension, ExtPrivA extracts the privacy policy URL from the extension's overview page. We use a clean instance of Chrome browser that fully executes JavaScript to extract privacy policies of dynamic web pages. ExtPrivA then extracts plain text from the HTML by using PolicyLint preprocessing tool [20]. The plain text is then segmented into sentences by using a transformer-based neural model *en_core_web_trf* included in the Spacy NLP library [9].

C.3 List of Data Types on Chrome Web Store

The list of data types used by the Chrome Web Store is shown in Table C.2 and Fig. C.1.

Category	URL

Search	`https://www.google.com/search?q=statistics&hl=en`
	`https://www.bing.com/search?q=statistics`

Shopping	`https://www.amazon.com/gp/product/B085TFF7M1`
	`https://www.amazon.com/dp/B07G7T3M6C`
	`https://www.ebay.com/itm/323879722346`
	`https://www.aliexpress.com/item/4000901174719.html`

Table C.1: Testing URLs for generating candidate URLs.

	Data Type	Example
1	Personally identifiable info.	Name, address, email address, age, identification number
2	Health information	Heart rate data, medical history, symptoms, diagnoses, procedures
3	Financial and payment info.	Transactions, credit card numbers, credit ratings, financial statements, payment history
4	Authentication information	Passwords, credentials, security question, personal identification number (PIN)
5	Personal communications	Emails, text or chat messages, social media posts, conference calls
6	Location	Region, IP address, GPS coordinates, information about things near the user's device
7	Web history	The list of web pages a user has visited, browsing-related data such as page title and time of visit
8	User activity	Network monitoring, clicks, mouse position, scroll, keystroke logging
9	Website content	Text, images, sounds, videos, hyperlinks

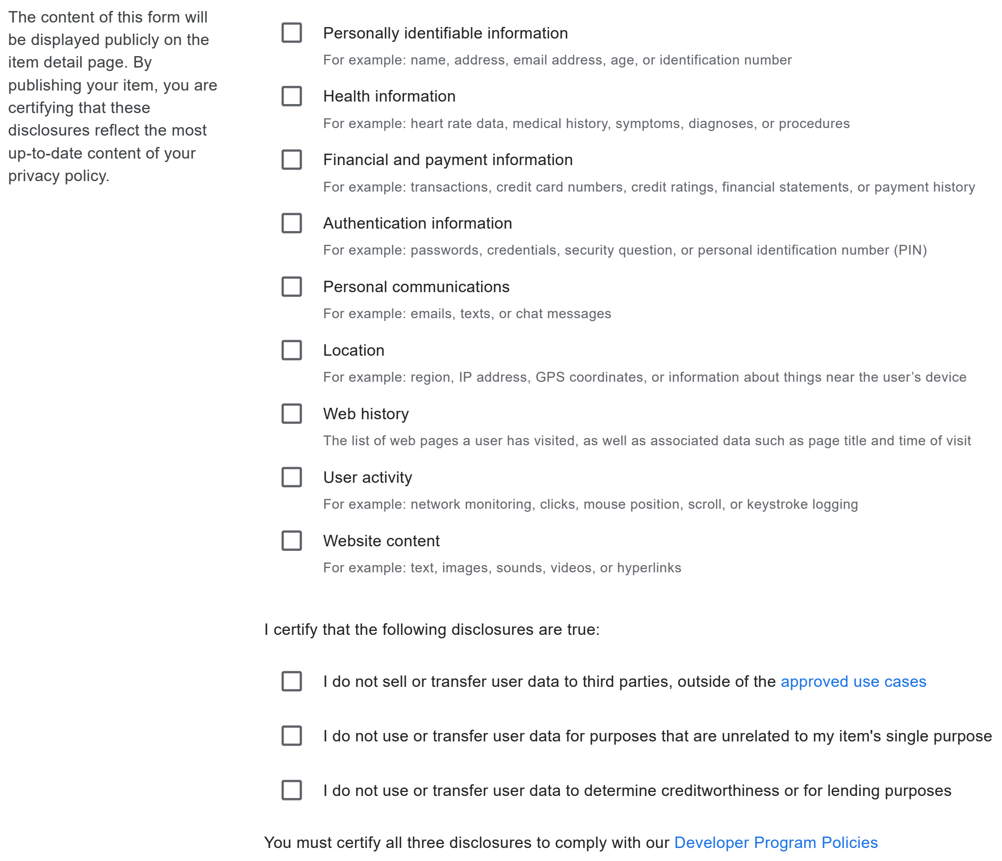
Table C.2: List of data-types and examples specified by the Chrome Web Store policies [156].

C.4 Precision of Data Type Extraction

The precision of data-type extraction is shown in Table C.3.

C.5 Distribution of Inconsistent Extensions

Distribution of inconsistent extensions is shown in Table C.4.



The content of this form will be displayed publicly on the item detail page. By publishing your item, you are certifying that these disclosures reflect the most up-to-date content of your privacy policy.

□ Personally identifiable information
For example: name, address, email address, age, or identification number

□ Health information
For example: heart rate data, medical history, symptoms, diagnoses, or procedures

□ Financial and payment information
For example: transactions, credit card numbers, credit ratings, financial statements, or payment history

□ Authentication information
For example: passwords, credentials, security question, or personal identification number (PIN)

□ Personal communications
For example: emails, texts, or chat messages

□ Location
For example: region, IP address, GPS coordinates, or information about things near the user's device

□ Web history
The list of web pages a user has visited, as well as associated data such as page title and time of visit

□ User activity
For example: network monitoring, clicks, mouse position, scroll, or keystroke logging

□ Website content
For example: text, images, sounds, videos, or hyperlinks

I certify that the following disclosures are true:

□ I do not sell or transfer user data to third parties, outside of the approved use cases

□ I do not use or transfer user data for purposes that are unrelated to my item's single purpose

□ I do not use or transfer user data to determine creditworthiness or for lending purposes

You must certify all three disclosures to comply with our Developer Program Policies

Figure C.1: Privacy-practice declaration on the Chrome Developer Dashboard.

Data	# Flows	# Samples	Precision (%)
Page URL	7,262	30	100.00
Page Hostname	2,256	30	100.00
Product ID	1,302	30	100.00
Website Text	1,054	30	100.00
Hyperlink	786	30	93.33
Region	404	30	93.33
IP Address	396	30	100.00
Page Title	279	30	96.67
Mouse Click	133	30	100.00
GPS Coordinate	68	30	100.00
Keystroke Logging	59	30	100.00
Overall	13,999	330	99.37

Table C.3: Precision of data-type extraction in data flows. The overall precision is a weighted average by the number of flows per data type.

Category	# Inconsistencies	% Inconsistencies	# Extensions
Productivity	557	43.18	351
Shopping	293	22.71	190
Developer Tools	138	10.70	66
Accessibility	84	6.51	52
Search Tools	62	4.81	51
Social & Communication	68	5.27	48
Fun	56	4.34	40
News & Weather	6	0.47	6
No-Category	10	0.78	6
Blogging	9	0.70	5
Photos	6	0.47	4
Sports	1	0.08	1
Total	1290	100.00	820

Table C.4: Distribution of detected inconsistent extensions over category.

APPENDIX D

ConsentChk

D.1 Cookie-Preference Button Dataset Creation

D.1.1 Hyperparameter Tuning Ranges

The hyperparameter tuning ranges of the ML models are shown in Table D.1.

Model	Hyper parameter	Search Range	Optimal Value
Logistic Regression	Regularization *C*	[0.01, 10]	5
MLP	Hidden layer size	[20, 200]	150
Random Forest	# decision trees	[20, 300]	100
SVM	Regularization *C*	[0.1, 100]	1
XGBoost	# decision trees	[20, 200]	50

Table D.1: Hyperparameters tuning range of ML models for classifying preference buttons.

D.1.2 Ablation Study of Preference Button Classifier

Fig. D.1 shows the top-k scores of the classifier using 10-fold validation on the training set with different feature dimensions.

D.2 Consent Cookie Decoding

To extract cookie consent preferences, we decode consent cookies basing on the documentation and analyzing their key-value pairs. OneTrust's consent cookie is called *Optanon-*

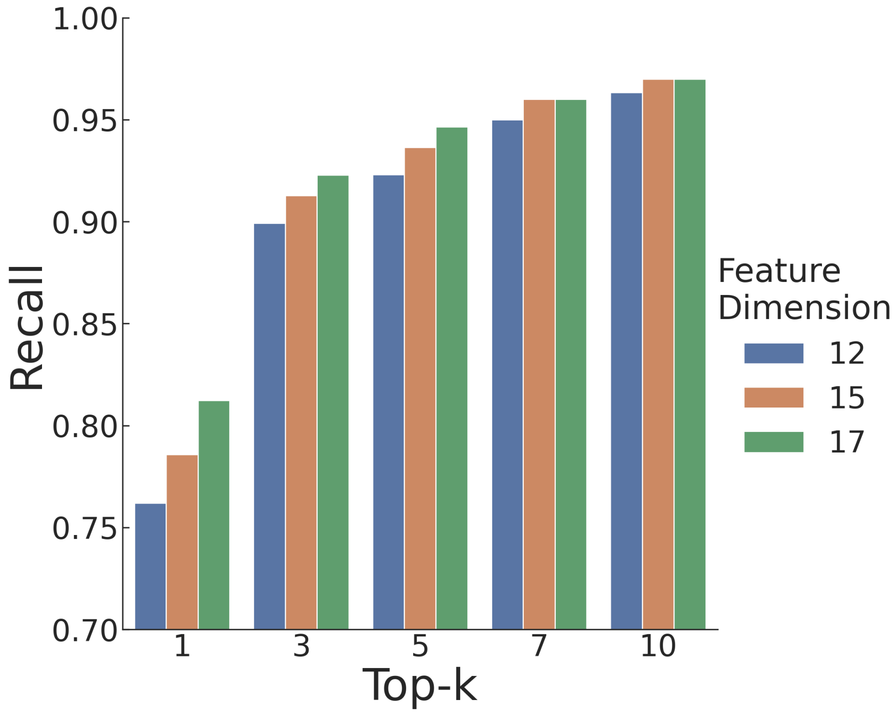


Figure D.1: Ablation study of effectiveness of feature groups on the preference button classifiers.

Consent [237] which stores the consent preference of each cookie category. For example, *groups=C1:1,C2:0* indicates that cookie category *C*1 is approved while *C*2 is rejected. Cookiebot's consent cookie is called *CookieConsent* storing consents for 4 fixed cookie categories: Necessary, Preferences, Statistics, and Marketing [76]. Similar to 'Necessary' cookies, 'Unclassified' cookies are not automatically blocked and cannot be denied by users so we set their consent to True [75]. Finally, Termly stores its categorical consent states as key-values in the *TERMLY_API_CACHE* local storage object.

D.3 Automatic Cookie Consent Approval and Rejection

To automate the experiments, we design `ConsentChk` to find and set cookie consent preferences via the cookie settings as follows.

