Natural Language Processing for Mobile App Privacy Compliance

Peter Story*[1], Sebastian Zimmeck*[2], Abhilasha Ravichander[1], Daniel Smullen[1], Ziqi Wang[1], Joel Reidenberg[3], N. Cameron Russell[3], and Norman Sadeh*[1]

[1]School of Computer Science, Carnegie Mellon University
[2]Department of Mathematics and Computer Science, Wesleyan University
[3]School of Law, Fordham University

Abstract

Many Internet services collect a flurry of data from their users. Privacy policies are intended to describe the services' privacy practices. However, due to their length and complexity, reading privacy policies is a challenge for end users, government regulators, and companies. Natural language processing holds the promise of helping address this challenge. Specifically, we focus on comparing the practices described in privacy policies to the practices performed by smartphone apps covered by those policies. Government regulators are interested in comparing apps to their privacy policies in order to detect non-compliance with laws, and companies are interested for the same reason.

We frame the identification of privacy practice statements in privacy policies as a classification problem, which we address with a three-tiered approach: a privacy practice statement is classified based on a data type (e.g., location), party (i.e., first or third party), and modality (i.e., whether a practice is explicitly described as being performed or not performed). Privacy policies omit discussion of many practices. With negative F1 scores ranging from 78% to 100%, the performance results of this three-tiered classification methodology suggests an improvement over the state-of-the-art.

Our NLP analysis of privacy policies is an integral part of our Mobile App Privacy System (MAPS), which we used to analyze 1,035,853 free apps on the Google Play Store. Potential compliance issues appeared to be widespread, and those involving third parties were particularly common.

1 Introduction

In the absence of a general privacy law in the United States, the Federal Trade Commission (FTC) is stepping into the void and is creating a "common law of privacy" (Solove and Hartzog 2014), which, to a large extent, is based on the notice and choice paradigm. In this paradigm, users are notified of a service's privacy practices and are given a choice to consent to those practices; if the user does not consent to the service's practices, they are not allowed to use the service. Natural language privacy policies are intended to notify users of privacy practices. Privacy policies are complex and lengthy documents: they are often vague, internally contradictory, offer little protection, or are silent on critical points (Marotta-Wurgler 2015). While there are other forms of privacy notification, such as mobile app permission requests, these are not a replacement for privacy policies; permission requests are generally insufficient to express what users agree to with sufficient clarity. Machine-readable privacy policies, such as P3P policies (Cranor et al. 2002), were suggested as replacements for natural language privacy policies. However, none of these replacements have gained widespread adoption. Thus, despite their shortcomings, natural language privacy policies are the standard instrument for effectuating notice and choice.

The FTC engages in enforcement actions against operators of apps that are non-compliant with their privacy policies. Such non-compliance is considered an unfair or deceptive act or practice in or affecting commerce in violation of Section 5(a) of the FTC Act (FTC 2014). In order to detect whether an app is potentially not compliant with its privacy policy, we built the Mobile App Privacy System (MAPS) (Zimmeck et al. 2019). MAPS is of interest to both government regulators and companies. For government regulators, MAPS can identify potential compliance issues, reducing the cost of investigations. For companies, MAPS can help them ensure that their privacy policies fully describe their apps' practices.

Our focus in this article is on the natural language analysis component of MAPS. We provide a detailed description of the design and performance of our three-tiered classifier design for identifying privacy practice statements (§ 3). We also provide a summary of findings from our recent scan of over 1 million mobile apps on the Google Play Store (§ 4). In this large scale analysis of policies and apps, we found widespread evidence of potential privacy compliance issues. In particular, it appears that many apps' privacy policies do not sufficiently disclose identifier and location data access practices performed by ad networks and other third parties.

2 Related Work

Our work leverages earlier studies on the automated analysis of privacy policy text as well as examinations of privacy in

*The first two authors contributed equally to this study. Previously, Sebastian Zimmeck was a postdoctoral associate at Carnegie Mellon's School of Computer Science. The corresponding authors' emails are: pstory@andrew.cmu.edu, szimmeck@wesleyan.edu, and sadeh@cs.cmu.edu.

the Android ecosystem.

Automated Privacy Policy Text Analysis

Privacy policies are the main instruments for disclosing and describing apps' or other software's privacy practices. However, the sheer volume of text an individual user would need to read for the software he or she is using makes privacy policies impractical for meaningfully conveying privacy practices (McDonald and Cranor 2008). Some research has focused on the structure of privacy policies. For example, the problem of identifying policy sections relating to different topics (Ramanath et al. 2014; Liu et al. 2018). Sathyendra et al. classified advertising opt outs and similar consumer choice options on websites (Sathyendra et al. 2017). Other work has focused on building tools for users. Using a simple naive Bayes classifier, Zimmeck and Bellovin provided a browser extension for identifying common privacy practices in policy text (Zimmeck and Bellovin 2014). Tesfay et al. used a machine learning-based approach to identify text addressing various GDPR provisions (Tesfay et al. 2018). Harkous et al. developed PriBot, a chatbot for answering questions about privacy policies (Harkous et al. 2018). Different from those studies, however, our domain consists of app policies instead of website policies.

Various studies analyzed privacy policies in specific domains. Cranor et al. evaluated financial institutions' privacy notices, which, in the US, nominally adhere to a model privacy form released by federal agencies (Cranor, Leon, and Ur 2016). They found clusters of institutions sharing consumer data more often than others. They also found institutions that do not follow the law, by disallowing consumers to limit such sharing. Further, Zhuang et al. aimed to help university researchers by automating enforcement of privacy policies of Institutional Review Boards (Zhuang et al. 2018). Auditing the disclosure of third party data collection practices on 200,000 website privacy policies, Libert found that the names of third parties are usually not explicitly disclosed in website privacy policies (Libert 2018). We focus on classifying first and third party access of contact, location, and unique identifier data in smartphone apps' privacy policies.

Android Privacy Studies

We are extending the emerging domain of verifying privacy practices of mobile apps against privacy requirements, notably privacy policies. The closest related work to ours analyzed the practices of 17,991 Android apps and determined whether those with a privacy policy adhered to it (Zimmeck et al. 2017). Several other studies have also compared privacy policies to apps' code (Yu et al. 2016; Slavin et al. 2016). Going beyond this work, our system is capable of large-scale analyses, which we demonstrate by an analysis of 1,035,853 free apps on the Google Play Store. Additionally, our analysis evaluates compliance issues at a finer granularity. This advance is notable because the access of coarse-grained location data (e.g., city) is far less privacy-invasive than the access of fine-grained data (e.g., latitude and longitude).

We are motivated to study privacy in the Android ecosystem due to numerous findings of potentially non-compliant privacy practices. Story et al. studied the metadata of over a million apps on the Play Store and found that many apps lack privacy policies, even when developers describe their apps as collecting users' information (Story, Zimmeck, and Sadeh 2018). Analyzing close to a million Android web apps (i.e., Android apps that use a WebView), Mutchler et al. found that 28% of those have at least one vulnerability, such as data leakage through overridden URL loads (Mutchler et al. 2015). Differences in how apps are treating sensitive data were used to identify malicious apps (Avdiienko et al. 2015). More recently, AppCensus revealed that many Android apps collect persistent device identifiers to track users, which is not allowed for advertising purposes according to the Google Play Developer Program Policy (Reyes et al. 2018; Google 2018b). The observation of 512 popular Android apps over eight years of version history by Ren et al. came to the conclusion that an important factor for higher privacy risks over time is the increased number of third party domains receiving personally identifiable information (Ren et al. 2018). In line with these observations, it is one of our goals in this study to examine apps' third party practices in the Android ecosystem.

3 Analysis Techniques

Our Mobile App Privacy System (MAPS) is comprised of separate modules for the analysis of privacy policies and apps. Our system compares the policy and app analyses in order to identify potential compliance issues.

Privacy Practices

Our system analyzes privacy practices. A *privacy practice*, or simply *practice*, describes a behavior of an app that can have privacy implications. Table 3 contains the list of practices we consider in our model.[1] We account for the fact that disclosures found in privacy policies can vary in specificity. For instance, for the access of location data our model includes practices that pertain to location in general (i.e., `Location`) as well as more specific practices that explicitly identify the type of access (i.e., `Location Cell Tower`, `Location GPS`, and `Location WiFi`). Our model distinguishes between first party access, where data is accessed by the code of the app itself, and third party access, where data is accessed by advertising or other third party libraries. Finally, our model also distinguishes between a policy describing the performance of a practice (e.g., "We access your location information.") and the description that a practice is not performed (e.g., "We do not access your location information."). When access is neither explicitly described nor explicitly denied, neither modality classifier flags the statement. Note that a given text fragment can refer to multiple practices.

[1] In preliminary tests we also considered city, ZIP code, postal address, username, password, ad ID, address book, Bluetooth, IP address (identifier and location), age, and gender practices. However, we ultimately decided against further pursuing those as we had insufficient data, unreliable annotations, or difficulty identifying a corresponding API for the app analysis.

Privacy Policy Analysis

We characterize the detection of privacy practice descriptions in privacy policy text as a classification problem.

Dataset We used the APP-350 corpus of 350 annotated mobile app privacy policies to train and test our classifiers (Zimmeck et al. 2019).[2] The corpus's policies were selected from the most popular apps on the Google Play Store. The policies were annotated by legal experts using a set of privacy practice annotation labels. As they were annotating the policies, the experts also identified the policy text fragments corresponding to the practice annotation labels they applied. All policies were comprehensively annotated. Consequently, it is assumed that all unannotated portions of text do not describe any of the practices and can be used as training, validation, and test data to detect the absence of statements on respective practices.

We randomly split the annotated privacy policies into training ($n = 188$), validation ($n = 62$), and test ($n = 100$) sets. We used the training and validation sets to develop our classifiers. The test set was set aside in order to prevent overfitting. We did not calculate performance using the test set until after we finished developing our classifiers.

Classification Task The goal of the classification task is to assign annotation labels to policy segments, that is, structurally related parts of policy text that loosely correspond to paragraphs (Wilson et al. 2016; Liu et al. 2018). We focus on segments instead of entire policies to make effective use of the annotated data and to identify the specific policy text locations that describe a certain practice.

The infrequent occurrence of certain types of statements makes the training of classifiers for some practices more challenging. In particular, statements on third party practices and statements explicitly denying that activities are performed are rare. For example, our training set only includes 7 segments saying that `Location Cell Tower` information is *not* accessed by third parties. To address this challenge, we decompose the classification problem into three subproblems, that is, classifying (1) data types (e.g., `Location`), (2) parties (i.e., `1stParty` or `3rdParty`)[3], and (3) modalities (i.e., whether a practice is explicitly described as being performed or not performed). For example, the `Location Cell Tower 3rdParty Not Performed` classification will be assigned to a segment if the `Location Cell Tower`, `3rdParty`, and `Not Performed` classifiers all return a positive result for the segment.