Cookie Setting Menu Extraction. On the home page, `ConsentChk` attempts to open the cookie setting menu by simulating a user's clicks on the preference button candidates and detects whether any cookie setting menu is open or not. ConsentChk extracts candidates of preference buttons by using the preference button classifier (Section 6.3.1).

Extraction of Cookie Setting Controls. From the initial cookie setting menu interface, if the interface contains only a welcome description, ConsentChk continues to navigate to the main cookie preference setting interface that contains cookie settings. ConsentChk then detects the layout of the menu to iterate through the cookie categories. For layouts that organize cookie categories into tabs (e.g., OneTrust's tab layout), ConsentChk switches to each cookie category tab by clicking on the category's tab header. On each cookie category, ConsentChk identifies the controls (e.g., toggle switches and check boxes) that set the consent choices.

Cookie Consent Approval and Rejection. ConsentChk simulates the user's clicks on the cookie consent UI controls identified in the prior step to set consent preferences (approval or rejection) and clicks on the preference saving/submission button to have the website save the preferences. A consent preference is determined by the state of the associated UI control. For example, a toggle switch's *checked* state indicates an *approved* consent.
ConsentChk determines the final cookie preferences recorded by the website by decoding the values of the cookie management libraries' consent cookies. The system also verifies that the automated approval/rejection is successful by matching the preferences of cookie categories set via the UI with the corresponding values stored in a consent cookie.

D.4 Cookie Setting Categories

The distribution of cookie declarations per category in cookie settings is shown in Fig. D.2.

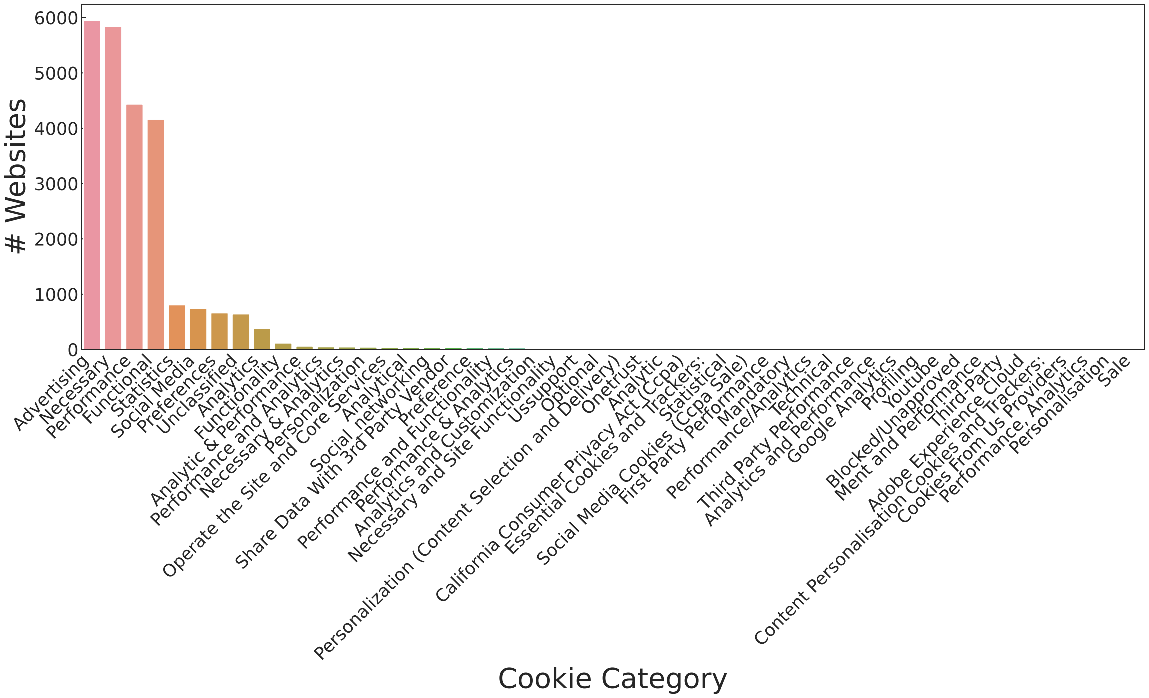


Figure D.2: The top 50 most common cookie categories.

APPENDIX E

OptOutCheck

E.1 Automatically Clicking a Button

To click on an opt-out button, we use two complement methods: the *click()* function provided by the automation tool and JavaScript *click()* API. The automation tool's version scrolls the button to the viewport and checks the visibility of the button before issuing a click event to the element. However, this method fails when the button is hidden by a popup menu such as an email-subscription banner when the page is first loaded. The JavaScript version bypasses the checks and can activate the opt-out button in this case. However, in some cases where the click is intercepted by an outer HTML element, the JavaScript version can still fail. When both clicking methods fail, `OptOutCheck` concludes the button is not clickable.

E.2 Web Crawler Timeouts

To avoid missing cookies due to too short experiment time on web pages with dynamically loaded resources, we design the following timeout values. For each web page, the crawler navigates to a URL and waits until there is no active connection (i.e., no active resource downloads) for 500ms [295]. This timeout is effective for dynamically loaded resources (e.g., advertisements in iframes) and typically much later than the completion of the page load event (i.e., when the entire page, including all dependent resources such as stylesheets and images, has been loaded [228]). If the page has always-active connections, the crawler stops loading after a final timeout of 30 seconds. This timeout is sufficient to avoid missing resources because the crawler runs on a server with a fast network speed while the top websites are commonly designed to reduce page load time for a large number of visitors [161]. In particular, the page load time (i.e., duration from Navigation Start to the Load Event End [224, 301]) of websites in the top 5k domains is 2.47 (2.28 SD) seconds. In the remainder of this paper, unless stated otherwise, we use these timeout heuristics in our experiments.

E.3 Opt-out Policy Corpus

E.3.1 Cookie Domain Selection

The cookie domains in the opt-out policy corpus are selected as follows. The data collection of the top 5k websites crawled 2.2M cookies from 38.8k web pages on 4,437 websites. Of these, 1.6M third-party cookies from 2,364 unique cookie domains were collected. We removed the cookie domains present on less than 100 websites to reduce noise from first-party cookies of a publisher placed on their own domains. We also excluded ad platforms which do not provide any English version of their websites and policies. After this removal, 180 cookie domains remained. We selected the top 100 cookie domains and randomly selected additional 20 domains from the rest of the list to cover both ends of the spectrum of ad platforms in terms of popularity.

E.3.2 Opt-out Button Identification

We searched opt-out settings of an ad platform as follows. We first looked for privacy policies of the advertiser. Starting from the home page, we find keywords describing privacy policies, such as the combination of {privacy, cookie} and {policy, notice, statement}.

In the policies found in the previous step, we searched for keywords "opt-out" and "opt out", and read through the surrounding text to check whether it mentioned the opt-out choices for users from OBA, data collection or tracking of the ad platforms, rather than on its website. It is not straightforward to identify the direct opt-out settings because they commonly provide separate privacy policies for their websites themselves (now they are the first party) and for the websites where they place their cookies as a third party.

If no opt-out option was found in the policies, we searched for the opt-out page from the home page. We looked for opt-out keywords such as "opt-out", "opt out", "consumer choice", "interest based ads", and "ad choice". These steps are necessary because the settings are not always in the privacy policy but could be located in a dedicated ad choice page. The websites sometimes placed a direct link with text "opt out" on their home pages. We also looked for opt-out choices in the "What choices do I have?" and the like.

If the opt-out button could not be found in the privacy policies and home page, we looked for "opt out opt-out site:*tracker_domain*" on a search engine to find the opt-out page in the tracker's website. This process was repeated until an opt-out setting was found or the advertiser was determined not to provide such a setting.

E.3.3 Opt-out Policy Classifier Performance

Table E.1 shows the performance of the opt-out policy classifiers on opt-out policy corpus.

E.4 Proof of Theorem 7.8.4

Proof of Theorem 7.8.4 is as follows.

Opt-out Policy	Precision	Recall	F1	Support
No-tracking	88.24	83.33	85.71	18
No-data-collection	90.48	82.61	86.36	23
Average	89.36	82.97	86.04	20

Table E.1: Performance of the opt-out policy classifiers on the opt-out policy corpus.

Proof. The collection of unique IDs for tracking purposes constitutes a data flow $f = (r, unique\_id, tracking)$ where $r \equiv_\delta$ "first party", according to Definition 7.7.1.

Informally, the data flow is inconsistent with *No-tracking* policy due to their contradictory data-usage purposes. From the definitions of policy classes in Table 7.1, there exists a policy statement $((r, collect, data), (data, not\_for, tracking))$. Since a unique ID is a type of user data, i.e., $unique\_id \sqsubseteq data$, the statement is flow-relevant to f but $k_t = not\_for$. So the policy is inconsistent with the flow by Definition 7.8.3.

Similarly, the data flow is inconsistent with *No-data-collection* policy because it collects a data type that is not allowed by the policy. By definition of the policy classes, there exists a statement $((r, not\_collect, data), None)$ in the *No-data-collection* opt-out class. Because a unique ID is a type of user data, i.e., $unique\_id \sqsubseteq data$, the policy is flow-relevant to f but $c_t = not\_collect$, and thus, is inconsistent with the flow by Definition 7.8.3. □

E.5 Detected Inconsistent Trackers

The extracted inconsistent data flows, opt-out policies, opt-out cookies and opt-out domains of the detected inconsistent trackers are listed in Tables E.2, E.3, E.4, and E.5, respectively.

Tracker	Opt-out Policies	Data Type	Cookie Name	Cookie Domain
adtriba.com	*No-tracking*	Unique ID	atbgdid	.adtriba.com
criteo.com	*No-data-collection*	Unique ID Unique ID	uid uid	.criteo.com .storetail.io
deepintent.com	*No-data-collection*	Unique ID	CDIUSER	.deepintent.com
dianomi.com	*No-tracking*	Unique ID	session[1]	.dianomi.com
dynad.net	*No-data-collection*, *No-tracking*	Unique ID	uid	.dynad.net
liveintent.com	*No-data-collection*	Unique ID	lidid	.liadm.com
onaudience.com	*No-data-collection*	Unique ID	cookie	.onaudience.com
reachlocal.com	*No-tracking*	Unique ID Unique ID	visitor_id visitor_id	04be4b16-c90f-4f1c-89a3-f7f1c516e394.rlets.com 789a4467-dcc5-452e-9434-e15256aed01b.rlets.com
sovrn.com	*No-data-collection*, *No-tracking*	Unique ID	vglnk.Agent.p	.viglink.com[2]
taboola.com	*No-tracking*	Unique ID	t_gid	.taboola.com
underdogmedia.com	*No-data-collection*	Location Unique ID Unique ID Unique ID	geode apnid pmid sncr	.udmserve.net .udmserve.net .udmserve.net .udmserve.net

Table E.2: Detected inconsistencies and data flows. [1] Despite the name, this is a persistent cookie. [2] Sovrn acquired VigLink and owned viglink.com [283].