The decomposition of the classification task allows for an economic use of annotated data. If the subproblems were tackled all at once, 68 monolithic classifiers would be needed, most of which would have to be trained on fewer than 100 positive training samples. By dividing the problem, only 22 classifiers are needed (18 "data type", 2 "party", and 2 "modality" classifiers). These classifiers have a much higher number of positive samples available for training, as shown in Figure 1.

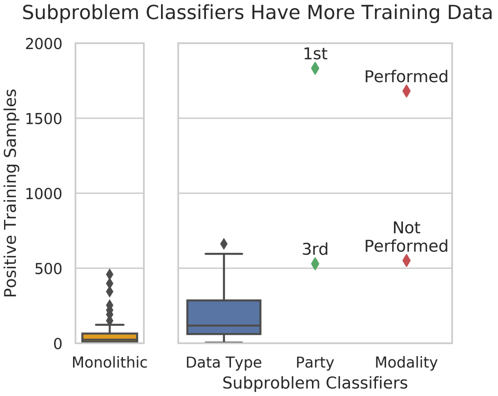

Figure 1: By decomposing the classification task into three subproblems more positive training samples are available than for monolithic classifiers.

Preprocessing As classifier performance depends on adequate preprocessing of policy text as well as domain-specific feature engineering, we normalize whitespace and punctuation, remove non-ASCII characters, and lowercase all policy text. Because stemming did not lead to performance improvements, we are omitting it. In order to run our classifiers on the most relevant set of features, we use an optional preprocessing step of sentence filtering. Based on a grid search, in cases where it improves classifier performance, we remove a segment's sentences from further processing if they do not contain keywords related to the classifier in question (Zimmeck et al. 2017). For example, the `Location` classifier is not trained on sentences which only describe cookies.

Vectorizing Prior to training, we generate vector representations of the segments. Specifically, we take the union of a TF-IDF vector and a vector of manually crafted features. Our TF-IDF vector is created using the TfidfVectorizer (scikit-learn developers 2016a) configured with English stopwords (`stop_words='english'`), unigrams and bigrams (`ngram_range=(1, 2)`), and binary term counts (`binary=True`). This configuration is similar to what was used in prior work (Liu et al. 2018). Our vector of manually crafted features consists of Boolean values indicating the presence or absence of indicative strings we observed in our training and validation data. For example, we include the string `not collect`, because we realized that it would be a strong indicator of the negative modality.

Training Using scikit-learn, version 0.18.1 (Pedregosa et al. 2011) we train binary classifiers for each data type, party, and modality. For all but four classifiers we use scikit-learn's SVC implementation (scikit-learn developers 2016b). We train those with a linear kernel (`kernel='linear'`), balanced class weights (`class_weight='balanced'`), and a grid search with

[2] The dataset is available at https://data.usableprivacy.org.

[3] Note that the `Single Sign On` and `Single Sign On: Facebook` practices do not use a party classifier, as all data is exchanged between the app developer as first party and the SSO provider as third party.

	Configuration				
Classifier	**Baseline**	**+ Bigrams**	**+ C.F.**	**+ S.F.**	**+ Final**
Contact	29%	**+42%**	+35%	+28%	+39%
Contact Email Address	73%	+9%	+8%	+12%	+11%
Contact Phone Number	76%	+11%	+2%	+11%	+9%
Identifier Cookie	88%	+2%	+2%	+3%	+2%
Identifier Device ID	74%	+9%	+8%	+12%	+13%
Identifier IMEI	77%	-19%	+20%	+17%	+17%
Identifier MAC	84%	-23%	+2%	-2%	-2%
Identifier Mobile Carrier	62%	-14%	-3%	0%	-14%
Location	80%	+4%	+3%	+9%	+6%
Location Cell Tower	62%	-4%	+4%	+12%	+10%
Location GPS	76%	-1%	+3%	+16%	+12%
Location WiFi	74%	+5%	+5%	+1%	+5%
Single Sign On	63%	+7%	+11%	-43%	+4%
Single Sign On: Facebook	75%	+3%	+5%	-64%	+6%
1stParty	95%	+0%	-1%	-1%	+0%
3rdParty	77%	+4%	+2%	+1%	+1%
Performed	90%	+1%	+5%	-2%	+6%
Not Performed	73%	-2%	+13%	+5%	+14%

Table 1: Effects of different preprocessing and feature configurations on our classifiers' F1 scores. Effects are calculated with regard to a baseline configuration (Baseline), in which the TF-IDF vectors only include unigrams. For the baseline, bigrams (Bigrams) and manually crafted features (C.F.) are not included, and keyword-based sentence filtering (S.F.) is not performed. For example, including bigrams in our TF-IDF vectors leads to an F1 score increase of 42% (from 29% to 71%) for the Contact classifier. Our final configuration (Final) includes bigrams as well as crafted features; sentence filtering is enabled on a per-classifier basis using a grid search.

Configuration	Improved	Avg. Improvement	Decreased	Avg. Decrease
+ Bigrams	**11/18**	**+8.8%**	**6/18**	**-10.5%**
+ Crafted Features	16/18	+8.0%	2/18	-2.0%
+ Sentence Filtering	12/18	+10.6%	5/18	-22.4%
Final	15/18	+10.3%	2/18	-8.0%

Table 2: Summary of effects of different preprocessing and feature configurations on F1 scores based on the data shown in Table 1. For example, adding bigrams (+ Bigrams) to our TF-IDF vectors improved the F1 scores of 11 classifiers by an average of 8.8%, but also decreased the F1 scores for 6 classifiers by an average of 10.5%.

five-fold cross-validation over the penalty (`C=[0.1, 1, 10]`) and gamma (`gamma=[0.001, 0.01, 0.1]`) parameters. We create rule-based classifiers for four data types (`Identifier`, `Identifier IMSI`, `Identifier SIM Serial`, and `Identifier SSID BSSID`) due to the limited amount of data and their superior performance. Our rule-based classifiers identify the presence or absence of a data type based on indicative text strings.

Table 1 shows the effects of our features and preprocessing steps on the F1 scores of our non-rule-based classifiers. The performance is calculated using our training and validation sets. We made sentence filtering an optional part of preprocessing because of the large detrimental effect it has on some of our classifiers, as highlighted in Table 2. In general, our results suggest that the chosen feature and preprocessing steps improve classifier performance. However, ideally they should be chosen on a per-classifier basis to avoid any negative performance impact.

Performance Analysis Table 3 shows the performance of the classifiers on the privacy policies of the test set. We say a policy describes a practice if at least one segment is flagged by the corresponding data type, party, and positive modality classifiers. Since our definition of potential compliance issues does not depend on the negative modality classifier, we do not include it in the table. Because detecting potential compliance issues is dependent on detecting when practices are *not* described in policies (Zimmeck et al. 2017), negative predictive value, specificity, and negative F1 scores are of particular importance.

In the closest related work (Zimmeck et al. 2017), classifiers for contact, identifier, and location data practices covered multiple specific practices. Thus, a direct performance comparison to our classifiers is not possible. However, with negative F1 scores ranging from 78% to 100%, 23 of our specific classifiers achieve better negative F1 scores than the corresponding course-grained classifiers, and 3 performed equally. These results demonstrate that our approach constitutes an overall improvement over the state of the art. We believe that decomposing the classification task into three subproblems increases performance as it allows for a better exploitation of training data compared to monolithic classifiers.

Our results reveal that generally + support is lower for third party practices; that is, third party practices are often not as extensively described in privacy policies as first party practices. It should be further noted that higher counts of - support generally correlate with higher performance scores. Intuitively, it is easier to classify a policy that does not describe a practice, which makes up the majority of - support instances.

We reviewed the errors made by our classifiers and identified several potential areas for improvement. First, approaching the classification task at the level of segments, as suggested by prior work (Wilson et al. 2016; Liu et al. 2018), can pose difficulties for our subproblem classifiers. For example, if a segment describes a `1stParty` performing the `Location` practice, and a `3rdParty` performing `Contact`, our classifiers cannot distinguish which party should be associated with which practice. Thus, performing classifications at the level of sentences may yield performance improvements. Second, the variety of technical language in privacy policies poses challenges. For example, we observed a false positive when "location" was used in the context of "co-location facility", and a false negative when "clear gifs" was used to refer to web beacons. Such errors might be prevented by training on more data or using domain-specific word embeddings (Kumar et al. 2019). Finally, a more sophisticated semantic representation might be necessary in certain cases. For example, we observed misclassification of a sentence which said that although the first party does not perform a practice, third parties do perform the practice.

App Analysis

MAPS detects apps' privacy practices at app store-wide scale. Detecting which practices an app performs relies on static code analysis, a relatively resource-efficient technique

Policy Classification	NPV	Specificity	Neg. F1	Precision	Recall	F1	+/- Support
Contact 1stParty	92%	96%	94%	89%	80%	84%	30/70
Contact 3rdParty	95%	96%	95%	43%	38%	**40%**	8/92
Contact Email Address 1stParty	78%	90%	84%	97%	94%	96%	80/20
Contact Email Address 3rdParty	91%	83%	87%	29%	46%	35%	13/87
Contact Phone Number 1stParty	93%	93%	93%	94%	94%	94%	54/46
Contact Phone Number 3rdParty	97%	93%	95%	22%	40%	29%	5/95
Identifier 1stParty	93%	68%	**78%**	38%	80%	52%	20/80
Identifier 3rdParty	97%	76%	85%	21%	75%	33%	8/92
Identifier Cookie 1stParty	100%	92%	96%	95%	100%	98%	63/37
Identifier Cookie 3rdParty	94%	92%	93%	92%	94%	93%	52/48
Identifier Device ID 1stParty	86%	96%	91%	96%	87%	91%	54/46
Identifier Device ID 3rdParty	97%	95%	96%	83%	90%	86%	21/79
Identifier IMEI 1stParty	99%	99%	99%	94%	94%	94%	17/83
Identifier IMEI 3rdParty	99%	100%	99%	100%	75%	86%	4/96
Identifier IMSI 1stParty	100%	100%	**100%**	100%	100%	100%	3/97
Identifier IMSI 3rdParty	99%	100%	99%	N/A	0%	0%	1/99
Identifier MAC 1stParty	95%	98%	96%	88%	79%	83%	19/81
Identifier MAC 3rdParty	99%	96%	97%	56%	83%	67%	6/94
Identifier Mobile Carrier 1stParty	90%	100%	95%	100%	57%	73%	21/79
Identifier Mobile Carrier 3rdParty	98%	97%	97%	25%	33%	29%	3/97
Identifier SIM Serial 1stParty	100%	97%	98%	73%	100%	84%	8/92
Identifier SIM Serial 3rdParty	100%	99%	99%	50%	100%	67%	1/99
Identifier SSID BSSID 1stParty	99%	100%	99%	100%	80%	89%	5/95
Identifier SSID BSSID 3rdParty	100%	99%	99%	N/A	N/A	N/A	0/100
Location 1stParty	92%	81%	86%	87%	95%	91%	58/42
Location 3rdParty	96%	83%	89%	61%	87%	71%	23/77
Location Cell Tower 1stParty	98%	94%	96%	71%	86%	77%	14/86
Location Cell Tower 3rdParty	98%	95%	96%	29%	50%	36%	4/96
Location GPS 1stParty	99%	94%	96%	88%	97%	92%	29/71
Location GPS 3rdParty	99%	94%	96%	45%	83%	59%	6/94
Location WiFi 1stParty	99%	86%	92%	48%	92%	63%	12/88
Location WiFi 3rdParty	100%	95%	97%	29%	100%	44%	2/98
Single Sign On	89%	90%	90%	83%	81%	82%	37/63
Single Sign On: Facebook	95%	84%	89%	72%	91%	81%	32/68