Tracker	Sentence	Opt-out Policy
adtriba.com	It is at all times possible to object to the data collection through this third party tracking and storage with effect for the future (Opt - Out) .	*No-tracking*
	In order to be excluded from Adtriba third party tracking , you can click the following button .	*No-tracking*
	[Opt-out Button] Opt - Out from Adtriba tracking	*No-tracking*
criteo.com	[Opt-out Button] Disable Criteo services	*No-data-collection*
deepintent.com	This page is intended to help you opt out of the use of cookies , and other data points .	*No-data-collection*
dianomi.com	Opted Out : you have opted out of tracking (behavioural targeting) , Dianomi will no longer serve you with personalized content recommendations based on your Internet history .	*No-tracking*
dynad.net	Opting out of DynAd services through this link from the User 's browser inserts a cookie on the User 's browser and clear all LSOs stored data by DynAd Service .	*No-data-collection*
	This means that we will not track an opted out User 's behavior or display customized ads to the User .	*No-tracking*
liveintent.com	You can opt - out of the cookie - based portion of the LiveIntent Advertising Program by clicking [Opt-out Button] here .	*No-data-collection*
onaudience.com	An " opt - out cookie " will be installed in your browser and block the placement of cookies from OnAudience .	*No-data-collection*
reachlocal.com	If you would like to opt out of tracking provided by the ReachLocal Tracking Code , click the button below .	*No-tracking*
sovrn.com	Opting out of Sovrn //Commerce cookies means that Sovrn //Commerce will stop placing cookies on your device when you browse Sovrn //Commerce enabled websites and/or links .	*No-data-collection*
	To honor your opt - out choice , a cookie is required on your device so that we know not to track your activity .	*No-tracking*
	[Opt-out Button] Disable Tracking	*No-tracking*
taboola.com	* Opted Out : You have opted out of tracking , Taboola will no longer serve you with personalized content recommendations based on your Internet use history .	*No-tracking*
underdogmedia.com	Your current status is to opt - out for Underdog Media hosted 3rd Party Cookies .	*No-data-collection*

Table E.3: Opt-out policies of the detected inconsistent trackers. *[Opt-out-Button]* indicates the occurrence of an opt-out button.

	Opt-out Cookies		
Tracker	Domain	Value	Names
adtriba.com	.adtriba.com	1	atboptout
criteo.com	.courses-en-ligne.carrefour.fr	1	STO_carrefourv2_optout, STO_carrefour_one_optout
	.criteo.com	1	optout
	.fnac.com	1	STO_fnac_optout
	.hlserve.com	1	oo
	.laredoute.fr	1	STO_laredoute_optout
	.storetail.io	1	STO_carrefour_espagne_v2_optout, STO_clarel_optout, STO_alcampo_optout, STO_dia_es_optout, STO_metro_en_optout, STO_metro_fr_optout, STO_phone_house_es_optout, STO_carrefour_espagne_optout, STO_fnac_espagne_optout, STO_ulabox_optout, STO_tudespensa_optout, STO_planetahuerto_optout, STO_pccomponentes_optout, STO_worten_es_optout, STO_worten_pt_optout, STO_primor_optout, STO_costco_optout, STO_auchandrive_v2_optout, STO_cora_optout, STO_auchan_optout, STO_darty_optout, STO_croquetteland_optout, STO_delhaize_fr_optout, STO_delhaize_nl_optout, STO_fnac_belgique_fr_optout, STO_fnacspectacles_optout, STO_fnac_portugal_optout, STO_intermarche_optout, STO_fnac_belgique_nl_optout, STO_leclercculture_optout, STO_sephora_optout, STO_leclerc_optout, STO_leclercpara_optout, STO_sephora_mobile_optout, STO_leclercportal_optout, STO_micromania_optout, STO_rewe_optout, STO_leroymerlin_optout, STO_kingjouet_optout, STO_auchan_v2_optout
deepintent.com	.deepintent.com	true	optout
dianomi.com	www.dianomi.com	1	dnt
dynad.net	.dynad.net	1	optout
liveintent.com	d.liadm.com	opt-out	tuuid
onaudience.com	www.onaudience.com	1	opt-out
reachlocal.com	.rlets.com	1	RlocalOptOut
sovrn.com	.sovrn.co	1	vglnk.OptOut.p
	.viglink.com	1	vglnk.OptOut.p
taboola.com	.taboola.com	1	DNT
underdogmedia.com	.udmserve.net	Thank_You	optout

Table E.4: Opt-out cookies of the detected inconsistent trackers. These cookies are grouped by their domains and values. All cookies have a "/" path.

Tracker	Opt-out Domains	Opt-out Page URL
adtriba.com	.adtriba.com	https://privacy.adtriba.com/
criteo.com	.carrefour.fr, .criteo.com, .fnac.com, .hlserve.com, .laredoute.fr, .storetail.io	https://www.criteo.com/privacy/disable-criteo-services-on-internet-browsers/
deepintent.com	.deepintent.com	https://option.deepintent.com/optout
dianomi.com	.dianomi.com	https://www.dianomi.com/legal/privacy.epl
dynad.net	.dynad.net	https://www.dynad.net/en/privacy-and-terms.html
liveintent.com	.liadm.com	https://www.liveintent.com/ad-choices/
onaudience.com	.onaudience.com	https://www.onaudience.com/opt-out
reachlocal.com	.rlets.com	https://www.reachlocal.com/us/en/legal/trackingopt-out
sovrn.com	.sovrn.co, .viglink.com	https://www.sovrn.com/legal/privacy-center/
taboola.com	.taboola.com	https://www.taboola.com/policies/privacy-policy
underdogmedia.com	.udmserve.net	https://underdogmedia.com/optout/

Table E.5: Opt-out domains and web page URLs of the detected inconsistent trackers.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] 42matters AG. *Google Play Categories | 42matters*. 2020.

[2] Gunes Acar, Steven Englehardt, and Arvind Narayanan. “No boundaries: data exfiltration by third parties embedded on web pages”. In: *Proceedings on Privacy Enhancing Technologies* 2020.4 (2020).

[3] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. “The Web Never Forgets: Persistent Tracking Mechanisms in the Wild”. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014.

[4] AdGuard. *AdGuard Ad Filters | AdGuard Knowledgebase*. 2021. URL: `https://kb.adguard.com/en/general/adguard-ad-filters` (visited on 03/12/2021).

[5] Anupama Aggarwal, Bimal Viswanath, Liang Zhang, Saravana Kumar, Ayush Shah, and Ponnurangam Kumaraguru. “I Spy with My Little Eye: Analysis and Detection of Spying Browser Extensions”. In: *2018 IEEE European Symposium on Security and Privacy (EuroS P)*. 2018.

[6] Charu C. Aggarwal and ChengXiang Zhai. “A Survey of Text Clustering Algorithms”. In: *Mining Text Data*. Ed. by Charu C. Aggarwal and ChengXiang Zhai. 2012.

[7] Eugene Agichtein and Luis Gravano. “Snowball: extracting relations from large plain-text collections”. In: *Proceedings of the fifth ACM conference on Digital libraries*. 2000.

[8] Wasi Ahmad, Jianfeng Chi, Tu Le, Thomas Norton, Yuan Tian, and Kai-Wei Chang. “Intent Classification and Slot Filling for Privacy Policies”. In: *ACL*. 2021.

[9] Explosion AI. *spaCy · Industrial-strength Natural Language Processing in Python*. 2020. URL: `https://spacy.io/` (visited on 01/08/2021).

[10] Alexa Internet, Inc. *Alexa - Top Sites in United States - Alexa*. URL: `https://www.alexa.com/topsites/countries/US`.

[11] AllAboutCookies.org. *What is an Opt Out Cookie? - All about Cookies*. 2021. URL: `https://www.allaboutcookies.org/manage-cookies/opt-out-cookies.html` (visited on 07/23/2021).

[12] AllenAI. *AllenNLP - Semantic Role Labeling*. 2020.

[13] Digital Advertising Alliance. *Consumer Assistance | WebChoices and AppChoices*. 2021. URL: https://youradchoices.com/choices-faq (visited on 05/03/2021).

[14] Digital Advertising Alliance. *WebChoices: 'Protect My Choices' Plug-Ins*. 2021. URL: https://youradchoices.com/pmc (visited on 05/03/2021).

[15] Amazon Mechanical Turk, Inc. *https://www.mturk.com/*. 2020.

[16] Ryan Amos, Gunes Acar, Elena Lucherini, Mihir Kshirsagar, Arvind Narayanan, and Jonathan Mayer. "Privacy Policies over Time: Curation and Analysis of a Million-Document Dataset". In: *arXiv:2008.09159 [cs]* (2020).

[17] Ryan Amos, Gunes Acar, Elena Lucherini, Mihir Kshirsagar, Arvind Narayanan, and Jonathan Mayer. "Privacy Policies over Time: Curation and Analysis of a Million-Document Dataset". In: *Proceedings of the Web Conference 2021*. 2021.

[18] Ben Andow. *HtmlToPlaintext*. 2020.

[19] Ben Andow. *PrivacyPolicyAnalysis*. 2020.

[20] Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. "PolicyLint: Investigating Internal Privacy Policy Contradictions on Google Play". In: *28th USENIX Security Symposium (USENIX Security 19)*. 2019.

[21] Benjamin Andow, Samin Yaseer Mahmud, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Serge Egelman. "Actions Speak Louder than Words: Entity-Sensitive Privacy Policy and Data Flow Analysis with POLICHECK". In: *29th USENIX Security Symposium (USENIX Security 20)*. 2020.

[22] Benjamin Andow, Samin Yaseer Mahmud, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Serge Egelman. "Actions Speak Louder than Words: Entity-Sensitive Privacy Policy and Data Flow Analysis with PoliCheck". In: *29th USENIX Security Symposium (USENIX Security 20)*. 2020.

[23] AppCensus, Inc. *AppCensus AppSearch*. 2020.

[24] Ashlea Cartee. *Say Hello to Cookie Auto-Blocking*. CookiePro. 2019. URL: https://www.cookiepro.com/blog/cookie-auto-blocking/ (visited on 05/03/2021).

[25] Aurore Fass, Dolière Francis Somé, Michael Backes, and Ben Stock. "DoubleX: Statically Detecting Vulnerable Data Flows in Browser Extensions at Scale". In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2021.

[26] Office of the Australian Information Commissioner. *What is personal information?* OAIC. 2021. URL: https://www.oaic.gov.au/privacy/guidance-and-advice/what-is-personal-information/ (visited on 04/21/2021).

[27] Rebecca Balebako, Pedro G. Leon, Richard Shay, Blase Ur, Yang Wang, and Lorrie Faith Cranor. "Measuring the effectiveness of privacy tools for limiting behavioral advertising". In: *In Web 2.0 Workshop on Security and Privacy*. 2012.

[28] Vinayshekhar Bannihatti Kumar, Roger Iyengar, Namita Nisal, Yuanyuan Feng, Hana Habib, Peter Story, Sushain Cherivirala, Margaret Hagan, Lorrie Cranor, Shomir Wilson, Florian Schaub, and Norman Sadeh. "Finding a Choice in a Haystack: Automatic Extraction of Opt-Out Statements from Privacy Policy Text". In: *Proceedings of The Web Conference 2020*. 2020.

[29] Adam Barth. *HTTP State Management Mechanism*. 2011. URL: `https://tools.ietf.org/html/rfc6265` (visited on 12/31/2020).

[30] Hannah Bast and Elmar Haussmann. "Open Information Extraction via Contextual Sentence Decomposition". In: *2013 IEEE Seventh International Conference on Semantic Computing*. 2013.

[31] Iz Beltagy, Kyle Lo, and Arman Cohan. "SciBERT: A Pretrained Language Model for Scientific Text". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019.

[32] Berstend. *puppeteer-extra-plugin-stealth*. npm. 2021. URL: `https://www.npmjs.com/package/puppeteer-extra-plugin-stealth` (visited on 06/30/2021).

[33] J. Bhatia and T. D. Breaux. "Towards an information type lexicon for privacy policies". In: *2015 IEEE Eighth International Workshop on Requirements Engineering and Law (RELAW)*. 2015.

[34] J. Bhatia, M. C. Evans, S. Wadkar, and T. D. Breaux. "Automated Extraction of Regulated Information Types Using Hyponymy Relations". In: *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*. 2016.

[35] Jaspreet Bhatia and Travis D. Breaux. "A Data Purpose Case Study of Privacy Policies". In: *2017 IEEE 25th International Requirements Engineering Conference (RE)* (2017).

[36] Jaspreet Bhatia and Travis D. Breaux. "Semantic Incompleteness in Privacy Policy Goals". In: *26th IEEE International Requirements Engineering Conference (RE'18)*. 2018.

[37] Sarah Bird, Ilana Segall, and Martin Lopatka. "Replication: Why We Still Can't Browse in Peace: On the Uniqueness and Reidentifiability of Web Browsing Histories". In: 2020.

[38] Jay Blanchard and Vincent Mikkelson. "Underlining Performance Outcomes in Expository Text". In: *The Journal of Educational Research* 80.4 (1987).

[39] European Data Protection Board. *The Belgian DPA has imposed a fine of €15000 on a website specialized in legal news | European Data Protection Board*. 2019. URL: `https://edpb.europa.eu/news/national-news/2019/belgian-dpa-has-imposed-fine-eu15000-website-specialized-legal-news_en` (visited on 06/02/2022).

[40] Dino Bollinger. "Analyzing Cookies Compliance with the GDPR". MA thesis. ETH Zurich, 2021.

[41] Dino Bollinger, Karel Kubicek, Carlos Cotrini, and David Basin. “Automating Cookie Consent and GDPR Violation Detection”. In: *31st USENIX Security Symposium (USENIX Security 22)*. 2022.

[42] Claire Bonial, Julia Bonn, Kathryn Conger, Jena Hwang, Martha Palmer, and Nicholas Reese. “English PropBank Annotation Guidelines”. In: (2015).

[43] Jasmine Bowers, Bradley Reaves, Imani Sherman, Patrick Traynor, and Kevin Butler. “Regulators, mount up! analysis of privacy policies for mobile money services”. In: *Proceedings of the Thirteenth USENIX Conference on Usable Privacy and Security*. 2017.

[44] Marc Brysbaert. “How many words do we read per minute? A review and meta-analysis of reading rate”. In: *Journal of Memory and Language* 109 (2019).

[45] Duc Bui, Kang G. Shin, Jong-Min Choi, and Junbum Shin. “Automated Extraction and Presentation of Data Practices in Privacy Policies”. In: *Proceedings on Privacy Enhancing Technologies* 2021.2 (2021).

[46] Duc Bui, Yuan Yao, Kang G. Shin, Jong-Min Choi, and Junbum Shin. “Consistency Analysis of Data-Usage Purposes in Mobile Apps”. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2021.

[47] BuiltWith. *Privacy Compliance Usage Distribution in the Top 1 Million Sites*. 2022. URL: `https://web.archive.org/web/20220528205303/https://trends.builtwith.com/widgets/privacy-compliance` (visited on 05/28/2022).

[48] Digital Advertising Alliance Of Canada. *Online Interest-Based Advertising FAQ*. AdChoices in Canada. 2020. URL: `https://youradchoices.ca/en/faq` (visited on 12/09/2020).

[49] capitaloneshopping.com. *Capital One Shopping: Add to Chrome for Free*. 2022. URL: `https://chrome.google.com/webstore/detail/capital-one-shopping-add/nenlahapcbofgnanklpelkaejcehkggg` (visited on 03/30/2022).

[50] F. H. Cate. “The Limits of Notice and Choice”. In: *IEEE Security Privacy* 8.2 (2010).

[51] Wentao Chang and Songqing Chen. “ExtensionGuard: Towards runtime browser extension information leakage detection”. In: *2016 IEEE Conference on Communications and Network Security (CNS)*. 2016.

[52] Quan Chen, Panagiotis Ilia, Michalis Polychronakis, and Alexandros Kapravelos. “Cookie Swap Party: Abusing First-Party Cookies for Web Tracking”. In: *Proceedings of the Web Conference 2021*. 2021.

[53] Quan Chen and Alexandros Kapravelos. “Mystique: Uncovering Information Leakage from Browser Extensions”. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 2018.

[54] Chris Coyier. *Make Entire Div Clickable*. CSS-Tricks. 2021. URL: `https://css-tricks.com/snippets/jquery/make-entire-div-clickable/` (visited on 03/01/2021).

[55] Chrome. *Content scripts*. Chrome Developers. 2019. URL: https://developer.chrome.com/docs/extensions/mv2/content_scripts/ (visited on 09/20/2021).

[56] Cisco Systems, Inc. *Consumer Privacy Survey: The growing imperative of getting data privacy right*. 2019.

[57] Jacob Cohen. *Statistical power analysis for the behavioral sciences*. 2nd ed. 1988. 567 pp.

[58] Federal Trade Commission. *Fair Information Practice Principles*. 2009. URL: https://web.archive.org/web/20090331134113/http://www.ftc.gov/reports/privacy3/fairinfo.shtm (visited on 08/22/2021).

[59] Federal Trade Commission. *Goldenshores Technologies, LLC, and Erik M. Geidl, In the Matter of*. Federal Trade Commission. 2013. URL: http://www.ftc.gov/legal-library/browse/cases-proceedings/132-3087-goldenshores-technologies-llc-erik-m-geidl-matter (visited on 05/22/2022).

[60] Federal Trade Commission et al. "Protecting consumer privacy in an era of rapid change: Recommendations for businesses and policymakers". In: *Washington, DC: Federal Trade Commission* (2012).

[61] Federal Trade Commission. *Snapchat, Inc., In the Matter of*. Federal Trade Commission. 2014. URL: http://www.ftc.gov/legal-library/browse/cases-proceedings/132-3078-snapchat-inc-matter (visited on 05/22/2022).

[62] United States Federal Trade Commission. *Privacy online: a report to Congress*. 1998.

[63] Complianz. *Cookie: GPS*. Cookiedatabase.org. 2021. URL: https://cookiedatabase.org/cookie/youtube/gps/ (visited on 04/30/2021).

[64] Mike Conley. *These Weeks in Firefox: Issue 70*. Firefox Nightly News. 2020. URL: https://blog.nightly.mozilla.org/2020/03/03/these-weeks-in-firefox-issue-70 (visited on 03/26/2022).

[65] *ConsentChk: URL will be released upon the completion of the double-blind review process*. 2021.

[66] Ninja Cookie. *Ninja Cookie*. Ninja Cookie. 2022. URL: https://ninja-cookie.com/ (visited on 05/31/2022).

[67] Cookiebot. *Functions | The Cookiebot CMP solution*. 2021.

[68] COOKIEPRO. *CookiePro Knowledge: Client-Side Cookie Management*. 2021. URL: https://bit.ly/3nqyVq7 (visited on 04/27/2021).

[69] CookiePro. *OneTrust Cookie Auto-Blocking™*. 2021. URL: https://community.cookiepro.com/s/article/UUID-c5122557-2070-65cb-2612-f2752c0cc4aa (visited on 08/14/2021).

[70] CookiePro. *What is an Opt-Out Cookie?* CookiePro. 2021. URL: https://www.cookiepro.com/knowledge/what-is-an-opt-out-cookie/ (visited on 07/23/2021).

[71] Elisa Costante, Jerry den Hartog, and Milan Petković. "What Websites Know About You". In: *Data Privacy Management and Autonomous Spontaneous Security*. Ed. by Roberto Di Pietro, Javier Herranz, Ernesto Damiani, and Radu State. 2013.

[72] Lorrie Faith Cranor, Candice Hoke, Pedro Giovanni Leon, and Alyssa Au. "Are They Worth Reading? An In-Depth Analysis of Online Trackers' Privacy Policies". In: *I/S: a journal of law and policy for the information society* (2015).

[73] Crownpeak Technology, Inc. *Global Consent Preferences*. Global Consent Preferences. 2020. URL: https://www.evidon.com/resources/global-opt-out/ (visited on 12/02/2020).

[74] Cybot. *Automatic Cookie Blocking - How does it work?* Cookiebot Support. 2019. URL: https://support.cookiebot.com/hc/en-us/articles/360009063100-Automatic-Cookie-Blocking-How-does-it-work- (visited on 08/14/2021).

[75] Cybot. *Cookiebot: Check your 'unclassified cookies'*. Cookiebot Support. 2021. URL: https://bit.ly/383S0aV (visited on 08/17/2021).

[76] Cybot. *Developer - setting up Cookiebot CMP*. 2021. URL: https://www.cookiebot.com/en/developer/ (visited on 08/17/2021).

[77] Cybot. *Duration of CookieConsent 2020*. Cookiebot Support. 2020. URL: https://bit.ly/3yWzpcE (visited on 08/16/2021).

[78] DataGuidance. *France: CNIL fines Société du Figaro €50,000 for placing advertising cookies without user consent*. DataGuidance. 2021. URL: https://www.dataguidance.com/news/france-cnil-fines-soci%C3%A9t%C3%A9-du-figaro-%E2%82%AC50000-placing (visited on 06/02/2022).

[79] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. "We Value Your Privacy ... Now Take Some Cookies: Measuring the GDPR's Impact on Web Privacy". In: *Proceedings 2019 Network and Distributed System Security Symposium*. 2019.