Table 3: Performance metrics of our classifiers for determining whether or not a privacy policy states that a practice is performed calculated on the test set ($n = 100$). The negative predictive value (NPV) is the precision for negative instances. Specificity is the recall for negative instances. Negative F1 (Neg. F1) is the F1 for negative instances. In the Support column, + is the number of ground-truth positive instances (cases where policies truly describe the practice being performed) and - is the number of ground-truth negative instances (cases where policies truly do not describe the practice being performed). With negative F1 scores ranging from 78% to 100%, the absence of all practices can be classified relatively accurately. Lower positive F1 scores, for example, 40% for access of contact information by third parties, could be the result of insufficient availability of data. N/A is shown where the metrics are undefined, or where a lack of positive ground truth instances would always make the metric zero.

compared to dynamic code analysis. Our system operates on four app resources: Android APIs, permissions, strings, and class structure. If a sensitive Android API is called, the app has the required permissions to make the call, and required string parameters (e.g., the `GPS_PROVIDER` string) are passed in, the system will flag the existence of a first or third party practice depending on the package name of the class from which the call originated. We assume a threat model which considers data as compromised from the moment a privacy-sensitive API appears to be called (Neisse et al. 2016).

After downloading an app from the Google Play Store our system decompiles it into Smali bytecode using Apktool.[4] It then searches through the bytecode, identifying APIs indicative of a privacy practice being performed. Generally, if a practice occurs in a package corresponding to the app's package ID, the practice is considered a first party practice; otherwise, it is considered a third party practice. In order to evaluate the performance of our system's app analysis, we compare its results against ground truth obtained by a manual dynamic analysis.

Compliance Analysis

Our system combines policy and app analysis results to identify potential compliance issues. We define a *potential compliance issue* to mean that an app is performing a practice (e.g., `Location GPS 1stParty`) while its associated policy does not disclose it either generally (e.g., "Our app accesses your *location* data.") or specifically (e.g., "Our app accesses your *GPS* data."). We chose this definition because we observed that policies generally either disclose that a practice is performed or omit discussion of the practice—statements denying practices are rare.

Table 4 shows our system's identification of potential compliance issues and its performance. For the 26 practices for which positive ground truth instances were present, we observe a mean F1 score of 71%. Many potential compliance issues relate to the access of identifiers. However, the three third party location practices Cell Tower, GPS, and WiFi account for 15, 10, and 12 respective findings as well. Notably, all first party practices exhibit a lower number of potential compliance issues than their third party counterparts.

[4] Apktool, https://ibotpeaches.github.io/Apktool/, accessed: March 18, 2019.

Potential Compliance Issue	Precision	Recall	F1	+/-/? Support
Contact Email Address 1stParty	75%	75%	75%	4/77/19
Contact Email Address 3rdParty	38%	71%	50%	7/74/19
Contact Phone Number 1stParty	100%	100%	100%	1/82/17
Contact Phone Number 3rdParty	29%	67%	40%	3/80/17
Identifier Cookie 1stParty	50%	100%	67%	1/70/29
Identifier Cookie 3rdParty	83%	87%	85%	23/48/29
Identifier Device ID 1stParty	70%	88%	78%	16/63/21
Identifier Device ID 3rdParty	96%	86%	91%	58/21/21
Identifier IMEI 1stParty	79%	65%	71%	17/64/19
Identifier IMEI 3rdParty	76%	85%	80%	26/55/19
Identifier IMSI 1stParty	33%	67%	44%	3/78/19
Identifier IMSI 3rdParty	69%	82%	75%	11/70/19
Identifier MAC 1stParty	83%	91%	87%	11/70/19
Identifier MAC 3rdParty	58%	78%	67%	23/58/19
Identifier Mobile Carrier 1stParty	78%	70%	74%	20/61/19
Identifier Mobile Carrier 3rdParty	92%	75%	83%	64/18/18
Identifier SIM Serial 1stParty	50%	50%	50%	2/81/17
Identifier SIM Serial 3rdParty	50%	88%	64%	8/75/17
Identifier SSID BSSID 1stParty	83%	56%	67%	9/74/17
Identifier SSID BSSID 3rdParty	53%	62%	57%	16/67/17
Location Cell Tower 1stParty	100%	100%	100%	2/76/22
Location Cell Tower 3rdParty	79%	73%	76%	**15**/63/22
Location GPS 1stParty	N/A	N/A	N/A	0/77/23
Location GPS 3rdParty	70%	70%	70%	**10**/67/23
Location WiFi 1stParty	50%	100%	67%	1/77/22
Location WiFi 3rdParty	75%	75%	75%	**12**/66/22
Single Sign On: Facebook	56%	45%	50%	11/72/17