[80] Detectify Labs. *Chrome Extensions - AKA Total Absence of Privacy*. Detectify Labs. 2015. URL: https://labs.detectify.com/2015/11/19/chrome-extensions-aka-total-absence-of-privacy/ (visited on 05/16/2021).

[81] Chrome Developers. *Chrome DevTools Protocol - Network.Initiator*. 2022. URL: https://chromedevtools.github.io/devtools-protocol/tot/Network/#type-Initiator (visited on 03/24/2022).

[82] Chrome Developers. *chrome.cookies API*. chrome.cookies API. 2021. URL: https://developer.chrome.com/docs/extensions/reference/cookies/ (visited on 01/09/2021).

[83] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019.

[84] Disconnect. *Disconnect Privacy Icons*. 2017.

[85] Disconnect. *disconnectme/disconnect-tracking-protection*. 2021. URL: `https://github.com/disconnectme/disconnect-tracking-protection` (visited on 06/27/2021).

[86] Disconnect. *Tracking Protection Lists*. 2021. URL: `https://disconnect.me/trackerprotection` (visited on 06/26/2021).

[87] Duc Bui. *PI-Extract Dataset `https://github.com/um-rtcl/piextract_dataset`*. 2020.

[88] Duc Bui. *PurPliance*. 2021. URL: `https://github.com/ducalpha/PurPlianceOpenSource` (visited on 09/10/2021).

[89] DuckDuckGo. *DuckDuckGo Tracker Radar*. 2022.

[90] EasyList. *EasyList - Overview*. 2021.

[91] Richard Eckart de Castilho, Eva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. "A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures". In: *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*. 2016.

[92] Elmar Haussmann. "Contextual Sentence Decomposition". Master's thesis. University of Freiburg, 2011.

[93] Steven Englehardt and Arvind Narayanan. "Online Tracking: A 1-million-site Measurement and Analysis". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 2016.

[94] Steven Englehardt, Dillon Reisman, Christian Eubank, Peter Zimmerman, Jonathan Mayer, Arvind Narayanan, and Edward W. Felten. "Cookies That Give You Away: The Surveillance Implications of Web Tracking". In: *Proceedings of the 24th International Conference on World Wide Web*. 2015.

[95] IAB Europe. *TCF v2.0 - IAB Europe*. https://iabeurope.eu/. 2021. URL: `https://iabeurope.eu/tcf-2-0/` (visited on 05/03/2021).

[96] IAB Europe. *Transparency and Consent Framework*. GitHub. 2020. URL: `https://github.com/InteractiveAdvertisingBureau/GDPR-Transparency-and-Consent-Framework` (visited on 03/14/2021).

[97] European Parliament and Council of the European Union. *General Data Protection Regulation*. 2016.

[98] M. C. Evans, J. Bhatia, S. Wadkar, and T. D. Breaux. "An Evaluation of Constituency-Based Hyponymy Extraction from Privacy Policies". In: *2017 IEEE 25th International Requirements Engineering Conference (RE)*. 2017.

[99] Evidon. *Opting-Out of Data Collection*. Evidon. 2021. URL: `https://www.evidon.com/opting-out/` (visited on 07/23/2021).

[100] Explosion AI. *Models & Languages · spaCy Usage Documentation*. 2020.

[101] Benjamin Fabian, Tatiana Ermakova, and Tino Lentz. "Large-scale readability analysis of privacy policies". In: *Proceedings of the International Conference on Web Intelligence*. 2017.

[102] Hossein Falaki, Dimitrios Lymberopoulos, Ratul Mahajan, Srikanth Kandula, and Deborah Estrin. "A first look at traffic on smartphones". In: *Proceedings of the 10th annual conference on Internet measurement - IMC '10*. 2010.

[103] *Findings of Inconsistent Trackers*. 2022. URL: `https://bit.ly/ccsdemo125`.

[104] Jenny Rose Finkel and Christopher D. Manning. "Nested Named Entity Recognition". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*. 2009.

[105] Karën Fort. *Collaborative Annotation for Reliable Natural Language Processing: Technical and Sociological Aspects*. 2016.

[106] Fortinet. *Web Filter Categories*. FortiGuard. 2022. URL: `https://fortiguard.com/webfilter/categories` (visited on 06/07/2022).

[107] Imane Fouad, Cristiana Santos, Arnaud Legout, and Nataliia Bielova. "My Cookie is a phoenix: detection, measurement, and lawfulness of cookie respawning with browser fingerprinting". In: *PETS 2022 - 22nd Privacy Enhancing Technologies Symposium*. 2022.

[108] Mozilla Foundation. *Public Suffix List*. 2020. URL: `https://publicsuffix.org/` (visited on 03/15/2021).

[109] Robert L. Fowler and Anne S. Barker. "Effectiveness of highlighting for retention of text material". In: *Journal of Applied Psychology* 59.3 (1974).

[110] Gertjan Franken, Tom Van Goethem, and Wouter Joosen. "Who Left Open the Cookie Jar? A Comprehensive Evaluation of Third-Party Cookie Policies". In: 2018.

[111] Jonathan Freeman. *What is JSON? A better format for data exchange*. InfoWorld. 2019. URL: `https://www.infoworld.com/article/3222851/what-is-json-a-better-format-for-data-exchange.html` (visited on 03/15/2022).

[112] FTC. *FTC Puts an End to Tactics of Online Advertising Company That Deceived Consumers Who Wanted to "Opt Out" from Targeted Ads*. 2011.

[113] FTC. *Google Will Pay $22.5 Million to Settle FTC Charges it Misrepresented Privacy Assurances to Users of Apple's Safari Internet Browser*. 2012.

[114] FTC. *Online Advertiser Settles FTC Charges ScanScout Deceptively Used Flash Cookies to Track Consumers Online*. 2011.

[115] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. "AllenNLP: A Deep Semantic Natural Language Processing Platform". In: *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*. 2018.

[116] Ghostery. *ghostery/autoconsent*. 2022.

[117] Vicki Silvers Gier, David S. Kreiner, and Amelia Natz-Gonzalez. "Harmful Effects of Preexisting Inappropriate Highlighting on Reading Comprehension and Metacognitive Accuracy". In: *The Journal of General Psychology* 136.3 (2009).

[118] Adtriba GmbH. *Adtribas Cookies*. 2021. URL: `http://help.adtriba.com/en/articles/3772978-adtribas-cookies` (visited on 07/20/2021).

[119] Cliqz GmbH. *Tracker Categories*. 2017. URL: `https://whotracks.me/blog/tracker_categories.html` (visited on 06/27/2021).

[120] Cliqz GmbH. *WhoTracks.me - Bringing Transparency to Online Tracking*. 2021. URL: `https://whotracks.me` (visited on 03/15/2021).

[121] R. Gonzalez, L. Jiang, M. Ahmed, M. Marciel, R. Cuevas, H. Metwalley, and S. Niccolini. "The cookie recipe: Untangling the use of cookies in the wild". In: *2017 Network Traffic Measurement and Analysis Conference (TMA)*. 2017.

[122] Google. *Catapult performance tools*. 2021. URL: `https://chromium.googlesource.com/catapult/` (visited on 02/26/2021).

[123] Google. *Changes to Cross-Origin Requests in Chrome Extension Content Scripts*. 2020. URL: `https://www.chromium.org/Home/chromium-security/extension-content-script-fetches` (visited on 09/20/2021).

[124] Google. *Chrome DevTools Protocol*. 2021. URL: `https://chromedevtools.github.io/devtools-protocol/` (visited on 09/18/2021).

[125] Google. *Extension Match Patterns*. Chrome Developers. 2021. URL: `https://developer.chrome.com/docs/extensions/mv3/match_patterns/` (visited on 09/16/2021).

[126] Google. *google/chrome-opt-out-extension*. 2015.

[127] Google. *Manifest - Key*. Chrome Developers. 2018. URL: `https://developer.chrome.com/docs/extensions/mv3/manifest/key/` (visited on 10/11/2021).

[128] Google. *Overview of Manifest V3*. Chrome Developers. 2020. URL: `https://developer.chrome.com/docs/extensions/mv3/intro/mv3-overview/` (visited on 09/20/2021).

[129] Google Chrome DevTools Team. *Puppeteer Tools for Web Developers*. 2020.

[130] Google Chromium. *Catapult performance tools*. 2021. URL: `https://chromium.googlesource.com/catapult/` (visited on 02/26/2021).

[131] Google Inc. *Compact Language Detector v3 (CLD3)*. 2021.

[132] Google Inc. *Custom Search JSON API Reference*. 2021. URL: `https://developers.google.com/custom-search/v1` (visited on 07/08/2021).

[133] Google Inc. *Method: cse.list | Custom Search JSON API*. Google Developers. 2021. URL: `https://developers.google.com/custom-search/v1/reference/rest/v1/cse/list` (visited on 07/09/2021).

[134] Google Inc. *Programmable Search Engine*. Programmable Search Engine by Google. 2021. URL: `https://programmablesearchengine.google.com/` (visited on 07/08/2021).

[135] Alessandra Gorla, Ilaria Tavecchia, Florian Gross, and Andreas Zeller. “Checking app behavior against app descriptions”. In: *Proceedings of the 36th International Conference on Software Engineering*. 2014.

[136] GuoShi. *HttpCanary — HTTP Sniffer/Capture/Analysis*. 2020.

[137] Hana Habib, Sarah Pearman, Jiamin Wang, Yixin Zou, Alessandro Acquisti, Lorrie Faith Cranor, Norman Sadeh, and Florian Schaub. “"It’s a scavenger hunt": Usability of Websites’ Opt-Out and Data Deletion Choices”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020.

[138] Hana Habib, Yixin Zou, Aditi Jannu, Neha Sridhar, Chelse Swoopes, Alessandro Acquisti, Lorrie Faith Cranor, Norman Sadeh, and Florian Schaub. “An empirical analysis of data deletion and opt-out choices on 150 websites”. In: *Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security*. 2019.

[139] Catherine Han, Irwin Reyes, Amit Elazari Bar On, Joel Reardon, Alvaro Feal, Serge Egelman, and Narseo Vallina-Rodriguez. “Do You Get What You Pay For? Comparing the Privacy Behaviors of Free vs. Paid Apps”. In: 2019.

[140] Abram Handler, Matthew Denny, Hanna Wallach, and Brendan O’Connor. “Bag of What? Simple Noun Phrase Extraction for Text Analysis”. In: *Proceedings of the First Workshop on NLP and Computational Social Science*. 2016.

[141] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G. Shin, and Karl Aberer. “Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning”. In: *27th USENIX Security Symposium (USENIX Security 18)*. 2018.

[142] Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. “Deep Semantic Role Labeling: What Works and What’s Next”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017.

[143] Yuyu He, Lei Zhang, Zhemin Yang, Yinzhi Cao, Keke Lian, Shuai Li, Wei Yang, Zhibo Zhang, Min Yang, Yuan Zhang, and Haixin Duan. “TextExerciser: Feedback-driven Text Input Exercising for Android Applications”. In: 2020.

[144] Marti A. Hearst. *Search User Interfaces*. 2009.

[145] Maximilian Hils, Daniel W. Woods, and Rainer Böhme. “Measuring the Emergence of Consent Management on the Web”. In: *Proceedings of the ACM Internet Measurement Conference*. 2020.

[146] Jeff Huang, Oren Etzioni, Luke Zettlemoyer, Kevin Clark, and Christian Lee. "RevMiner: an extractive interface for navigating reviews on a smartphone". In: *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 2012.

[147] Jeff Huang, Patrick O'Neil Meredith, and Grigore Rosu. "Maximal sound predictive race detection with control flow abstraction". In: *ACM SIGPLAN Notices* 49.6 (2014).

[148] Ilya Sergey. *What Does It Mean for a Program Analysis to Be Sound?* SIGPLAN Blog. 2019. URL: `https://blog.sigplan.org/2019/08/07/what-does-it-mean-for-a-program-analysis-to-be-sound/` (visited on 10/03/2020).

[149] AdGear Technologies Inc. *Adgear Opt-out*. AdGear. 2021. URL: `https://adgear.com/en/corporate-privacy/optout/` (visited on 05/03/2021).

[150] Apple Inc. *App Store Review Guidelines - Apple Developer*. 2021. URL: `https://developer.apple.com/app-store/review/guidelines/#privacy` (visited on 05/22/2021).

[151] Google Inc. *Chrome Web Store sitemap*. 2021. URL: `https://chrome.google.com/webstore/sitemap` (visited on 05/28/2021).

[152] Google Inc. *cookie_util.cc - Chromium Code Search*. 2021. URL: `https://bit.ly/2LbBRYJ` (visited on 08/16/2021).

[153] Google Inc. *CookieMonster - The Chromium Projects*. 2021. URL: `https://www.chromium.org/developers/design-documents/network-stack/cookiemonster` (visited on 08/16/2021).

[154] Google Inc. *Developer Program Policies*. Chrome Developers. 2020. URL: `https://developer.chrome.com/docs/webstore/program_policies/` (visited on 05/22/2021).

[155] Google Inc. *Robots.txt Introduction & Guide | Google Search Central*. Google Developers. 2021. URL: `https://developers.google.com/search/docs/advanced/robots/intro` (visited on 07/09/2021).

[156] Google Inc. *Transparent privacy practices for Chrome Extensions*. Chromium Blog. 2020. URL: `https://blog.chromium.org/2020/11/transparent-privacy-practices.html` (visited on 05/13/2021).

[157] Google Inc. *Updated Privacy Policy & Secure Handling Requirements*. Chrome Developers. 2020. URL: `https://developer.chrome.com/docs/webstore/user_data/` (visited on 05/20/2021).

[158] Adblock Inc. *AdBlock Privacy Policy*. 2022. URL: `https://web.archive.org/web/20220318225253/https://getadblock.com/en/privacy/` (visited on 03/18/2022).

[159] Docker Inc. *Swarm mode*. Docker Documentation. 2021. URL: `https://docs.docker.com/engine/swarm/` (visited on 08/01/2021).

[160] Functional Software Inc. *Application Monitoring and Error Tracking Software*. Sentry. 2022. URL: https://sentry.io/welcome/ (visited on 03/20/2022).

[161] Moz Inc. *Page Speed — 2021 Website Best Practices*. Moz. 2021. URL: https://moz.com/learn/seo/page-speed (visited on 03/14/2021).

[162] Network Advertising Initiative. *FAQ | NAI: Network Advertising Initiative*. 2020. URL: https://www.networkadvertising.org/faq (visited on 03/14/2021).

[163] Network Advertising Initiative. *NAI Consumer Opt Out*. 2021. URL: https://optout.networkadvertising.org/ (visited on 04/15/2021).

[164] Costas Iordanou, Georgios Smaragdakis, Ingmar Poese, and Nikolaos Laoutaris. "Tracing Cross Border Web Tracking". In: *Proceedings of the Internet Measurement Conference 2018*. 2018.

[165] Grant Jenks. *grantjenks/python-wordsegment*. 2020.

[166] Haojian Jin, Minyi Liu, Kevan Dodhia, Yuanchun Li, Gaurav Srivastava, Matthew Fredrikson, Yuvraj Agarwal, and Jason I. Hong. "Why Are They Collecting My Data?: Inferring the Purposes of Network Traffic in Mobile Apps". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.4 (2018).

[167] Haojian Jin, Tetsuya Sakai, and Koji Yatani. "ReviewCollage: a mobile interface for direct comparison using online reviews". In: *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*. 2014.

[168] Garrett A. Johnson, Scott K. Shriver, and Shaoyin Du. "Consumer Privacy Choice in Online Advertising: Who Opts Out and at What Cost to Industry?" In: *Marketing Science* 39.1 (2020).

[169] Judith K. Jones. *Purpose Clauses: Syntax, Thematics, and Semantics of English Purpose Constructions*. 1991.

[170] Jordan Jueckstock, Shaown Sarker, Peter Snyder, Aidan Beggs, Panagiotis Papadopoulos, Matteo Varvello, Ben Livshits, and Alexandros Kapravelos. "Towards Realistic and Reproducible Web Crawl Measurements". In: *Proceedings of the The Web Conference (WWW)*. 2021.

[171] Jordan Jueckstock, Shaown Sarker, Peter Snyder, Aidan Beggs, Panagiotis Papadopoulos, Matteo Varvello, Benjamin Livshits, and Alexandros Kapravelos. "Towards Realistic and ReproducibleWeb Crawl Measurements". In: *Proceedings of the Web Conference 2021*. 2021.

[172] Daniel Jurafsky and James H. Martin. *Speech and Language Processing, 3rd edition*. Third Edition draft. 2019.

[173] Department of Justice. *Privacy Act of 1974, as amended, 5 U.S.C. § 552a*. 1974. URL: https://www.justice.gov/opcl/privacy-act-1974 (visited on 08/23/2021).

[174] Kaggle. *Data Leakage*. 2020. URL: https://kaggle.com/alexisbcook/data-leakage (visited on 10/16/2020).

[175] Alexandros Kapravelos, Chris Grier, Neha Chachra, Christopher Kruegel, Giovanni Vigna, and Vern Paxson. “Hulk: Eliciting Malicious Behavior in Browser Extensions”. In: 2014.

[176] Arjaldo Karaj, Sam Macbeth, Rémi Berson, and Josep M. Pujol. “WhoTracks .Me: Shedding light on the opaque world of online tracking”. In: *arXiv:1804.08959 [cs]* (2019). arXiv: `1804.08959`.

[177] Matthew Kay and Michael Terry. “Textured Agreements: Re-envisioning Electronic Consent”. In: *Proceedings of the Sixth Symposium on Usable Privacy and Security*. 2010.

[178] Patrick Gage Kelley, Joanna Bresee, Lorrie Faith Cranor, and Robert W. Reeder. “A "Nutrition Label" for Privacy”. In: *Proceedings of the 5th Symposium on Usable Privacy and Security*. 2009.

[179] Rishabh Khandelwal, Asmit Nayak, Hamza Harkous, and Kassem Fawaz. *CookieEnforcer: Automated Cookie Notice Analysis and Enforcement*. Tech. rep. arXiv:2204.04221. arXiv, 2022.

[180] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel*. Naval Technical Training Command Millington TN Research Branch, 1975.

[181] Chisato Kitagawa. “Purpose expressions in english”. In: *Lingua* 34.1 (1974).

[182] Bart Knijnenburg and David Cherry. “Comics as a Medium for Privacy Notices”. In: 2016.

[183] Saranga Komanduri, Richard Shay, Greg Norcie, and Blase Ur. “AdChoices - Compliance with Online Behavioral Advertising Notice and Choice Requirements”. In: *I/S: A Journal of Law and Policy for the Information Society* 7 (2011).

[184] Philip Kotler and Gary Armstrong. *Principles of marketing*. 2017.

[185] Kuba Suder. *Meet Safari Web Extensions*. WWDC NOTES. 2020. URL: `https://www.wwdcnotes.com/notes/wwdc20/10665` (visited on 09/20/2021).

[186] John Kurkowski. *tldextract*. 2020. URL: `https://github.com/john-kurkowski/tldextract` (visited on 03/15/2021).

[187] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. “Neural Architectures for Named Entity Recognition”. In: *arXiv:1603.01360 [cs]* (2016).

[188] Daniil Larionov, Artem Shelmanov, Elena Chistova, and Ivan Smirnov. “Semantic Role Labeling with Pretrained Language Models for Known and Unknown Predicates”. In: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. 2019.

[189] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. “Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation”. In: *Proceedings of the 26th Annual Network and Distributed System Security Symposium*. 2019.

[190] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4 (2020).

[191] Pedro Leon, Blase Ur, Richard Shay, Yang Wang, Rebecca Balebako, and Lorrie Cranor. "Why Johnny can't opt out: a usability evaluation of tools to limit online behavioral advertising". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2012.

[192] Adam Lerner, Anna Kornfeld Simpson, Tadayoshi Kohno, and Franziska Roesner. "Internet Jones and the Raiders of the Lost Trackers: An Archaeological Study of Web Tracking from 1996 to 2016". In: *25th USENIX Security Symposium (USENIX Security 16)*. 2016.

[193] Qi Li. "Literature survey: domain adaptation algorithms for natural language processing". In: *Department of Computer Science The Graduate Center, The City University of New York* (2012).

[194] Yuanchun Li, Ziyue Yang, Yao Guo, and Xiangqun Chen. "DroidBot: a lightweight UI-Guided test input generator for android". In: *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*. 2017.

[195] Yuanchun Li, Ziyue Yang, Yao Guo, and Xiangqun Chen. "Humanoid: A Deep Learning-Based Approach to Automated Black-box Android App Testing". In: *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 2019.

[196] Jialiu Lin, Shahriyar Amini, Jason I. Hong, Norman Sadeh, Janne Lindqvist, and Joy Zhang. "Expectation and Purpose: Understanding Users' Mental Models of Mobile App Privacy Through Crowdsourcing". In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. 2012.

[197] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv:1907.11692 [cs]* (2019).

[198] Zengrui Liu, Umar Iqbal, and Nitesh Saxena. *Opted Out, Yet Tracked: Are Regulations Enough to Protect Your Privacy?* Tech. rep. arXiv:2202.00885. arXiv, 2022.

[199] Will Bontrager Software LLC. *Linking Without an 'A' Tag*. 2021. URL: https://www.willmaster.com/library/web-development/linking-without-an-a-tag.php (visited on 03/01/2021).

[200] SimilarWeb LTD. *Top Websites in United States - Website Ranking | SimilarWeb*. 2020. URL: https://www.similarweb.com/top-websites/united-states/ (visited on 11/28/2020).