Table 4: Performance metrics of our system's ability to detect potential compliance issues on our test set of app/policy pairs ($n = 100$). In the Support column, + is the number of ground-truth positive instances of potential compliance issues, - is the number of ground-truth negative instances, and ? is the number of instances where missing ground truth data from our app analyses makes it unclear whether or not potential compliance issues exist.

4 Privacy Compliance in the Play Store

Our large-scale analysis of free apps in the Google Play Store provides us with a rich dataset for evaluating the state of privacy in a substantial part of the Android ecosystem. Here, we summarize our findings, with a focus on our privacy policy analysis. For a complete description of our findings, please see (Zimmeck et al. 2019).

Analyses at Scale

Designing and implementing a robust system to identify potential compliance issues for large app populations presents challenges of scale. We address those with a pipeline of distributed tasks implemented in a containerized software stack. We performed our Play Store analysis from April 6 to May 15, 2018. Out of 1,049,790 retrieved free apps, 1,035,853 (98.67%) were analyzed successfully. Of the apps which were not analyzed successfully, 1.03% failed to download, 0.21% failed in the static analysis, 0.08% failed in the policy analysis, and 0.01% failed during our re-analysis.[5]

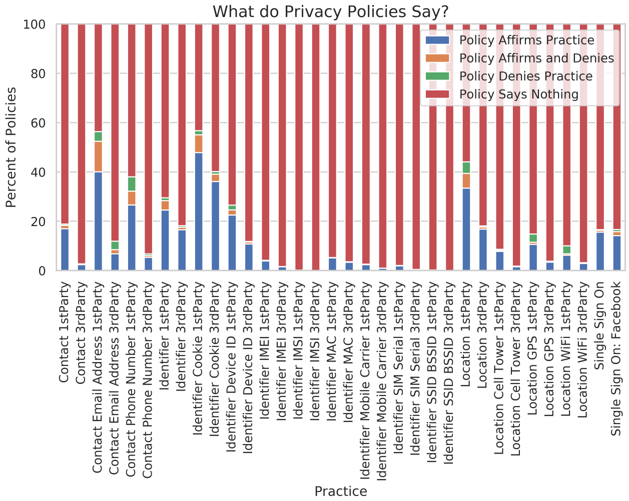

Figure 2: Third party practices are discussed less frequently than first party practices. Given that users often have less opportunity to observe third party practices directly, it is unfortunate that they are not more widely discussed. A policy both affirming and denying a practice does not necessarily imply a contradiction (e.g., "We disclose your phone number to advertisers, but not to data brokers.").

Which Practices are Described in Policies?

35.3% of the apps we analyzed had privacy policies.[6] For apps with privacy policies, Figure 2 depicts the occurrence of policy statements relating to the practices we examine. It can be observed that most practices are described only infrequently; that is, a policy does not mention it at least once. Further, the statements that are present typically affirm that a practice is occurring. This finding reveals that users seem to be given little assurance of potentially objectionable practices not being performed (e.g., disclosing users' phone numbers to third parties). Silence about privacy practices in privacy policies is problematic because there are no clear statutory default rules of what the privacy relationship between a user and a service should be, in the absence of explicit statements in the policy (Marotta-Wurgler 2015).

Prevalence of Potential Compliance Issues

Our system detects potential compliance issues by comparing the privacy policy analysis to the static analysis. A potential compliance issue is detected when an app performs a practice that is not described in the app's privacy policy (if the app even has a privacy policy). Note that when our system finds multiple privacy policies for a given app, it pools

[5] After the completion of the Play Store analysis we noticed a bug in our static analysis code. As a result, we re-performed the static analyses and re-calculated all statistics. 135 additional analyses failed, yielding a final total of 1,035,853 successfully analyzed apps.

[6] This only counts English-language privacy policies: our system does not identify policies in other languages.

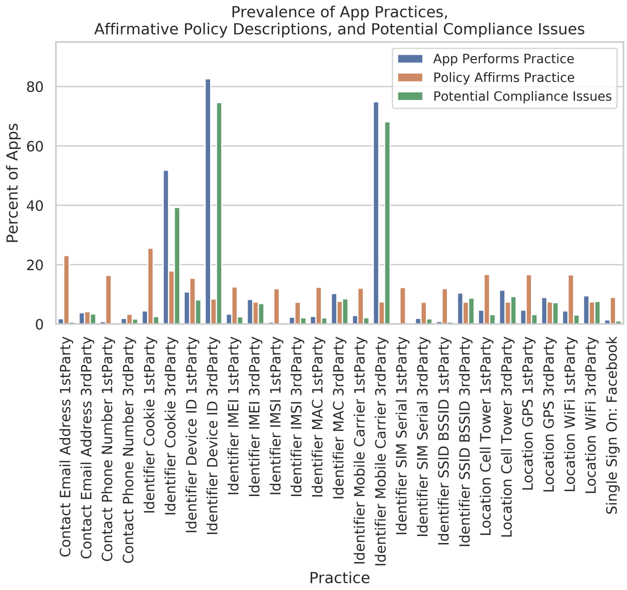

Figure 3: The percents of apps which perform different practices, have policies that affirmatively describe practices as performed, and have potential compliance issues. In this graph, general descriptions of practices are counted with specific descriptions.

the practice descriptions discovered across all those policies. This pooling has the effect of making our results rather conservative. One policy may disclose a particular practice while another policy discloses another practice, and together they may cover all practices performed by the associated app. Overall, the average number of potential compliance issues per app is 2.89 and the median is 3.

Figure 3 shows the percent of apps that perform various practices and the respective percent of apps with potential compliance issues. The figure demonstrates that in many cases the performance of a practice is strongly associated with the occurrence of a potential compliance issue: if a practice is performed, there is a good chance a potential compliance issue exists as well. This result suggests a broad level of potential non-compliance. Identifier-related potential compliance issues are the most common. Three different types of identifiers make up most potential compliance issues: cookies, device IDs, and mobile carriers. In particular, the use of device IDs may constitute a misuse for purposes of ad tracking (Google 2018b). In addition, there are also elevated levels of location-related potential compliance issues. 15.3% of apps perform at least one location-related practice, and 12.1% of apps have at least one location-related potential compliance issue.

For all data types, third party practices are more common than first party practices and so are third party-related potential compliance issues. One reason for the prevalence of potential compliance issues for third party practices could be that app developers are unaware of the functionality of the libraries they integrate. Perhaps they also hold the mistaken belief that it is not their responsibility but the responsibility of the library developers to disclose to users the practices the libraries are performing. Some libraries' terms of services—for example, the Google Analytics Terms of Service (Google 2018a)—obligate the developer integrating it to explicitly disclose the integration in the developer's privacy policy. However, this type of information transfer from the third party via the developer to the user may be susceptible to omissions and mistakes.

5 Conclusions

Natural language privacy policies are intended to communicate how a service collects, shares, uses, and stores user data. However, as they are generally lengthy and difficult to read, the average user often struggles to understand which privacy practices apply. Leveraging natural language processing techniques in the policy domain holds the promise to extract policy content and convert it to a format that is easier to comprehend. In this study, we reported on our development of a three-tiered classification model to classify a variety of privacy practices and their omissions in policy text. Compared to a monolithic classifier for a privacy practice, using data type, party, and modality classifiers allows for economic use of training and test data—which is oftentimes expensive to obtain—as well as good performance.

The classification model we are proposing here is an integral part of the Mobile App Privacy System (MAPS) (Zimmeck et al. 2019). Many mobile apps are reliant on the collection and use of a wide range of data for purposes of their functionality and monetization. MAPS presents one use case for implementing the suggested privacy policy classification model. MAPS pairs our policy analysis with static analysis of mobile apps to identify possible discrepancies between the two and flag potential compliance issues. Our results from analyzing 1,035,853 free apps on the Google Play Store suggest that potential compliance issues are rather common, particularly, when it comes to the disclosure of third party practices. These and similar results may be of interest to app developers, app stores, privacy activists, and regulators.

Recently enacted laws, such as the General Data Protection Directive, impose new obligations and provide for substantial penalties for failing to properly disclose privacy practices. We believe that the natural language analysis of privacy policies, in tandem with mobile app analysis, for example, has the potential to improve privacy transparency and enhance privacy levels overall.

Acknowledgments

We would like to thank Jongkyu Talan Baek, Sushain Cherivirala, Roger Iyengar, Pingshan Li, Shaoyan Sam Li, Kanthashree Sathyendra, Florian Schaub, Xi Sun, and Shomir Wilson for their help with this research. This study was supported in part by the NSF Frontier grant on Usable Privacy Policies (CNS-1330596, CNS-1330141, and CNS-1330214) and a DARPA Brandeis grant on Personalized Privacy Assistants (FA8750-15-2-0277). The US Government is authorized to reproduce and distribute reprints for Governmental purposes not withstanding any copyright notation. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NSF, DARPA, or the US Government. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing high performance computing resources that have contributed to the research results reported within this paper.

References

Avdiienko, V.; Kuznetsov, K.; Gorla, A.; Zeller, A.; Arzt, S.; Rasthofer, S.; and Bodden, E. 2015. Mining apps for abnormal usage of sensitive data. In *37th IEEE/ACM International Conference on Software Engineering, ICSE 2015, Florence, Italy, May 16-24, 2015, Volume 1*, 426–436.

Cranor, L. F.; Langheinrich, M.; Marchiori, M.; Presler-Marshall, M.; and Reagle, J. M. 2002. The Platform for Privacy Preferences 1.0 (P3P1.0) specification. World Wide Web Consortium, Recommendation REC-P3P-20020416.

Cranor, L. F.; Leon, P. G.; and Ur, B. 2016. A large-scale evaluation of U.S. financial institutions standardized privacy notices. *ACM Trans. Web* 10(3):17:1–17:33.

FTC. 2014. Complaint Goldenshores Technologies. https://www.ftc.gov/system/files/documents/cases/140409goldenshorescmpt.pdf. accessed: March 18, 2019.