[201] Lucy Cui. *MythBusters: Highlighting helps me study*. Psychology In Action. 2018. URL: https://www.psychologyinaction.org/psychology-in-action-1/2018/1/8/mythbusters-highlighting-helps-me-study (visited on 08/22/2020).

[202] Xuezhe Ma and Eduard Hovy. “End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF”. In: 2016.

[203] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008).

[204] D. Malandrino, V. Scarano, and R. Spinelli. “How Increased Awareness Can Impact Attitudes and Behaviors toward Online Privacy Protection”. In: *2013 International Conference on Social Computing*. 2013.

[205] Christopher Manning. *Representations for Language: From Word Embeddings to Sentence Meanings | Simons Institute for the Theory of Computing*. 2017.

[206] Christopher Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. 1999. 657 pp.

[207] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. 2008.

[208] C. Matte, N. Bielova, and C. Santos. “Do Cookie Banners Respect my Choice? : Measuring Legal Compliance of Banners from IAB Europe’s Transparency and Consent Framework”. In: *2020 IEEE Symposium on Security and Privacy (SP)*. 2020.

[209] Thor May. “Purposive constructions in English”. In: *Australian Journal of Linguistics* 10.1 (1990).

[210] Erika McCallister, Tim Grance, and Karen Scarfone. *Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)*. Tech. rep. NIST Special Publication (SP) 800-122. National Institute of Standards and Technology, 2010.

[211] Steve McConnell. *Code Complete, Second Edition*. 2004.

[212] A.M. McDonald and L.F. Cranor. “The cost of reading privacy policies”. In: *I/S: A Journal of Law and Policy for the Information Society* 4 (2008).

[213] Aleecia McDonald and Lorrie Faith Cranor. *Beliefs and Behaviors: Internet Users’ Understanding of Behavioral Advertising*. SSRN Scholarly Paper ID 1989092. Social Science Research Network, 2010.

[214] Aleecia M. McDonald, Robert W. Reeder, Patrick Gage Kelley, and Lorrie Faith Cranor. “A Comparative Study of Online Privacy Policies and Formats”. In: *Privacy Enhancing Technologies*. Ed. by Ian Goldberg and Mikhail J. Atallah. 2009.

[215] Hassan Metwalley, Stefano Traverso, and Marco Mellia. “Unsupervised Detection of Web Trackers”. In: *2015 IEEE Global Communications Conference (GLOBECOM)*. 2015.

[216] Microsoft. *Docker | Playwright Python*. 2021. URL: `https://playwright.dev/python/docs/docker/` (visited on 10/05/2021).

[217] Microsoft. *microsoft/playwright-python*. 2020.

[218] Microsoft. *Tracking Prevention in Microsoft Edge*. 2021. URL: `https://docs.microsoft.com/en-us/microsoft-edge/web-platform/tracking-prevention` (visited on 06/26/2021).

[219] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. "Distributed Representations of Words and Phrases and Their Compositionality". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. 2013.

[220] Minimum Wage | U.S. Department of Labor. *https://www.dol.gov/agencies/whd/minimum-wage*. 2020.

[221] Moz Inc. *URL Structure*. Moz. 2021. URL: https://moz.com/learn/seo/url (visited on 07/09/2021).

[222] Mozilla. *Add-on Policies*. Firefox Extension Workshop. 2019. URL: https://extensionworkshop.com/documentation/publish/add-on-policies/ (visited on 05/22/2021).

[223] Mozilla. *Enhanced Tracking Protection in Firefox for desktop*. 2021. URL: https://support.mozilla.org/en-US/kb/enhanced-tracking-protection-firefox-desktop#w_what-enhanced-tracking-protection-blocks (visited on 06/26/2021).

[224] Mozilla. *Navigation Timing API - Web APIs | MDN*. 2021. URL: https://developer.mozilla.org/en-US/docs/Web/API/Navigation_timing_API (visited on 03/14/2021).

[225] Mozilla. *Node.textContent - Web APIs | MDN*. 2021. URL: https://developer.mozilla.org/en-US/docs/Web/API/Node/textContent (visited on 07/09/2021).

[226] Mozilla. *Origin - HTTP request header*. 2021. URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Origin (visited on 09/27/2021).

[227] Mozilla. *WebExtensions - MozillaWiki*. 2018. URL: https://wiki.mozilla.org/WebExtensions (visited on 05/22/2021).

[228] Mozilla. *Window: load event - Web APIs | MDN*. 2021. URL: https://developer.mozilla.org/en-US/docs/Web/API/Window/load_event (visited on 03/14/2021).

[229] Kanthashree Mysore Sathyendra, Shomir Wilson, Florian Schaub, Sebastian Zimmeck, and Norman Sadeh. "Identifying the Provision of Choices in Privacy Policy Text". In: 2017.

[230] Niclas. *Clefspeare13/pornhosts*. 2020.

[231] Helen Nissenbaum. "PRIVACY AS CONTEXTUAL INTEGRITY". In: *Washington Law Review* 79 (2004).

[232] Sherrie L. Nist and Mark C. Hogrebe. "The Role of Underlining and Annotating in Remembering Textual Information". In: *Reading Research and Instruction* 27.1 (1987).

[233] OECD, OCDE. "The OECD principles of corporate governance". In: *Contaduria y Administracion* 216 (2004).

[234] The U.S. Government Publishing Office. *United States Code, 2006 Edition, Supplement 5, Title 15 - COMMERCE AND TRADE*. 2011.

[235] Ehimare Okoyomon, Nikita Samarin, Primal Wijesekera, Amit Elazari Bar On, Narseo Vallina-Rodriguez, Irwin Reyes, Alvaro Feal, and Serge Egelman. “On The Ridiculousness of Notice and Consent: Contradictions in App Privacy Policies”. In: 2019.

[236] CookiePro by OneTrust. *What is a Third-Party Cookie?* CookiePro. 2020. URL: `https://www.cookiepro.com/knowledge/what-is-a-third-party-cookie/` (visited on 03/15/2021).

[237] LLC OneTrust. *OneTrust Cookies*. 2021. URL: `https://bit.ly/3nDfUAV` (visited on 05/02/2021).

[238] Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. “A Span Selection Model for Semantic Role Labeling”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018.

[239] Xiang Pan, Yinzhi Cao, Xuechao Du, Boyuan He, Gan Fang, and Yan Chen. “FlowCog: context-aware semantics extraction and analysis of information flow leaks in android apps”. In: *Proceedings of the 27th USENIX Conference on Security Symposium*. 2018.

[240] Rahul Pandita, Xusheng Xiao, Wei Yang, William Enck, and Tao Xie. “{WHYPER}: Towards Automating Risk Assessment of Mobile Applications”. In: 2013.

[241] Emmanouil Papadogiannakis, Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. “User Tracking in the Post-cookie Era: How Websites Bypass GDPR Consent to Track Users”. In: *Proceedings of the Web Conference 2021*. 2021.

[242] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos Markatos. “Cookie Synchronization: Everything You Always Wanted to Know But Were Afraid to Ask”. In: *The World Wide Web Conference*. 2019.

[243] European Parliament. *Regulation (EU) 2016/679*. 2016. URL: `https://web.archive.org/web/20210813201707/https://eur-lex.europa.eu/eli/reg/2016/679/oj` (visited on 08/22/2021).

[244] Paul E. Black. *Ratcliff/Obershelp pattern recognition*. Dictionary of Algorithms and Data Structures. 2004. URL: `https://xlinux.nist.gov/dads/HTML/ratcliffObershelp.html` (visited on 01/02/2021).

[245] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. “GloVe: Global Vectors for Word Representation”. In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014.

[246] Raffaello Perrotta and Feng Hao. “Botnet in the Browser: Understanding Threats Caused by Malicious Browser Extensions”. In: *IEEE Security Privacy* 16.4 (2018).

[247] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. “Deep contextualized word representations”. In: *Proc. of NAACL*. 2018.

[248] Sarah E. Peterson. “The cognitive functions of underlining as a study technique”. In: *Reading Research and Instruction* 31.2 (1991).

[249] Robert Pitofsky, Sheila F Anthony, Mozelle W Thompson, Orson Swindle, and Thomas B Leary. “Privacy Online: Fair Information Practices in the Electronic Marketplace: A Federal Trade Commission Report to Congress”. In: *PRIVACY ONLINE: FAIR INFORMATION PRACTICES IN THE ELECTRONIC MARKETPLACE* (2000).

[250] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. “Towards Robust Linguistic Analysis using OntoNotes”. In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. 2013.

[251] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. “CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes”. In: *Joint Conference on EMNLP and CoNLL-Shared Task*. Association for Computational Linguistics. 2012.

[252] Python-Markdown. *markdown*. 2021.

[253] Zhengyang Qu, Vaibhav Rastogi, Xinyi Zhang, Yan Chen, Tiantian Zhu, and Zhong Chen. “AutoCog: Measuring the Description-to-permission Fidelity in Android Applications”. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014.

[254] Qualtrics. *Online Survey Software https://www.qualtrics.com/*. Qualtrics. 2020. URL: https://www.qualtrics.com/core-xm/survey-software/ (visited on 08/29/2020).

[255] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. *OntoNotes Release 5.0 LDC2013T19. Web Download. Philadelphia: Linguistic Data Consortium*. 2013.

[256] Ashwini Rao, Florian Schaub, Norman Sadeh, Alessandro Acquisti, and Ruogu Kang. “Expecting the Unexpected: Understanding Mismatched Privacy Expectations Online”. In: 2016.

[257] Joel R. Reidenberg, Travis Breaux, Lorrie Faith Cranor, Brian French, Amanda Grannis, James T. Graves, Fei Liu, Aleecia McDonald, Thomas B. Norton, Rohan Ramanath, N. Cameron Russell, Norman Sadeh, and Florian Schaub. “Disagreeable Privacy Policies: Mismatches Between Meaning and Users’ Understanding”. In: *Berkeley Technology Law Journal* 30.1 (2015).

[258] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 2019.

[259] Irwin Reyes, Primal Wiesekera, Abbas Razaghpanah, Joel Reardon, Narseo Vallina-Rodriguez, Serge Egelman, and Christian Kreibich. “"Is Our Children’s Apps Learning?" Automatically Detecting COPPA Violations”. In: 2017.

[260] Irwin Reyes, Primal Wijesekera, Joel Reardon, Amit Elazari Bar On, Abbas Razaghpanah, Narseo Vallina-Rodriguez, and Serge Egelman. “"Won’t Somebody Think of the Children?" Examining COPPA Compliance at Scale”. In: *Proceedings on Privacy Enhancing Technologies* 2018.3 (2018).

[261] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. “Detecting and Defending Against Third-Party Tracking on the Web”. In: 2012.

[262] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. “A Primer in BERTology: What we know about how BERT works”. In: *arXiv:2002.12327 [cs]* (2020).

[263] Lior Rokach and Oded Maimon. “Clustering methods”. In: *Data mining and knowledge discovery handbook*. 2005.