Google. 2018a. Google analytics terms of service. https://www.google.com/analytics/terms/us.html. accessed: March 18, 2019.

Google. 2018b. Play Console Help. https://support.google.com/googleplay/android-developer/answer/6048248?hl=en. accessed: March 18, 2019.

Harkous, H.; Fawaz, K.; Lebret, R.; Schaub, F.; Shin, K. G.; and Aberer, K. 2018. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *USENIX Security '18*.

Kumar, V. B.; Ravichander, A.; Story, P.; and Sadeh, N. 2019. Quantifying the effect of in-domain distributed word representations: A study of privacy policies. *AAAI Spring Symposium on Privacy-Enhancing Artificial Intelligence and Language Technologies*.

Libert, T. 2018. An automated approach to auditing disclosure of third-party data collection in website privacy policies. In *WWW '18*.

Liu, F.; Wilson, S.; Story, P.; Zimmeck, S.; and Sadeh, N. 2018. Towards Automatic Classification of Privacy Policy Text. Technical Report CMU-ISR-17-118R and CMU-LTI-17-010, School of Computer Science Carnegie Mellon University, Pittsburgh, PA.

Marotta-Wurgler, F. 2015. Does "notice and choice" disclosure regulation work? An empirical study of privacy policies. accessed: March 18, 2019.

McDonald, A. M., and Cranor, L. F. 2008. The cost of reading privacy policies. *I/S: A Journal of Law and Policy for the Information Society* 4(3):540–565.

Mutchler, P.; Doupé, A.; Mitchell, J.; Kruegel, C.; and Vigna, G. 2015. A large-scale study of mobile web app security. In *MoST '15*.

Neisse, R.; Steri, G.; Geneiatakis, D.; and Fovino, I. N. 2016. A privacy enforcing framework for android applications. *Computers & Security* 62:257 – 277.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*.

Ramanath, R.; Liu, F.; Sadeh, N.; and Smith, N. A. 2014. Unsupervised alignment of privacy policies using hidden markov models. In *ACL '14*.

Ren, J.; Lindorfer, M.; Dubois, D.; Rao, A.; Choffnes, D.; and Vallina-Rodriguez, N. 2018. Bug fixes, improvements, ... and privacy leaks – a longitudinal study of PII leaks across android app versions. In *NDSS '18*.

Reyes, I.; Wijesekera, P.; Reardon, J.; On, A. E. B.; Razaghpanah, A.; Vallina-Rodriguez, N.; and Egelman, S. 2018. "Won't somebody think of the children?" Examining COPPA compliance at scale. In *PETS '18*.

Sathyendra, K. M.; Wilson, S.; Schaub, F.; Zimmeck, S.; and Sadeh, N. 2017. Identifying the provision of choices in privacy policy text. In *EMNLP '17*.

scikit-learn developers. 2016a. sklearn.feature_extraction.text.tfidfvectorizer. http://scikit-learn.org/0.18/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html. Accessed: March 18, 2019.

scikit-learn developers. 2016b. sklearn.svm.svc. http://scikit-learn.org/0.18/modules/generated/sklearn.svm.SVC.html. Accessed: March 18, 2019.

Slavin, R.; Wang, X.; Hosseini, M.; Hester, W.; Krishnan, R.; Bhatia, J.; Breaux, T.; and Niu, J. 2016. Toward a framework for detecting privacy policy violation in android application code. In *ICSE '16*.

Solove, D. J., and Hartzog, W. 2014. The FTC and the new common law of privacy. *Columbia Law Review* 114:583–676.

Story, P.; Zimmeck, S.; and Sadeh, N. 2018. Which apps have privacy policies? In *APF '18*.

Tesfay, W. B.; Hofmann, P.; Nakamura, T.; Kiyomoto, S.; and Serna, J. 2018. I read but don't agree: Privacy policy benchmarking using machine learning and the EU GDPR. In *WWW '18*.

Wilson, S.; Schaub, F.; Dara, A. A.; Liu, F.; Cherivirala, S.; Leon, P. G.; Andersen, M. S.; Zimmeck, S.; Sathyendra, K. M.; Russell, N. C.; Norton, T. B.; Hovy, E.; Reidenberg, J.; and Sadeh, N. 2016. The creation and analysis of a website privacy policy corpus. In *ACL '16*.

Yu, L.; Luo, X.; Liu, X.; and Zhang, T. 2016. Can we trust the privacy policies of android apps? In *DSN '16*.

Zhuang, Y.; Rafetseder, A.; Hu, Y.; Tian, Y.; and Cappos, J. 2018. Sensibility testbed: Automated IRB policy enforcement in mobile research apps. In *HotMobile '18*.

Zimmeck, S., and Bellovin, S. M. 2014. Privee: An architecture for automatically analyzing web privacy policies. In *USENIX Security '14*.

Zimmeck, S.; Wang, Z.; Zou, L.; Iyengar, R.; Liu, B.; Schaub, F.; Wilson, S.; Sadeh, N.; Bellovin, S. M.; and Reidenberg, J. 2017. Automated analysis of privacy requirements for mobile apps. In *NDSS '17*.

Zimmeck, S.; Story, P.; Smullen, D.; Ravichander, A.; Wang, Z.; Reidenberg, J.; Russell, N. C.; and Sadeh, N. 2019. MAPS: Scaling privacy compliance analysis to a million apps. In *PETS '19*.