[264] T. Sakamoto and M. Matsunaga. “After GDPR, Still Tracking or Not? Understanding Opt-Out States for Online Behavioral Advertising”. In: *2019 IEEE Security and Privacy Workshops (SPW)*. 2019.

[265] Iskander Sanchez-Rola, Matteo Dell’Amico, Platon Kotzias, Davide Balzarotti, Leyla Bilge, Pierre-Antoine Vervier, and Igor Santos. “Can I Opt Out Yet? GDPR and the Global Illusion of Cookie Control”. In: *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*. 2019.

[266] Alireza Savand. *html2text*. 2021.

[267] Shlomo Sawilowsky. “New Effect Size Rules of Thumb”. In: *Journal of Modern Applied Statistical Methods* 8.2 (2009).

[268] F. Schaub, R. Balebako, and L. F. Cranor. “Designing Effective Privacy Notices and Controls”. In: *IEEE Internet Computing* 21.3 (2017).

[269] Florian Schaub. *Nobody reads privacy policies – here’s how to fix that*. 2017.

[270] searchpreview.de. *SearchPreview*. 2021. URL: `https://chrome.google.com/webstore/detail/searchpreview/hcjdanpjacpeeppdjkppebobilhaglfo` (visited on 08/01/2021).

[271] *seatgeek/fuzzywuzzy*. 2020.

[272] SEOPressor. “Does URL Structure Affect SEO? Here’s What Google Thinks”. In: (2019).

[273] Peng Shi and Jimmy Lin. “Simple BERT Models for Relation Extraction and Semantic Role Labeling”. In: *arXiv:1904.05255 [cs]* (2019).

[274] Laura Shipp and Jorge Blasco. “How private is your period?: A systematic analysis of menstrual app privacy policies”. In: *Proceedings on Privacy Enhancing Technologies* 4 (2020).

[275] Anastasia Shuba and Athina Markopoulou. "NoMoATS: Towards Automatic Detection of Mobile Tracking". In: *Proceedings on Privacy Enhancing Technologies* 2020.2 (2020).

[276] Yan Shvartzshnaider, Noah Apthorpe, Nick Feamster, and Helen Nissenbaum. "Going against the (Appropriate) Flow: A Contextual Integrity Approach to Privacy Policy Analysis". In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* 7.1 (2019).

[277] Vicki L. Silvers and David S. Kreiner. "The effects of pre-existing inappropriate highlighting on reading comprehension". In: *Reading Research and Instruction* 36.3 (1997).

[278] Jordan Sissel. *xdotool - x11 automation tool*. 2021.

[279] Rocky Slavin, Xiaoyin Wang, Mitra Bokaei Hosseini, James Hester, Ram Krishnan, Jaspreet Bhatia, Travis D. Breaux, and Jianwei Niu. "Toward a Framework for Detecting Privacy Policy Violations in Android Application Code". In: *Proceedings of the 38th International Conference on Software Engineering*. 2016.

[280] Yannis Smaragdakis, Jacob Evans, Caitlin Sadowski, Jaeheon Yi, and Cormac Flanagan. "Sound predictive race detection in polynomial time". In: *Proceedings of the 39th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*. 2012.

[281] Konstantinos Solomos, Panagiotis Ilia, Soroush Karami, Nick Nikiforakis, and Jason Polakis. "The Dangers of Human Touch: Fingerprinting Browser Extensions through User Actions". In: *Proceedings of the 31st USENIX Security Symposium (USENIX Security 22)*. 2022.

[282] Dolière Francis Somé. "EmPoWeb: Empowering Web Applications with Browser Extensions". In: *2019 IEEE Symposium on Security and Privacy (SP)*. 2019.

[283] Sovrn. *Sovrn Acquires VigLink to Expand Publisher Services*. Sovrn, Inc. 2018. URL: `https://www.sovrn.com/blog/sovrn-acquires-viglink/` (visited on 05/01/2022).

[284] Mukund Srinath, Shomir Wilson, and C. Lee Giles. "Privacy at Scale: Introducing the PrivaSeer Corpus of Web Privacy Policies". In: *arXiv:2004.11131 [cs]* (2020). arXiv: `2004.11131`.

[285] Gaurav Srivastava, Kunal Bhuwalka, Swarup Kumar Sahoo, Saksham Chitkara, Kevin Ku, Matt Fredrikson, Jason Hong, and Yuvraj Agarwal. "PrivacyProxy: Leveraging Crowdsourcing and In Situ Traffic Analysis to Detect and Mitigate Information Leakage". In: *arXiv:1708.06384 [cs]* (2018).

[286] Oleksii Starov and Nick Nikiforakis. "Extended Tracking Powers: Measuring the Privacy Diffusion Enabled by Browser Extensions". In: *Proceedings of the 26th International Conference on World Wide Web*. 2017.

[287] Statista. *Desktop internet browser market share 2015-2021*. Statista. 2022. URL: `https://www.statista.com/statistics/544400/market-share-of-internet-browsers-desktop/` (visited on 03/16/2022).

[288] Statista. *Internet users in the world 2022*. Statista. 2022. URL: https://www.statista.com/statistics/617136/digital-population-worldwide/ (visited on 04/28/2022).

[289] Pontus Stenetorp, Goran Topić, Sampo Pyysalo, Tomoko Ohta, Jin-Dong Kim, and Jun'ichi Tsujii. “BioNLP Shared Task 2011: Supporting Resources”. In: *Proceedings of BioNLP Shared Task 2011 Workshop*. 2011.

[290] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. 2009.

[291] Sweepatic. *Sweepatic releases GDPR cookie violation detection feature*. 2022.

[292] Madiha Tabassum, Abdulmajeed Alqhatani, Marran Aldossari, and Heather Richter Lipford. “Increasing User Attention with a Comic-based Policy”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018.

[293] TermsFeed. *Personal vs. Sensitive Information*. TermsFeed. 2021. URL: https://www.termsfeed.com/blog/personal-vs-sensitive-information/ (visited on 04/21/2021).

[294] The European Parliament and the Council of the European Union. *Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications)*. 2002.

[295] Tim Nolet. *Navigating & waiting*. 2020. URL: https://theheadless.dev/posts/basics-navigation/ (visited on 03/14/2021).

[296] Erik F. Tjong Kim Sang and Fien De Meulder. “Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition”. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*. 2003.

[297] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “Well-Read Students Learn Better: On the Importance of Pre-training Compact Models”. In: *arXiv:1908.08962 [cs]* (2019).

[298] Aarhus University. *cavi-au/Consent-O-Matic*. 2021.

[299] Tobias Urban, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. “Beyond the Front Page:Measuring Third Party Dynamics in the Field”. In: *Proceedings of The Web Conference 2020*. 2020.

[300] Pelayo Vallina, Álvaro Feal, Julien Gamba, Narseo Vallina-Rodriguez, and Antonio Fernandez Anta. “Tales from the Porn: A Comprehensive Privacy Analysis of the Web Porn Ecosystem”. In: *Proceedings of the Internet Measurement Conference*. 2019.

[301] W3C. *Navigation Timing W3C Recommendation*. 2012. URL: https://www.w3.org/TR/navigation-timing/ (visited on 03/14/2021).

[302] X. Wang, X. Qin, M. Bokaei Hosseini, R. Slavin, T. D. Breaux, and J. Niu. “GUILeak: Tracing Privacy Policy Claims on User Input Data for Android Applications”. In: *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)*. 2018.

[303] Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N. Cameron Russell, Thomas B. Norton, Eduard Hovy, Joel Reidenberg, and Norman Sadeh. “The Creation and Analysis of a Website Privacy Policy Corpus”. In: 2016.

[304] Shomir Wilson, Florian Schaub, Rohan Ramanath, Norman Sadeh, Fei Liu, Noah A. Smith, and Frederick Liu. “Crowdsourcing Annotations for Websites’ Privacy Policies: Can It Really Work?” In: *Proceedings of the 25th International Conference on World Wide Web - WWW ’16*. 2016.

[305] Stephanie Winkler and Sherali Zeadally. “Privacy Policy Analysis of Popular Web Platforms”. In: *IEEE Technology and Society Magazine* 35.2 (2016).

[306] Daniel W. Woods and Rainer Böhme. “The commodification of consent”. In: *Computers & Security* 115 (2022).

[307] Mengfei Xie, Jianming Fu, Jia He, Chenke Luo, and Guojun Peng. “JTaint: Finding Privacy-Leakage in Chrome Extensions”. In: *Information Security and Privacy*. Ed. by Joseph K. Liu and Hui Cui. 2020.

[308] Vikas Yadav and Steven Bethard. “A Survey on Recent Advances in Named Entity Recognition from Deep Learning models”. In: *Proceedings of the 27th International Conference on Computational Linguistics*. 2018.

[309] Zhiju Yang and Chuan Yue. “A Comparative Measurement Study of Web Tracking on Mobile and Desktop Environments”. In: *Proceedings on Privacy Enhancing Technologies* 2020.2 (2020).

[310] Le Yu, Xiapu Luo, Xule Liu, and Tao Zhang. “Can We Trust the Privacy Policies of Android Apps?” In: *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 2016.

[311] Hui Zhang. *Beyond Query-Oriented Highlighting: Investigating the Effect of Snippet Text Highlighting in Search User Behavior*. Research Article. 2018.

[312] Yufei Zhao, Longtao He, Zhoujun Li, Liqun Yang, Hao Dong, Chao Li, and Yu Wang. “Large-scale Detection of Privacy Leaks for BAT Browsers Extensions in China”. In: *2019 International Symposium on Theoretical Aspects of Software Engineering (TASE)*. 2019.

[313] Yufei Zhao, Liqun Yang, Zhoujun Li, Longtao He, and Yipeng Zhang. “Privacy Model: Detect Privacy Leakage for Chinese Browser Extensions”. In: *IEEE Access* 9 (2021).

[314] Sebastian Zimmeck and Steven M. Bellovin. “Privee: An Architecture for Automatically Analyzing Web Privacy Policies”. In: 2014.

[315] Sebastian Zimmeck, Peter Story, Daniel Smullen, Abhilasha Ravichander, Ziqi Wang, Joel Reidenberg, N. Cameron Russell, and Norman Sadeh. “MAPS: Scaling Privacy Compliance Analysis to a Million Apps”. In: *Proceedings on Privacy Enhancing Technologies* 2019.3 (2019).

[316] Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman Sadeh, Steven M. Bellovin, and Joel Reidenberg. “Automated Analysis of Privacy Requirements for Mobile Apps”. In: *Proceedings 2017 Network and Distributed System Security Symposium*. 2017.

[317] Martin Zurowietz, Daniel Langenkämper, Brett Hosking, Henry A. Ruhl, and Tim W. Nattkemper. “MAIA—A machine learning assisted image annotation method for environmental monitoring and exploration”. In: *PLOS ONE* 13.11 (2018).